

Modeling and Simulation of Variational Rigid and Compliant Assembly for Tolerance Analysis

by

Eng. Pasquale Franciosa
Department of Mechanical Engineering and Energetics

Research Doctorate
Mechanical Engineering Systems – XXII cycle

at

University of Naples Federico II



Coordinator of the PhD School: Prof. Eng. Raffaele Tuccillo
University of Naples Federico II, School of Engineering

Supervisor: Prof. Eng. Francesco Caputo
University of Naples Federico II, School of Engineering

Supervisor: Prof. Eng. Salvatore Gerbino
University of Molise, School of Engineering

November 2009

a mio Padre e a mio Nonno
to my Father and my Grandfather

Acknowledgments

Vorrei innanzitutto ringraziare il prof. Francesco Caputo ed il prof. Antonio Lanzotti per avermi accolto nel loro gruppo di ricerca, credendo fin da subito nelle mie capacità.

Un ringraziamento particolare va al prof. Salvatore Gerbino che mi ha seguito in qualità di *supervisor* fin dai tempi della tesi specialistica ed ha sempre saputo incoraggiarmi e stimolarmi per farmi dare il meglio di me stesso. La sua passione per il lavoro e per la ricerca ed il suo profondo senso critico sono stati di forte insegnamento. Non dimenticherò mai le discussioni e le riunioni avute con lui durante questi anni. A lui devo veramente tanto sia dal punto di vista scientifico sia morale.

Desidero ringraziare i colleghi ed amici del DPGI, con i quali ho condiviso gli ultimi tre anni: Prof. Giuseppe Di Gironimo, Prof. Massimo Martorelli, Prof. Stanislao Patalano, Prof. Fabrizio Renno, Adelaide Marzano, Stefano Papa, Domenico Speranza e Andrea Tarallo. Un ringraziamento particolare va anche ai colleghi del DIAS: Giovanna Matrone e Mariangela Trotta.

Desidero poi ringraziare i tesisti che ho seguito in questi tre anni. Le loro curiosità e loro domande mi hanno spesso fatto riflettere e sono servite per arricchire la mia ricerca. Un ringraziamento particolare va a Vito Rufrano ed a Domenico e Giovanni Tuccillo.

Ringrazio la mia famiglia per avermi aiutato e sostenuto, non solo dal punto di vista economico, ma soprattutto dal punto di vista morale. Il loro affetto ha contato e conta molto per me. Un ringraziamento particolare va a mia madre, mia zia Giovanna e mio fratello. Ringrazio, poi, mio padre e mio nonno per avermi insegnato il significato profondo del rispetto e dell'onore.

Nell'ultimo anno ho riscoperto la gioia di vivere grazie ad una persona *speciale*: Nisia. A lei sono grato per avermi incoraggiato ed essermi stata vicino nei momenti difficili (e sono stati tanti) e per aver avuto tanta pazienza durante il mio soggiorno a Boston.

Last but not least, I would like to thank prof. Alain Riviere, for hosting me as visiting student at SUPMECA-Paris. My gratitude also goes to prof. Daniel Whitney who improved the quality of my research by critically reviewing some chapters of this dissertation.

Modeling and Simulation of Variational Rigid and Compliant Assembly for Tolerance Analysis

by

Eng. Pasquale Franciosa
Department of Mechanical Engineering and Energetics

Abstract

Modern manufacturing processes are strongly affected by part and process variations. Understanding the final shape of the assembly is a crucial task to be achieved during the design stage to reduce cost and production time. Typically, when parts are put together variations propagate part-to-part. This stack-up effect is strictly related to part deviations, assembly sequence, assembly constraints, part flexibility. All these factors combine into a non-linear way. Technical literature provides valid methodologies to analyze tolerance stack-up problems. In this contest, assemblies are usually classified into two main categories: rigid assemblies and compliant assemblies. In the first case, parts are assumed ideal-rigid and the additional variation due to elastic or plastic deformation is not accounted. When the flexibility of parts is not negligible, as into manufacturing processes involving sheet-metal parts, its effect has to be accounted into numerical simulations. This dissertation provides a contribution to the modeling and the simulation both of rigid and compliant assemblies.

With respect to rigid assembly, a general methodology, called SVA-TOL (Statistical Variation Analysis for Tolerancing), is proposed. The methodology is based into two main steps. Tolerance specifications are modeled in the first step following International Standard rules. Variational features are so generated. Then, by using these variational features, assembly constraints are modeled as combination of elementary geometry entities: points, lines and planes. Two assembly solvers are illustrated: the sequential solver and the least squares solver.

The sequential solver allows to analyze assembly constraints taking into account the assembly sequence. Specific mathematical tools, such as Screw Theory and Graph Theory, are here adopted to automatically calculate the list of degrees of freedom allowed for a specific part being assembled.

The least squares solver, instead, permits to analyze all assembly constraint si-

multaneously by best fitting all mating conditions. Case studies have pointed out the field of applicability of proposed assembly solvers.

With respect to compliant assembly, a general frame-work, called SVA-FEA (Statistical Variation Analysis & Finite Element Analysis), is presented. SVA-FEA allows to statistically simulate single- and multi-station assembly processes under linear assumptions. A *Global Sensitivity Matrix* is introduced to link input part or process deviations and output assembly deviations. In this way, no Monte Carlo simulation is needed. The whole assembly process is based on the Place, Clamp, Fasten and Release (PCFR) cycle. Parts are positioned on fixturing frames, where are clamped and fastened. Then, they are released reaching their final assembly configuration. To numerically calculate the sensitivity matrix two FEA runs, performed on the nominal geometry, are required. The first one calculates the fixturing and fastening forces, by applying the method of influence coefficients. These forces are then applied into the second FEA run to simulate the final elastic spring-back. SVA-FEA methodology has been implemented into a friendly MatLAB®'s GUI, allowing to define the whole assembly process, with its variability, and to visualize the final assembly deviations, in terms on mean and standard deviations. Two significant case studies have highlighted how SVA-FEA allows to simulate both single- and multi-station assembly processes. Results have been compared with ones coming from commercial CAT software, showing a good numerical correlation.

SVA-FEA assumes that input statistical deviations are independent among them. This means that no covariance effect is accounted. To overcome this weakness, a new non-linear methodology is also proposed. The non-linear methodology, to do variational analysis of compliant assembly, uses a Monte Carlo approach to statistically generate input free shape geometry. The methodology can be described as follows. For each Monte Carlo step, input geometry is generated according to an automatic morphing mesh procedure. Then, the PCFR cycle is simulated. The geometry is so updated for each phase of the PCFR cycle. To reach more realistic results, contact pairs, defined as surface-to-surface type, may be introduced into the numerical simulation. The morphing mesh approach allows to generated free shape parts starting from few control points, defined on the nominal geometry, and setting the tolerance error value.

Finally, critical remarks are outlined, highlighting future directions of research in the field of tolerance analysis of rigid and compliant assembly processes.

Table of Contents

<u>Abstract</u>	I
<u>List of Figures</u>	VII
<u>List of Tables</u>	XI
<u>List of Annexes</u>	XII
<u>1. Thesis Overview</u>	
1.1 Introduction	1
1.2 Motivations	2
1.2.1 Variational Feature Modeling	3
1.2.2 Tolerance Analysis for Rigid Part Assemblies	3
1.2.3 Tolerance Analysis for Compliant Part Assemblies	4
1.2.4 Tolerance Analysis and Simulation Tools	4
1.3 Goals of Research	5
1.4 Thesis Organization	5
<u>2. Review of Previous Works</u>	
2.1 Introduction	7
2.2 Variational Feature Modeling	8
2.2.1 TTRS Classification	8
2.2.2 Geometric Covariance	9
2.3 Tolerance Analysis for Rigid Part Assemblies	11
2.3.1 One-dimensional Tolerance Chart	12
2.3.2 Vector Loop-based Tolerance Analysis	12
2.3.3 4x4 Chain Transformation Matrix	13
2.3.4 Tolerance Domain Analysis	14
2.4 Tolerance Analysis for Compliant Part Assemblies	15

2.4.1 PCFR Cycle	15
2.4.2 Method of Influence Coefficients	16
2.4.3 Flexible Beam Elements	18
2.4.4 FASTA Framework	18
2.4.5 Multi-station Assembly Modeling	19
2.4.6 Including Non-Linear Effects	20
2.5 Tolerance Analysis and Simulation Tools	21
2.6 Summary	22
3. <u>Graph Theory for Tolerancing</u>	
3.1 Introduction	23
3.2 Graph Definition and Manipulation	24
3.2.1 Adequacy Analysis	24
3.2.2 Consistency Analysis	27
3.3 Implementation	30
3.4 Examples	32
3.4.1 Two-part Assembly	32
3.4.2 Graph Managing	33
3.5 Summary	35
4. <u>Assembly Constraint Analysis</u>	
4.1 Introduction	37
4.2 Methodology Overview	39
4.3 Exploded Graph-based Assembly	40
4.4 Graph-based Manipulation	41
4.4.1 Path Detection	41
4.4.2 Loop Detection	42
4.5 Evaluating Twist Matrix	44
4.6 Davies's Law and Constraint Analysis	46
4.6.1 Motion Analysis	46
4.6.2 Constraint Analysis	48
4.7 Examples	50
4.7.1 Four-bar Assembly	50
4.7.2 Parallelogram Mechanism	51
4.7.3 Landing Wheel Mechanism	52
4.8 Summary	55
5. <u>Rigid Assemblies</u>	
5.1 Introduction	57
5.2 Methodology Overview	58

5.3 Variational Feature Modeling	59
5.3.1 Feature Modeling	59
5.3.2 Statistical Simulation	64
5.4 Assembly Constraint Modeling	66
5.5 Sequential Solver	69
5.5.1 Procedure Overview	69
5.5.2 Solving Mate Joint	71
5.5.3 Solving Fit Joint	74
5.5.4 Updating Joint Matrix	75
5.5.5 Updating DoFs List	75
5.6 Least Squares Solver	78
5.7 Data Structure for Tolerance Analysis Software	78
5.7.1 Variational Features	79
5.7.2 Assembly Constraints	80
5.8 Examples	82
5.8.1 Sequential Solver: Three-mate Joints	82
5.8.2 Sequential Solver: Mate and Fit Joints	84
5.8.3 Least Squares Solver Example	86
5.9 Summary	88

6. Linear Analysis of Compliant Assemblies

6.1 Introduction	90
6.2 Methodology Overview	90
6.3 Assembly Process	92
6.3.1 Fixturing Point Modeling	92
6.3.2 Fastening Point Modeling	93
6.3.3 Contact Point Modeling	94
6.3.4 Statistical Input Data	96
6.3.5 Assembly Operation	97
6.4 Solution	98
6.4.1 The Global Sensitivity Matrix	98
6.4.2 Method of Influence Coefficient	100
6.4.3 Calculating the Sensitivity Matrix	100
6.5 Data Structure for Tolerance Analysis Software	104
6.6 Implementation	105
6.7 Examples	108
6.7.1 Single-station Simulation	108
6.7.2 Multi-station Simulation	112
6.8 Summary	122

<u>7. Non-Linear Analysis of Compliant Assemblies</u>	
7.1 Introduction	124
7.2 Free Shape Feature Modeling	125
7.2.1 Morphing Mesh Procedure	125
7.2.2 Statistical Simulation	128
7.3 PCFR Simulation	129
7.3.1 Positioning/Clamping Phase	131
7.3.2 Fastening Phase	132
7.3.3 Releasing Phase	136
7.4 Implementation	136
7.5 Examples	138
7.5.1 Free Shape Feature Modeling	138
7.5.2 Free Shape Part Modeling	139
7.5.3 PCFR Simulation	141
7.6 Summary	146
<u>8. Conclusions and Future Works</u>	
8.1 Conclusions	147
8.2 Future Works	151
<u>Annexes</u>	155
<u>References</u>	191

List of Figures

Figure 2.1: general work-flow of the state space model (Camelio, 2002)	20
Figure 3.1: Assembly-based Graph	25
Figure 3.2: Exploded Assembly-based Graph	28
Figure 3.3: parametric CAD model to check tolerance specification mistakes	29
Figure 3.4: Graphical User Interface	30
Figure 3.5: example. Tolerance specification and graph representation	32
Figure 3.6: example. analysis results	32
Figure 3.7: example. Graph representation	33
Figure 3.8: example. Analysis results	33
Figure 4.1.a: Graph-based Assembly representation	40
Figure 4.1.b: Exploded Graph-based Assembly representation	40
Figure 4.2: main graph and secondary graph	43
Figure 4.3: example. Four bar assembly	50
Figure 4.4: example. Parallelogram mechanism and EGrA representation	51
Figure 4.5: example. Landing wheel mechanism	53
Figure 5.1: variational feature modeling	60
Figure 5.2.a: planar feature	62
Figure 5.2.b: cylindrical feature	63
Figure 5.3: hyper-box and hyper-domain (shaded) for cylindrical tolerance zone	65
Figure 5.4.a: <i>mate</i> assembly joint. Before assembling	66
Figure 5.4.b: <i>mate</i> assembly joint. After assembling	66
Figure 5.5.a: <i>fit</i> assembly joint. Before assembling	67
Figure 5.5.b: <i>fit</i> assembly joint. After assembling	67
Figure 5.6.a: ideal mating features	67
Figure 5.6.b: variational mating features	67
Figure 5.7: assembly constraint modeling for sequential solver	70
Figure 5.8: sequential solver work-flow	71
Figure 5.9: mate joint condition	72

Figure 5.10: mapped mesh for the evaluation of assembly translational parameters in the mate joint condition	73
Figure 5.11: target and object planes for different initial configurations	73
Figure 5.12: fit joint condition	74
Figure 5.13: graph and definition of coordinate frames for updating DoFs	76
Figure 5.14: data structure for variational feature generation	79
Figure 5.15: data structure for assembly constraint simulation	80
Figure 5.16: example: three mate joints	82
Figure 5.17: three mate joints: assembly geometry for different assembly sequence variation scale factor = 100	83
Figure 5.18: analysis of assembly sequence "V"	83
Figure 5.19: example. Fit and mate joints	84
Figure 5.20: fit and mate joints. Assembly geometry configurations. Variation scale factor = 20	84
Figure 5.21: fit and mate joints. Analysis of points P_0 and P_1 when object part rotates around the ISA axis	85
Figure 5.22: fit and mate joints. Histograms of frequencies. 1000 Monte Carlo simulations	85
Figure 5.23: example. Least squares solver application	86
Figure 5.24: least squares solver application: assembly geometry configurations. Variation scale factor = 20	87
Figure 6.1: general work-flow of the methodology	91
Figure 6.2: fastening modeling	94
Figure 6.3: linear contact modeling	95
Figure 6.4: constitutive law for linear contact modeling	96
Figure 6.5: combination of form defects and positioning errors	97
Figure 6.6: assembly operation modeling	98
Figure 6.7: general work-flow for sensitivity matrix evaluation	100
Figure 6.8: constraint configurations for the first FEA run	101
Figure 6.9: constraint configurations for the second FEA run	103
Figure 6.10: data structure for key point definition	104
Figure 6.11.a: SVA-FEA Graphical User Interface: Main GUI	105
Figure 6.11.b: SVA-FEA Graphical User Interface. Main menus	106
Figure 6.12: two-part assembly and monitoring points	108
Figure 6.13: SVA-FEA and TAA models	109
Figure 6.14: contour plot for mean displacements along global X axis	109
Figure 6.15.a: mean results along global X axis	110
Figure 6.15.b: standard deviation results along global X axis	110
Figure 6.16.a: mean results along global X axis. Contact vs no-contact analysis	112
Figure 6.16.b: Standard deviation results along global X axis. Contact vs no-contact analysis	112

Figure 6.17: four-part assembly and monitoring points	113
Figure 6.18: SVA-FEA and TAA models	113
Figure 6.19: initial deviations of part A calculated in SVA-FEA	114
Figure 6.20: initial deviations of part D calculated in SVA-FEA	114
Figure 6.21: SVA-FEA results in the first assembly configuration	116
Figure 6.22: TAA results in the first assembly configuration. Releasing phase in the third station	116
Figure 6.23.a: mean deviations at monitoring points. Euclidean norm is adopted. First assembly configuration. $d=5$ mm	117
Figure 6.23.b: standard deviations at monitoring points. Euclidean norm is adopted. First assembly configuration. $d=5$ mm	117
Figure 6.24.a: mean deviations at monitoring points. Euclidean norm is adopted. Second assembly configuration	119
Figure 6.24.b: standard deviations at monitoring points. Euclidean norm is adopted. Second assembly configuration	119
Figure 6.25: SVA-FEA results in the second assembly configuration (final releasing). Spot weld diameter is 100 mm	120
Figure 6.26: TAA results in the second assembly configuration (final releasing). Spot weld diameter is 100 mm	120
Figure 7.1: general work-flow of the methodology	124
Figure 7.2: definition of control points and relative influence hulls	126
Figure 7.3: definition of the basic function, f	127
Figure 7.4: work-flow of the non-linear methodology proposed for PCFR cycle simulation	130
Figure 7.5: positioning modeling	132
Figure 7.6: closing gap phase	133
Figure 7.7: line-mesh intersection procedure	134
Figure 7.8: Graphical User Interface	136
Figure 7.9: free shape feature modeling. Nominal model and influence hulls	138
Figure 7.10: free shape feature modeling. Influence hulls	138
Figure 7.11: free shape feature modeling. Six sampled features variation scale factor = 10	139
Figure 7.12: free shape part modeling input statistical values	140
Figure 7.13: free shape part modeling. MMP_I results	140
Figure 7.14: free shape part modeling. MMP_{II} - influence hulls assigned at control points for the local influence of the geometry	141
Figure 7.15: free shape part modeling. MMP_{II} results	141
Figure 7.16: PCFR case study. Nominal geometry, control points ($P_{c,j}$), spot welds (SP_i) and inspection points (MS_i)	142
Figure 7.17: PCFR case study. Results	143
Figure 7.18: mean and standard deviation results	144
Figure A.1: frame definition	155
Figure A.2: composition of transformations	158

Figure D.1.a: graph representation. Non-oriented graph	169
Figure D.1.b: graph representation. Oriented graph	169
Figure E.1: tolerance zone. Planar feature-two parallel planes	174
Figure E.2: tolerance zone. Cylindrical feature-cylindrical zone	175
Figure E.3: tolerance zone. Cylindrical feature-two coaxial cylinders	176
Figure E.4: tolerance zone. Cylindrical feature-two parallel planes	177
Figure E.5: tolerance zone. Cylindrical feature-parallelepiped zone	178

List of Tables

Table 2.1: recent publications in variation propagation	19
Table 5.1: joint definition	68
Table 5.2: three-mate joints. Feasible assembly sequences	82
Table 6.1: statistical input variability	108
Table 6.2: correlation indexes	111
Table 6.3: percentage RSS errors	111
Table 6.4: percentage RSS error. Contact vs no-contact analysis	111
Table 6.5: statistical input variability	114
Table 6.6: assembly configurations	115
Table 6.7: correlation indexes. First assembly configuration	118
Table 6.8: RSS errors. First assembly configuration	118
Table 6.9: RSS errors. Influence of spot weld diameter. First assembly configuration	118
Table 6.10: correlation indexes. Second assembly configuration. $d=100$ mm	121
Table 6.11: RSS errors. Second assembly configuration. $d=100$ mm	121
Table 7.1: statistical values assigned at control points. Gaussian distribution assumed	142
Table 7.2: contact pair vs no-contact pair	145
Table E.1: tolerance zones-summary	179

List of Annexes

<u>A. 4x4 Transformation Matrix</u>	
A.1 Transformation Matrix	155
A.2 Composition of Transformations	157
<u>B. Linear Algebra Overview</u>	
B.1 Systems of Linear Equations	159
B.2 Eigenvalues and Eigenvectors	162
B.3 Singular Value Decomposition	163
<u>C. Mex File in MatLAB®</u>	
C.1 Introduction	165
C.2 Write and Compile Mex File	166
<u>D. Introduction to Graph Theory</u>	
D.1 Edge Matrix and Incidence Matrix	169
D.2 Graph Matrix	171
<u>E. Tolerance Zones</u>	
E.1 Introduction	173
E.2 Tolerance Zone Representation	174
E.3 Summary	179
<u>F. Boundary Integral Definition</u>	
F.1 Integral Definition	181
<u>G. Bulk Data File</u>	
G.1 Introduction	183
G.2 bdf Structure	184
G.3 Examples	189

Chapter 1.

THESIS OVERVIEW

Process and product quality control involve different engineering fields and knowledge, from statistical process control to robust design and tolerance design.

This dissertation focuses on latter topic: tolerance design and methodologies for predicting, during the design stage, the amount of variation involved into real assembly processes.

1.1 Introduction

It was reported that in aerospace and automobile industries 65-70% of all design changes and failures are related to dimensional or geometrical variation caused by lack of technology or knowledge for accurate prediction of process variation during the product design stage (Ceglarek, 2009). It is well recognized that geometrical accuracy and dimensional variations are two of the most influence factors in several manufacturing processes.

Generally speaking, when parts are put together, manufacturing and assembling errors affect final Key Characteristics (KCs). How to account and numerically simulate errors occurring into mechanical assembly processes is investigated in the present dissertation.

Two key words are here introduced: *variation* and *tolerance*. *Variation* is the physical deviation from the nominal size of a part due to manufacturing or assembling errors. *Tolerance*, instead, is the amount of variation which the designer may tolerate assuring a given KC (Whitney, 2004).

In the mechanical assembly field, the process aiming to assign a tolerance value,

Chapter 1. Thesis Overview

on a given set of features is called *tolerance specification*. Nowadays, the tolerance specification process is achieved taking into account International Standards, such as ISO-GPS (ISO, 2004) or ANSI-GD&T (ASME, 2004).

During the tolerance design stage, two processes are typically involved: *tolerance analysis*, which allows to determine the amount of variation on specific KCs for a given set of tolerance specifications, and *tolerance synthesis*, which aims to evaluate the amount of variation of functional features assuring that KCs do not exceed target design values.

Technical literature on tolerances is worth of methods about specification, modeling and analysis of tolerances, and they mainly fall into two categories of assemblies: rigid body assemblies and flexible body assemblies. In the first category all parts are assumed rigid, so no variation occurs for part deformation; in the second case, instead, parts are assumed compliant - see sheet-metal components - and the additional effect due to part deformation during the assembly process is taken into account in the tolerance analysis.

In this dissertation a general methodologies to accomplish tolerance analysis, both for rigid and compliant assemblies, from a statistical point of view, is discussed. Final results are expressed as displacements, in terms of mean and standard deviation, for a specific set of points or, generally speaking, for given functional features.

1.2 Motivation

Variation occurring during the manufacturing or assembling phases causes the assembly, or a set of functional features, to be a wrong shape or to be in a wrong position and orientation with respect to a global reference frame.

Considering all sources of variation, involved during an assembly process, is not a trivial task. All this is especially true for a general 3D tolerance stack-up, where variations propagate part-to-part into a non-linear way, and general constraint conditions among part being assembled are defined. Moreover, often, assemblies exhibit a flexible behavior and then their deformation should be introduced into the analysis.

Over the years, researchers have addressed their attention on the tolerance analysis topic. The literature covers four main fields:

- variational feature modeling;
- tolerance analysis for rigid part assemblies;
- tolerance analysis for compliant part assemblies; and,
- tolerance analysis and simulation tools.

1.2.1 Variational Feature Modeling

Tolerance analysis process begins with the definition and the assignment of variational features. Researchers have been focusing their efforts to define, into a mathematical way, 3D tolerance zone for each tolerance specification. Typically, shape errors are neglected and variational features are modeled by introducing small rotational and small translational parameters. These parameters describe all configurations of that feature within the tolerance zone. The rigid body motion theory is here applied to account such variational parameters. This issue is well known in literature, where two main mathematical approaches are adopted: the small displacement *torsor* proposed in (Bourdet, 1988) and the 4x4 matrix transformation method as in (Whitney, 1994). The main idea is to parameterize every variational feature with a set of parameters. Generally speaking, these parameters are dependent among them. How to account this is achieved with a parametric domain, enveloping all feature configurations within the tolerance zone. The domain method is then proposed in (Giordano, 2005a) to perform tolerance analysis on a given mechanical assembly. Based on the same principle, (Shah, 2007) suggested to use the T-map space as a representation of all feature configurations.

More sophisticated models were proposed into (Huang, 2002) or (Samper, 2007), where shape errors were also accounted by using a modal decomposition analysis based on real measurement data.

1.2.2 Tolerance Analysis for Rigid Part Assemblies

Traditionally, tolerance stack-up is solved under the hypothesis of ideal rigid parts. Variational features are used to simulate the accumulation of deviations during the whole assembly phase. Three main types of variability are here identified: tooling and fixturing variability, part variability and assembly sequence (Huang, 2007a; Huang, 2007b).

Typically, assembly processes are made in multi-station phases. For each phase, parts are positioned, joined, clamped and then released. Obviously, parts may deform and final assembly shape is influenced by flexibility of components. Under the hypothesis of ideal rigid parts, different methodologies have been proposed. Researchers at Brigham Young University (USA) developed a robust method, called Direct Linearization Method (DLM) and based on a Taylor's series expansion (Chase, 1997). French researchers (Giordano, 2007), developed the domain method as a general methodology to do tolerance analysis. By using the small displacement torsor, feature domains were combined, accounting assembly features, and then the global assembly domain was compared with one related to functional requirements.

1.2.3 Tolerance Analysis for Compliant Part Assemblies

Over the last decade, variational compliant assembly analysis has been becoming a critical topic in those assembly processes involving parts highly deformable, such as sheet-metal parts, widely used in aerospace and mechanical applications. Due to part deformation, combined with process variability, final assembly shape cannot be predicted using rigid assembly methodologies. To overcome these limits, many valid methodologies, mainly based on FEA approach, have been suggested.

At Brigham Young University, researchers developed a general framework, called FASTA (Flexible Assemblies Statistical Tolerance Analysis) allowing to statically predict variability for flexible assemblies starting from experimental measurements (Mortensen, 2002).

(Liu, 1997) developed a general methodology to numerical simulate part deviations. The method of influence coefficient was there introduced: deviations at part level were related to deviations at assembly level by means of the so-called sensitivity matrix. Two FEA runs were required to numerically evaluate the sensitivity matrix. The method of influence coefficient was then extended to multi-station processes involving fixturing and tooling deviations (Camelio, 2004b).

Since its origin, compliant variational analysis was based on linear assumption. Non-linear phenomena, such as part-to-part contacts, were not accounted. Recently, researchers have shown how contacts between parts being assembled highly influence final simulation results (Xie, 2007). Therefore, to achieve results closer as much as possible to real assembly processes, non-linear contact should be introduced into simulation models.

1.2.4 Tolerance Analysis and Simulation Tools

Tolerance analysis aims to numerically evaluate the amount of deviations when parts are assembled. Typically, two main approach are adopted: worst case simulation and statistical simulation.

Many commercial CAT software, such as VisVSA® (by UGS Co.), eM-TolMate® (by TecnoMatix/UGS Co.), CETOL 6 σ ® (by Sigmetrix LLC.), 3DCS® (by Dimensional Systems Inc.), Mechanical Advantage® (by Cognition Co.) and Sigmund® (by Varatech Co.), are available nowadays to do tolerance analysis of rigid part assemblies. Generally speaking, these CAT packages allow to do both worst case and statistical analyses. Instead, only three CAT tools to do tolerance analysis of compliant part assemblies are available: TAA® (by Dassault Systemes), 3DCS-FEA® (by Dimensional Systems Inc.), VisVSA-FEA® (by UGS Co.).

1.3 Goals of Research

The present dissertation, starting from the state of the art, provides a methodology to numerically perform tolerance analysis both of rigid and compliant assembly, taking into account non-linear stack-up effects.

For those assemblies which may be assumed rigid, the SVA-TOL (Statistical Variation Analysis for Tolerancing) methodology has been developed. The general SVA-TOL work-flow can be summarized as follows. Nominal assembly geometry is imported from a CAD system. Tolerance specifications are modeled for each part being assembled, according to ISO or ANSI specifications, and then assembly constraints are introduced among assembly features. Each constraint equation is considered as a combination of point, line and plane entities. A numerical solution, based on iterative algorithms and Screw Theory, for the assembly constraint problem is provided. In particular, Screw Theory is used to determine the list of Degree of Freedoms (DoFs) allowed to the part being assembled.

When deformation needs to be accounted, the SVA-FEA (Statistical Variation Analysis & Finite Element Analysis) methodology is introduced. SVA-FEA allows to simulate single- and multi-station assembly processes, taking into account both part and fixture variations. Here, a *Global Sensitivity Matrix* is defined to calculate the influence, at assembly level, due to input deviations. This matrix is numerically evaluated performing two consecutive FEA runs. All sources of variation are assumed statistical independent variables and the small displacement hypothesis is adopted. All these assumptions preserve the linearity of the model.

Assuming independent statistical variations does not allow to well-simulate the geometry's real shape. This issue is known in literature as *geometric covariance*, which relates geometrically neighboring points on the same surface. Geometric covariance assures surface continuity and smoothness. In the present dissertation, a morphing mesh-based approach is used to generate variational shapes, according to deviations occurring in a small set of points defined on the nominal geometry. Having these variational parts, the assembly process is then simulated.

1.4 Thesis Organization

After an overview on the main methodologies available in literature, two chapters focus on tolerance specification and on constraint analysis for rigid assemblies. Then, SVA-TOL and SVA-FEA methodologies are illustrated. The main weaknesses and limits of SVA-FEA are pointed out and then a morphing mesh procedure is de-

Chapter 1. Thesis Overview

scribed to overcome them. Finally, some applications show the applicability of proposed numerical methodologies to real assembly case studies. Also, comparative studies with a commercial CAT package are described.

The dissertation is arranged as follows:

- Chapter 2 gives an overview of the proposed methods available in literature to do tolerance analysis of rigid and compliant assemblies;
- Chapter 3 focuses on the *tolerance specification* topic, highlighting the need to have a global consistency specification;
- Chapter 4 shows how to mathematically analyze constraints into assemblies by combining Screw Theory and Graph Theory;
- Chapter 5 develops the SVA-TOL methodology, highlighting the main features and the limits;
- Chapter 6 describes the SVA-FEA methodology, with a special reference to the Graphical User Interface (GUI), implemented into MatLAB® environment;
- Chapter 7 shows the actual limit of SVA-FEA and provides a non-linear methodology to do tolerance analysis of compliant assemblies through a morphing mesh approach; and,
- Chapter 8 remarks final conclusions and a critical discussion of future directions for the research in tolerance field.

Chapter 2.

REVIEW OF PREVIOUS WORKS

In Chapter 1 four key topics have been identified as most relevant in the field of tolerance design: (I) variational feature modeling, (II) tolerance analysis for rigid part assemblies, (III) tolerance analysis for compliant part assemblies, and (IV) tolerance analysis and simulation tools. This Chapter revisits related works available in literature about these key topics.

2.1 Introduction

When working with tolerance analysis, the main scope is to find out the functional relationship between independent and dependent variables. From a mathematical point of view, all this may be expressed as in equation (2.1):

$$\mathbf{z} = G(\mathbf{u}) \quad (2.1)$$

where \mathbf{z} is the KC (dependent variable) aimed to evaluate, while \mathbf{u} is the list of assigned deviations (independent variables). Function G accounts all phenomena which may occur during the assembly process: assembly constraints, assembly sequence, flexibility of parts being assembled, variability of fixturing and tooling systems, non-linear stack-up conditions. G is, in general, a non-linear function.

During last years, researchers have addressed their attention on proposing methods to evaluate G . One of the most relevant framework to do tolerance analysis is the Stream-of-Variation (SOVA) methodology. SOVA is a general methodology to model and analyze into a systematic way complex rigid and compliant assemblies (Hu, 1998; Huang, 2004b; Huang, 2007a; Huang, 2007b; Huang, 2009; Ceglarek, 2009). The gen-

eral framework includes: (I) Statistical Modal Analysis (SMA) methodology, which establishes a mathematical method to represent free shape form variability of a geometrical feature; (II) mechanistic models for stream of variation analysis both for rigid and compliant assembly; and, (III) computational algorithms for statistical tolerancing, needed to overcome computational effort of Monte Carlo-based simulations.

2.2 Variational Feature Modeling

The allowable deviations from the nominal geometry and dimensions are specified during the design stage by defining tolerances on the functional features using International Standards, (ISO, 2004) or (ASME, 1994)¹.

Traditionally, variational features dealt with parametric models to be included into CAD modeler (Requicha, 1986; Gupta, 1991; Chase, 1991): tolerances were viewed as a set of parameters. Strictly speaking, variations were introduced into variational CAD modelers (based on Constructive Solid Geometry (CSG) and feature-based approaches) to parameterize the whole geometry.

The actual trend is to incorporate International Standard specifications into mathematical models able to simulate variational features (Pasupathy, 2003; Loose, 2009; Kong, 2009).

Generally speaking, tolerance specifications may be classified into two main groups: tolerances requiring a *datum reference frame*, and tolerances of form which *do not* require any datum reference frame (Meadows, 1997). Typically, for each tolerance specification a 3D tolerance zone is built. This zone defines the 3D region within which any feature may vary.

2.2.1 TTRS Classification

A very powerful methodology was developed by the researchers at SUPMECA (Institut Supérieur de Mécanique de Paris, Paris-France). Any feature and the related constraints, for a specific datum reference frame (Clément, 1998), may be accounted reducing the analyzed feature to a combination of elementary geometrical entities: point, line and plane. In this way, the so-called Minimum Geometrical Reference Element (MGRE) is uniquely defined for each type of feature, allowing to define its positioning into the 3D space.

Seven key features were then proposed: planar, cylindrical, revolution, spheri-

¹ It should be noted that there is no specific Standard referring to tolerance design for assemblies.

Chapter 2. Review of Previous Works

cal, prismatic, helicoidal and complex feature (Desrochers, 1999). This general classification, called Topologically and Technologically Related Surfaces (TTRS), provides a methodology to manage complex parts by composing their simple elements while assuring the coherence of geometrical tolerancing methods (Chiabert, 2004).

One of the main applications of TTRS Theory is the mathematical definition of 3D tolerance zones, established by geometrical tolerances. The main idea is that the position of each feature (or each MGRE) may be represented by means of small rigid body displacements. In order to simulate and manipulate these small displacements the *torsor* operator² (Bourdet, 1988) was adopted.

Starting from the TTRS Theory, Whitney (Whitney, 1994) suggested to model GD&T specifications introducing the concept of 4x4 variational matrix. By using a notation well-known in Robotics, he derived the 4x4 variational matrix under the hypothesis of small rotational and translational displacements. What happens at the assembly level by modifying the nominal 4x4 matrices and introducing the variational matrices is so investigated.

TTRS Theory became a reference for the ISO/TC-213 project, in the frame of the ISO-GPS Standard. This methodology was then successfully implemented into CATIA® v5 CAD system to manage assembly constraints and tolerance annotations.

TTRS Theory provides a valid representation and classification of functional features, but it does not allow to model free shape errors. In (Choley, 2007) this issue is managed by introducing the concept of feature association: the real feature, with shape, orientation and position errors, is associated to an *ideal* feature, parameterized with small rotational and translational parameters.

2.2.2 Geometric Covariance

Despite many efforts made into the field of variational modeling, one need is the representation of free shape errors. Free shape errors are often neglected. However, for certain assembly processes, such as those involving sheet metal parts, these kinds of errors may strongly affect final assembly deviations.

In the literature, free shape errors are treated by introducing the *geometric covariance* concept. The geometric covariance states the geometrical relation among the neighboring points on the same surface. Geometric covariance assures surface continuity and smoothness.

Researchers at Brigham Young University focused on this topic developing dif-

² Strictly speaking, the *torsor* operator is a screw operator (see Chapter 4), defined as $[\mathbf{tr}, \mathbf{rt}]$, where \mathbf{tr} and \mathbf{rt} are the column vectors of translational and rotational parameters, respectively.

Chapter 2. Review of Previous Works

ferent methodologies. Merkley (Merkley, 1998) proposed to use bounded random Bézier curves to model and numerically evaluate the covariance matrix, given the tolerance error. Shape deviations were parameterized by constraining the displacement of the control points of Bézier curves.

$$\Sigma = \sigma^2 \cdot (\mathbf{A}^T \cdot \mathbf{A})^{-1} \quad (2.2)$$

In this way, he derived the covariance matrix, stated into equation (2.2), by using a least squares approach. σ is the standard deviation associated to the tolerance error³, while \mathbf{A} is a rectangular matrix related to Bernstein polynomials, defining the Bezier curve. Σ is the covariance matrix.

This method may be also extended to rectangular Bezier patches. However, for complex shapes the parameterization of the patch becomes not a trivial task, so this method may be inadequate for real implementation.

Bihlmaier (Bihlmaier, 1999) extended the Merkley's work, and proposed to model the variations of a surface as a finite summation of sinusoidal waves, each with a different amplitude and wavelength. In this way, any surface profile is modeled as a summation of sinusoids, having different wavelengths and amplitudes, represented in the frequency domain using the Fourier transform.

A more general method was proposed in (Tonks, 2002). To account for the surface variation, a hybrid method was used to model the surface covariance. Legendre polynomials were used to model the long wavelengths, and the frequency spectrum was used to model the shorter wavelengths. The hybrid method for geometric covariance was validated by means of experimental data.

Starting from the Merkley's work, Camelio (Camelio, 2002; Camelio, 2004a) discussed the effect of geometric covariance in the calculation of assembly variation of compliant parts. He combined the use of statistical Principal Component Analysis (PCA) and FEA in estimating the effect of part/component variation on assembly variation. PCA was used to extract deformation patterns from production data, decomposing the component covariance into the individual contributions of these deformation patterns. FEA was adopted to determine the effect of each deformation pattern over the assembly variation.

Over the years, authors have also developed the main idea of modal error decomposition to model the geometric covariance.

Srinivasan (Srinivasan, 1997) introduced a mathematical association between free shape errors and fractals. He used fractal and wave theory (typically applied into signal processing), to reconstruct the principal deformation mode of any geometrical feature. He tested his theory on real case studies.

³ Typically, for normal assembly processes, one may assume $T=6\cdot\sigma$, where T is the tolerance error.

Chapter 2. Review of Previous Works

Huang (Huang , 2002; Huang, 2004a) suggested to decompose the shape deviation by using the DCT (Direct Cosine Transform) technique (DCT is typically used in image processing for filtering image purpose). The field of variation was divided into a set of independent defects. Huang defined the Statistical Modal Analysis as a general methodology to generate variational features starting from measurement data.

Similar to DCT decomposition, another interesting approach is in (Samper, 2007), where the modal decomposition analysis is presented. Firstly, the nominal CAD feature is meshed. By using an analogy with a mass-spring system, the following differential equation is solved for:

$$\mathbf{M} \cdot \ddot{\mathbf{u}} + \mathbf{K} \cdot \mathbf{u} = 0 \quad (2.3)$$

which expresses the dynamic equilibrium for a conservative mass-spring system (\mathbf{M} and \mathbf{K} are the squared mass and stiffness matrices, respectively). Solution to equation (2.3) is achieved evaluating eigenvectors and eigenvalues related to the matrix $\mathbf{K}^{-1} \cdot \mathbf{M}$. It may be shown that eigenvectors corresponding to different eigenvalues are linearly independent. In this way, the orthogonal modal matrix (in which each column contains the related eigenvector) is built. This matrix is then used to extract the principal modal shapes. To reach more accuracy, during the modal decomposition a large amount of measurement data is needed. Moreover, a linear modal solver is required to calculate the orthogonal modal matrix.

A similar method was proposed also in (Ungemach, 2009), where the first eigenvector mode, resulting from a preliminary buckling analysis, was adopted to generate the initial variational geometry to be used in an assembly process simulation.

Both modal decomposition and buckling eigenvector analyses have no physical significance. However, they give a good approximation of the geometrical covariance.

In summary, in the field of variational feature modeling, variational features may be modeled by combining rigid body motion and modal decomposition to account location, position and free shape errors. However, all proposed methods requires input measurement data. This means that variational matrix or covariance matrix may be constructed after parts have been produced and sampled, but it could not be done during designing phases.

2.3 Tolerance Analysis for Rigid Part Assemblies

Traditionally, tolerance analysis has been performed under the assumption of ideal rigid parts. Assembly deviations are so evaluated accounting variational features and constraints among the same parts.

Chapter 2. Review of Previous Works

Methodologies proposed in literature to accomplish tolerance analysis of rigid parts may be distinguished into (I) one-dimensional tolerance charts, (II) vector loop-based tolerance analysis, (III) 4x4 chain transformation matrix, and (IV) tolerance domain analysis, as suggested also in (Shen, 2005), where a wide survey of the current computer based methods to capture tolerance zones is presented.

2.3.1 One-dimensional Tolerance Chart

One-dimensional tolerance chart is a manual procedure for 1D stack-up problems, typically used during the early design stage to understand the overall behavior of the assembly (Chase, 1991).

2.3.2 Vector Loop-based Tolerance Analysis

The vector loop-based tolerance analysis method is a well documented approach developed by researchers at Brigham Young University (Chase, 1995a; Chase, 1995b; Chase, 1996a; Chase, 1996b; Chase, 1997). Three types of variation are considered: dimensional, kinematic and geometric variations. In a vector loop representation, dimensions are represented by vectors (the magnitude of each vector corresponds to the length of each dimensional variable). Kinematic variations are small adjustments between joints features, which occur at the assembly level in response to the dimensional and geometric variations. Geometric tolerances are considered by adding small degrees of freedom to specific joints. The methodology may be summarized in three main steps.

First of all, one should create the assembly vector loops. The whole assembly is modeled with a graph representation, in which each edge correspond to a joining feature, while each vertex is a part being assembled.

Then, equations are written for each independent loop. Assembly constraints for each vector loop may be expressed as a concatenation of homogeneous rigid body transformation matrices, which results in a set of non-linear equations. These equations are linearized using Taylor's series expansion (Direct Linearization Method - DLM).

Finally, loop equations are solved for. The first order Taylor's series expansion of the loop equations can be expressed into a matrix notation: the unknowns kinematic variables are here calculated. Then, the basic formulation of the DLM is:

$$\mathbf{A} \cdot \Delta x + \mathbf{F} \cdot \Delta \alpha + \mathbf{B} \cdot \Delta u = 0 \quad (2.4)$$

where \mathbf{A} , \mathbf{F} and \mathbf{B} are the Jacobian matrices related to dimensional variation, Δx , geometrical variation, $\Delta \alpha$, and kinematic variation, Δu , respectively.

Chapter 2. Review of Previous Works

Equation (2.4) may be solved for with respect to Δu , as into equation (2.5). \mathbf{S}_x and \mathbf{S}_α are the sensitivity matrices due to dimensional and geometrical variations. From equation (2.5) it should be clear that once dimensional and geometrical variations are defined, the kinematic variable is univocally determined since \mathbf{S}_x and \mathbf{S}_α are constant.

$$\begin{cases} \Delta u = \mathbf{S}_x \cdot \Delta x + \mathbf{S}_\alpha \cdot \Delta \alpha \\ \mathbf{S}_x = -\mathbf{B}^{-1} \cdot \mathbf{A} \\ \mathbf{S}_\alpha = -\mathbf{B}^{-1} \cdot \mathbf{F} \end{cases} \quad (2.5)$$

The DLM procedure allows to solve into a closed form any mechanical assembly for a given set of tolerances: no Monte Carlo simulation is strictly required. However, 3D tolerance zones are not fully-integrated. In addition, it does not allow to simulate different assembly sequences: assembly constraints among mating features are modeled through *equivalent* joints, which do not consider the real behaviour at feature-to-feature interface.

Old versions of CETOL 6 σ [®] were based on the DLM approach. Due to some of the above limitations, Sigmetrix has apparently switched to a parametric approach, integrating also Screw Theory.

2.3.3 4x4 Chain Transformation Matrix

Whitney (Whitney, 1999; Whitney, 2004) adopted 4x4 homogeneous transformation matrices to model variational assemblies. The whole assembly is described as a chain of coordinate frames, attached to functional features, designed to deliver a specific KC.

For a nominal assembly, the following 4x4 equation may be written:

$$\mathbf{T}_{1,n} = \mathbf{T}_{1,2} \cdot \mathbf{T}_{2,3} \cdot \dots \cdot \mathbf{T}_{i-1,i} \cdot \dots \cdot \mathbf{T}_{n-1,n} \quad (2.6)$$

where $\mathbf{T}_{i,j}$ is the transformation matrix defining the coordinate frame j into i one. In summary, equation (2.6) allows to refer the coordinate frame, attached to the feature n , to one attached to feature 1. Assembly constraints and feature-to-datum relationships are captured in this formulation.

Equation (2.6) may be modified into equation (2.7) when tolerance deviations are considered. $\mathbf{DT}_{i-1,i}$ is the variational 4x4 transformation matrix, derived in the small displacement hypothesis. The triplets (α, β, γ) and $(\Delta x, \Delta y, \Delta z)$ represent the small rotational and translational variables, respectively.

$$\mathbf{T}_{1,n'} = \mathbf{T}_{1,1'} \cdot \mathbf{DT}_{1',2} \cdot \dots \cdot \mathbf{T}_{i-1,i-1'} \cdot \mathbf{DT}_{i-1',i} \cdot \dots \cdot \mathbf{T}_{n-1,n-1'} \cdot \mathbf{DT}_{n-1',n}$$

$$\mathbf{DT}_{i-1',i} = \begin{bmatrix} 1 & -\gamma & \beta & \Delta x \\ \gamma & 1 & -\alpha & \Delta y \\ -\beta & \alpha & 1 & \Delta z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.7)$$

Multivariate joint probability density functions and direct Monte Carlo simulations are then used to numerically perform tolerance analysis of assemblies.

The Whitney's method allows to parameterize the whole assembly as a set of variational features. However, it does not provide an efficient solution to handle closed loop assemblies. In addition, this method does not allow to take into account different assembly sequences.

2.3.4 Tolerance Domain Analysis

The tolerance domain-based method (Giordano, 2005a; Giordano, 2007) was developed by researchers at University of Savoie (Annecy-France).

The main idea is to model any tolerance, related to a specific functional feature, and any mating joint by means of the small displacement torsor. Then, considering that any point of the feature must be inside the related tolerance zone, the components of the torsor operator are constrained each-other: in this phase inequalities are derived. These inequalities describes, into the 6D space⁴, the so-called hyper-domain.

Once hyper-domains are derived for any feature and for any mating joint, they are combined to obtain the assembly hyper-domain. To numerically evaluate it, authors suggested to use the Minkowski sum operator⁵. However, the Minkowski sums are very complex to be numerically evaluated.

Finally, the assembly hyper-domain is compared with the functional hyper-domain, corresponding to a specific KC: if the assembly hyper-domain is completely included inside the functional hyper-domain, then that KC is satisfied.

In an independent way, at Arizona State University (USA), researchers (Jian, 2005; Shah, 2007) developed the Tolerance-Map (T-Map) model which reflects the hyper-domain approach. Generally speaking, T-Map is a hypothetical euclidean space, whose size and shape reflect all variational possibilities for a feature. A tolerance analysis method utilizing T-Map is based on two phases. First of all, part variations, consider-

⁴ It should be noted that the torsor operator is made of 6 components.

⁵ In geometry, the Minkowski sum of two sets A and B, defined into Euclidean space, is the result of adding every element of A to every element of B.

Chapter 2. Review of Previous Works

ing the interaction of all geometric deviations, are modeled. Then, deviations on a part or assembly are correlated. T-Map model satisfies all the requirements embedded in the ASME Standard: it incorporates the different classes of geometric tolerances, Rule #1, floating zones, bonus tolerances, and datum precedence assignment.

In summary, in the field of tolerance analysis of rigid assemblies, mechanistic accumulation models are mainly based on non-linear hypotheses. To make easier their managing, simplified linear approach are often introduced. This assumption may be valid under the small displacement hypothesis. However, two important issues need to be still investigated. First of all, variational feature models should be fully-integrated into models for tolerance accumulation. Then, 3D joining conditions among parts being assembled should be investigated to reach numerical results closer as much as possible to real assembly processes. All this allows to analyze and simulate the behavior of different assembly sequences.

2.4 Tolerance Analysis for Compliant Part Assemblies

Mechanistic models for rigid assemblies may be inadequate when parts exhibit a high deformation. This is particularly true for products of some industries, such as automotive and aerospace industries, where sheet-metal parts are very common. The high flexibility of such parts may cause wide shape variations during assembly process combined with tolerances on parts and fixtures, so causing uncertainty to predicting the real shape configuration of the final released assembly.

To analyze compliant part assemblies researchers have proposed, during last ten years, interesting methodologies, combining mechanistic models for rigid part assemblies and FE approaches (Caputo, 2006). Generally speaking, equation (2.1) is solved for including part deformation.

2.4.1 PCFR Cycle

Chang (Chang, 1997) proposed a methodology based on the so-called Place, Clamp, Fasten and Release (PCFR) cycle. In real assembly processes involving deformable components, parts are firstly placed onto the fixturing frame, then clamped, next fastened, and finally released. He calculated the final assembly deviation by combining elastic constitutive relations, geometric compatibility and force continuity. Constitutive relations are, for example, $\mathbf{F}=\mathbf{K}\cdot\mathbf{v}$, where \mathbf{F} is the general force vector, \mathbf{v} is the displacement vector, and \mathbf{K} the squared stiffness matrix. Chang used an assembly graph-

based representation to analyze how errors accumulate. Linearity, rigid fixture, and no friction were assumed in his model. Linearity means that the constitutive equation is linear (\mathbf{K} matrix is constant with respect to forces and displacements).

2.4.2 Method of Influence Coefficients

The milestone in the field of tolerance analysis of compliant assemblies is the methodology proposed by Liu and Hu (Liu, 1997) and based on the concept of the sensitivity matrix.

Starting from the PCFR assembly cycle, Liu and Hu suggested a new methodology to simulate compliant part assemblies. Instead of using direct Monte Carlo simulation, that requires many thousands of FEA runs for every random part shape configuration within the tolerance specifications, they provided a method based on the concept of influence coefficients. The method of influence coefficient establishes, under linear hypotheses, a relationship between part deviations and assembly spring-back deviations. Only two FE runs are needed: one for the unit force responses of parts (from which the matrix of influence coefficients is derived), the other for the spring-back deviation of the assembly (from which the sensitivity matrix is derived). In this way the computational effort was strongly reduced.

The suggested methodology may be summarized as follows. When parts are put together, the gap existing between mating features must be closed by applying tooling forces. The linear relationship between force and displacement may be written as:

$$\mathbf{F}_u = \mathbf{K}_u \cdot \mathbf{v} \quad (2.8)$$

where \mathbf{K}_u and \mathbf{v} are the stiffness matrix and the deviation vector of parts before assembling (unfastened parts), respectively. \mathbf{F}_u is the general force vector supported by tooling guns. Once parts are fastened, the elastic constitutive equation becomes:

$$\mathbf{F}_f = \mathbf{K}_f \cdot \mathbf{u} \quad (2.9)$$

where \mathbf{K}_f and \mathbf{u} are the stiffness matrix and the deviation vector at assembly level (fastened parts), respectively. \mathbf{F}_f is the equivalent vector force related to spring-back phenomena.

Liu and Hu proposed, under the small displacement hypothesis, that $\mathbf{F}_u = \mathbf{F}_f$. Then, it is:

$$\begin{aligned} \mathbf{F}_f = \mathbf{F}_u &\rightarrow \mathbf{K}_u \cdot \mathbf{v} = \mathbf{K}_f \cdot \mathbf{u} \rightarrow \mathbf{u} = \mathbf{K}_f^{-1} \cdot \mathbf{K}_u \cdot \mathbf{v} \\ \mathbf{u} &= \mathbf{S} \cdot \mathbf{v} \end{aligned} \quad (2.10)$$

where \mathbf{S} is the *sensitivity matrix*, linking part deviations, \mathbf{v} , and assembly spring-back

Chapter 2. Review of Previous Works

deviations, \mathbf{u} .

The sensitivity matrix may be calculated once the stiffness matrices, \mathbf{K}_f and \mathbf{K}_u , are available. However, Liu and Hu noted that commercial FEA software often do not make available such as matrices. Therefore, he suggested a general procedure to calculate the \mathbf{S} matrix, which may implemented in any FEA solver, by using two consecutive FEA runs.

Once the FE model of nominal parts has been created, in the first FEA run, unit forces are applied at deviation points. The matrix of influence coefficients, \mathbf{C} , linking the unit force at j -th source of variation to the deviation at the i -th point is then derived as in equation (2.11).

$$\mathbf{v} = \mathbf{C} \cdot \mathbf{F}_u \quad (2.11)$$

The inverse of \mathbf{C} gives the stiffness matrix \mathbf{K}_u . Entries into the \mathbf{K}_u matrix are the tooling forces related to input deviations, \mathbf{v} . These forces are then applied to the assembled parts (Liu and Hu used MPC - Multi Point Constraint to joint parts) in the second FEA run. Displacements, calculated in the second FEA run, correspond to the entries of the sensitivity matrix.

The Liu and Hu's method is valid under the hypothesis of independent sources of variation. If dependences exist, a covariance matrix must be introduced but they did not specify it. Moreover, the suggested method is valid only for single-station assembly processes.

The last release of VisVSA® offers an integrated FEA module which apparently implements this method.

Starting from the same hypotheses of Liu and Hu, Sellem *et al.* (Sellem, 1998; Sellem, 1999; Sellem, 2001) developed a method accounting two of the most relevant source of variability: positioning and geometric deviation of parts being assembled. In addition, contacts among mating feature were assumed as *not* significant.

The theoretical basis for the Sellem *et al.*'s method is represented by the so-called *influence matrix* (similar to the sensitivity matrix), which represents how the assembly deforms when a unit displacement is applied in a specific point. This method was then implemented into TAA® module, integrated in CATIA®⁶ v5 CAD system.

By using TAA®, Sellem *et al.* tested the theoretical model on real sheet-metal part assemblies. Numerical results were well-correlated to experimental data. Their model allows to simulate multi-station assembly processes: the PCFR cycle is here adopted.

⁶ CATIA® is a trademark of Dassault Systemes

2.4.3 Flexible Beam Elements

In (Shiu, 1997) an interesting simplified model of the assembly structure for dimensional control is described. The original structure is subdivided in beam elements according to some decomposition rules, strictly related to the assembly process (*simplified* 1D beams must be derived for 3D sheet-metal panels). Other principles are described to connect those elements by beam-to-beam joints and to locate parts/subassemblies to each other. Based on this model a tolerance analysis study can be performed based on the knowledge of the statistical distribution of the part and assembly tooling variability.

This method well-works for simple structures, but it is quite hard to adopt for complex assemblies. Anyway, it allows to understand the overall behavior of the assembly process very quickly.

2.4.4 FASTA Framework

Researchers at Brigham Young University extended the DLM approach to flexible assembly processes. A general overview of the Flexible Assembly Statistical Tolerance Analysis (FASTA) approach is in (Mortensen, 2002).

The FASTA framework may be summarized as follows. Starting point is the nominal CAD model of parts. Measurements from real assemblies are collected to get statistical data. Also rigid body translations and rotations may be statistically evaluated as they add variability to the assembly geometry (see DLM approach). All these items statistically characterize the input deviations at part level. Parts are considered rigid during this first step. In addition, FASTA focuses only on the gap at mating interface. Then, by using a FEA approach, the stiffness matrix of the assembly describing equilibrium conditions of the assembly gap is calculated (to speed-up the calculation the super-element method is here adopted). Finally, combining FEA results and statistical data (in terms of mean and covariance) of the input deviations, the final assembly shape may be described in terms of mean and covariance.

One of the main limitations of the FASTA method is the assumption that the final state of deformation is independent of fixtures, as proposed in Stewart (Stewart, 2004). Moreover, FASTA may be applied only for single-station assembly processes and contacts at mating features are not considered.

2.4.5 Multi-station Assembly Modeling

Variation modeling and analysis for multi-station manufacturing processes has been developed mainly for rigid parts. However, in real industrial applications a large group of multi-station assembly processes consider non-rigid parts. For example, 37% of all assembly stations in automotive body structure manufacturing involve non-rigid parts (Ceglarek, 2009). Variation propagation analysis for a multi-station assembly process introduces new modeling challenges. In comparison to the single-station approach, it is necessary to define an appropriate variation representation in order to account the variation propagation station-to-station.

Table 2.1 summarizes some of recent publications involving single- and multi-station modeling both for rigid and compliant parts.

	Rigid Parts	Compliant Parts
Single-Station Level	(Chase, 1991) (Chase, 1995a)	(Shiu, 1996) (Liu, 1997) (Long, 1998) (Cai, 2006)
Multi-Station Level	(Shiu, 1996) (Mantripragada, 1999) (Jin, 1999)	(Camelio, 2002)

Table 2.1: recent publications in variation propagation

Comparatively, few researches have been done in multi-station systems considering compliant, non-rigid parts. Generally speaking, assembly processes of compliant parts are based on the following steps: a part or sub-assembly is released from a generic station and then it is re-positioned into the next one. Here it is clamped and fastened to new parts or sub-assemblies.

Camelio *et al.* (Camelio, 2002; Camelio, 2003; Camelio, 2004b) extended the Liu and Hu's method, based on the sensitivity matrix, to multi-station assembly processes. They accounted part variation, fixturing variations and tooling guns variations. Moreover, starting from the concept of *state space vector*, originally introduced in (Mantripragada, 1999) for rigid part assemblies, they considered the variation propagation process as a linear-time varying discrete time system, where the variable time represents the generic station location. In this way, they defined a sensitivity matrix for each step of the state vector (Figure 2.1) by using a FEM approach.

Originally, the state vector model was based on linear assumptions and all input variable were assumed independent each-other. In (Camelio, 2004a) the covariance matrix was introduced to consider the dependency between deviation variables. Starting

from production data and adopting the principal component analysis, deformation patterns were extracted and used for the multi-station assembly process simulation.

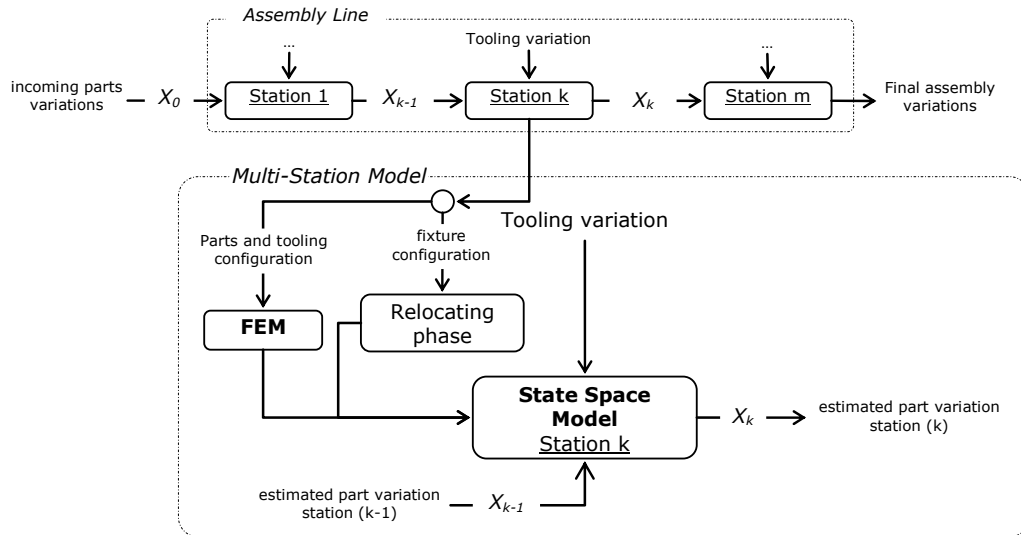


Figure 2.1: general work-flow of the state space model (Camelio, 2002)

2.4.6 Including Non-Linear Effects

Since its origin the tolerance analysis of compliant assemblies has been based on linear assumption. However, linear approaches do not provide adequate results when large deformations occur, or part-to-part contact conditions have to be taken into account. (Liao, 2007) and (Xie, 2007) showed how contact among parts being assembled highly influences final assembly shape. They proposed to use a non-linear FEA approach to solve the contact problem. Their methods are more accurate than a linear one, but they are very time consuming especially if combined with Monte Carlo-based simulations. To overcome this lack, an interesting linear contact algorithm was proposed both in (Dalhlström, 2007) and (Ungemach, 2009), by combining a linear contact search and a contact equilibrium criterion. Their approaches were integrated with the method of influence coefficient proposed by Liu and Hu.

In summary, in the field of tolerance analysis of compliant assemblies, mechanistic accumulation models are mainly based on linear hypotheses. All this allows to quickly analyze the problem and to evaluate the influence of each source of variation.

How to model multi-station assembly processes is still a need, which should be investigated. Moreover, to avoid part-to-part penetration contacts should be integrated within the propagation models.

2.5 Tolerance Analysis and Simulation Tools

Tolerance analysis aims to numerically evaluate the amount of deviations when parts are put together. Typically, two main approaches are here adopted: worst case simulation and statistical simulation. In worst case approach each source of variation is assumed at its limits and, then, final assembly result gives information only on the maximum and minimum deviations. This kind of analysis is usually called *deterministic tolerance analysis*. On the other hand, *statistical tolerance analysis* aims to evaluate the amount of deviation, modeling each source of deviation as a random statistical variable (such as normal, uniform and so on).

To perform a deterministic tolerance analysis few runs⁷ are required: each tolerance is assumed one time at its maximum and then at its minimum. Statistical tolerance analysis, instead, requires a more expensive simulation. Typically a Monte Carlo approach is used to randomly generate each source of variation. How to improve statistical tolerance analysis performances is still investigated in literature. For example, in (Huang, 2004a) an innovative and robust method, called Number-Theoretical net method (NT-net), is proposed and tested.

Many commercial CAT packages, as discussed in Chapter 1, are available today to do tolerance analysis of rigid part assemblies. All those CAT packages allow to do both worst case and statistical analyses. On the other hand, only three CAT tools to accomplish tolerance analysis of compliant part assemblies are available: TAA® (by Dassault Systemes), 3DCS-FEA® (by Dimensional Systems Inc.), VisVSA-FEA® (by UGS Co.).

⁷ Generally speaking, if there are n contributing dimensions, 2^n simulations are required.

2.6 Summary

The actual literature covers the following key topics: (I) variational feature modeling, (II) tolerance analysis for rigid part assemblies, (III) tolerance analysis for compliant part assemblies, and (IV) tolerance analysis and simulation tools. Several limits and weaknesses have been pointed out.

Relating to (I), a robust variational model is needed to generate variational features without using production data. Moreover, Standard specifications, such as geometric tolerances, Rule #1, floating zones, bonus tolerances, material modifiers and datum precedence assignment have to be included into variational procedures.

Relating to (II), assembly constraints must be investigated and correctly modeled. All this allows to analyze different assembly sequence.

Relating to (III), models for multi-station assembly process simulations need to be improved. Then, how contacts, defined at mating features, influence final assembly results should be verified.

Relating to (IV), an integrated tool, taking into account all above critical aspects, able to do tolerance analysis both of rigid and compliant parts is not available today.

Chapter 3.

GRAPH THEORY FOR TOLERANCING¹

In a design contest where tolerances have to be set to achieve crucial KCs, choosing the right tolerance specification scheme is not a trivial task, especially for industrial applications, where assemblies are made of many and many parts.

This Chapter focuses on the main concept of global consistency and describes a general methodology to automatically detect the global consistency for a given assembly. Proposed algorithms are based on Graph Theory. The assembly is modeled as a set of vertices and edges. Vertices are parts or features while edges are joining conditions among parts or datum assignments among features.

Two examples are illustrated and analyzed to show how the methodology works.

3.1 Introduction

Graph Theory is a very powerful tool used in engineering applications to model and manage real systems (West, 2001; Falgarone, 2006). In the tolerance mechanical field, Graph Theory is a support for the analysis and synthesis of dimensional and geometrical specification.

A very interesting application of *Graph Theory for Tolerancing* can be found in (Ballu, 1999; Giordano, 2005b) where a methodology, mainly based on “hyper-graphs”,

¹ This Chapter is based on: Franciosa P., Patalano S., Riviere A., *3D Tolerance Specification: an Approach for the Analysis of the Global Consistency based on Graphs*, Int. Journal on Interactive Design and Manufacturing, DOI: 10.1007/s12008-009-0067-7, 2009.

Chapter 3. Graph Theory for Tolerancing

is proposed. Here, the assembly is assumed a hyper-graph, where vertices are the sub-graphs related to parts. For each sub-graph, functional features are vertices, while datum assignments among features are edges.

An approach based on Graph Theory is also in (Whitney, 2004). Whitney defines the so-called Datum Flow Chain (DFC) as a directed *acyclic* graphical representation of an assembly, in which vertices represent the parts, while edges represent the mate conditions among them. The DFC is a general tool which may be used as support during the whole design stage.

(Kandikjan, 2003) proposed a graph-based method to completely check the validity of a given set of geometric tolerances, while (Clément, 1999) suggested a mathematical model, mainly based on the TTRS Theory, to check the global consistency of a set of dimensional and geometric specifications.

Recently, some tools for tolerance specifications and annotations have been implemented into parametric 3D CAD systems. FTA® module, available into the CAD system CATIA® V5, by Dassault Systemes, addresses the definition and the editing of tolerance specifications of 3D parts, based on the latest revisions of standards ISO-GPS (ISO, 2004; ISO, 2006) and ANSI-GD&T (ASME, 1994; ASME, 2004). This tool allows to automatically verify the semantic association between functional features. However, no specific tool is available today to check the global consistency of tolerance assignment sets.

Two key words are here introduced: *adequacy* and *consistency*. *Adequacy* allows to address parts that influence each KC of the assembly. In this way, it is possible to tackle the complexity of the assembly focusing the subset of parts that influences the specific KC. *Consistency* focuses on tolerance assignment accounting datum and features for a specific part, and assembly links among parts. Mistakes in tolerance specification may be detected when tolerance specifications do not fulfill geometric rules.

Starting from these key words, two kinds of graphs are considered: the *Assembly-based Graph* (AGr) and the *Part-based Graph* (PGr). The first one is used to model and analyze the assembly in a functional point of view. The second one describes the tolerance-based links among features of a same part.

3.2 Graph Definition and Manipulation

3.2.1 Adequacy Analysis

The aim of adequacy analysis is to detect parts that influence KCs. In this way it is possible to reduce the assembly graph, focusing the subset of parts that influences specific KCs. To perform this kind of analysis, the AGr is introduced.

Chapter 3. Graph Theory for Tolerancing

In the AGr, each vertex represents a part, while each edge represents the *assembly link* between two parts. Generally speaking, each edge may describe multi-joint conditions. For example, the j -th part may be linked to the i -th part by means of several joints, such as planar mating, axial alignment and so on. For the purpose of the present analysis multi-joint edges are reduced to single-edges – only the functional link between two parts is aimed to detect.

Three kinds of parts are here defined:

- *target part*: a part whose features are used to constrain another part;
- *object part*: a part assembled according to the features of a target part; and,
- *datum part*: a part having no target part.

The AGr is assumed as a directed graph: the direction of each edge specifies which part (object part) is located by means of another part (target part). For example, in Figure 3.1, the part 2 is located with respect to the part 1; so a directed edge from part 1 to part 2 is drawn. Moreover, part 1 is a datum part because it has no target parts.

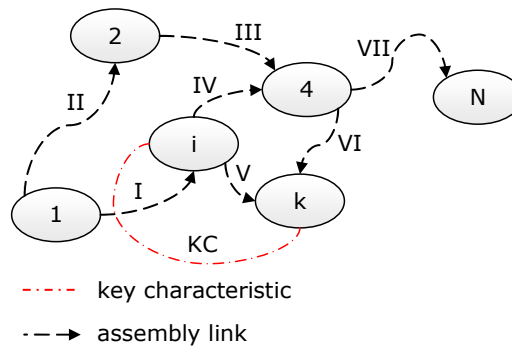


Figure 3.1: Assembly-based Graph

To easily evaluate all datum parts, one should find out those rows in the graph incidence matrix, \mathbf{I} (see Annex D for more details on graph matrices), having all entries greater or equal to zero. Indices of those rows correspond to datum parts.

Let (V_i, V_j) be the KC connecting vertices V_i and V_j . For each vertex defining the KC, the *target chain* is introduced and it is defined as the path connecting that vertex with any datum vertex. Therefore, looking at Figure 3.1, with respect to the couple (i, k) - assumed as KC -, four target chains are counted: $[I]$ for the vertex i and $[I V]$, $[II, III, VI]$, $[I, IV, VI]$ for the vertex k . It is clear that part N does not belong to any target chains, so it may be removed from the AGr.

Below the pseudo-code to automatically detect target chains inside the AGr.

```

%- find all paths connecting vertex Vi to Vj into AGr
function [pathedge,allpath,flag]=paths2graph(I,E,Vi,Vj,edgein,...
                                             pathedge,allpath,flag)

%- I: incidence matrix (for oriented graph)
%- E: edge matrix
  
```

Chapter 3. Graph Theory for Tolerancing

```
%- Vi: start vertex
%- Vj: end vertex
%- edgein: list of connected edges
%- pathedge: local path variable
%- allpath: list of all paths detected
%- flag: boolean variable for cyclic loops detection

%- check for cyclic loops
for k=1:length(pathedge)
    if edgein==pathedge(k);
        flag=false;
        return %- cyclic loop detected and return function
    end
end

%- update locale path
pathedge=[pathedge,edgein];

%- return path when Vi=Vj
if Vi==Vj
    allpath=[allpath,pathedge];
end

%- update start vertex and edgein
Viupdate=[];
edgein=[];
for j=1:size(I,2) %-count all vertices
    if I(Vi,j)==-1 %- Vi is object with respect to edge j
        edgein=[edgein,j];
        Viupdate=[Viupdate,E(j,1)]; %- E(j,1) is the target vertex
    end
end

%- call function in a recursive way
for i=1:length(edgein)
    [tempe,allpath,flag]=paths2graph(I,E,Viupdate(i),Vj,edgein(i),...
        pathedge,allpath,flag);
    if flag==false
        return
    end
end
```

A recursive procedure is used. During each step, all vertices connected to V_i are evaluated. Here, V_i is assumed as a list of indices recursively updated. The first entry of this list is the starting vertex of the path (to do it a search within the incidence matrix, I , is performed). Then, the V_i index is updated in a recursively way and the path is returned when the first entry in the V_i index becomes equal to V_j . The algorithm is stopped when all paths connecting V_i to V_j are detected.

It should be noted that the proposed algorithm allows to detect also cyclic loops inside the AGr. A cyclic loop is defined as an oriented path connecting, cyclically, a vertex with itself.

Here, the *target chain* matrix, TC , ($1 \times N_e$ row vector - where N_e is the number of edges counted into the AGr) is introduced. This matrix is defined as follows:

$$\begin{cases} \mathbf{TC}(1,i) = 1 \rightarrow i \text{ belongs to target chain} \\ 0 \rightarrow \text{otherwise} \end{cases} \quad (3.1)$$

$$\forall i = 1, 2, \dots, N_e$$

For example, looking at Figure 3.1, the target chain matrix related to the KC linking vertex i to vertex k is $\mathbf{TC}=[1 \ 1 \ 1 \ 1 \ 1 \ 0]$.

Once all target chains are detected, the *assembly functional chain (AFC)* may be calculated. **AFC** is union of the single target chain matrices, as stated in the relationship (3.2).

$$\begin{cases} \mathbf{TC}_1 \\ \mathbf{TC}_2 \\ \dots \\ \mathbf{TC}_{N_{KC}} \end{cases} \rightarrow \mathbf{AFC} = \bigcup_{i=1}^{N_{KC}} \mathbf{TC}_i \quad (3.2)$$

$$\begin{cases} \mathbf{AFC}(i,1) = 1 \rightarrow i \text{ belongs to AFC} \\ 0 \rightarrow \text{otherwise} \end{cases}$$

$$\forall i = 1, 2, \dots, N_e$$

Finally, the adequacy at assembly level aims to:

- *validate* the AGr: cyclic loops have to be avoided; and,
- *reduce* the AGr: all parts, which do not belong to the **AFC**, can be removed.

Next, same examples will show how the proposed procedure works.

3.2.2 Consistency Analysis

The representation of geometric tolerances is based on the concept of *Datum Reference Frame (DRF)*. Geometric tolerances related to a DRF are tolerances of positions, orientation, run-out, and, sometimes, profile. Tolerances of form (such as flatness, circularity and cylindricity) and profile do not require any DRF (Meadows, 1997). The DRF can be defined by using a combination of datum features, which are theoretically exact geometry, such as point, line and plane (Clément, 1998). The mathematical representation of DRFs is addressed by means of a point, defining the origin of the frame and three unit vectors, orthogonal among them, defining the orientation of the same frame, with respect to an another one (see Annex A).

Generally speaking, functional features of a part come in contact with the functional faces of the mating parts or are used for mating conditions against fixtures, during manufacturing and inspection phases. The *tolerance-based links* among features are represented into a graph, called *part-based graph*. The *part-based graph* is a directed

graph, such as the AGr, in which vertices are the features, while edges are the tolerance links among them. For example, in Figure 3.2, the feature F_{i-t} is related to the feature F_{i-s} by means of the tolerance specification T_{i-k} : in this case the feature F_{i-t} is classified as *object feature*. Instead, the feature F_{i-k} is a *datum feature*.

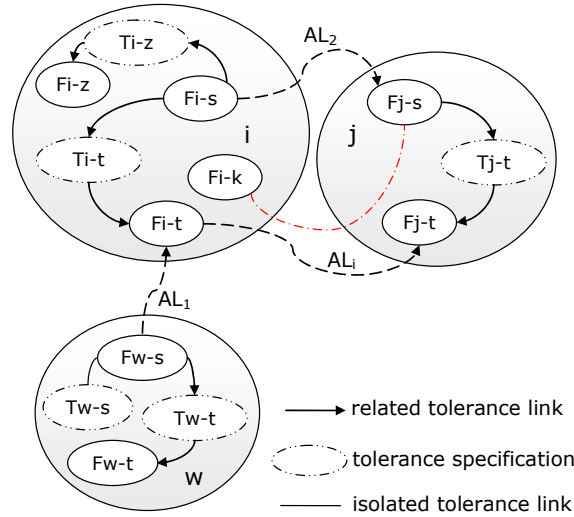


Figure 3.2: Exploded Assembly-based Graph

Therefore, similarly to AGr, three kinds of features are here defined:

- *target feature*: a feature used to relate an another feature;
- *object feature*: a feature related to a target feature; and,
- *datum feature*: a feature having no target features.

The complete representation of all part-based graphs gives the so-called *Exploded Assembly-based Graph*, EAGr (Figure 3.2). It can be pointed out that multi-joint conditions are now introduced since the direct functional or assembly link between features is now available. For example, in Figure 3.2 the part i has two assembly links with respect to part j : features F_{i-s} and F_{i-t} , for part i , and features F_{j-s} and F_{j-t} , for part j , achieve this.

$$\left\{ \begin{array}{l} KC_1 \\ \dots \\ KC_{N_{KC}} \\ \dots \\ AL_1 \\ \dots \\ AL_{N_{AL}} \end{array} \right. \rightarrow HKC = \bigcup \left(\bigcup_{i=1}^{N_{KC}} KC_i, \bigcup_{j=1}^{N_{AL}} AL_j \right) \quad (3.3)$$

Algorithms proposed for managing AGr are here again applied. In particular, the *Hyper-Key Characteristic* (HKC) is now introduced, as defined into relationship (3.3),

Chapter 3. Graph Theory for Tolerancing

where N_{AL} is the total number of assembly links.

Target chains defined above may be extended into *hyper-target chain*, **HTC**. One can write:

$$\begin{cases} \mathbf{HTC}(1,i) = 1 \rightarrow i \text{ belongs to hyper - target chain} \\ 0 \rightarrow \text{otherwise} \end{cases} \quad (3.4)$$

$$\forall i = 1, 2, \dots, N_{e,EAGr}$$

where $N_{e,EAGr}$ is the total number of edges counted into the EAGr.

Once all hyper-target chains are detected, the *exploded assembly functional chain* (**EAFC**) may be calculated. **EAFC** is union of the single hyper-target chains, as stated in the relationship (3.5).

$$\begin{cases} \mathbf{HTC}_1 \\ \mathbf{HTC}_2 \\ \dots \\ \mathbf{HTC}_{N_{HKC}} \end{cases} \rightarrow \mathbf{EAFC} = \bigcup_{i=1}^{N_{HKC}} \mathbf{HTC}_i \quad (3.5)$$

$$\begin{cases} \mathbf{EAFC}(i,1) = 1 \rightarrow i \text{ belongs to EAFC} \\ 0 \rightarrow \text{otherwise} \end{cases}$$

$$\forall i = 1, 2, \dots, N_{e,EAGr}$$

where N_{HKC} is the total number of HKCs.

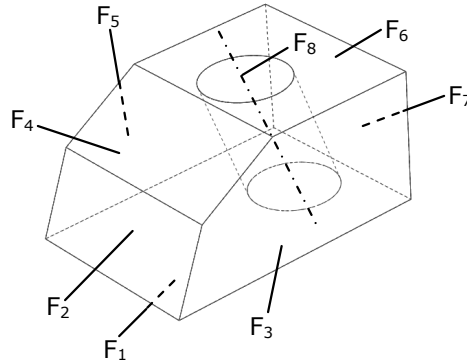


Figure 3.3: parametric CAD model to check tolerance specification mistakes

Finally, the consistency analysis aims to:

- *validate* the EAGr: cyclic loops have to be avoided;
- *reduce* the EAGr: all features, which do not belong to the **EAFC**, can be removed;
- and,
- *check mistakes* for tolerance specification set.

To check tolerance specification mistakes, the integration of specifications and

annotations into parametric CAD systems may be adopted. Geometric and dimensional constraints, used during the modeling phase of a preliminary CAD model, may be used to validate the geometric and dimensional tolerance specifications.

For example, looking at Figure 3.3, the face F_6 is parallel to F_1 , the face F_7 is perpendicular to F_1 , the line F_8 is parallel to F_3 and it is oblique with respect to F_1 .

Therefore, geometric rules can be derived in order to automatically detect tolerance specification mistakes, according to geometric and dimensional constraints available into CAD model. This strategy was implemented into FTA[®] module, integrated into CATIA[®] V5 CAD system.

3.3 Implementation

The discussed methodology was implemented into a GUI, developed in MatLAB^{®2} environment (Figure 3.4). Vertices (which may represent a part or a feature) of the graphs can be manually created, by picking any point from the *GRAPH MODEL* interface. Links among vertices can be added. Moreover, user can set and modify (*FLIP*) their orientation (during this phase the graph edge matrix, E , is updated). Assembly links or tolerance-based links can be created, where necessary.

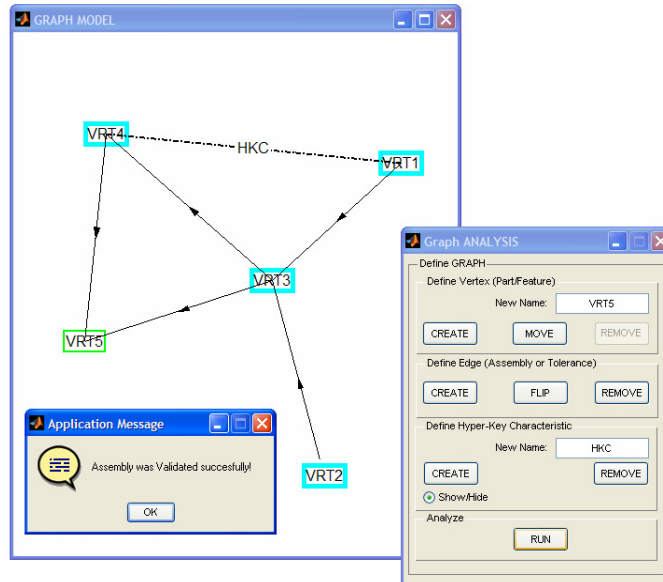


Figure 3.4: Graphical User Interface

Once hyper-key characteristics have been assigned among vertices, assembly adequacy or consistency is performed (*RUN*), by applying algorithms summarized in

² MatLAB[®] is a registered trademark of MathWorks.

Chapter 3. Graph Theory for Tolerancing

Section 3.2 . Parts (vertices) influencing each hyper-key characteristic are highlighted into the *GRAPH MODEL* interface.

3.4 Examples

3.4.1 Two-part Assembly

A two-part assembly has been analyzed. Figure 3.5 shows tolerance specifications and the EAGr. KC is the radial distance between pin and hole.

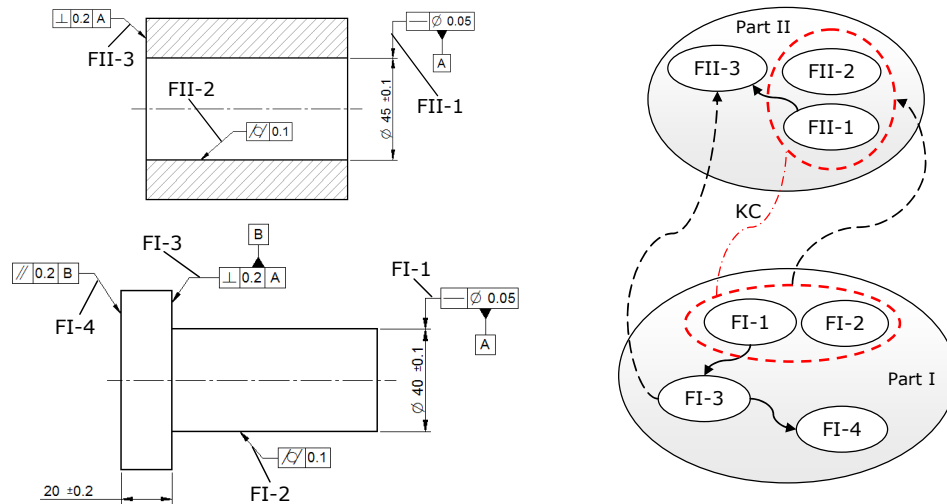


Figure 3.5: example. Tolerance specification and graph representation

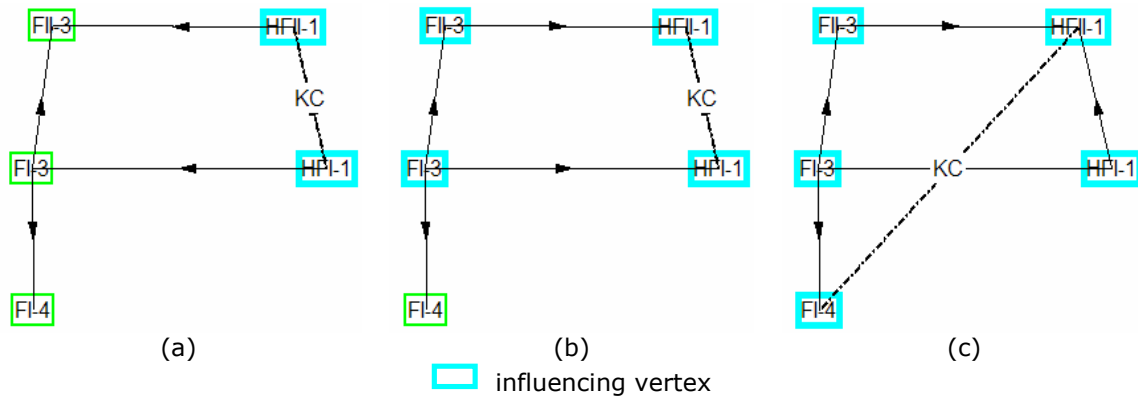


Figure 3.6: example. Analysis results

Two *hyper-features* (HFI-1 and HFII-1 in Figure 3.6) are here introduced. For instance, with respect to Part I (the same may be said for part II), the axis of the pin and the cylindrical surface of pin are geometrically dependent. Therefore, groups (FI-1 – FI-2) and (FII-1 – FII-2) are here introduced and called hyper-features. KC connects HFI-1 to HFII-1.

Simulations say, as expected, that the assembly graph and the tolerance links are correctly defined. Moreover, as depicted in Figure 3.6.a, only features HFI-1 and HFII-1

influence the specified KC. It is of interest changing the datum assignment. For instance, feature HFII-1 becomes object feature with respect to FII-3 one and feature HFI-1 is now object feature with respect to FI-3. Assuming the same KC of the previous analysis, simulation gives results shown in Figure 3.6.b.

Finally, Figure 3.6.c shows influencing parts, when the KC is the angle between feature FI-4 and HFII-1. In this case, all features are classified as influencing features.

3.4.2 Graph Managing

This example show how the proposed methodology allows to detect cyclic closed inside the assembly graph. In order to make the example more general, a generic assembly graph (Figure 3.7) is proposed, with no reference to a specific assembly.

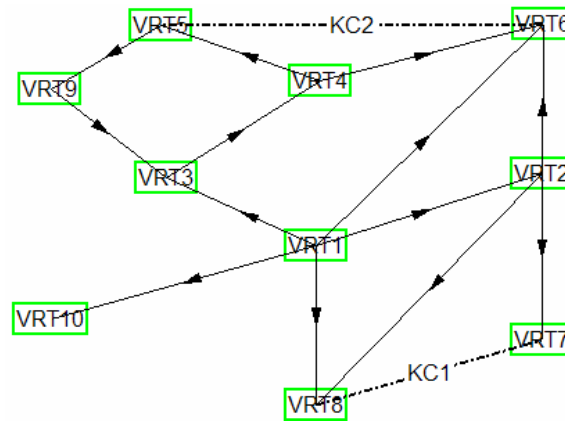
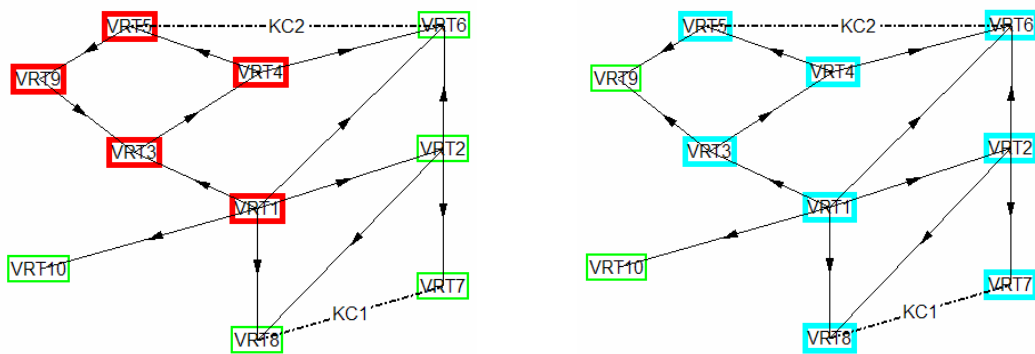


Figure 3.7: example. Graph representation

Two KCs are set: KC1 linking vertices VRT7 and VRT8, and KC2 linking vertices VRT5 and VRT6.



(a) - vertices belonging to cyclic loop (b) - influencing vertices

Figure 3.8: example. Analysis results

Graph analysis gives results depicted into Figure 3.8.a: vertices VRT3, VRT4,

Chapter 3. Graph Theory for Tolerancing

VRT5, VRT9 belongs to a cyclic loop. If vertices are assumed as parts being assembled, then, a cyclic loop means that a part locates itself (the same conclusion may be reached for features-tolerance links). Flipping the orientation of the edge connecting VRT9 to VRT3, the graphs becomes consistent. In fact, Figure 3.8.b shows influencing vertices, with respect to KC1 and KC2.

3.5 Summary

This Chapter presented a general methodology to analyze the global consistency of a given assembly graph. In this context, an oriented assembly graph was used to model tolerance specifications. Generally speaking, each vertex corresponds to a functional feature, while each edge is a functional link among features. Adequacy analysis was introduced, allowing to check the assembly graph from a global assembly point of view. Then, the more general consistency analysis was presented, allowing to check the assembly graph, accounting feature tolerance links and assembly links.

To easily manage such graphs, a MatLAB®'s graphical user interface was illustrated. User can interactively create the assembly graph, set assembly or tolerance links and run the analysis. Not-adequate or not-consistent vertices are automatically drawn into the graphical interface.

The proposed methodology for graph analysis may be a powerful support during the design stage, when designer should assign tolerance specifications on assembly made of many and many parts. To improve the efficiency of such methodology, geometrical and functional links should be directly imported from a CAD modeler.

Next Chapter develops the constraint analysis methodology, by partially using the same graph approach adopted for the global consistency analysis.

Chapter 4.

ASSEMBLY CONSTRAINT ANALYSIS¹

This Chapter describes a general procedure to analyze constraints for a given assembly combining Screw Theory, well-known in Kinematics and Robotics, and Graph Theory.

Generally speaking, mechanical assemblies may be classified into (I) under-constrained, (II), properly-constrained, and (III) over-constrained. Under-constrained assemblies are mechanism with some DoFs. In over-constrained assemblies, one or more DoFs are locked by different assembly features (redundant constraints are introduced into the assembly). If an assembly is not under-constrained and it is not over-constrained, it is properly-constrained (Whitney, 2008). It should be noted that the same assembly may be under- and over-constrained with respect to a specific DoF.

The aim of this Chapter is to provide a numerical methodology able to automatically detect the constraint state of a given assembly using geometrical information, imported from any CAD system.

4.1 Introduction

When working with assemblies with some mobility among parts, one may have interest to finding all the degrees of freedom (DoFs) of rigid body, and then looking at the under-constrained conditions of the whole assembly. In that contest, constraint

¹ This Chapter is based on: Franciosa, P., Gerbino S., *A CAD-based Methodology for Motion and Constraint Analysis According to Screw Theory*, Proc. of the ASME-IMECE'09, Lake Buena Vista, Florida (USA), November 13-19, 2009.

Chapter 4. Assembly Constraint Analysis

analysis is used to find under-constraints. Moreover, constraint analysis, is also related to check of the over-constrained status of the assembly. While un-constrained DoFs may cause unwanted motions, over-constrained DoFs may make difficult assembling parts or induce dangerous mechanical stresses. Choosing the right assembly constraint scheme is a crucial task to be achieved to assure the success of the assembly (Whitney, 1999).

Over the years many procedures have been suggested. A wide overview on geometric constraint solving problems is in (Gogu, 2005). Graph-based approaches and algebraic methods are the most common used to solve geometric constraint problems, and are dominant in 2D CAD. They have been also extended more recently to 3D cases where handling constraints and finding solutions is much more complex.

From 2D CAD point of view, the algebraic approach by D-Cubed, the so-called Dimensional Constraint Manager, DCM, is de-facto an industrial standard in constraint-based sketching. The more recent 3D version of this software, 3D DCM, based on a fast non sequential solver, is used to constraint parts in assemblies and mechanisms. Similar solution is offered by Leda Geometric solver, LGS 3D, a variational geometry engine used by several CAx systems.

All these solutions recognize some over-constraints also allowing redundant constraints in 2D. But they do not evaluate, at 3D level, all the real over-constraint status from a mathematical standpoint.

Screw Theory can be adopted since it allows to mathematically analyze all DoFs of an assembly and to model kinematic joints.

Since its origin, Screw theory has been used to analyze mechanisms. Waldron (Waldron, 1966) firstly introduced the concept of twist and wrench matrices, and the series and parallel laws of instantaneous kinematics. In (Davies, 1971) an extension of the Waldron theory to analyze planar linkages with cross coupling was described, while Baker (Baker, 1980) proposed an algorithm to represent 3D assembly joints, but only for closed chain problems. Davies (Davies, 1981a; Davies, 1981b; Davies, 1981c) used the similarity between joint loops and electrical circuits to apply the Kirchoff's loop rule to analyze mechanisms in terms of relative velocities of assembly joints.

Adams (Adams, 1999a; Adams, 1999b; Adams, 2001) gave a methodology to apply Screw Theory to perform constraint analysis of assemblies of rigid parts. Shukla (Shukla, 2001a; Shukla, 2001b) improved Adams's approach, with a new method, able to be used both for open and closed chains, and based on an assembly graph representation.

All the proposed methods still have some limits analyzing 3D assemblies of any complexity. A more general procedure can be found in the (Whitney, 2004), where Davies' approach, based on Kirchoff's loop rule law, is generalized to analyze constraints

in any assemblies.

This approach has been used in this dissertation to describe a methodology able to automatically analyze the constraint status of a given CAD assembly, combining Screw Theory and Graph Theory.

4.2 Methodology Overview

The proposed methodology may be summarized as follows. Starting from the assembly CAD model, assembly features are parameterized and the *Exploded Graph-based Assembly* (EGrA) is created, accordingly. Constraint analysis is accomplished by manipulating the graph related to the EGrA. In this phase, kinematic equations are automatically written, for each loop detected into the graph.

The methodology is mainly based on the following features:

- 4x4 homogenous transformation matrices are used to transform any screw matrix from a reference frame to another one;
- a global coordinate frame, Ω_0 , and a feature coordinate frame, attached to any feature, Ω_{Fi} , are defined;
- graphs are used to model assembly parts or features; and
- parts are assumed rigid. Screw Theory can be applied only under the hypothesis of ideal-rigid parts.

In the following the list of DoFs allowed for a given kinematic joint is given as $[\alpha, \beta, \gamma, \Delta x, \Delta y, \Delta z]$, where the triplet (α, β, γ) states the rotational DoFs, while $(\Delta x, \Delta y, \Delta z)$ is related to the translational DoFs.

Generally speaking, a screw matrix is a six-tuple that may represent either twist or wrench. A *twist* matrix is a screw that describes the instantaneous motion of a rigid body. One can write:

$$\mathbf{TS}_i = [\boldsymbol{\omega}_i \quad \mathbf{v}_i] \quad (4.1)$$

where $\boldsymbol{\omega}_i$ is the angular velocity vector. \mathbf{v}_i is the linear velocity vector of that point on the body or its extension that is instantaneously located at the origin of the global coordinate frame (Whitney, 2004).

A *wrench* matrix is a screw that describes the resultant force and moment for a system force acting on a rigid body. One can write:

$$\mathbf{W}_i = [\mathbf{f}_i \quad \mathbf{m}_i] \quad (4.2)$$

where \mathbf{f}_i the resultant force, while \mathbf{m}_i is the resultant moment calculated with respect to the origin of the global coordinate frame (Whitney, 2004).

First Davies's law states that the sum of wrenches exerted at any vertex, into the

Chapter 4. Assembly Constraint Analysis

motion graph, is equal to the net wrench exerted by that vertex (which corresponds to the first Kirchhoff's law - at any vertex in an electrical circuit the sum of currents flowing into that vertex is equal to the sum of currents flowing out of that vertex).

Second Davies's law states that changing in relative motion around any loop in a motion graph must be zero (which corresponds to the second Kirchhoff's law - the sum of the electrical potential differences around any closed circuit must be zero). Therefore, one can say that twist is associated to electrical potential difference, while wrench is associated to electrical current.

Both Davies's laws are mainly based on equations written for each independent loop counted into the EGrA.

To implement Davies's laws one has to:

- calculate independent loops into the EGrA;
- evaluate twist matrix for each edge counted into the EGrA, with respect the global coordinate frame;
- write Davies's laws accounting twist matrices;
- perform constraint analysis: solving constraint equations; and,
- analyze results, critically.

4.3 Exploded Graph-based Assembly

Kinematic joints among mating parts are modeled with a graph. Here, the *Graph-based Assembly* is introduced. It consists in a set of vertices and edges among them (Matripragada, 1998a; Matripragada, 1998b). Each vertex represents a part, while each edge represents mating conditions between two parts. The GrA is assumed as a non-oriented graph (Figure 4.1.a).

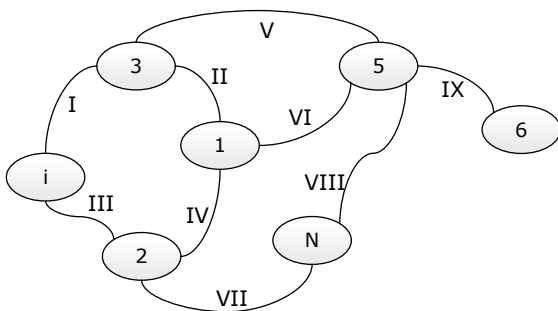


Figure 4.1.a: Graph-based Assembly representation

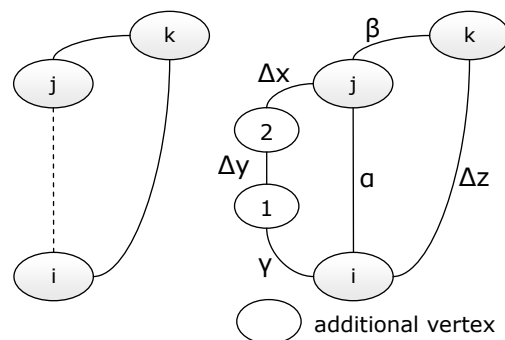


Figure 4.1.b: Exploded Graph-based Assembly representation

Generally speaking, each edge may describe multi-joint conditions. Part i may be referred to part j by means of several kinematic joints, such as planar mating, axial

alignment and so on. In these cases, the GrA is modified by introducing additional vertices (see Figure 4.1.b). All this allows to define the *Extended Graph-based Assembly*. In this way, each edge of the EGrA represents a specific DoF related to a kinematic joint. For example, in Figure 4.1.b part j is linked to part i by means of two kinematic joints: $[i, 1, 2, j]$ and $[i, j]$ (1 and 2 are additional vertices).

To easily manipulate the EGrA, the *edge* matrix, \mathbf{E} , and the *incidence* matrix, \mathbf{I} , are calculated (see Annex D for more details). \mathbf{E} is a $N_e \times 2$ rectangular matrix, where N_e is the total number of edges counted into the EGrA. \mathbf{I} is a $N_v \times N_e$ rectangular matrix, where N_v is the total number of vertices counted into the EGrA. It should be noted that \mathbf{I} is the incidence matrix for non-oriented graph.

4.4 Graph-based Manipulation

4.4.1 Path Detection

It is of interest evaluating the path connecting a vertex V_i to vertex V_j into the EGrA. This task is well-known in combinatorial research, where the minimum path is aimed to be detected. To do it, the Dijkstra's algorithm is usually applied (West, 2001). However, for the scope of the constraint analysis any path linking V_i to V_j can be calculated.

Below the MatLAB® pseudo-code to find out all paths connecting V_i to V_j .

```
%- find all paths connecting vertex Vi to Vj into EGrA
function[pathedge,P]=evalPaths(I,E,Vi,Vj,edgein,pathedge,P)

%- I: incidence matrix (for non-oriented graph)
%- E: edge matrix
%- Vi: start vertex
%- Vj: end vertex
%- edgein: list of connected edges
%- pathedge: local path variable
%- P: path matrix

%- return only first path detected
if size(P,1)==1
    return
end

%- update locale path
pathedge=[pathedge,edgein];

%- return path when Vi=Vj
if Vi==Vj
    allpath=[allpath,pathedge];
end

%- update start vertex and edgein
Viupdate=[];
edgein=[];
for j=1:size(I,2) %-count all vertices
```

Chapter 4. Assembly Constraint Analysis

```

    if I(Vi,j)==1 %- Vi is linked to edge j
        edgein=[edgein,j];
        Viupdate=[Viupdate,E(j,1)]; %- E(j,1) is the target vertex
    end
end

%- call function in a recursive way
for i=1:length(edgein)
    [tempe,P]= evalPaths(I,E,Viupdate(i),Vj,edgein(i),...
        pathedge,P);

    if flag==false
        return
    end
end
end

```

The proposed algorithm is similar to the “paths2graph” function adopted to detect target chains into functional graphs (see Chapter 3). In this case, the non-oriented incidence matrix, \mathbf{I} , is used and only the first path is returned.

Here, the *path* matrix, \mathbf{P} , ($1 \times N_e$ row vector) is introduced. The i -th entry is related to the i -th edge of the graph. The \mathbf{P} matrix is defined as follows:

$$\begin{cases} \mathbf{P}(1,i) = 1 \rightarrow i \text{ belongs to the path} \\ 0 \rightarrow \text{otherwise} \end{cases} \quad (4.3)$$

$$\forall i = 1, 2, \dots, N_e$$

The entry $(1, j)$ is equal to 1 if the edge belongs to the path; zero otherwise.

4.4.2 Loop Detection

Detecting loops inside the EGrA is needed to implement Davies’s laws. The aim is to evaluate all independent loops. A loop is a particular path connecting a vertex with itself through other vertices.

Generally speaking, two loops, A and B, are assumed independent if their difference is not empty. The difference operator between two sets, A and B, is a set having the items of A which are not included in B. In other words, loops A and B are independent if they differ in at least one edge. For instance, looking at Figure 4.1.a, four loops may be extracted: $\mathbf{L}_1=[\text{I II IV III}]$, $\mathbf{L}_2=[\text{II VI V}]$, $\mathbf{L}_3=[\text{IV VI VIII VII}]$ and $\mathbf{L}_4=[\text{I V VIII VII III}]$. Obviously, loop \mathbf{L}_4 depends on the first three ones because the difference between this set and the union of the first three sets is empty, according to the previous definition. An efficient way to accomplish loop detection is to divide the assembly graph into two sub-graphs: the *main graph* and the *secondary graph*. The main graph is directly derived from the EGrA. It contains those edges connecting all vertices of the EGrA without redundancies (main graph is known in electrical network as *spanning tree*). The secondary graph, instead, is the difference set between the EGrA and the

Chapter 4. Assembly Constraint Analysis

main graph.

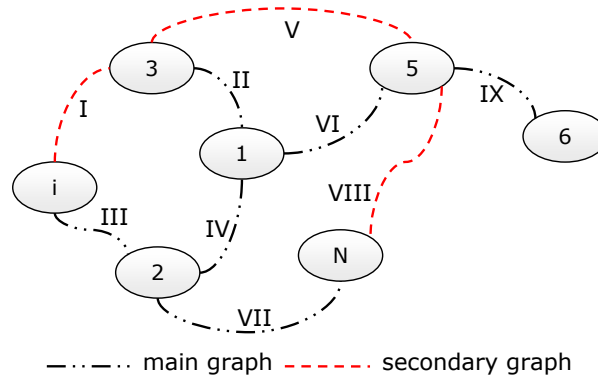


Figure 4.2: main graph and secondary graph

For example, Figure 4.2 shows the main graph and the secondary graph of EGrA depicted into Figure 4.1.a. It should be noted that, for a given graph, the main graph is not univocally defined. However, for the aim of the present analysis any main graph is acceptable.

Below the MatLAB® pseudo-code to evaluate the main graph and the secondary graph.

```
%- find out main and secondary graphs
function [maingraph,secgraph]=evalMainSecGraph(I)

%-I: incidence matrix
%-maingraph: list of edge belonging to main graph
%-secgraph: list of edge belonging to secondary graph

maingraph=[];
secgraph=[];
stnode=1; %-begin from vertex 1-seed vertex
visitedSeed=1; %-visited vertex

Flag=true; %-boolean variable to break loop
while Flag==true

    stnode=length(stnode);
    for I=1:stnode
        [tempseed,edge]=findSeed(stnode(I),I,visitedSeed); %-find new seed vertex

        maingraph=[maingraph,edge]; %-update

        visitedSeed=[visitedSeed,tempseed]; %-update list of visited vertex
    end
    stnode=tempseed; %-update seed list

    if length(allseed)==size(I,1);
        Flag=false; %-return when all vertices have been visited
    end
end

nEdge=size(1,2);
secgraph=diffSet([1:nEdge],maingraph); %-return the secondary graph
```

Chapter 4. Assembly Constraint Analysis

A growing procedure is adopted. Initially, the list of *visited* vertices contains only the first node counted into the EGrA. In an iteratively way, for each item belonging that list, all connected vertices are calculated (do to it, a search within the incidence matrix is performed - “findSeed” function). Then, the main graph list is updated, accordingly. The algorithm is stopped when the number of visited vertices becomes equal to N_v . This assures that all vertices of the EGrA are counted into the main graph, without redundancies.

Now, independent loops can be easily evaluated, by adding in main graph, step by step, edges of the secondary graph. For the i -th iteration, the i -th loop is calculated as the path connecting vertices belonging to the i -th edge of secondary graph. For example, looking at Figure 4.2, the main graph and the secondary graph are [II, III, IV, VI, VII, IX] and [I, V, VIII], respectively. With respect to the edge I, one possible path connecting vertices i to 3 is [III, IV, II]. Thus, the loop becomes [III, IV, II, I].

Here, one can introduce the *loop* matrix, \mathbf{L} , as a $N_{loop} \times N_e$ rectangular matrix, where N_{loop} is the number of independent loops. \mathbf{L} matrix is defined as follows:

$$\begin{cases} \mathbf{L}(i, j) = 1 \rightarrow j \text{ belongs to the loop } i - \text{th} \\ 0 \rightarrow \text{otherwise} \end{cases} \quad (4.4)$$

$$\forall i = 1, 2, \dots, N_{loop}, \forall j = 1, 2, \dots, N_e$$

It can be pointed out that all entries of matrix \mathbf{L} are calculated taking into account the same approach used for the \mathbf{P} matrix.

4.5 Evaluating Twist Matrix

Any assembly feature is parameterized through a feature coordinate frame, attached to the same feature (Adams, 1998). Let $\mathbf{T}_{0,Fi}$ be the 4x4 transformation matrix, allowing to express the feature coordinate frame into the global coordinate frame (see Annex A for details on how calculate it by using geometrical data). One can write:

$$\mathbf{T}_{0,Fi} = \begin{bmatrix} \mathbf{R} & \mathbf{d} \\ \mathbf{0} & 1 \end{bmatrix} \quad (4.5)$$

where \mathbf{R} and \mathbf{d} are the rotation matrix and the position vector of the origin of the feature frame with respect the global one.

As it is, twist matrix is written into the feature coordinate frame. The aim is to express the same twist into the global coordinate frame.

From the definition stated into equation (4.1), it has:

Chapter 4. Assembly Constraint Analysis

$$\mathbf{TS}_{Fi} = [\boldsymbol{\omega}_{Fi} \quad \mathbf{v}_{Fi}] \quad (4.6)$$

For rotational DoF one can write:

$$\begin{aligned} \mathbf{v}_{Fi} &= \boldsymbol{\omega}_{Fi} \wedge \mathbf{r} = \boldsymbol{\omega}_{Fi} \wedge (-\mathbf{R}^T \cdot \mathbf{d}) = \mathbf{R}^T \cdot \boldsymbol{\omega}_0 \wedge (-\mathbf{R}^T \cdot \mathbf{d}) = \dots \\ &\dots = \mathbf{R}^T \cdot (\mathbf{d} \wedge \boldsymbol{\omega}_0) = \mathbf{R}^T \cdot \mathbf{v}_0 \rightarrow \mathbf{v}_0 = \mathbf{d} \wedge \boldsymbol{\omega}_0 \end{aligned} \quad (4.7)$$

where \mathbf{r} is the position of the origin of the global coordinate frame with respect the feature coordinate frame.

Thus, the twist matrix for rotational DoF becomes:

$$\begin{cases} \boldsymbol{\omega}_0 = \mathbf{R} \cdot \boldsymbol{\omega}_{Fi} \\ \mathbf{v}_0 = \mathbf{d} \wedge \boldsymbol{\omega}_0 \end{cases} \rightarrow \mathbf{TS}_0 = [\boldsymbol{\omega}_0 \quad \mathbf{v}_0] \quad (4.8)$$

while, for translational DoF, it has:

$$\begin{cases} \boldsymbol{\omega}_0 = \mathbf{0} \\ \mathbf{v}_0 = \mathbf{R} \cdot \mathbf{v}_{Fi} \end{cases} \rightarrow \mathbf{TS}_0 = [\mathbf{0} \quad \mathbf{v}_0] \quad (4.9)$$

Therefore, given the rotation matrix, \mathbf{R} , and the position vector, \mathbf{d} , of the origin of the feature frame with respect to the global frame, equations (4.8) and (4.9) allow to completely define the twist matrix of any kinematic joint.

It is of interest evaluating the *Instantaneous Screw Axis* (ISA) starting from the twist matrix, \mathbf{TS}_0 . Here, the ISA is represented through a point, \mathbf{P}_{ISA} , and a unit vector, \mathbf{N}_{ISA} . By using the definition in (4.1), the first triplet of the twist matrix defines the \mathbf{N}_{ISA} direction, while the second triplet is associated to the position of such axis.

$$\begin{cases} \mathbf{N}_{ISA} = \frac{\boldsymbol{\omega}_0}{\|\boldsymbol{\omega}_0\|} \\ \mathbf{v}_0 = \mathbf{d} \wedge \boldsymbol{\omega}_0 \rightarrow \mathbf{P}_{ISA} = \mathbf{d} = \frac{1}{\|\boldsymbol{\omega}_0\|^2} \cdot \boldsymbol{\omega}_0 \wedge \mathbf{v}_0 \end{cases} \quad (4.10)$$

Thus, for rotational DoFs, one can use the relationship (4.10). Instead, for translation DoFs, it has:

$$\begin{cases} \mathbf{N}_{ISA} = \mathbf{0} \\ \mathbf{P}_{ISA} = \frac{\mathbf{v}_0}{\|\mathbf{v}_0\|} \end{cases} \quad (4.11)$$

The above parameterization has to be applied for any DoF, related to any kinematic joint, counted into the EGrA. Kinematic joints could be imported from a CAD feature-based assembly, via scripting. In this dissertation, the proposed algorithm has

been implemented and tested in MatLAB® environment, where geometrical data are manually assigned.

4.6 Davies's Law and Constraint Analysis

4.6.1 Motion Analysis

Starting from the EGrA correctly defined, assembly twist matrix ($N_e \times 6$ matrix), \mathbf{TS} , can be introduced as in (4.12), where $\mathbf{TS}_{0,i}$ is the twist matrix associated to the i -th edge (for instance, the i -th edge corresponds to the i -th DoF for a specific joint) of the EGrA and defined into the global coordinate frame, Ω_0 .

$$\mathbf{TS} = \begin{bmatrix} \mathbf{TS}_{0,1} \\ \dots \\ \mathbf{TS}_{0,i} \\ \dots \\ \mathbf{TS}_{0,N_e} \end{bmatrix} \quad (4.12)$$

Motion analysis is based on the second Davies's law, stated above. Therefore, taking into account the loop matrix, \mathbf{L} , and the assembly twist matrix, \mathbf{TS} , one can write:

$$\begin{cases} \mathbf{TS}(1, j) \cdot \mathbf{L}(i, 1) \cdot X_{m,1} + \mathbf{TS}(2, j) \cdot \mathbf{L}(i, 2) \cdot X_{m,2} + \dots \\ \dots + \mathbf{TS}(k, j) \cdot \mathbf{L}(i, k) \cdot X_{m,k} + \dots \\ \dots + \mathbf{TS}(N_e, j) \cdot \mathbf{L}(i, N_e) \cdot X_{m,N_e} = 0 \\ \forall i = 1, 2, \dots, N_{loop} \\ \forall j = 1, 2, \dots, 6 \end{cases} \quad (4.13)$$

where $X_{m,k}$ is the k -th unknown velocity magnitude. Equation (4.13) states that, with respect to the i -th loop, the net j -th component of velocity is zero.

Equation (4.13) can be generalized in a matrix form as in equation (4.14), where \mathbf{M}_m is the $6 \cdot N_{loop} \times N_e$ *assembly motion* matrix, while \mathbf{X}_m is a $N_e \times 1$ vector containing the unknown velocity magnitudes.

$$\mathbf{M}_m \cdot \mathbf{X}_m = \mathbf{0} \quad (4.14)$$

Solutions of the linear homogeneous system stated in (4.14) can be obtained by evaluating the rank, r^2 , of the rectangular matrix \mathbf{M}_m (see Annex B for more details on

² It should be noted that the rank of an m by n matrix is less or equal to $\min(m, n)$.

Chapter 4. Assembly Constraint Analysis

systems of linear equations and their managing).

Therefore, if $r=N_e$ then system (4.14) has only a trivial solution ($\mathbf{X}_m=\mathbf{0}$). In this case, the assembly has no motion. Thus, one can say that the degree of mobility, d_m , of the assembly is zero ($d_m=0$).

Instead, when $0<r\leq N_e$ system (4.14) is underdetermined and it has N_e-r solutions. In this case, the number of unknowns is greater than the number of independent equations. The degree of mobility becomes $d_m=N_e-r$, while the space of solutions is given by the *reduced motion* matrix, $\mathbf{X}_{m,r}$ ($N_e \times d_m$ rectangular matrix). To numerically calculate it, the nullspace of the motion matrix, \mathbf{M}_m , must be evaluated. Any solver of system of linear equations may be adopted, such as the “null” function available in MatLAB®. The $\mathbf{X}_{m,r}$ matrix contains the magnitude velocities for each DoF allowed for a specific joint of the analyzed assembly.

In mechanical field or Robotics, evaluating the motion of a part (*object* part) with respect to another one (*target* part) may be of interest. To do this, the twist matrix related the object part should be calculated. The procedure may be accomplished into two consecutive steps: (I) finding a path matrix³, \mathbf{P} , linking the object part to the target one; (II) calculating the motion (if any) and the ISA of the object part.

Therefore, with respect to the i -th degree of mobility, the j -th component of velocity becomes:

$$\left\{ \begin{array}{l} \mathbf{U}_{O,T}(i,j) = \mathbf{TS}(1,j) \cdot \mathbf{P}(1,1) \cdot \mathbf{X}_{m,r}(1,i) + \dots \\ \dots + \mathbf{TS}(2,j) \cdot \mathbf{P}(1,2) \cdot \mathbf{X}_{m,r}(2,i) + \dots \\ \dots + \mathbf{TS}(k,j) \cdot \mathbf{P}(1,k) \cdot \mathbf{X}_{m,r}(k,i) + \dots \\ \dots + \mathbf{TS}(N_e,j) \cdot \mathbf{P}(1,N_e) \cdot \mathbf{X}_{m,r}(N_e,i) \end{array} \right. \quad (4.15)$$

$$\left\{ \begin{array}{l} \forall i = 1, 2, \dots, d_m \\ \forall j = 1, 2, \dots, 6 \end{array} \right.$$

where $\mathbf{U}_{O,T}(i,j)$ is the j -th component of velocity related to the i -th independent degree of mobility of the assembly for the object part with respect to the target one. $\mathbf{U}_{O,T}$ is here called *object motion* matrix ($d_m \times 6$ matrix). As stated above, starting from the so-calculated object motion matrix, the ISA may be easily derived (see Section 4.5).

It can be pointed out that, if no loop exists within the EGrA, then the Davies’s law cannot be applied. Typically, all this happens for serial linkages assemblies (a robot arm, for example). For such assemblies, motion analysis can be performed by applying the serial law, as proposed in (Shukla, 2001b), where a stack-up of twist matrices is used (union algorithm for twist matrices).

³ To better understand this issue, the analogy between twist and potential difference for an electrical network may be here adopted.

4.6.2 Constraint Analysis

Constraint analysis aims to detect over-constraints existing from a part to another and it is based on the first Davies's law. Looking at the assembly motion matrix, \mathbf{M}_m , the number of dependent rows gives information on the degree of over-constraint.

Thus, one can introduce the degree of over-constraint, d_o , which is equal to $6 \cdot N_{loop} - r$. The degree of over-constraint is zero when all equations are independent among them ($r = 6 \cdot N_{loop}$). In this case the assembly presents no over-constraint. Thus, one can say that an assembly with serial linkages (for which all kinematic equations are independent – no loop exists) exerts no over-constraint. Instead, a more general assembly with cross coupling linkages (for which loops may be detected) may present over-constraints if at least two kinematic equations are dependent among them.

As stated above for the assembly motion matrix, here, the *assembly over-constraint* matrix, \mathbf{M}_o ($N_e \times 6 \cdot N_{loop}$), is introduced accordingly. One can write, briefly:

$$\begin{cases} \mathbf{M}_o \cdot \mathbf{X}_o = 0 \\ \mathbf{M}_o = \mathbf{M}_m^T \end{cases} \quad (4.16)$$

The space of solutions of the linear homogeneous system stated in (4.16) is called *reduced over-constraint* matrix, $\mathbf{X}_{o,r}$ ($6 \cdot N_{loop} \times d_o$ rectangular matrix).

Once the reduced over-constraint matrix is established, over-constraints transmitted by a specific joint to a part (*object* part) can be calculated, by applying the first Davies's law. The procedure is based on two phases: (I) all joints (*over-constraint joints*) related to the object part are determined (into the EGrA, all edges connected to the *object* vertex are evaluated); (II) for each over-constraint joint, relative over-constraints are evaluated by adding or subtracting over-constraints for each loop counted into the EGrA.

With respect to the i -th over-constraint joint, transmitted over-constraints can be calculated as (below “ $i:j$ ” is a vector which ranges from “ i ” to “ j ” with unitary step) in equation (4.17), where $\mathbf{W}_{O,i}$ is the over-constraint matrix ($d_o \times 6$ rectangular matrix) related to the i -th joint for the specific object part. $\mathbf{W}_{O,i}$ corresponds to the wrench space according to the twist space stated into relationship (4.15).

$$\begin{cases} \mathbf{W}_{O,i} = \mathbf{X}_{o,r}(1:6,:) \cdot \mathbf{L}_{ov}(1,i) + \dots \\ \dots \mathbf{X}_{o,r}(7:12,:) \cdot \mathbf{L}_{ov}(2,i) + \dots \\ \dots + \mathbf{X}_{o,r}(6 \cdot N_{loop} - 5 : 6 \cdot N_{loop}, :) \cdot \mathbf{L}_{ov}(N_{loop}, i) \\ \forall i = 1, 2, \dots, N_{joint,o} \end{cases} \quad (4.17)$$

Moreover, \mathbf{L}_{ov} is the sub-matrix extracted from the loop matrix, \mathbf{L} . \mathbf{L}_{ov} contains columns of \mathbf{L} with respect to the over-constraint joints, related to the part being analyzed.

Chapter 4. Assembly Constraint Analysis

Then, \mathbf{L}_{ov} is a $N_{loop} \times N_{joint,o}$ matrix, where $N_{joint,o}$ is the number of over-constraint joints for the specific object part.

4.7 Examples

To better understand the proposed algorithm, some numerical examples will be shown in this Section. All numerical analyses were performed in MatLAB®.

4.7.1 Four-bar Assembly

Figure 4.3 shows a four-bar planar assembly. Each assembly feature is a pin joint which allows only the rotation around the local Z axis (all feature coordinate frames have the same orientation of the global frame). Translation along global Z axis is locked. Bar sizes are: $a=150$, $b=40$, $c=100$ and $d=90$.

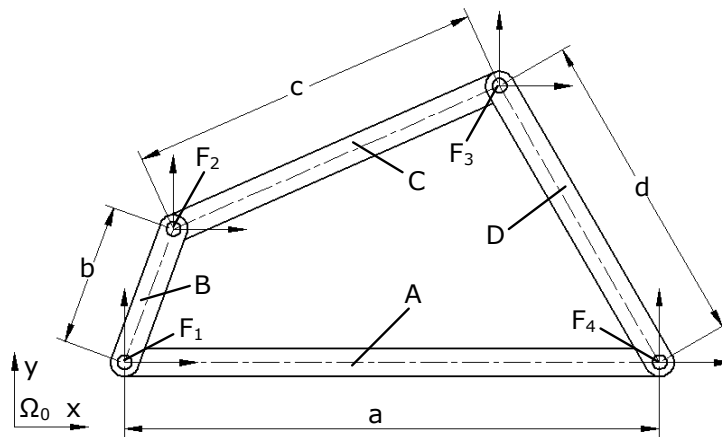


Figure 4.3: example. Four bar assembly

By using the described procedure, the edge matrix and the incidence matrix were automatically calculated (EGrA is made of 4 vertices and 4 edges).

It is of interest evaluating the mobility of bar C, with respect to bar A. Numerical analysis give the following results:

$$\left\{ \begin{array}{l} \mathbf{U}_{C,A} = [0.0, 0.0, 1.0 \mid 159.76, -58.15, 0.0] \\ \quad \quad \quad \downarrow \\ \mathbf{N}_{ISA} = [0.0, 0.0, 1.0] \\ \mathbf{P}_{ISA} = [58.15, 159.76, 0.0] \end{array} \right.$$

The object motion matrix, $\mathbf{U}_{C,A}$, has a single row. This means that the assembly has only one degree of mobility. Moreover, the ISA of part C is located into point \mathbf{P}_{ISA} and directed along \mathbf{N}_{ISA} . This result completely agrees with theoretical considerations (assuming that the angle between bar B and A is 70° and the global coordinate frame is coincident with the frame related to feature F_1).

Chapter 4. Assembly Constraint Analysis

Over-constraint analysis gives the following results (to make results more readable, the MatLAB® “rref” function has been applied):

$$\left\{ \begin{array}{l} \mathbf{W}_{C,F2} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \\ \mathbf{W}_{C,F3} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \end{array} \right.$$

where $\mathbf{W}_{C,F2}$ and $\mathbf{W}_{C,F3}$ are the over-constraint matrices transmitted by joints F_2 and F_3 to bar C, respectively. This result shows that out-of-plane over-constraints exist for this mechanism (the degree of over-constraint is 3). In fact, the first row of $\mathbf{W}_{C,F2}$ (the same may be said for $\mathbf{W}_{C,F3}$) corresponds to a reaction force along the global Z axis, whereas the second and the third rows correspond to reaction moments along the global X and Y axes, respectively. Therefore, the translation along the global Z axis and the rotation around global X and Y axes are over-constrained in this assembly. This result may be extended for any planar single-loop mechanism with pin joints: out-of-plane DoDs are always over-constrained.

4.7.2 Parallelogram Mechanism

Figure 4.4 shows a planar parallelogram mechanism. In this case, the EGrA is composed of 6 vertices (6 parts) and 7 edges (7 pin joints - all feature coordinate frames have the same orientation of the global frame). Sizes of bars are: $a=130$ and $b=75$. This mechanism has two independent loops (see Figure 4.4 – EGrA representation).

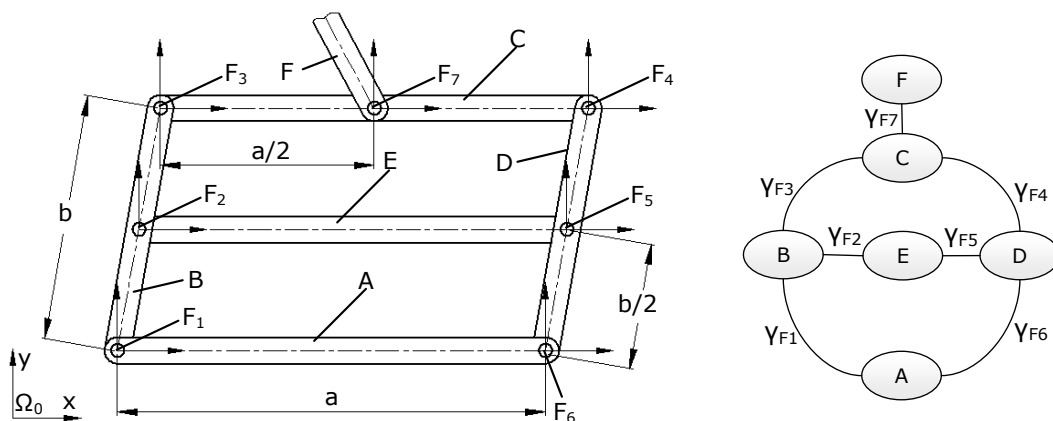


Figure 4.4: example. Parallelogram mechanism and EGrA representation

Chapter 4. Assembly Constraint Analysis

Motion analysis performed on bar F, with respect to bar A, gives the following results:

$$\mathbf{U}_{F,A} = \begin{bmatrix} 0.0, 0.0, 0.0 & | & 73.86, -13.02, 0.0 \\ 0.0, 0.0, 1.0 & | & 73.86, -78.02, 0.0 \end{bmatrix}$$

The object motion matrix has two rows: this means that the assembly has 2 independent pin joints ($d_m=2$). In particular, the first row says that, with respect the first independent pin joint (the feature F_1), bar F can translate along the global \mathbf{X} and \mathbf{Y} axes. Instead, the second row corresponds to the rotation of bar F with respect to pin F_7 . In this case the ISA becomes (assuming that the angle between bar B and A is 80°): $\mathbf{P}_{ISA}=[78.02, 73.86, 0]$, which exactly corresponds to the origin of the Ω_{F7} frame, as expected, and $\mathbf{N}_{ISA}=[0.0, 0.0, 1.0]$. Obviously, if the object motion matrix is evaluated with respect to bar C, it has $\mathbf{U}_{F,C}=[0.0, 0.0, 1.0 | 73.86, -78.02, 0.0]$. This means that bar F has only one rotational DoF with respect to bar C. Moreover, bar F has no over-constraint since it does not belongs to any loop.

Bar E belongs to two independent loops. Thus, an over-constrained status is expected. First of all, under-constraint analysis of bar E relative to bar A, gives:

$$\mathbf{U}_{E,A} = [0.0, 0.0, 0.0 | 36.93, -6.51, 0.0]$$

This means that bar E can translate along global \mathbf{X} and \mathbf{Y} axes. From over-constraint analysis, it has (the same result is reached for the joint F_5):

$$\mathbf{W}_{E,F2} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 & 0 & 36.96 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

This means that both joints F_2 and F_5 transmit to bar E moments around global \mathbf{X} and \mathbf{Y} axes and forces along global \mathbf{X} and \mathbf{Z} axes.

4.7.3 Landing Wheel Mechanism

Figure 4.5 shows a landing wheel mechanism, used in retractable aircraft systems. Cylinder E is connected to piston F by means of a cylindrical joint (translation and rotation DoFs are allowed). All other joints are revolute pin joints.

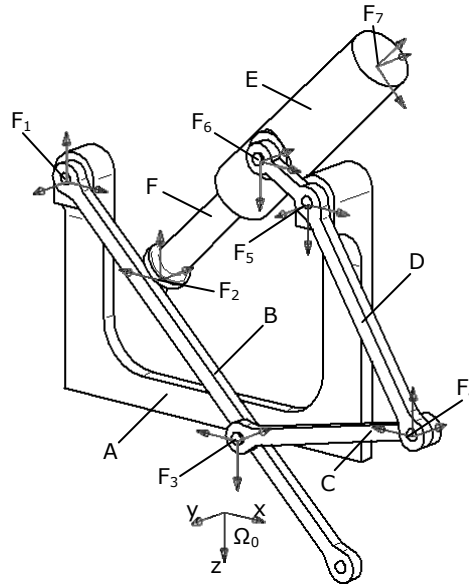


Figure 4.5: example. Landing wheel mechanism

The degree of mobility and the degree of over-constraint are 1 and 4, respectively. Assuming frame A as fixed part, motion analysis gives the following results for bar C and cylinder F:

$$\left\{ \begin{array}{l} \mathbf{U}_{C,A} = [0.00, -0.0029, -0.0 \mid 0.65, -0.0, -1.27] \\ \quad \downarrow \\ \mathbf{N}_{ISA} = [0.0, -1.0, 0.00] \\ \mathbf{P}_{ISA} = [437.93, 0.0, 224.14] \end{array} \right.$$

$$\left\{ \begin{array}{l} \mathbf{U}_{F,A} = [0.0, -0.0008, -0.00 \mid 0.25, 0.0, -0.44] \\ \quad \downarrow \\ \mathbf{N}_{ISA} = [0.0, -1.0, 0.0] \\ \mathbf{P}_{ISA} = [550.00, 0.0, 312.50] \end{array} \right.$$

It is of interest evaluating the over-constraints related to cylinder F. Constraint analysis gives (the same result is reached for joint F7):

$$\mathbf{W}_{F,F2} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Chapter 4. Assembly Constraint Analysis

This means that both pin joint F_2 and cylindrical joint F_7 transmit translational over-constraints along the global \mathbf{Y} axis and rotational over-constraints around global \mathbf{X} , \mathbf{Y} and \mathbf{Z} axes.

4.8 Summary

This Chapter presented a general numerical methodology do to motion and constraint analysis starting from geometrical data available from a CAD assembly.

Assembly features are detected and the *Exploded Graph-based Assembly* is introduced to model kinematic joints among mating parts. Then, graphs manipulations are performed to detect loops. By using the analogy between electrical and mechanical graphs, proposed by Davies, kinematic equations are written for each loop extracted from the EGrA. The degree of mobility and the degree of over-constraints of the analyzed assembly is calculated by evaluating the null space of a system of linear equations. Finally, three assembly CAD models were described and analyzed. Simulation results gave information on the state of constraint of assemblies. All algorithms were tested and implemented in MatLAB® environment, where input geometrical data were manually assigned.

To improve the efficiency and to make fully automatic the proposed methodology, geometrical data should be directly imported from a CAD modeler.

Next Chapter develops the SVA-TOL methodology to perform tolerance analysis of rigid assemblies. Constraint analysis, illustrated here, will be used to detect the constraint status and to simulate the assembly process.

Chapter 5.

RIGID ASSEMBLIES¹

This Chapter describes a general procedure to do tolerance analysis of rigid part assemblies. The methodology, called SVA-TOL (Statistical Variation Analysis for Tolerancing), aims to (I) model variational features, according to GD&T or ISO specifications, and (II) simulate 3D assembly joints.

4x4 homogenous transformation matrices are adopted to capture geometrical rules, provided by Standards. Once variational features are generated, assembly constraints among variational parts are introduced.

Two solvers for assembly constraints are proposed: the sequential solver and the least squares solver. The first one solves one-by-one assembly joints. All this allows to simulate different assembly sequences. The least squares approach, instead, solves all assembly constraints simultaneously.

5.1 Introduction

During assembly phase, variations propagate part-to-part. This propagation is strictly related to the constraint state of the assembly (see also Chapter 4).

From a mathematical point of view, the assembly model is obtained by specifying assembly joints and solving for the specified assembly constraints to find out the relative positions of parts (Kim, 2004; Kim, 2005a; Kim, 2005b).

¹ This Chapter is partially based on: Franciosa P., Gerbino S., Patalano S., *Variational Modeling and Assembly Constraints in Tolerance Analysis of Rigid Part Assemblies: Planar and Cylindrical Features*, Int. Journal of Advanced Manufacturing Technology, DOI: 10.1007/s00170-009-2400-5, 2009.

Chapter 5. Rigid Assemblies

Several methods have been proposed over the years related to the assembly constraint problem. The modified Newton-Raphson method or the Levenberg-Marquardt method allows to solve for non-linear assembly equations in a simultaneous way by using iterative algorithms. Other approaches are based on algebraic procedures (Turner, 1992; Tanaka, 2001), in which each assembly constraint is subdivided into a sequence of rotation and translation operations by using a reduction of DoFs.

Specific commercial software, such as Working Model® and Adams®, provides a joint mating method that uses joint mating constraint to define relations between components and solve for constraint equations (Kramer, 1992). Nowadays, the modern CAD systems integrate motion-based solver to perform the assembly motion analysis, mainly based on an assembly constraint solver.

In the tolerance analysis field, the evaluation of geometric feature variations, for given assembly constraint sets, is of interest. Typically, in the field of rigid part assemblies, only small rotational and translational displacements of features being assembled are considered.

As noted in Chapter 2, one urgent need is the modeling of variational features accounting GD&T or ISO specifications. In addition, mating joints between variational features have to be correctly introduced. The present Chapter gives a contribution to the solution to all of these needs.

5.2 Methodology Overview

The general SVA-TOL work-flow is based on two main steps. In the first step, variational features are modeled; then, assembly constraints among variational features are introduced.

Starting from the nominal assembly geometry, imported from a CAD modeler, tolerance specifications are modeled for each part being assembled, according to GD&T or ISO rules. By using 4x4 homogenous transformation matrices, each functional feature is parameterized into a local reference frame, attached to the feature, using the geometrical parameters of the same feature. Variational transformation matrices are then introduced to consider small translational and rotational displacements. These displacements are dependent among them. The envelope of all small displacement parameters is described through a hyper-domain, which is - generally speaking - a domain of the R^n parametric space and represents all sources of variation. Given this domain, variational features can now be generated. To do it, a Monte Carlo simulation is adopted.

Once variational features have been defined, assembly constraints among mat-

ing parts are introduced. By using point, line and plane entities and their combinations, kinematic joints are modeled. The proposed assembly model allows to take into account different assembly sequences. The best fit alignment among mating features is performed by using iterative optimization algorithms.

5.3 Variational Feature Modeling

5.3.1 Feature Modeling

The proposed variational feature methodology is based on the following hypotheses:

- any feature preserves its original shape, so planar faces remain always planar, cylindrical faces remain always cylindrical; and,
- small displacements are assumed, that is in accordance with the tolerance specification phase in which small deviations are usually considered.

In this dissertation, 4x4 homogeneous transformation matrices are used to completely define the location of a part or a feature, with respect to the global coordinate frame. For instance, by using the notation of Annex A, the homogeneous matrix, $\mathbf{T}_{0,j}$, from the frame Ω_j to Ω_0 one, can be expressed as in (5.1), where \mathbf{d} is the 3x1 position vector, while \mathbf{R} is the 3x3 rotation matrix.

$$\mathbf{T}_{0,j} = \begin{bmatrix} \mathbf{R} & \mathbf{d} \\ \mathbf{0} & 1 \end{bmatrix} \quad (5.1)$$

As shown in Figure 5.1, three main features are introduced:

- *nominal feature*: as described above, the nominal feature is coincident with the associated feature and it is imported directly from the CAD model;
- *actual feature*: it is the “perturbed” feature, according to tolerance deviations; and,
- *derived feature*: it is the one derived from the actual datum feature (“A” in Figure 5.1), with respect to the specified tolerance (“&” in Figure 5.1). “&” is any tolerance specification, which requires a *datum reference frame*.

Obviously, for form tolerance, which does not require any *datum reference frame*, the derived and the actual features are coincident each-other.

In addition, each elementary feature is parameterized by means of:

- *geometric parameters*: radius and height of a cylinder, width and length of a plane, and so on;
- *variational parameters*: small displacements due to small deviations between the actual and the derived features. These parameters depend on the local reference frame attached to the feature. TTRS classification is here used; and,

Chapter 5. Rigid Assemblies

- *datum constraints*: the derived feature is constrained to the datum features, if assigned.

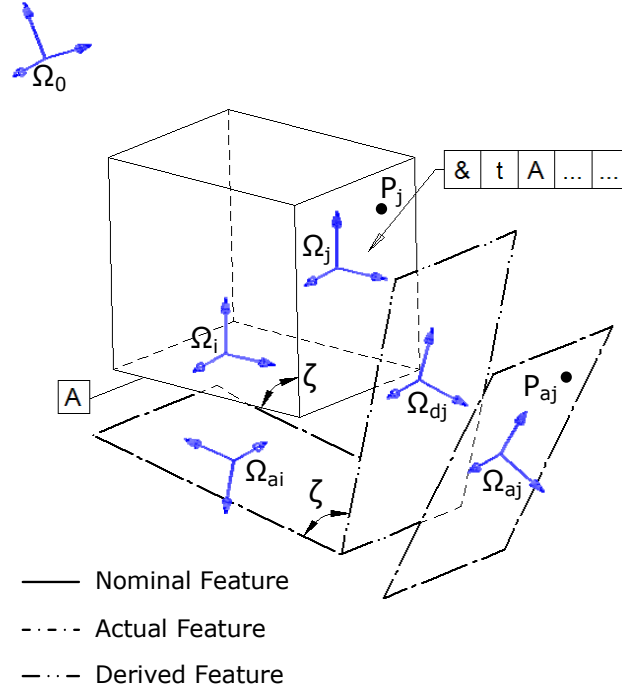


Figure 5.1: variational feature modeling

In the next, j is the actual feature, while i is the datum feature. Looking at Figure 5.1, the aim is to define any point, \mathbf{P}_{aj} , belonging to the actual feature Ω_{aj} , into the global frame Ω_0 , taking into account geometric parameters, variational parameters and datum constraints. It should be noted that the point \mathbf{P}_{aj} is initially defined into the global frame Ω_0 (CAD geometry is referred to that frame). Since the point \mathbf{P}_{aj} corresponds to point \mathbf{P}_j , defined into frame Ω_j (nominal feature), then one can write:

$$\mathbf{P}_{aj} \equiv \mathbf{P}_j = \mathbf{T}_{0,j}^{-1} \cdot \mathbf{P}_{0j} \quad (5.2)$$

where \mathbf{P}_{0j} , is the point \mathbf{P}_j defined into the global frame, Ω_0 .

Given these assumptions, the global variational 4x4 matrix (see Annex A for composition of transformations) can be expressed as in equation (5.3):

$$\mathbf{T}_{0,aj} = \mathbf{T}_{0,i} \cdot \mathbf{T}_{i,ai} \cdot \mathbf{T}_{ai,dj} \cdot \mathbf{T}_{dj,aj} \quad (5.3)$$

where $\mathbf{T}_{i,ai}$ and $\mathbf{T}_{dj,aj}$ are the variational matrices related to the feature Ω_j and to the datum feature Ω_i , respectively. The transformation $\mathbf{T}_{ai,dj}$ is equal to $\mathbf{T}_{i,j}$ one (it should be noted that rigid transformations are considered). Thus, it can be written:

$$\mathbf{T}_{ai,dj} = \mathbf{T}_{i,j} = \mathbf{T}_{i,0} \cdot \mathbf{T}_{0,j} \rightarrow \mathbf{T}_{ai,dj} = \mathbf{T}_{0,i}^{-1} \cdot \mathbf{T}_{0,j} \quad (5.4)$$

Chapter 5. Rigid Assemblies

Equation (5.4) states the geometrical relationship between the analyzed feature and the datum feature.

Finally, considering equations (5.2), (5.3) and (5.4) it is:

$$\begin{aligned}\bar{\mathbf{P}}_{0j} &= \mathbf{T}_{0,aj} \cdot \mathbf{P}_{aj} = \mathbf{T}_{0,aj} \cdot \mathbf{T}_{0,j}^{-1} \cdot \mathbf{P}_{0j} = \dots \\ &\dots = \mathbf{T}_{0,i} \cdot \mathbf{T}_{i,ai} \cdot \mathbf{T}_{0,i}^{-1} \cdot \mathbf{T}_{0,j} \cdot \mathbf{T}_{dj,aj} \cdot \mathbf{T}_{0,j}^{-1} \cdot \mathbf{P}_{0j} \rightarrow \bar{\mathbf{P}}_{0j} = \mathbf{T}_{0,0} \cdot \mathbf{P}_{0j}\end{aligned}\quad (5.5)$$

thus:

$$\mathbf{T}_{0,0} = \mathbf{T}_{0,i} \cdot \mathbf{T}_{i,ai} \cdot \mathbf{T}_{0,i}^{-1} \cdot \mathbf{T}_{0,j} \cdot \mathbf{T}_{dj,aj} \cdot \mathbf{T}_{0,j}^{-1} \quad (5.6)$$

where $\mathbf{T}_{0,0}$ is the global variational 4x4 matrix which transforms any point, belonging to the feature j and initially defined into Ω_0 , into frame Ω_0 , taking into account geometric parameters ($\mathbf{T}_{0,i}$ and $\mathbf{T}_{0,j}$), variational parameters ($\mathbf{T}_{i,ai}$ and $\mathbf{T}_{dj,aj}$) and datum constraints ($\mathbf{T}_{0,i}^{-1} \cdot \mathbf{T}_{0,j}$).

When no datum is assigned (tolerance of form), the global variational matrix, expressed in (5.6), becomes:

$$\mathbf{T}_{0,0} = \mathbf{T}_{0,j} \cdot \mathbf{T}_{j,aj} \cdot \mathbf{T}_{0,j}^{-1} \quad (5.7)$$

Variational matrices are defined by using the Whitney's notation. Then, one can write:

$$\mathbf{T}_{j,aj} = \begin{bmatrix} 1 & -\gamma_F & \beta_F & \Delta x_F \\ \gamma_F & 1 & -\alpha_F & \Delta y_F \\ -\beta_F & \alpha_F & 1 & \Delta z_F \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.8)$$

where $\mathbf{T}_{j,aj}$ is the generic variational matrix, from feature Ω_{aj} to Ω_j . In addition, $\mathbf{\Delta}_F = (\Delta x_F, \Delta y_F, \Delta z_F)$ and $\mathbf{\Theta}_F = (\alpha_F, \beta_F, \gamma_F)$ are the small translational and small rotational vectors², respectively. TTRS Theory provides an exhaustive classification of variational parameters, by combining elementary geometrical entities (point, line and plane). In the present approach, by using the TTRS classification, variational matrices may be evaluated for any functional feature.

Feature variational parameters are dependent among them. In fact, if all parameters are at their maximum value, a portion of the feature would be outside the geometrical tolerance zone. Therefore, for each tolerance zone, variational parameter constraints must be derived. Generally speaking, variational parameter constraints may be calculated by evaluating the small displacement, \mathbf{D}_{Pj} , of the j -th point, \mathbf{P}_j , belonging to the

² $\mathbf{\Delta}_F$ and $\mathbf{\Theta}_F$ are called torsor parameters into TTRS Theory.

Chapter 5. Rigid Assemblies

analyzed feature. To do it, one can write:

$$\mathbf{D}_{P_j} = \mathbf{T}_{j,aj} \cdot \mathbf{P}_j - \mathbf{P}_j \quad (5.9)$$

and by using the notation of Annex A, it has:

$$\mathbf{D}_{P_j} = \Delta_F + \begin{bmatrix} 1 & -\gamma_F & \beta_F \\ \gamma_F & 1 & -\alpha_F \\ -\beta_F & \alpha_F & 1 \end{bmatrix} \cdot \mathbf{P}_j - \mathbf{P}_j \quad (5.10)$$

which may be re-written as:

$$\mathbf{D}_{P_j} = \Delta_F + \Theta_F \wedge \mathbf{P}_j \quad (5.11)$$

Variational parameter constraints should be derived for any feature. In this dissertation, two particular cases are considered: planar and cylindrical features.

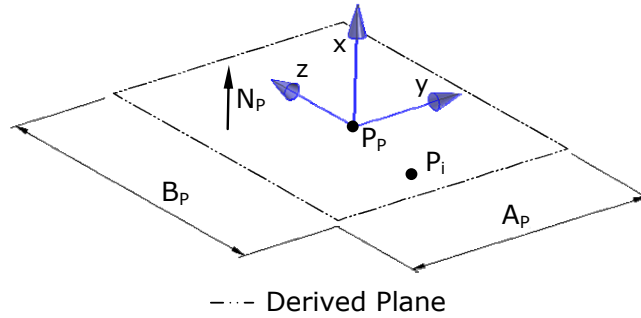


Figure 5.2.a: planar feature

A planar feature is parameterized through a vector, \mathbf{N}_p , a point, \mathbf{P}_p , the length, A_p and the width, B_p , (Figure 5.2.a). Then, the mathematical representation of a planar feature becomes:

$$\text{Plane: } (\mathbf{P}_i - \mathbf{P}_p) \cdot \mathbf{N}_p = 0 \quad (5.12)$$

Equation (5.12) states that any point \mathbf{P}_i belongs to the plane (\mathbf{N}_p , \mathbf{P}_p) if the above condition is satisfied. It can be pointed out that the plane (\mathbf{N}_p , \mathbf{P}_p) corresponds to the derived feature, which may vary within the tolerance zone.

General speaking, planar features may have no rectangular boundary. The relative bounding rectangle is here associated for any planar feature. In addition, this rectangle is oriented along the local \mathbf{Y} and \mathbf{Z} directions (see Figure 5.2.a). Thus, in the following, each planar feature will be assumed as a rectangular planar feature.

For planar features, translational and rotational small displacement parameters become $\Delta_F = (\Delta x_F, 0, 0)$ and $\Theta_F = (0, \beta_F, \gamma_F)$, respectively.

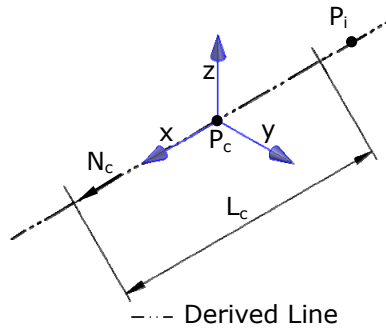


Figure 5.2.b: cylindrical feature

A cylindrical feature is defined by a vector, \mathbf{N}_c , and a point, \mathbf{P}_c and the height, L_c (Figure 5.2.b). The parametric representation is:

$$\text{Line: } \mathbf{P}_i(t) = \mathbf{P}_c + t \cdot \mathbf{N}_c \quad (5.14)$$

where “ t ” is the axis parameter, which represents any point belonging to the cylindrical feature’s axis.

For cylindrical features, translational and rotational small displacement parameters become $\Delta_F=(0, \Delta y_F, \Delta z_F)$ and $\Theta_F=(0, \beta_F, \gamma_F)$, respectively.

Choosing the position and the orientation of the frame attached to the feature is a critical task. Generally speaking, the local \mathbf{X} axis is always coincident with the normal unit vector of the plane, \mathbf{N}_p , or the cylinder axis, \mathbf{N}_c .

The following may happen:

- if the local \mathbf{X} axis of the first datum feature (\mathbf{X}_d) is orthogonal to the \mathbf{X} axis of the analyzed feature, then, the local \mathbf{Y} axis of the analyzed feature corresponds to \mathbf{X}_d . The local \mathbf{Z} axis is the cross vector between \mathbf{X} and \mathbf{Y} ;
- if no datum is assigned or the above condition is not satisfied, then the \mathbf{Y} local axis is any vector belonging to the plane, normal to the same \mathbf{X} axis.

In the latter case, the geometrical condition may be written as:

$$\mathbf{X} \cdot \Psi = 0 \quad (5.16)$$

where Ψ is any vector of the plane having \mathbf{X} as normal vector. The solution to equation (5.16) is achieved by evaluating the *null space* of vector \mathbf{X} (see Annex B). Assuming \mathbf{X} as 1x3 row vector, the null space is a 3x2 matrix. The two columns of such matrix are orthogonal among them and correspond to the \mathbf{Y} and \mathbf{Z} local directions.

Finally, the position of the local frame is the area centroid of the planar feature or the middle point of the cylindrical feature’s axis. Then, assuming \mathbf{P}_m as the position point of the local frame, the 4x4 transformation matrix $\mathbf{T}_{0,j}$ becomes:

$$\begin{cases} \mathbf{T}_{0,j} = \begin{bmatrix} \mathbf{R} & \mathbf{P}_m \\ \mathbf{0} & 1 \end{bmatrix} \\ \mathbf{R} = [\mathbf{X}, \mathbf{Y}, \mathbf{Z}] \end{cases} \quad (5.17)$$

Annex E illustrates how to mathematically describe tolerance zones related to planar and cylindrical features. In addition, a graphical representation is accomplished by means of hyper-domains. Hyper-domains envelop all variational feature configurations, according to tolerance specifications and geometrical parameters. Generally speaking, each point inside the hyper-domain corresponds to a specific determination of Δ_F and Θ_F vectors.

Therefore, having the rotational and translational parameters, the global variational 4x4 matrix, expressed into equation (5.6), is univocally determined.

Next Section provides a numerical procedure to statistically sample points inside the hyper-domain.

5.3.2 Statistical Simulation

The proposed numerical procedure allowing to statistically sample points inside the hyper-domain is based on the following steps (*reject method*):

- evaluating the bounding hyper-box, including the hyper-domain;
- assigning statistical sampling law;
- generating points inside the bounding hyper-box; and,
- rejecting those points which do not belong to the hyper-domain.

Hyper-box is evaluated by calculating the minimum and maximum extends along the parameter space axes. For example, for the cylindrical tolerance zone (see Figure E.2 and Table E.1) the hyper-box is depicted into Figure 5.3.

It should be noted that rotational and translational parameters become independent when related to the hyper-box.

Having the bounding hyper-box, points are sampled by using a Monte Carlo approach. Depending on the adopted statistical law, each Monte Carlo step calculates a point inside the bounding box. If this point belongs also to the hyper-domain, then it is stored. This procedure is stopped when the number of sampled points inside the hyper-domain becomes equal to the number of Monte Carlo simulations³.

This dissertation focuses only on uniform and gaussian distributions (however, other statistical laws may be easily implemented using the proposed procedure).

³ This dissertation does not focus on the efficiency of the reject method.

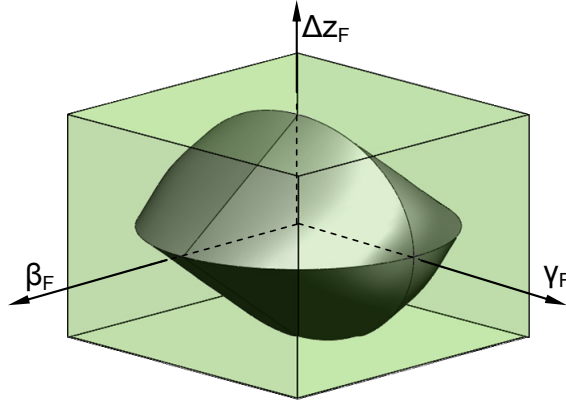


Figure 5.3: hyper-box and hyper-domain (shaded) for cylindrical tolerance zone

For uniform distribution one can write (Rotondi, 2005):

$$\begin{cases} \eta_F = \eta_{F,\min} - \xi \cdot (\eta_{F,\max} - \eta_{F,\min}) \\ \forall \eta_F = \Delta x_F, \Delta y_F, \Delta z_F, \alpha_F, \beta_F, \gamma_F \end{cases} \quad (5.18.a)$$

while for gaussian distribution it is:

$$\begin{cases} \xi = \int f(\eta_F) \cdot d\eta_F \\ \forall \eta_F = \Delta x_F, \Delta y_F, \Delta z_F, \alpha_F, \beta_F, \gamma_F \end{cases} \quad (5.18.b)$$

where ξ is any random value belonging to $[0, 1]$ interval and $f(\eta_F)$ is the gaussian probability density function with zero mean and standard deviation equal to:

$$\begin{cases} \sigma_i = \frac{L_{HB,i}}{T_n} \\ \forall i = \Delta x_F, \Delta y_F, \Delta z_F, \alpha_F, \beta_F, \gamma_F \end{cases} \quad (5.19)$$

where T_n is the natural tolerance range⁴, while $L_{HB,i}$ is the i -th edge length of the hyper-box along the i -th parametric space dimension. For example, for the cylindrical tolerance zone, it is:

$$\begin{cases} \sigma_{\Delta y_F} = \sigma_{\Delta z_F} = \frac{T}{T_n} \\ \sigma_{\beta_F} = \sigma_{\gamma_F} = \frac{2 \cdot T / L_c}{T_n} \end{cases}$$

Equation (5.18.b) is solved for by using the “randn” MatLAB® routine. Moreover, ran-

⁴ Typically, for normal assembly processes, one may assume $T=T_n \cdot \sigma$, where T is the error and σ the standard deviation.

dom values, ξ , are generated by using the “rand” function.

Once all features have been parameterized, the assembly can be made up by defining constraints among parts. In this context, the assembly modeling is accomplished into two main steps. In the first one, a DoFs analysis is performed. In the second step, assembly constraints among parts are introduced, accordingly.

Next Section gives details of the proposed methodology to model and simulate assembly constraints among variational features.

5.4 Assembly Constraint Modeling

In general, during assembly operations an *Object Part* (OP) must be moved to satisfy constraints of *Target Parts* (TP), which are assumed fixed.

In this dissertation, two particular assembly joints are considered: *mate* and *fit* alignments (the following definitions have been derived from (Kim, 2005b)).

Mate condition holds between two planar features and requires that features come in contact. This condition is accomplished by constraining the two unit normal vectors to be opposite each other, and a point of the object plane to belong to the target one (see Figure 5.4).

Fit condition, instead, holds between two cylindrical surfaces. This constraint is accomplished by imposing that target and object axes are parallel, and a point on the object axis belongs to the target one (see Figure 5.5).

Above definitions are valid for ideal mating features. If variational features are introduced, a contact search algorithm has to be adopted.

In fact, looking at Figure 5.6 the mating joint *plane-to-plane II* (full plane-to-plane constraint) becomes a *line-to-plane* constraint in the variational assembly.

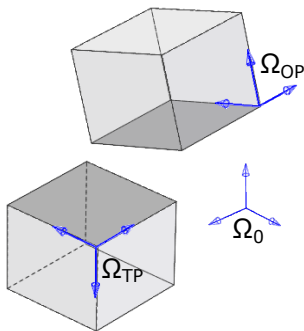


Figure 5.4.a: *mate* assembly joint. Before assembling

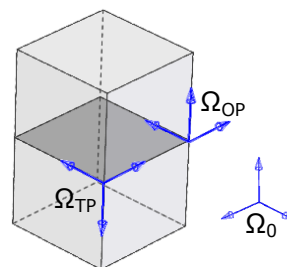


Figure 5.4.b: *mate* assembly joint. After assembling

Chapter 5. Rigid Assemblies

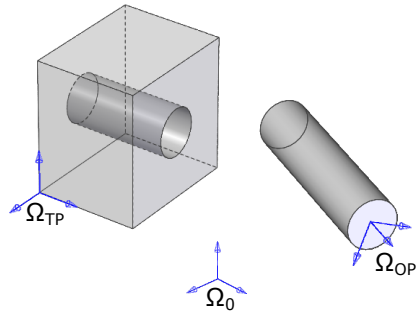


Figure 5.5.a: fit assembly joint. Before assembling

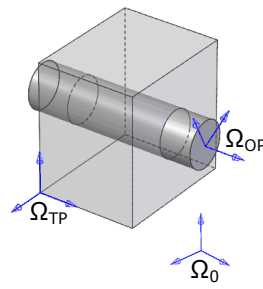


Figure 5.5.b: fit assembly joint. After assembling

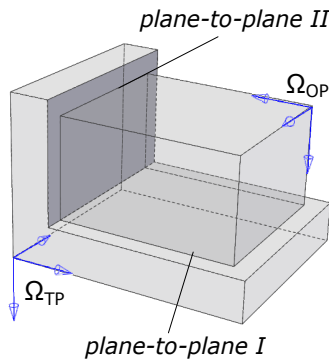


Figure 5.6.a: ideal mating features

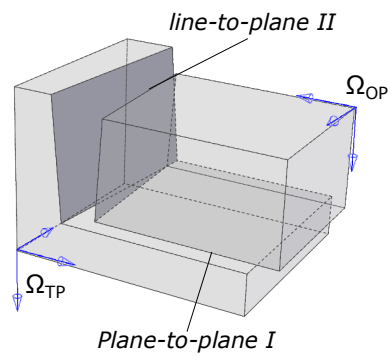


Figure 5.6.b: variational mating features

Therefore, an automatic algorithm able to detect how each constraint updates, when variational features are introduced, is needed.

Generally speaking, in the proposed approach mate and fit constraints are considered as combination of elementary geometry entities (points, lines and planes). Table 5.1 summarizes the five constraint conditions derived as combination of such entities. In this classification, the first and the second features are the object and the target features, respectively. For example, for the line-to-plane joint, line is the object while plane is the target. In addition, twist matrices are written, accordingly, using the notation introduced in Chapter 4.

$$\mathbf{T}_{\text{joint,OP}} = \begin{bmatrix} \mathbf{R}_A & \mathbf{d}_A \\ \mathbf{0} & 1 \end{bmatrix} \quad (5.20)$$

The relative location of the object part, with respect to the target is represented by a 4x4 transformation matrix, as in equation (5.20), where $\mathbf{T}_{\text{joint,OP}}$ is the assembly transformation matrix (in the next Section the meaning of subscripts will be clarified).

Chapter 5. Rigid Assemblies

Joint Type	Twist Matrix	Constraints
<p><i>plane-to-plane</i></p>	$TS_F = \begin{bmatrix} \omega_{F,x} & \mathbf{0} \\ \mathbf{0} & \mathbf{v}_{F,y} \\ \mathbf{0} & \mathbf{v}_{F,z} \end{bmatrix}$	$\begin{cases} \mathbf{x} \cdot \mathbf{N}_P = 1 \\ \forall \mathbf{P} \in \Pi \end{cases}$
<p><i>line-to-plane</i></p>	$TS_F = \begin{bmatrix} \omega_{F,x} & \mathbf{0} \\ \omega_{F,y} & \mathbf{0} \\ \mathbf{0} & \mathbf{v}_{F,x} \\ \mathbf{0} & \mathbf{v}_{F,z} \end{bmatrix}$	$\begin{cases} \mathbf{x} \cdot \mathbf{N}_L = 1 \\ \mathbf{y} \cdot \mathbf{N}_P = 1 \\ \forall \mathbf{P} \in L \end{cases}$
<p><i>point-to-plane</i></p>	$TS_F = \begin{bmatrix} \omega_{F,x} & \mathbf{0} \\ \omega_{F,y} & \mathbf{0} \\ \omega_{F,z} & \mathbf{0} \\ \mathbf{0} & \mathbf{v}_{F,y} \\ \mathbf{0} & \mathbf{v}_{F,z} \end{bmatrix}$	$\begin{cases} \mathbf{x} \cdot \mathbf{N}_P = 1 \\ \forall \mathbf{P} \in \Pi \end{cases}$
<p><i>line-to-line</i></p>	$TS_F = \begin{bmatrix} \omega_{F,x} & \mathbf{0} \\ \mathbf{0} & \mathbf{v}_{F,x} \end{bmatrix}$	$\begin{cases} \mathbf{x} \cdot \mathbf{N}_L = 1 \\ \forall \mathbf{P} \in L \end{cases}$
<p><i>point-to-line</i></p>	$TS_F = \begin{bmatrix} \omega_{F,x} & \mathbf{0} \\ \omega_{F,y} & \mathbf{0} \\ \omega_{F,z} & \mathbf{0} \\ \mathbf{0} & \mathbf{v}_{F,x} \end{bmatrix}$	$\begin{cases} \mathbf{x} \cdot \mathbf{N}_L = 1 \\ \forall \mathbf{P} \in L \end{cases}$

Table 5.1: joint definition

Chapter 5. Rigid Assemblies

In the small displacement hypotheses, equation (5.20) is rearranged as in equation (5.21):

$$\mathbf{T}_{\text{joint,OP}} = \begin{bmatrix} 1 & -\gamma_A & \beta_A & \Delta x_A \\ \gamma_A & 1 & -\alpha_A & \Delta y_A \\ -\beta_A & \alpha_A & 1 & \Delta z_A \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.21)$$

where the triplets $\Delta_A=(\Delta x_A, \Delta y_A, \Delta z_A)$ and $\Theta_A=(\alpha_A, \beta_A, \gamma_A)$ are the small translational and small rotational vectors at assembly level, respectively. Here, one should note an analogy with equation (5.8), defined at part level.

Triplets Δ_A and Θ_A are initially unknown and depend on the six rigid small motion parameters (three rotations and three translations).

This dissertation presents two different procedures allowing to determine the unknown assembly parameters: a *sequential* solver and a *least squares* solver. In first approach, constraint conditions are solved one-by-one into an iterative way. Instead, all constraints are solved for simultaneously in the second approach. The two procedures are analyzed separately in the next.

5.5 Sequential Solver

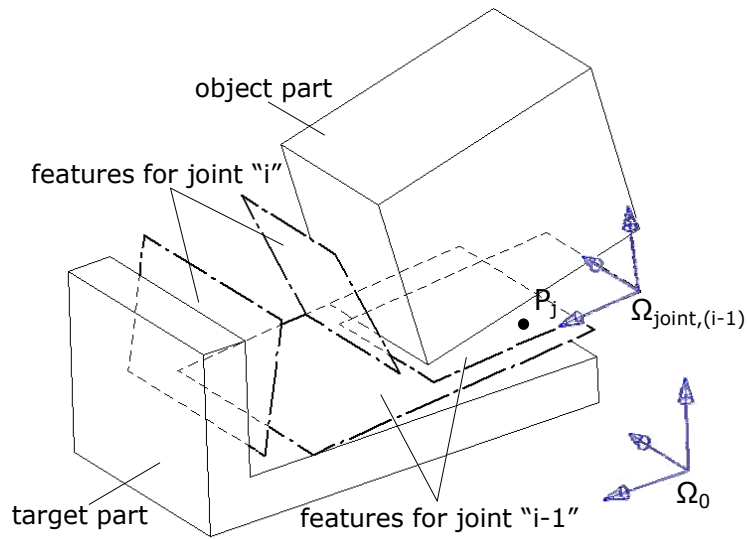
5.5.1 Procedure Overview

Look at Figure 5.7. With respect to the joint “i” (N_{joint} is the total number of joint conditions), a 4x4 transformation matrix, $\mathbf{T}_{0,\text{joint,(i-1)}}$, is calculated, following the joint classification adopted in Table 5.1.

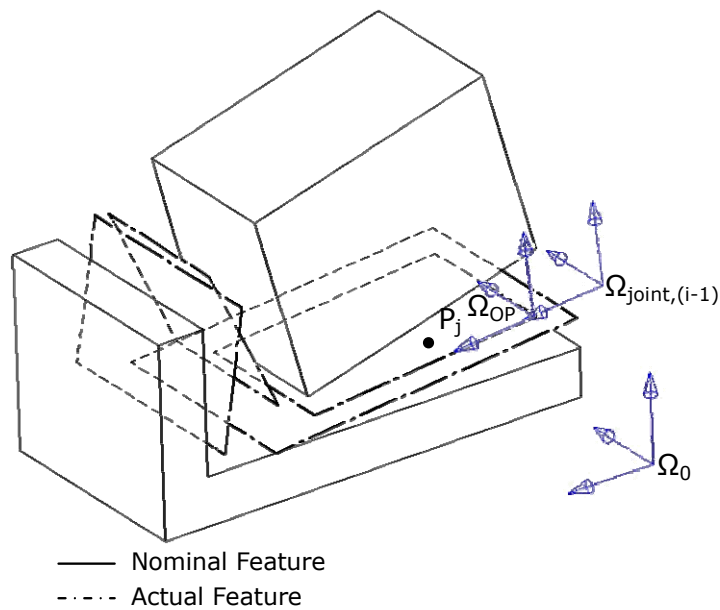
$\mathbf{T}_{0,\text{joint,(i-1)}}$ expresses the local coordinate frame, attached to the target joint feature “i-1”, $\Omega_{\text{joint,(i-1)}}$, into the global coordinate frame, Ω_0 . In addition, an initial list of degrees of freedom, DoF_{i-1} , allowed for the object part, is known with respect to the frame $\Omega_{\text{joint,(i-1)}}$. Once the 4x4 assembly transformation matrix, $\mathbf{T}_{\text{joint,(i-1),OP}}$, is solved, one can write:

$$\mathbf{T}_{0,OP} = \mathbf{T}_{0,\text{joint,(i-1)}} \cdot \mathbf{T}_{\text{joint,(i-1),OP}} \quad (5.22)$$

Equation (5.22) states that any point, \mathbf{P}_j , defined into frame Ω_{OP} is transformed into frame Ω_0 , accounting the assembly joint constraint.



(a) before assembling



(b) after assembling

Figure 5.7: assembly constraint modeling for sequential solver

Since the point $P_{OP,j}$ corresponds to point $P_{\text{joint},(i-1),j}$, defined into frame $\Omega_{\text{joint},(i-1)}$ (before assembling – see Figure 5.7.a), then one can write:

$$P_{OP,j} \equiv P_{\text{joint},(i-1),j} = T_{0,\text{joint},(i-1)}^{-1} \cdot P_{0j} \quad (5.23)$$

and taking into account the global variational 4x4 matrix, $T_{0,0}$:

$$\mathbf{P}_{OP,j} \equiv \mathbf{P}_{\text{joint}(i-1),j} = \mathbf{T}_{0,\text{joint}(i-1)}^{-1} \cdot \mathbf{T}_{0,0} \cdot \mathbf{P}_{0j} \quad (5.24)$$

Finally, considering equations (5.22) and (5.24) one can write:

$$\mathbf{T}_{0,0} = \mathbf{T}_{0,\text{joint}(i-1)} \cdot \mathbf{T}_{\text{joint}(i-1),OP} \cdot \mathbf{T}_{0,\text{joint}(i-1)}^{-1} \cdot \mathbf{T}_{0,0} \quad (5.25)$$

Equation (5.25) represents the global 4x4 transformation matrix which takes into account: (I) joint operation sequence ($\mathbf{T}_{0,\text{joint}(i-1)}$), (II) joint constraint ($\mathbf{T}_{\text{joint}(i-1),OP}$), and (III) part variability ($\mathbf{T}_{0,0}$).

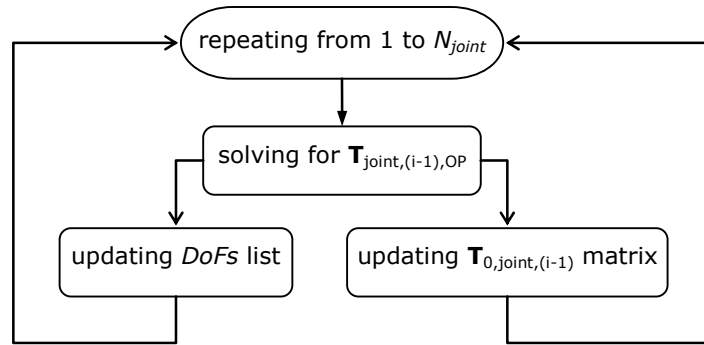


Figure 5.8: sequential solver work-flow

Given equation (5.25), the whole object part geometry may be now transformed and the list of degree of freedom, DoF_i , is updated. Figure 5.8 summarizes the proposed procedure. Next Sections focus on how calculating the $\mathbf{T}_{\text{joint}(i-1),OP}$ matrix and how updating the list of DoFs.

5.5.2 Solving Mate Joints

Looking at Figure 5.9, the aim is to best align the object plane (defined by means of the normal vector \mathbf{N}_{OP} and the point \mathbf{P}_{OP}) with respect to the target one (defined by means of the normal vector \mathbf{N}_{TP} and the point \mathbf{P}_{TP}).

Initially, object geometry is expressed into frame $\Omega_{\text{joint}(i-1)}$. The proposed procedure calculates the assembly parameters separately. In the first step rotational parameters, Θ_A , are solved. Then, translational parameters, Δ_A , are calculated.

$$\min_{\alpha_A, \beta_A, \gamma_A} (J_{\text{mate}}(\alpha_A, \beta_A, \gamma_A)) \quad (5.26)$$

$$J_{\text{mate}}(\alpha_A, \beta_A, \gamma_A) = \|\mathbf{N}_{TP} + \mathbf{N}_{OP}\|$$

With respect to the rotational parameters, two planes are best aligned if the relative angle is minimum. The metric in (5.26) is adopted to account the angle between

vectors \mathbf{N}_{TP} and \mathbf{N}_{OP} . The unit vector \mathbf{N}_{OP} is iteratively updated by means of:

$$\mathbf{N}_{OP} = \begin{bmatrix} 1 & -\gamma_A & \beta_A \\ \gamma_A & 1 & -\alpha_A \\ -\beta_A & \alpha_A & 1 \end{bmatrix} \cdot \begin{Bmatrix} N_{OPx} \\ N_{OPy} \\ N_{OPz} \end{Bmatrix} \quad (5.27)$$

In the mate joint, target and object vectors are opposite each other. Thus, equation (5.26) states that the relative angle is minimum when the norm of the resultant vector between \mathbf{N}_{TP} and \mathbf{N}_{OP} becomes minimum. Moreover, the function J_{mate} , defined in (5.26), is a scalar function of three variables, α_A , β_A , γ_A . MatLAB®'s "fminunc"⁵ routine is here adopted to find out the minimum of that function.

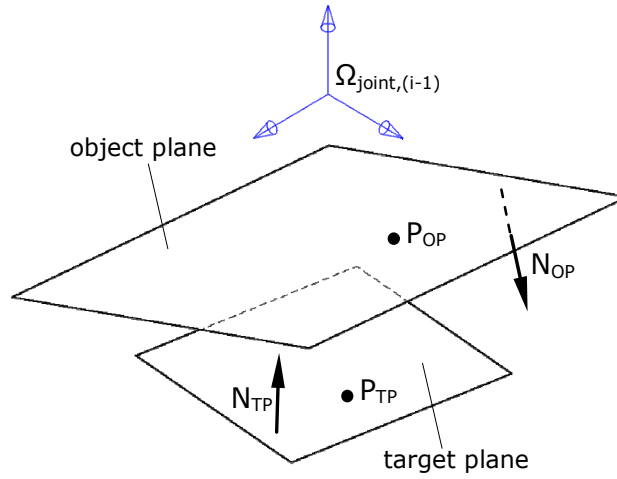


Figure 5.9: mate joint condition

In the second phase, assembly translational parameters are calculated by evaluating on object plane the point $\mathbf{P}_{OP,j}$ closest to target plane.

$$d_j = \frac{(\mathbf{P}_{TP} - \mathbf{P}_{OP,j}) \cdot \mathbf{N}_{TP}}{\mathbf{i} \cdot \mathbf{N}_{TP}} \quad (5.28)$$

$$\Delta \mathbf{i}_A = \begin{cases} \mathbf{i} \cdot \mathbf{N}_{TP} \geq 0 \rightarrow \max_j(d) \\ \mathbf{i} \cdot \mathbf{N}_{TP} < 0 \rightarrow \min_j(d) \end{cases}$$

$$\forall \mathbf{i} = \mathbf{X}, \mathbf{Y}, \mathbf{Z}, \forall j = 1, \dots, N_{map}$$

⁵ Generally speaking, this MatLAB®'s routine finds out the minimum of an un-constrained scalar function of n variables.

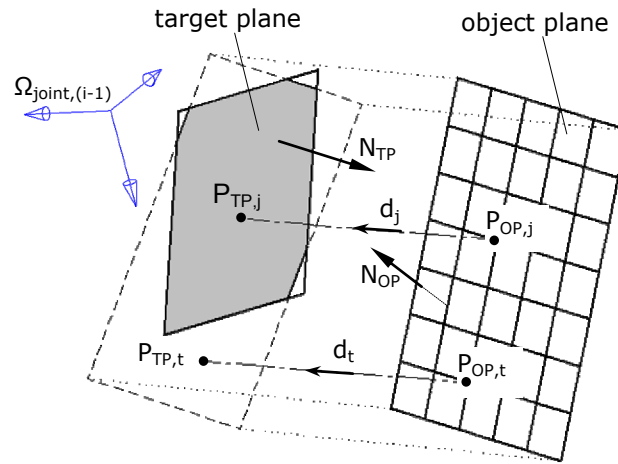
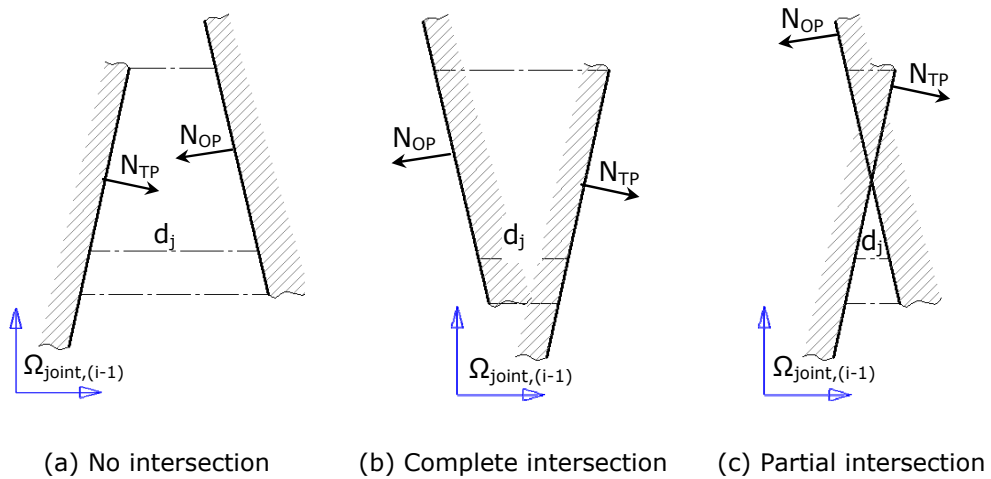


Figure 5.10: mapped mesh for the evaluation of assembly translational parameters in the mate joint condition



(a) No intersection (b) Complete intersection (c) Partial intersection

Figure 5.11: target and object planes for different initial configurations

For this calculation, a mapped mesh is created on object plane (see Figure 5.10). Each point of this mesh is projected on the target plane along the i -th coordinate ($i = X, Y, Z$). If point $\mathbf{P}_{TP,j}$ belongs to the bounding rectangle of target plane, then the i -th translation parameter is obtained by equation (5.28), where N_{map} is the total number of points of the mapped mesh belonging to the object plane, and d_j is the oriented distance from $\mathbf{P}_{OP,j}$ to the target plane.

This procedure assures no intersection between the two planes. In fact, if planes are initially not intersecting each other (Figure 5.11.a), then the oriented maximum distance is negative. Instead, for completely or partially intersecting planes (Figures 5.11.b and 5.11.c, respectively), the oriented maximum distance corresponds to a positive distance. Once both rotational and assembly translational parameters have been calculated, the assembly transformation matrix, $\mathbf{T}_{joint,(i-1),OP}$, is known.

5.5.3 Solving Fit Joints

Now the aim is to align the object axis on the target one (Figure 5.12).

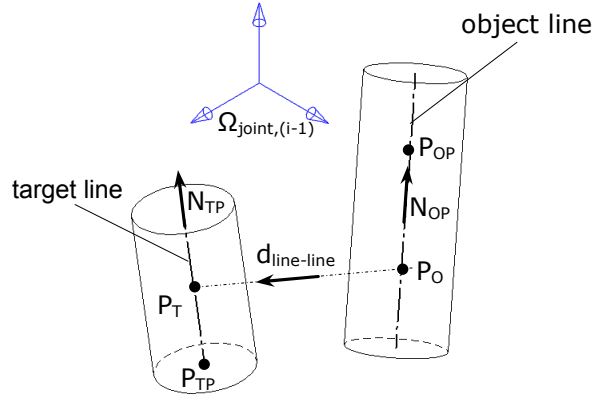


Figure 5.12: fit joint condition

As proposed for the mate alignment, the rotational assembly parameters can be calculated by applying relationship (5.26). Similarly, one can write:

$$\min_{\alpha_A, \beta_A, \gamma_A} (J_{\text{fit}}(\alpha_A, \beta_A, \gamma_A))$$

$$J_{\text{fit}}(\alpha_A, \beta_A, \gamma_A) = \|\mathbf{N}_{\text{TP}} + \mathbf{N}_{\text{OP}}\| \quad (5.29)$$

In this case, vectors \mathbf{N}_{OP} and \mathbf{N}_{TP} define object and target axis directions, respectively.

Assembly translational parameters are calculated by evaluating the minimum distance between two axes. Then, by using the axis definition given into equation (5.14), one can write:

$$\begin{cases} \mathbf{P}_O = \mathbf{P}_{\text{OP}} + t_o \cdot \mathbf{N}_{\text{OP}} \\ \mathbf{P}_T = \mathbf{P}_{\text{TP}} + t_t \cdot \mathbf{N}_{\text{TP}} \end{cases} \quad (5.30)$$

and the distance becomes:

$$d_{\text{line-line}}(t_o, t_t) = \|\mathbf{P}_O - \mathbf{P}_T\| \quad (5.31)$$

The minimum value of the scalar function in (5.31) can be found out by calculating partial derivatives with respect to t_o and t_t , as proposed in (Lengyel, 2003).

Once t_o and t_t are calculated, assembly translational parameters become:

$$\Delta \mathbf{i}_A = (\mathbf{P}_T(t_t) - \mathbf{P}_O(t_o)) \cdot \mathbf{i}, \forall \mathbf{i} = \mathbf{X}, \mathbf{Y}, \mathbf{Z} \quad (5.32)$$

Equation (5.32) states that assembly translational parameters are the components, along the joint coordinate frame, of the vector $\mathbf{P}_T(t_t) - \mathbf{P}_O(t_o)$.

5.5.4 Updating Joint Matrix

At the beginning (when no joint is assigned), the 4x4 transformation matrix, $\mathbf{T}_{0,\text{joint},(i-1)}$, is an identity matrix. Then, step-by-step, that matrix is updated by considering the number of contact points between the object and the target planes (for mate conditions), or between the object and target lines (for fit joint conditions).

With respect to the mate joint, at maximum three not aligned points are counted. In particular, for the point-to-plane condition there is a single point contact. Whereas, two points and three points may be counted for the line-to-plane and plane-to-plane joint conditions, respectively.

In order to calculate the contact points, \mathbf{P}_{cnt} , between the object and target plane, the following equation is checked for:

$$\mathbf{P}_{\text{cnt}} : \left| (\mathbf{P}_{\text{TP}} - \mathbf{P}_{\text{OP},j}) \cdot \mathbf{N}_{\text{TP}} \right| \leq \varepsilon, \forall j = 1, \dots, N_{\text{map}} \quad (5.33.a)$$

where ε^6 is a small error allowed to account numerical errors into calculation. It is clear that the size of the mesh, belonging to the object plane, could influence final results. To reach more accuracy a finer mesh is needed.

With respect to the fit joint, at maximum two not aligned points are counted. In particular, one point and two points for the point-to-line and line-to-line joints, respectively.

If object and target lines are parallel and coincident each-other, then, the two contact points are defined as:

$$\mathbf{P}_{\text{cnt}} = \mathbf{P}_{\text{TP}} + t \cdot \mathbf{N}_{\text{TP}}, \forall t = 0,1 \quad (5.33.b)$$

whereas, for two non-parallel lines the geometric relationships stated into equations (5.30) and (5.31) are here used. For instance, one can write:

$$d_{\text{line-line}}(t_o, t_t) = \|\mathbf{P}_O - \mathbf{P}_T\| \leq \varepsilon \rightarrow \mathbf{P}_{\text{cnt}} = \mathbf{P}_O \equiv \mathbf{P}_T \quad (5.33.c)$$

Once contact points have been calculated, the geometric rules expressed into Table 5.1 are here applied to determine the 4x4 joint matrix.

5.5.5 Updating DoFs List

DoFs allowed for the object part are updated step-by-step by performing a motion analysis as illustrated in Chapter 4. Therefore, Screw Theory is here used to find out the DoFs allowed for the object part. Generally speaking, with respect to the joint

⁶ Typically, ε is equal to 10^{-6} or less.

“i”, the Graph-based Assembly is made of i-1 independent loops (see Figure 5.13.a). Looking at Figure 13.b, the proposed procedure is based on the following steps:

- writing twist matrices for all joints from 1 to “i”;
- expressing above twist matrices into coordinate frame $\Omega_{\text{joint},(i)}$;
- performing motion analysis by using the numerical procedure described into Chapter 4 (getting the object motion matrix, $U_{O,T}$); and,
- updating list of DoFs by analyzing the $U_{O,T}$ matrix.

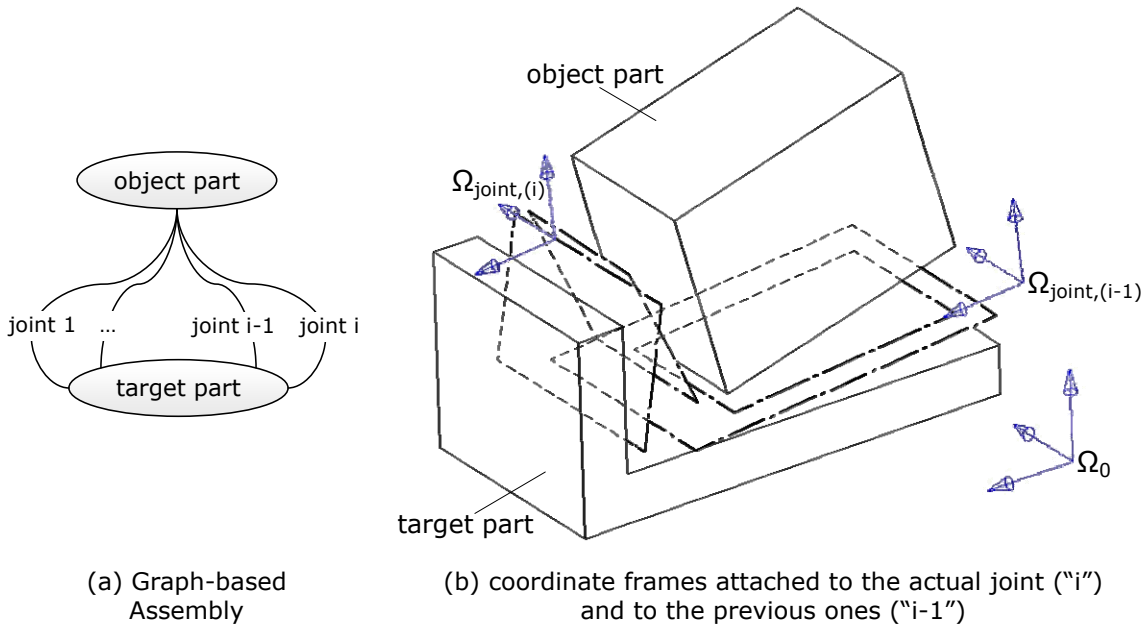


Figure 5.13: graph and definition of coordinate frames for updating DoFs

A critical task is the analysis of the motion matrix, $U_{O,T}$, to automatically obtain the list of independent DoFs. Generally speaking, $U_{O,T}$ is a 6x6 square matrix, as expressed into equation (5.34). Each row corresponds to a specific DoF.

$$U_{O,T} = \begin{bmatrix} \omega_{1i} & \mathbf{0} \\ \omega_{2i} & \mathbf{0} \\ \omega_{3i} & \mathbf{0} \\ \mathbf{0} & \mathbf{v}_{1i} \\ \mathbf{0} & \mathbf{v}_{2i} \\ \mathbf{0} & \mathbf{v}_{3i} \end{bmatrix} \quad (5.34)$$

where ω_{1i} , ω_{2i} , ω_{3i} , and \mathbf{v}_{1i} , \mathbf{v}_{2i} , \mathbf{v}_{3i} are the directions around and along which the object part can move, respectively. Generally speaking, these directions do not coincide with those of the coordinate frame.

Therefore, a rotation matrix, here called *canonical* rotation matrix, $R_{\text{joint},(i),c}$,

Chapter 5. Rigid Assemblies

must be calculated. First of all, the maximum number of independent directions, \mathbf{V} , is extracted from the $\mathbf{U}_{O,T}$ matrix. If this number is equal to three, then the canonical rotation matrix becomes:

$$\mathbf{R}_{\text{joint}(i),c} = \mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3] \quad (5.35)$$

where \mathbf{v}_j is the j -th independent direction. When the number of independent directions is less than three (this means that the rank of \mathbf{V} is less than three), then the canonical rotation matrix can be calculated by evaluating the null space of \mathbf{V} .

The transpose of the canonical rotation matrix allows to express the motion matrix $\mathbf{U}_{O,T}$ into the so-called *canonical coordinate frame*. Formally, one can write:

$$\mathbf{U}_{O,T,c} = \mathbf{R}_{c,\text{joint}(i)} \cdot \mathbf{U}_{O,T} \quad (5.36)$$

where $\mathbf{U}_{O,T,c}$ is the canonical motion matrix.

Below the MatLAB® pseudo-code to calculate the canonical matrix and the list of DoFs starting from any motion matrix.

```
%- find the list of DoFs and the canonical matrix
function [DoF, Rjoint_c]=DoFfromMotion(U)

%- find the V matrix
V=[];
for I=1:size(U,1)
    if any(U(I,1:3)~=0) %- rotational
        if rank([V;U(I,1:3)])==size([V;U(I,1:3)],1)
            V=[V; U(I,1:3)];
        end
    else %- translational
        if rank([V;U(I,4:6)])==size([V;U(I,4:6)],1)
            V=[V; U(I,4:6)];
        end
    end
end

%- create the 3x3 canonical rotation matrix
if size(V,1)==3
    Rjoint_c=V';
else
    nullSpace=null(V);
    Rjoint_c=V;
    Rjoint_c=[Rjoint_c; nullSpace'];
    Rjoint_c=Rjoint_c';
end

%- transform the motion matrix into the canonical frame
Uc=getcanonicalU(U, Rjoint_c);

%- get list of DoFs into the canonical frame
DoF=getDoF(Uc);
```

5.6 Least Squares Solver

The aim is to determine the best assembly configuration of the object part with respect to all assembly joint constraints. Therefore, an optimization algorithm is used. The objective function is built up by evaluating the distances of object plane and object axis with respect to target plane and target axis, respectively.

Looking at Figures 5.9 and 5.12, distances from object and target planes are:

$$\text{dist}_{\text{mate},i} = \left| (\mathbf{P}_{\text{TP}} - \mathbf{P}_{\text{OP},i}) \cdot \mathbf{N}_{\text{TP}} \right| \quad \forall \mathbf{P}_{\text{OP},i} \in \text{object plane}, \forall i = 1,2,3 \quad (5.37)$$

whereas, distances from object and target axes are:

$$\text{dist}_{\text{fit},i} = \left\| (\mathbf{P}_{\text{TP}} - \mathbf{P}_{\text{OP},i}) \wedge \mathbf{N}_{\text{TP}} \right\| \quad \forall \mathbf{P}_{\text{OP},i} \in \text{object line}, \forall i = 1,2 \quad (5.38)$$

With respect to equation (5.37), the three selected points on the object plane must not be aligned.

The point $\mathbf{P}_{\text{OP},i}$, belonging to object plane or object axis, is iteratively updated by means of relation (5.39).

$$\mathbf{P}_{\text{OP},i} = \begin{bmatrix} 1 & -\gamma_A & \beta_A & \Delta x_A \\ \gamma_A & 1 & -\alpha_A & \Delta y_A \\ -\beta_A & \alpha_A & 1 & \Delta z_A \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{Bmatrix} \mathbf{P}_{\text{OPx},i} \\ \mathbf{P}_{\text{OPy},i} \\ \mathbf{P}_{\text{OPz},i} \\ 1 \end{Bmatrix} \quad (5.39)$$

Thus, in the least squares sense, the assembly objective function may be written as in equation (5.40), where N_{fit} and N_{mate} are the number of fit and mate joint conditions, respectively.

$$J_{\text{over}}(\alpha_A, \beta_A, \gamma_A, \Delta x_A, \Delta y_A, \Delta z_A) = \sum_{i=1}^{2 \cdot N_{\text{fit}}} \text{dist}_{\text{fit},i}^2 + \sum_{j=1}^{3 \cdot N_{\text{mate}}} \text{dist}_{\text{mate},j}^2 \quad (5.40)$$

Equation (5.40) takes into account simultaneously all small motion parameters. This function can be minimized by using any non-linear least squares routine, such as the MatLAB®'s "lsqnonlin".

5.7 Data Structure for Tolerance Analysis Software

As mentioned above in the dissertation, all numerical procedures have been tested and implemented in MatLAB® environment. This Section shows the general structure to manage input data and store simulation results. One efficient way to handle such a data in MatLAB® is by using structured array.

5.7.1 Variational Features

Figure 5.14 shows the main structure to manage input data needed to generate variational features. For each part (“part i”), “plane” and “line” are given in input.

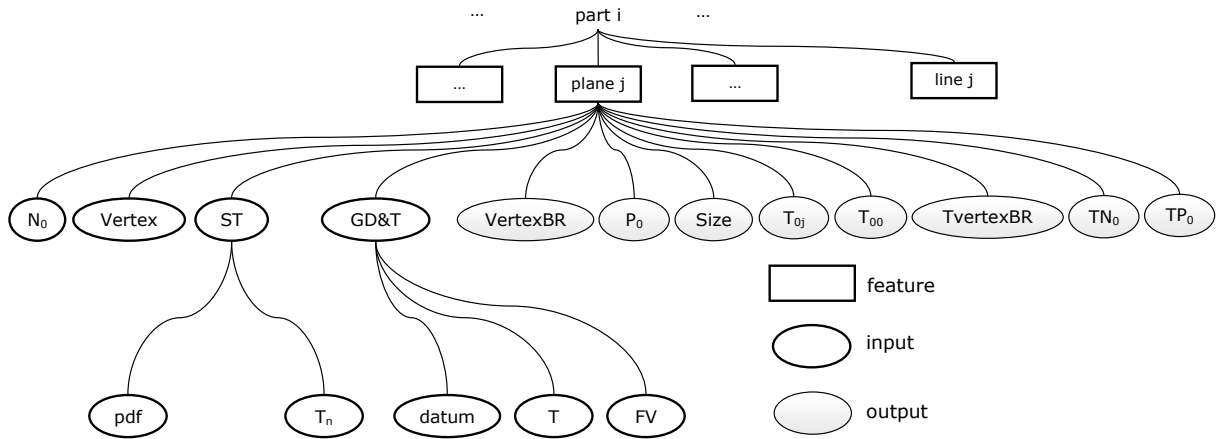


Figure 5.14: data structure for variational feature generation

Then, for each feature (Figure 5.14 depicts the structure just only for “plane” features; the same structure may be drawn for “line” features) input and output fields are available. They are:

- N₀: feature unit vector (normal vector to the plane of line vector) - input;
- Vertex: list of vertices - input;
- pdf: probability density function used to perform the statistical analysis - input;
- T_n: tolerance natural range - input;
- datum: list of datum - input;
- T: tolerance value - input;
- FV: list of feature variational parameters (Δ_F and Θ_F) - input;
- VertexBR: list of vertices of the boundary rectangle (only for planar features) - output;
- P₀: point defining the location of the feature - output;
- Size: feature sizes - output;
- T_{0j}: 4x4 transformation matrix defining the frame attached to the feature into the global one - output;
- T₀₀: global 4x4 variational transformation matrix - output;
- TvertexBR: list of “VertexBR” transformed by means of T₀₀ matrix - output;
- TN₀: N₀ transformed by means of T₀₀ matrix - output; and,
- TP₀: P₀ transformed by means of T₀₀ matrix - output.

As example, below a MatLAB® code defining a planar feature, with unit normal vector directed along global **Z** axis, four input vertices and a datum assignment, is

Chapter 5. Rigid Assemblies

shown. The name of the feature is “PL1”, while the related datum name is “LN1”. This feature belongs to the part identified with the integer “1”.

```
%- MatLAB® code for feature definition
InputData(1).PLANE.PL1.NO=[0 0 1];
InputData(1).PLANE.PL1.VERTEX=[0 0 0;100 0 0;100 100 0;0 100 0];
InputData(1).PLANE.PL1.PDF='Normal';
InputData(1).PLANE.PL1.TN=3;
InputData(1).PLANE.PL1.DATUM='LN1';
InputData(1).PLANE.PL1.TOL=0.5;
InputData(1).PLANE.PL1.FV=[1 0 0 0 1 1];
InputData(1).PLANE.PL1.VERTEXBR=[ ];
InputData(1).PLANE.PL1.PO=[ ];
InputData(1).PLANE.PL1.SIZE=[ ];
InputData(1).PLANE.PL1.TOj=[ ];
InputData(1).PLANE.PL1.T00=[ ];
InputData(1).PLANE.PL1.TVERTEXBR=[ ];
InputData(1).PLANE.PL1.TN0=[ ];
InputData(1).PLANE.PL1.TP0=[ ];
```

5.7.2 Assembly Constraints

Figure 5.15 shows the main structure to manage input data to perform the assembly simulation.

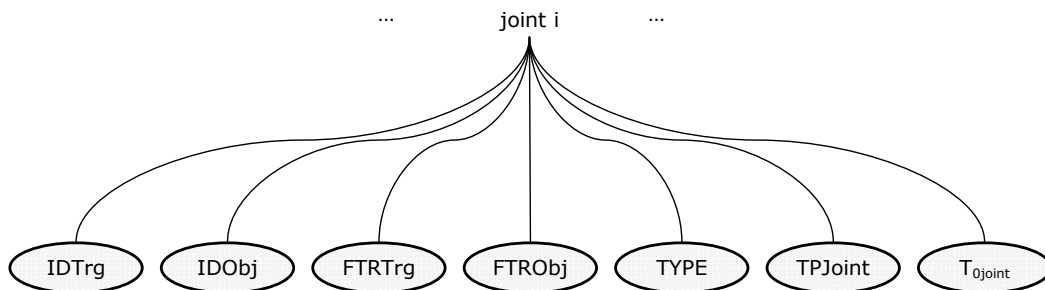


Figure 5.15: data structure for assembly constraint simulation

With respect to the joint “i”, the following fields are defined:

- IDTrg: identification number for target part - input;
 - IDObj: identification number for object part - input;
 - FTRTrg: feature name for target part - input;
 - FTRObj: feature name for object part - input;
 - TYPE: mate or fit joint - input;
 - TPJoint: joint type calculated following the classification of Table 5.1 - output;
- and,

Chapter 5. Rigid Assemblies

- $T_{0\text{joint}}$: 4x4 transformation matrix defining the frame attached to the joint to the global one - output.

As example, below a MatLAB® code defining a fit joint between the features “LN1” (belonging to part “1”) and “LN2” (belonging to part “2”) is shown. The joint is identified with the integer “3”.

```
%- MatLAB® code for assembly constraint definition
AssemblyData(3).IDTrg=1;
AssemblyData(3).IDObj=2;
AssemblyData(3).FTRTrg='LN1';
AssemblyData(3).FTRObj='LN2';
AssemblyData(3).TYPE='Fit';
AssemblyData(3).TYPEJOINT='';
AssemblyData(3).TOJOI=[ ];
```

5.8 Examples

5.8.1 Sequential Solver: Three-mate Joints

Figure 5.16.a shows a two-part assembly. Three mate joints are defined.

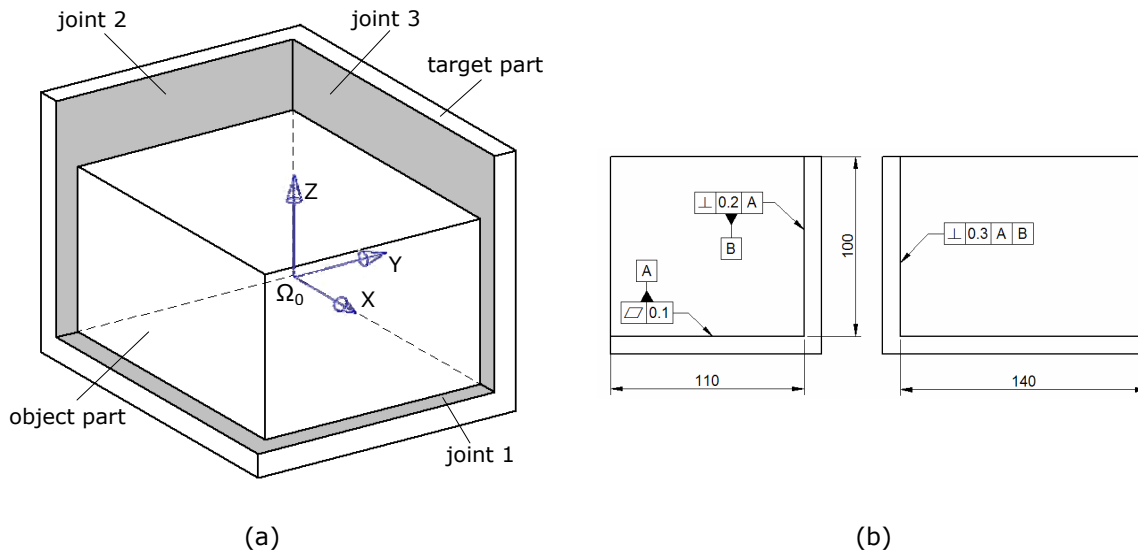


Figure 5.16: example. Three-mate joints

The aim of this example is to show how the proposed sequential solver allows to simulate different assembly sequences. Table 5.2 shows the six feasible assembly sequences.

ID	Joint Order
I	joint 1+joint 2+joint 3
II	joint 1+joint 3+joint 2
III	joint 2+joint 1+joint 3
IV	joint 1+joint 3+joint 1
V	joint 3+joint 1+joint 2
VI	joint 3+joint 2+joint 1

Table 5.2: three-mate joints.
Feasible assembly sequences

Mating features on object part are supposed ideal (no input variation is assigned), while the GD&T tolerance scheme for the target part is drawn in Figure 5.16.b.

Only one Monte Carlo simulation was performed. This means that just only one variational geometry configuration was adopted. Figure 5.17 depicts final assembly geometry for all six assembly sequences (only mating features are drawn for target parts). It should be noted how final results are strongly different each-other, as expected.

It is of interest analyzing step-by-step what happens for a specific assembly sequence. Figure 5.18 shows the three configurations related to the sequence “V”. It can

Chapter 5. Rigid Assemblies

be noted that joint 3 is a plane-to-plane type, while joint 1 and 2 becomes line-to-plane and point-to-plane types, respectively. In addition, the contact at mating features interfaces is assured: no penetration is allowed.

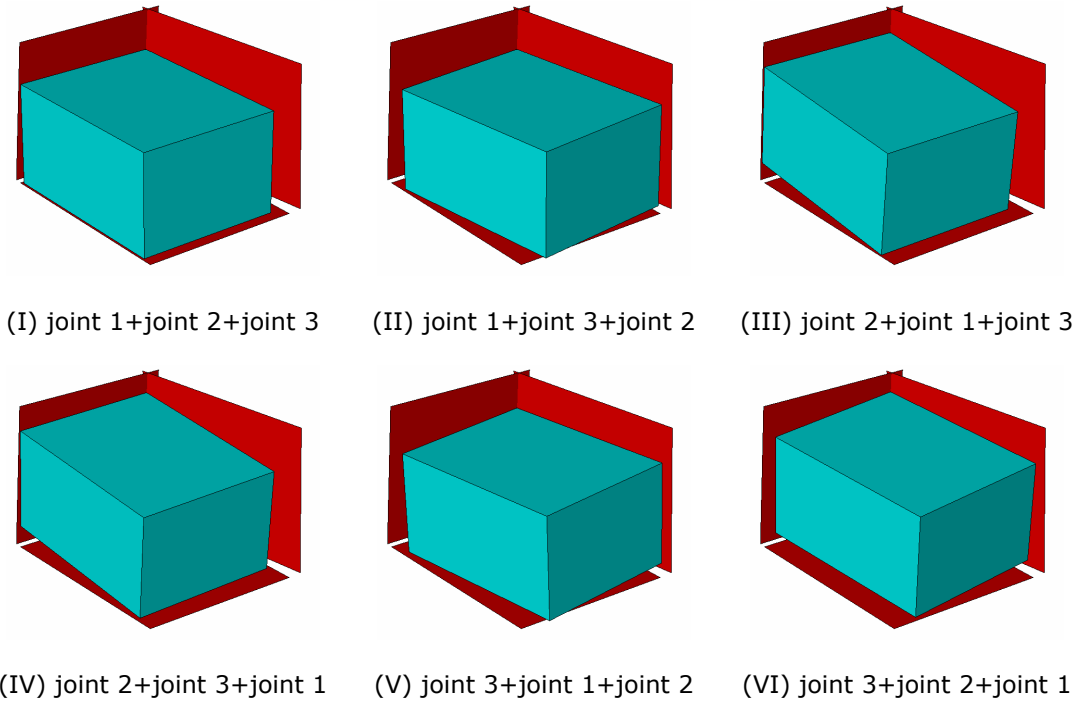


Figure 5.17: three-mate joints. Assembly geometry for different assembly sequences. Variation scale factor = 100

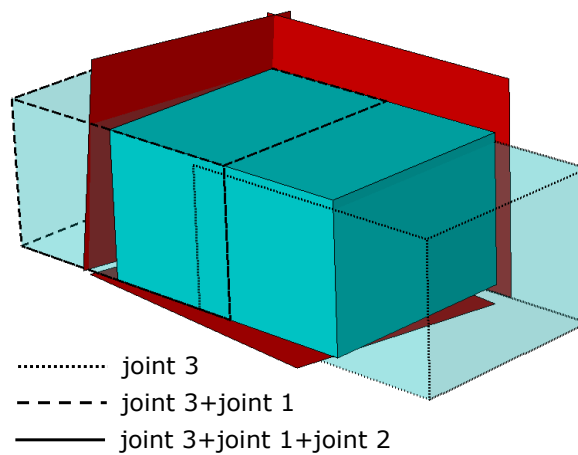


Figure 5.18: analysis of assembly sequence "V"

5.8.2 Sequential Solver: Mate and Fit Joints

Figure 5.19.a shows a two-part assembly. One fit and one mate joint are defined.

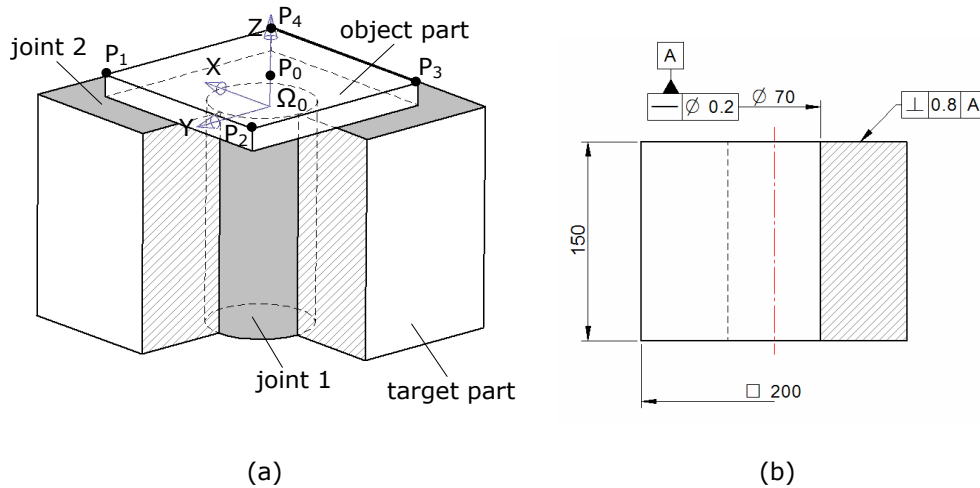


Figure 5.19: example. Fit and mate joints

Mating features on object part are supposed ideal (no input variation is assigned), while the GD&T tolerance scheme for the target part is drawn in Figure 5.19.b.

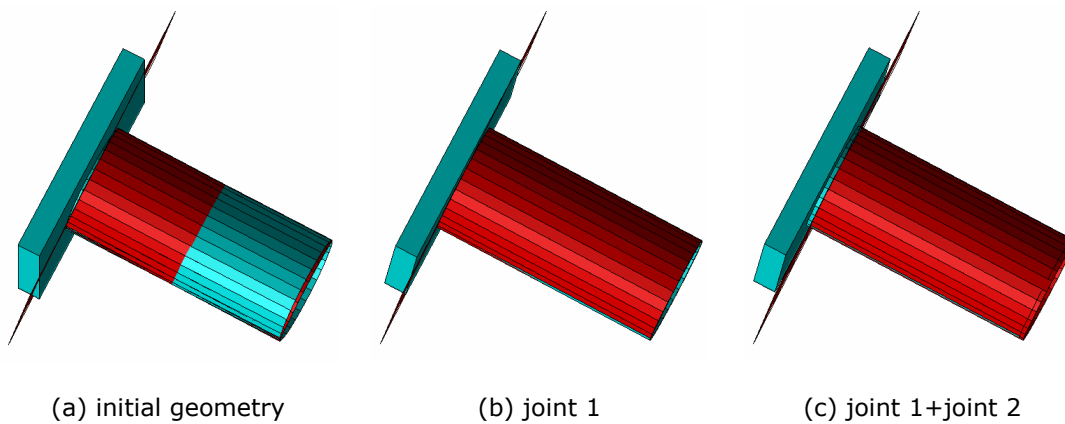


Figure 5.20: fit and mate joints. Assembly geometry configurations.
Variation scale factor = 20

First of all, Figure 5.20 shows the only one feasible assembly sequence: joint 1+joint 2, with respect to one specific Monte Carlo simulation. It can be noted that parts are initially intersecting each other (see Figure 5.20.a). Once the fit joint 1 is introduced, axes of pin and hole are best aligned and then the joint becomes a line-to-line type (see Figure 5.20.b). Moreover, the mate between the two planar features is updated accordingly into a point-to-plane type (see Figure 5.20.c).

Chapter 5. Rigid Assemblies

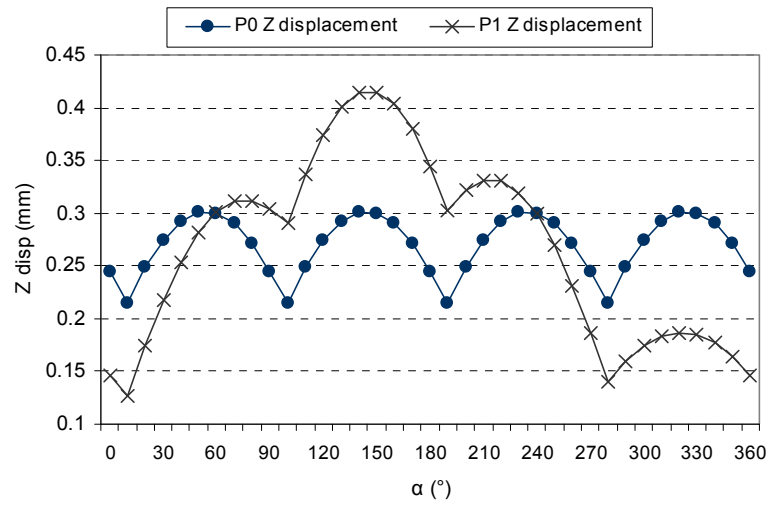


Figure 5.21: fit and mate joints. Analysis of points P_0 and P_1 when object part rotates around the ISA axis

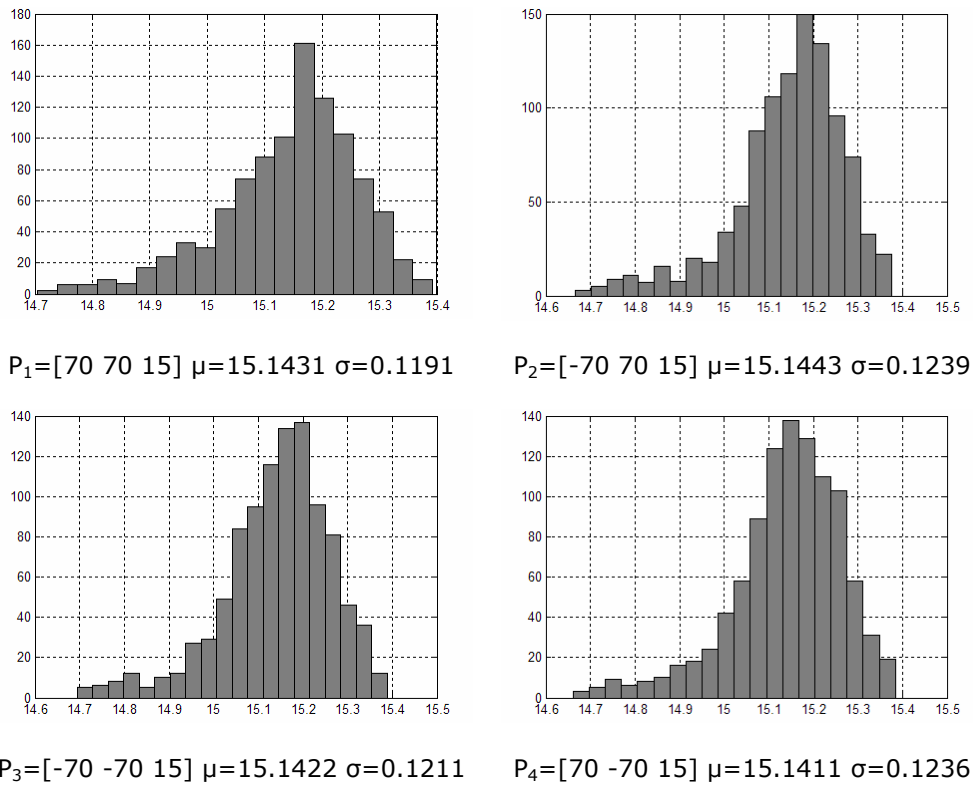


Figure 5.22: fit and mate joints. Histograms of frequencies. 1000 Monte Carlo simulations

The motion matrix, $U_{O,T}$, is not empty for this combination of joints. Therefore, it is of interest analyzing what happens when the object part rotates around the ISA axis

(corresponding to the target axis). Figure 5.21 depicts the displacements of two points, having nominal position $P_0=[0 \ 0 \ 15]$ and $P_1=[70 \ 70 \ 15]$, along the global Z direction. “ α ” is the object part rotation angle.

It can be pointed out that in order to avoid penetration between target and object mating planes, the object part translates along the ISA axis. For example, the point P_0 moves from a minimum of 0.21 mm to a maximum of 0.32 mm. The point P_1 moves from a minimum of 0.13 mm to a maximum of 0.41 mm. This result is an important issue when the KC to be delivered is directed along the global Z axis of the assembly.

Finally, a Monte Carlo analysis, with 1000 simulations, has been performed (no object part rotation has been assumed). Figure 5.22 shows the histograms related to the four corners P_1, P_2, P_3 and P_4 (see also Figure 5.19.a).

5.8.3 Least Squares Solver Example

The least squares solver has been tested on a two-part assembly. The object part has three pins fitting the related holes on the target part. Moreover, a mate joint among planar surfaces has to be satisfied (Figure 5.23.a).

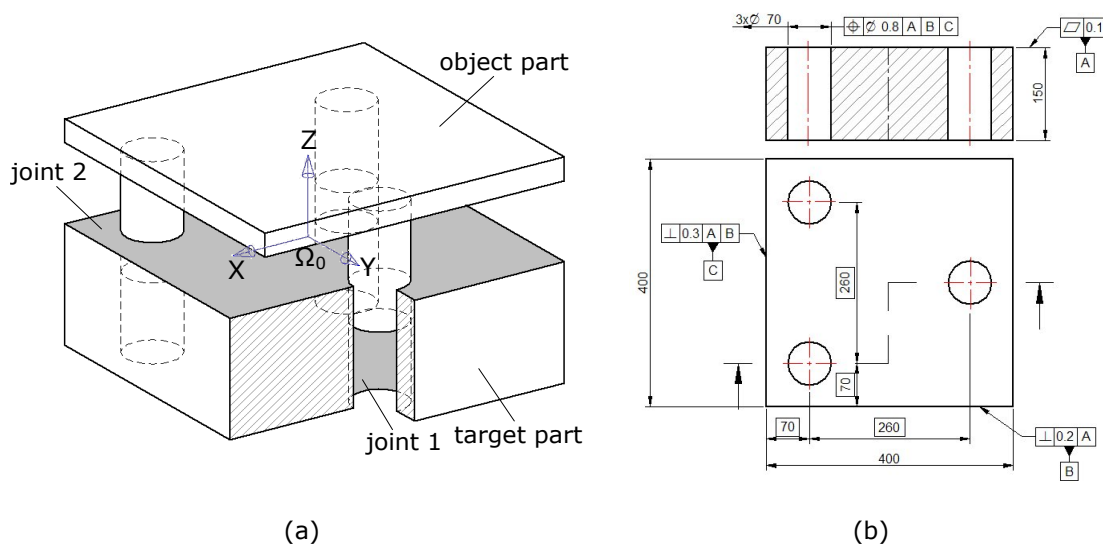


Figure 5.23: example. Least squares solver application

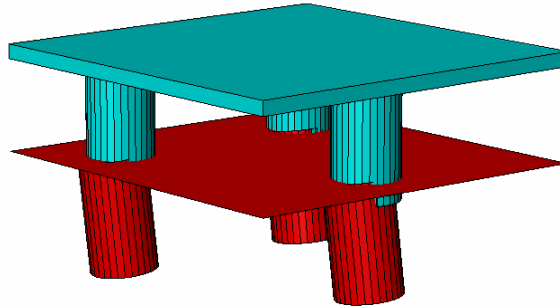
The aim of this example is to fit *simultaneously* the three pins into the related holes. Obviously, the sequential solver is now inadequate.

Since the assembly is over-constrained with respect to translations and rotations along and around global X and Y axes, respectively, the least squares solver gives only the best fit configuration. In addition, assembling features may intersect each-other.

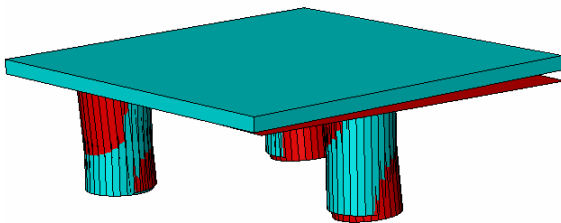
Two assembly configurations have been analyzed. In first one (I) the object part

Chapter 5. Rigid Assemblies

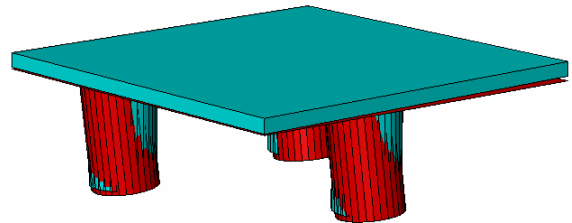
has initially all six DoFs allowed. Instead, in configuration (II), object part may translate only along the global **Z** direction. Moreover, holes present a position error with respect to the related datum as in Figure 5.23.b.



(a) initial geometry



(b) assembly configuration "I"



(c) assembly configuration "II"

Figure 5.24: least squares solver application. Assembly geometry configurations.
Variation scale factor = 20

Figures 5.24 show the final assembly configurations. It can be highlighted that parts may penetrate each other. Generally speaking, all this is true for all over-constrained assemblies. To reach results closer to real phenomena, compliant of components should be introduced into the numerical model.

5.10 Summary

This Chapter presented a methodology, called SVA-TOL, to do tolerance analysis of rigid part assemblies. The methodology is based on two main steps.

In the first one, variational features are modeled according to GD&T or ISO tolerance specifications. In this way the whole assembly is modeled as a set of parameterized variational features. In particular, the focus is on two specific features: planar and cylindrical.

Once variational features have been created, constraints among assembly features are introduced. In this dissertation mate and fit joints have been modeled among planar and cylindrical geometries, respectively. A sequential solver and a least squares solver have been illustrated.

With respect to the sequential solver, each assembly constraint is solved for according to DoFs allowed to the object part. Best alignment among object and target features is performed in two consecutive steps: firstly, assembly rotational parameters are calculated by minimizing a scalar function; then, assembly translational parameters are evaluated by using a linear contact algorithm. The list of DoFs allowed to the object part is automatically updated by performing a motion analysis by means of Screw Theory. The linear contact detection algorithm requires the definition of a mesh on the object feature. Results may be affected by the user-selected mesh size. To reach more accuracy a finer mesh is required. Obviously, finer meshes are much more expensive in terms of simulation run-time. This procedure, here applied only for planar features, could be extended to any 3D complex surface. Triangular or mixed mesh could be generated in that case.

With respect to the least squares solver, all constraint equations are solved simultaneously. This approach may allow penetration of part being assembled. The least squares solver is typically used for those assemblies having at least one over-constrained direction. To get results closer to real phenomena also flexibility of parts should be included into the model.

Chapter 6 will show the SVA-FEA methodology do to tolerance analysis of compliant assembly. Moreover, the concept of variational feature will be applied also in Chapter 7 to model and generate free shape features.

Chapter 6.

LINEAR ANALYSIS OF COMPLIANT ASSEMBLIES¹

The general hypothesis of ideal rigid-part assembly may be inadequate when parts exhibit a wide flexibility. All this is especially true for real assembly processes involving sheet-metal parts, as in aerospace or automobile applications.

This Chapter focuses on the topic of tolerance analysis of compliant part assemblies and provides a methodology, called SVA-FEA (Statistical Variation Analysis & Finite Element Analysis), aiming to simulate both single- and multi-station assembly configurations, accounting flexibility of components.

Two FEA runs are required to solve each assembly sub-station. In particular, in the first one fastening and fixturing forces are calculated. Then, these forces are applied into the second run to simulate the final elastic spring-back.

Two significant case studies are presented. Results are compared with ones coming from a commercial CAT package, working on the same assemblies.

¹ This Chapter is based on:

(I) Gerbino S., Patalano S., Franciosa P., *Statistical Variation Analysis of Multi-Station Compliant Assemblies based on Sensitivity Matrix*, Int. Journal Computer Applications in Technology, Vol. 33, No. 1, pp. 12-23, 2008.

(II) Franciosa P., Gerbino S., Patalano S., *Variation Analysis of Compliant Assemblies: A Comparative Study of a Single-Station Assembly*, Proc. of the XX Int. INGEGRAF, Valencia (Spain), June 4-6, 2008.

(III) Franciosa P., Gerbino S., Patalano S., *Variation Analysis of Compliant Assemblies: A Comparative Study of a Multi-Station Assembly*, Proc. of the Int. ADM-INGEGRAF, Lugo (Spain), June 10-12, 2009.

6.1 Introduction

As stated more than once in the present dissertation, during assembly phase, variations propagate part-to-part. This propagation is strongly related to the constraint state of the assembly. Chapter 4 and 5 pointed out that over-constraint configurations are dangerous because they make difficult assembling parts or they induce stresses and deformations into parts being assembled.

However, in real industrial applications over-constraints are often needed for those parts exhibiting high flexibility (see sheet-metal parts). In fact, just the gravity load may deform parts when located on fixturing frames. Over-constraints are so necessary to force components to their nominal configuration.

On the other hand, it was reported (Camelio, 2004b) that fixturing variability highly influences final assembly deviations. Therefore, well-understanding what is the final behavior of the final assembly during the early design stage, accounting any constraint scheme and part deformation, is a crucial task to be achieved.

This Chapter focuses on the modeling of single- and, more in general, multi-station processes involving sheet-metal parts, taking into account part and process variability and flexibility of parts being assembled. The latter need is accomplished through a FEM approach: forces and elastic displacements are calculated by solving for a linear FE model.

Starting from the classical *Stream of Variation* approach, the aim is to numerically evaluate what happens into a specific assembly station considering the propagation of variability from one station to another.

Each assembly station is solved by means of two consecutive FEA runs. The first one calculates the forces at fixturing and fastening points, while the second run simulates the elastic spring-back.

6.2 Methodology Overview

The general SVA-FEA work-flow is depicted into Figure 6.1. Starting from the nominal assembly geometry, imported from a CAD modeler, the FE model is created accordingly: shell elements with bending and extensional behavior are created. Let \mathbf{V} and \mathbf{F} be the matrix of nodes and elements, respectively; \mathbf{V} is a $N_n \times 3$ rectangular matrix, where N_n is the total number of nodes.

For each sub-station, assembly operations have to be defined. In particular, four sets of *Key Points* (KPs) are identified: *fixturing points* to model fixture tools; *fastening points* to model fastening operations; *contact points* to model the contact between parts to avoid part-to-part penetrations; and, *inspection points* to model the fixture frame used

to do final measurements on the assembly. For each sub-station, these points must be assigned, accordingly. Moreover, statistical input data are provided in terms of mean and standard deviation.

Once input data are correctly assigned, output data, in terms of statistical displacements, are given by solving two consecutive FEA runs. It should be noted that the final spring-back at assembly level, in terms of mean and standard deviations, is calculated in a closed-form without using the expensive Monte Carlo simulation: in the proposed approach, input data are linearly related to output data.

The whole software architecture is based on MatLAB® environment which drives, in background mode, the MSC NASTRAN®² solver.

The assembly process adopted in this dissertation is the classical PCFR cycle, proposed also in (Chang, 1997). With respect to the specific assembly station, parts are positioned onto the fixture frame, then they are clamped and fastened and, finally, they are released, reaching the final sub-assembly configuration.

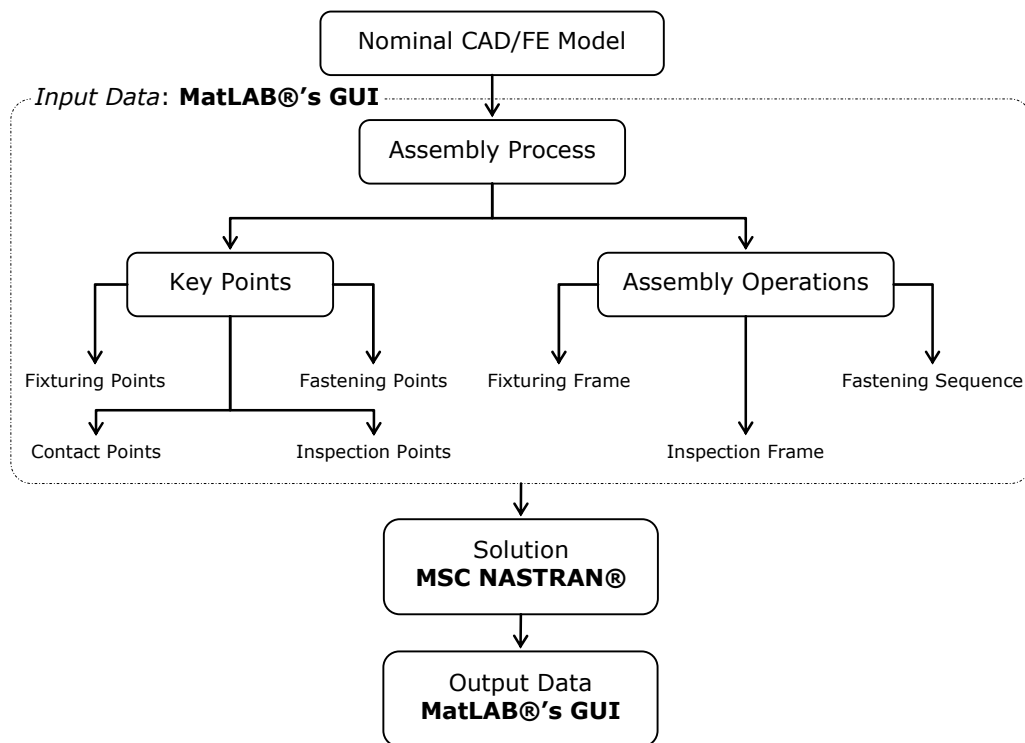


Figure 6.1: general work-flow of the methodology

As described in Chapter 2, one actual need is the station-to-station modeling: analyzing and correctly modeling what happens when parts are released from one sta-

² MSC NASTRAN® is a trademark of MSC Software Corporation.

tion and positioned in the next one is a critical task. This Chapter gives also a contribution to this aspect.

In addition, the proposed methodology assumes the following hypotheses:

- all phases of the assembly processes are assumed linear: elastic materials and small displacement hypotheses are adopted;
- fixtures are assumed rigid with respect to parts being assembled;
- friction among parts is neglected;
- multiphysics effects, such as local thermo-structure interaction or thermo-electrical-structure interaction, are not considered. Only a local thermal load, as deterministic load, may be applied at fastening points;
- input statistical variabilities are assumed independent among them. This means that no statistical covariance effect is considered; and,
- KPs are defined at mesh node level. This means that the mesh must be created accordingly in order to fit the location of KPs.

Generally speaking, the elastic constitutive law, linking displacements and forces, may be expressed as into equation (6.1), where \mathbf{u} and \mathbf{F} are the vectors of displacements and forces, respectively, while \mathbf{K} is the squared stiffness matrix. Under linear hypotheses, \mathbf{K} is constant and depends only on the nominal geometry and the material assignment.

$$\mathbf{F} = \mathbf{K} \cdot \mathbf{u} \quad (6.1)$$

6.3 Assembly Process

This Section focuses on the definition of KPs and how to model assembly operations. The MSC NASTRAN® syntax is also provided (see also Annex G for further details) for a quick implementation.

6.3.1 Fixturing Point Modeling

In SVA-FEA approach kinematic constraints are modeled by means of their relative DoFs. This means that real 3D mating conditions among mating features are not accounted: just an *equivalent* point is considered and the related DoFs are associated. For example, for a pin-hole joint, the middle point of the pin/hole axis, having as DoF the rotation around that axis, is associated. Such constraints may be defined into MSC NASTRAN® as *Single Point Constraint* (SPC).

For example, assuming that the point location is given by $\mathbf{P}=[10.0, 20.0, 10.0]$

Chapter 6. Linear Analysis of Compliant Assemblies

and the related DoF is the rotation around the global **Z** axis, one can write the following MSC NASTRAN®'s code :

```
$- creating the GRID entry (point P)
GRID, 1, , 10.0, 20.0, 10.0
$- creating the SPC (rotation around global Z axis)
SPC, 1, 1, 12345, 0.00
```

It should be noted that fixturing constraints may be seen as kinematic joints acting between parts and fixtures. The latter ones are locked and rigid.

6.3.2 Fastening Point Modeling

Real assembly processes involve different fastening operations: welding, bolting, riveting, gluing.

Generally speaking, this dissertation treats the fastening points as kinematic joints acting between two parts. In particular, their behavior, and thus the relative DoFs, depends on the history of the assembly process. For instance, they are assumed as kinematic joints, allowing some DoFs, before their acting, and as elastic or rigid beams once they have been applied. Looking at Figure 6.2, the *plane-to-plane* joint may be used to model welding (lap or butt joints) and gluing operations; the *hole-to-hole* and *hole-to-slot* joints, instead, may describe bolting operations. The related DoFs are specified with respect to the local coordinate frame attached to each joint. By substituting the 3D mating features with equivalent points, belonging to mating parts, SPCs may be defined to constraint those points before the effective fastening operation. Once the fastening joint has been realized, an elastic or a rigid beam, connecting the two equivalent nodes, is introduced. To do this, CWELD or MPC elements, available in MSC NASTRAN® are used to model elastic and rigid joints, respectively.

For example, assuming that a welding operation is assigned between points $\mathbf{P}_1=[0.0, 0.0, 0.0]$ and $\mathbf{P}_2=[10.0, 0.0, 0.0]$, the following MSC NASTRAN® code may be written:

```
$$- before welding operation
$- creating the GRID entry (point P1)
GRID, 5, , 0.0, 0.0, 0.0, 3 $- "3" is the local coordinate frame
$- creating the GRID entry (point P2)
GRID, 6, , 10.0, 0.0, 0.0, 3
$- creating the SPC for point P1
SPC, 1, 5, 156, 0.00 $- DoF=234
$- creating the SPC for point P2
SPC, 1, 6, 156, 0.00 $- DoF=234

$$- after welding operation
$- solution 1: CWELD entry
CWELD, 1, 1, , ALIGN, 5, 6
```

Chapter 6. Linear Analysis of Compliant Assemblies

```

$- solution 2: MPC entry
MPC, 1, 5, 1, 1.00, 6, 1, -1.00
MPC, 1, 5, 2, 1.00, 6, 2, -1.00
MPC, 1, 5, 3, 1.00, 6, 3, -1.00
MPC, 1, 5, 4, 1.00, 6, 4, -1.00
MPC, 1, 5, 5, 1.00, 6, 5, -1.00
MPC, 1, 5, 6, 1.00, 6, 6, -1.00
    
```

The need to separate the behavior of fastening points before and after their acting will be clarified later when the multi-station assembly model will be illustrated.

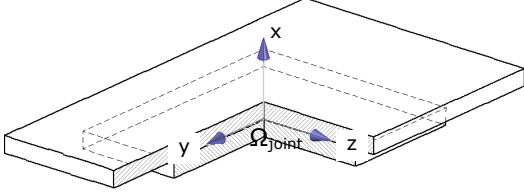
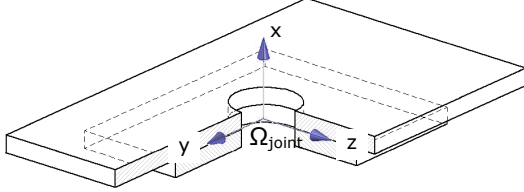
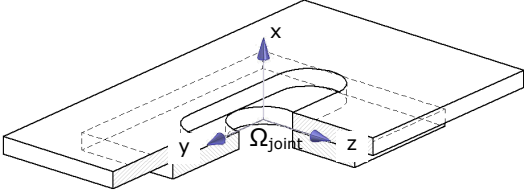
Joint Type	Before assembling	After assembling
<p><i>plane-to-plane</i></p> 	$\text{DoF} = \begin{bmatrix} a & 0 \\ 0 & \Delta y \\ 0 & \Delta z \end{bmatrix}$	<p>elastic/rigid beam</p>
<p><i>hole-to-hole</i></p> 	$\text{DoF} = \begin{bmatrix} a & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$	<p>elastic/rigid beam</p>
<p><i>hole-to-slot</i></p> 	$\text{DoF} = \begin{bmatrix} a & 0 \\ 0 & \Delta y \\ 0 & 0 \end{bmatrix}$	<p>elastic/rigid beam</p>

Figure 6.2: fastening modeling

6.3.3 Contact Point Modeling

To avoid part-to-part intersection, contact points must be defined. This Section provides a simplified model to take into account such contact points.

A special feature of MSC NASTRAN® has been here adopted. Linear contacts may be implemented in a linear static analysis (SOL 101 solver) by using MPCs to de-

Chapter 6. Linear Analysis of Compliant Assemblies

fine the gaps³. Because MPCs are used, there is no gap stiffness. In addition, friction is not available. MPCs are solved for through an *iterative* technique that is built into SOL 101. The solution converges when there is no penetration of MPCs and there are no tensile forces.

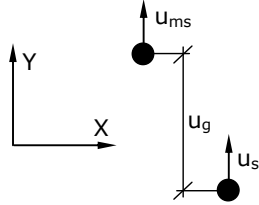


Figure 6.3: linear contact modeling

Looking at Figure 6.3, the following equation may be written:

$$\mathbf{S} = \mathbf{u}_{ms} - \mathbf{u}_{sl} + \mathbf{u}_g \quad (6.2)$$

where \mathbf{u}_{ms} and \mathbf{u}_{sl} are the displacements related to the master and slave nodes, respectively. \mathbf{u}_g is the initial gap distance between the two nodes.

The following may happen:

$$\begin{aligned} \text{a): } \mathbf{S} \leq \mathbf{u}_g &\rightarrow \mathbf{u}_{ms} - \mathbf{u}_{sl} = \mathbf{0} \\ \text{b): } \mathbf{S} > \mathbf{u}_g &\rightarrow \mathbf{u}_{ms} \text{ and } \mathbf{u}_{sl} \text{ are independent} \end{aligned} \quad (6.3)$$

If the condition a) is satisfied than the displacements \mathbf{u}_{ms} and \mathbf{u}_{sl} are dependent and equal each-other. Otherwise, the two nodes become independent.

Equation (6.2) may be rearranged as into relationship (6.4):

$$\mathbf{S} - \mathbf{u}_{ms} + \mathbf{u}_{sl} - \mathbf{u}_g = \mathbf{0} \quad (6.4)$$

which corresponds to the canonical form of the MPC condition (see also Annex G). Obviously, equation (6.4) should be specified for every DoF related to the master and slave nodes.

The general formulation stated into equation (6.4) allows to simulate: (I) parts initially in contact ($\mathbf{u}_g=0$); parts initially intersecting each-other ($\mathbf{u}_g<0$); and, (III) parts initially non in contact ($\mathbf{u}_g>0$).

Typically, \mathbf{u}_g is a user input value. When contact points are used to simulate the assembly process, \mathbf{u}_g should be set for each assembly station. However, this value depends on the deformation state, in terms of statistical mean and standard deviation, of parts related to that specific station. Because the statistical deformation is calculated af-

³ Generally speaking, when working with contact pairs, iterative numerical procedures are needed. More realistic results are reached when surface-to-surface pairs are introduced. This Chapter provides a linear solution to the contact problem through a point-to-point contact definition. A more sophisticated model will be presented in Chapter 7.

ter performing the FEA run, the \mathbf{u}_g value is here assumed equal to zero: this means that parts are already assumed to be initially in contact, when contact points are defined.

Another significant aspect to be accounted is the changing in the constitutive law when contact points are introduced. Looking at Figure 6.4, a positive force, \mathbf{F}_j , applied in the j-th node produces a displacement \mathbf{u}_i at the i-th node; instead, the same i-th node does not move if a negative force is applied at the j-th node.

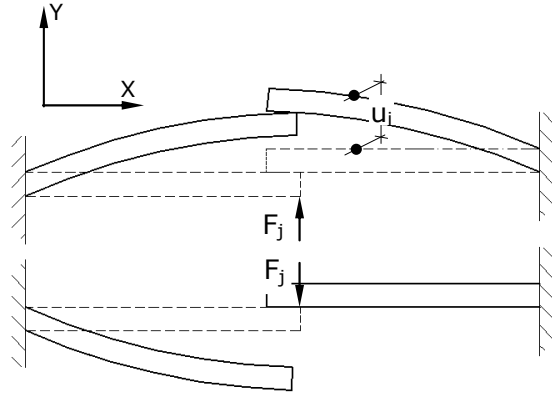


Figure 6.4: constitutive law for linear contact modeling

Therefore, the linear constitutive relationship between force and displacement becomes:

$$\begin{cases} \mathbf{F} \geq \mathbf{0} \rightarrow \mathbf{F} = \mathbf{K}_+ \cdot \mathbf{u} \\ \mathbf{F} < \mathbf{0} \rightarrow \mathbf{F} = \mathbf{K}_- \cdot \mathbf{u} \end{cases} \quad (6.5)$$

where, \mathbf{K}_+ and \mathbf{K}_- represent the stiffness matrices due to positive and negative contributions.

6.3.4 Statistical Input Data

Generally speaking, considering only the variability at point level, two main sources of variability may occur when assembling compliant parts: variability due to form defects of parts, and variability due to positioning errors of fixtures and fastening tools (such as weld guns).

Figure 6.5 shows four combinations of such variability. In the cases (a) and (b) only form defects on parts are considered, whereas fixturing and fastening tools act in the nominal position. In the cases (c) and (d) parts are nominal and positioning errors occur in fixtures and fastening tools.

In this dissertation, only variability at point level corresponding to the cases (a) and (c) is modeled. For these points, statistical data must be defined in terms of mean

Chapter 6. Linear Analysis of Compliant Assemblies

and standard deviation: statistical data may be acquired from real measurements data.

In the next, fastening variability point will be named as FSV, while fixturing variability point as FXV. The total number of FSV and FXV are N_{FSV} and N_{FXV} , respectively.

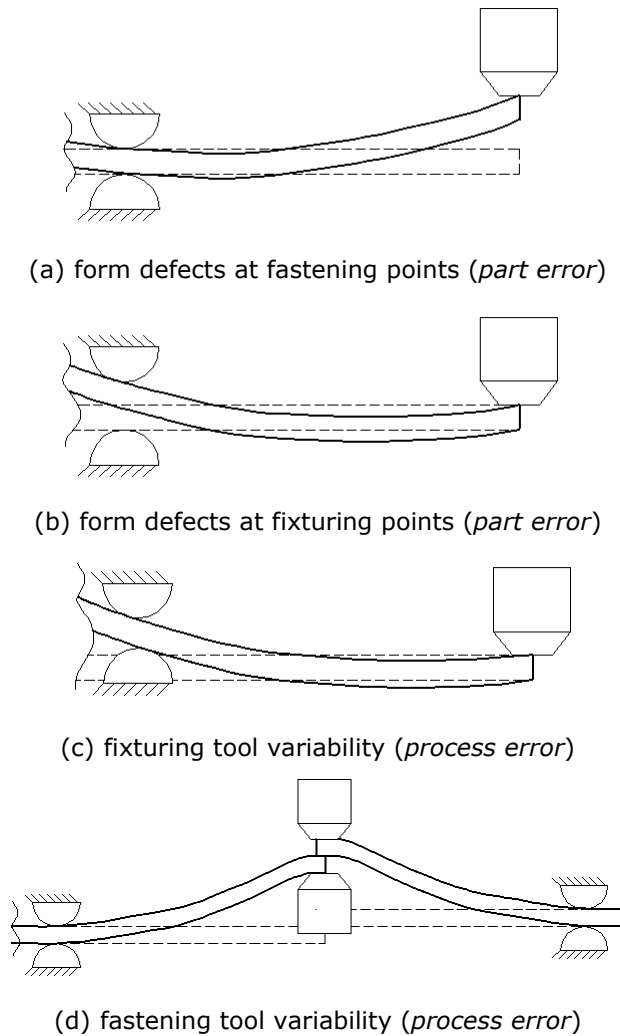


Figure 6.5: combination of form defects and positioning errors

6.3.5 Assembly Operation

An efficient way to model assembly operations is by using a graph representation. As depicted into Figure 6.6, each vertex of the graph corresponds to a part or, more generally, to a specific sub-station assembly, while each edge is a station-to-station connection.

The mathematical representation of such graph may be achieved by evaluating the incidence matrix, for oriented graph (see also Annex D). In this way, with respect to specific sub-station, previous and next stations can be easily evaluated (the algorithm to solve this task is formally equal to “paths2graph” presented into Chapter 3). For example, looking at Figure 6.6, with respect to the “station i”, “station 1” and “station N_s” are the previous and next stations, respectively. N_s is the total number of assembly stations.

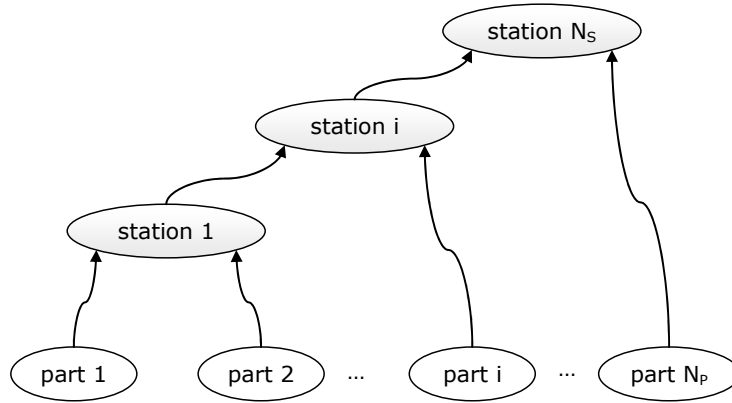


Figure 6.6: assembly operation modeling

In the following, the previous and next stations are shortly named as PSt and NSt, respectively. In addition, ASt is the actual station.

6.4 Solution

6.4.1 The Global Sensitivity Matrix

The assembly process simulation is based on the mechanistic variation simulation methodology developed by Liu and Hu (Liu, 1997).

The linear relationship between variations at part level, \mathbf{u} , and deviations at assembly level, \mathbf{v} , can be formally written as follows:

$$\mathbf{v} = \mathbf{S} \cdot \mathbf{u} \quad (6.6)$$

where \mathbf{u} is a $N_{\text{var}} \times 1$ vector while \mathbf{v} is a $N_n \times 3$ (three translations for each node⁴) vector. \mathbf{S} is the global sensitivity matrix for displacement evaluation.

In the same fashion, the global sensitivity matrix for force evaluation, \mathbf{R} , can be defined as:

$$\mathbf{f} = \mathbf{R} \cdot \mathbf{u} \quad (6.7)$$

⁴ It should be noted that for shell elements every node is characterized with six DoFs: three translations and three rotations. In the proposed approach, only translations are considered.

Chapter 6. Linear Analysis of Compliant Assemblies

where \mathbf{f} is the $N_n \times 6$ (three translations and three rotations for each node) vector of forces.

Generally speaking, one can write:

$$\begin{aligned} N_{\text{var},i} &= \sum_{k=1}^{N_{\text{PSt}}} (2 \cdot N_{\text{FSV}} + 2 \cdot N_{\text{FXV}})_k + (2 \cdot N_{\text{FSV}} + 2 \cdot N_{\text{FXV}})_i \\ N_{\text{var}} &= \sum_{i=1}^{N_S} N_{\text{var},i} \end{aligned} \quad (6.8)$$

where $N_{\text{FSV},i}$ and $N_{\text{FXV},i}$ is the total number of FSVs and FXVs acting into station i -th, respectively. N_{PSt} is the total number of PSt influencing the station i -th. Moreover, in relationship (6.8) each source of variation is counted twice: the elastic behavior of the structure must be evaluated both for positive and negative displacements at critical nodes as stated above in Section 6.3.3 (obviously, if contact points are not defined, each source of variation is counted once).

Moreover, global sensitivity matrices \mathbf{S} and \mathbf{R} can be decomposed into N_S sub-matrices, as described into relationship (6.9).

$$\begin{cases} \mathbf{S} = [\mathbf{S}_1, \mathbf{S}_2, \dots, \mathbf{S}_{N_S}] \\ \mathbf{R} = [\mathbf{R}_1, \mathbf{R}_2, \dots, \mathbf{R}_{N_S}] \end{cases} \quad (6.9)$$

Statistical distributions of displacements and forces, at assembly level, can be carried out by relationships (6.6) and (6.7). Applying the expected operator, $\boldsymbol{\mu}$, formally, one can write:

$$\begin{cases} \boldsymbol{\mu}_v = \mathbf{S} \cdot \boldsymbol{\mu}_u \\ \boldsymbol{\mu}_f = \mathbf{R} \cdot \boldsymbol{\mu}_u \end{cases} \quad (6.10)$$

In addition, by applying the standard deviation operator, $\boldsymbol{\sigma}$, it is:

$$\begin{cases} \boldsymbol{\sigma}_v = \sqrt{\mathbf{S}^2 \cdot \boldsymbol{\sigma}_u^2} \\ \boldsymbol{\sigma}_f = \sqrt{\mathbf{R}^2 \cdot \boldsymbol{\sigma}_u^2} \end{cases} \quad (6.11)$$

with obvious meaning of adopted symbols.

Finally equation (6.10) and (6.11) allow to statistically calculate, in terms of mean and standard deviation, final assembly reaction forces, acting on fixturing and fastening tools, and node displacements.

How to numerically calculate the sensitivity matrices will be presented in next Sections.

6.4.2 Method of Influence Coefficient

As seen in Chapter 2, several authors proposed to evaluate the elastic behavior of the parts by means of the *Method of Influence Coefficient* (MIC).

Generally speaking, in the present dissertation the influence coefficient matrix α is defined as a square matrix whose element α_{ij} represents the reaction force at the i -th node due to a unit displacement applied at the j -th node. For instance, the MIC is here used to evaluate the stiffness of the parts or sub-assemblies (in the literature, this issue is also called material covariance (Merkely, 1996)).

Moreover, as noted in Section 6.3.3, when contact points are defined MIC must be applied both for positive and negative unit displacements.

6.4.3 Calculating the Sensitivity Matrix

Figure 6.7 depicts the general work-flow to perform the evaluation of the global sensitivity matrices.

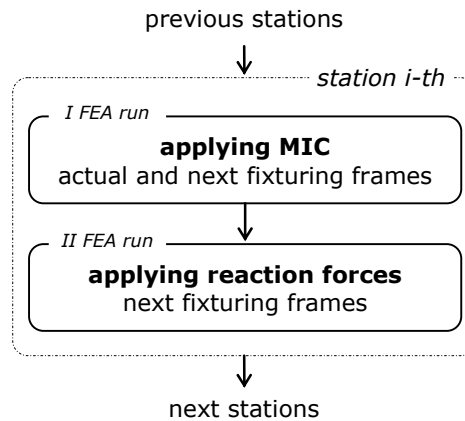


Figure 6.7: general work-flow for sensitivity matrix evaluation

The i -th station, which influences the next station, may be influenced by the previous ones. Two consecutive FEA runs are required. Generally speaking, the first FEA run is used to evaluate the fixturing and fastening forces (in this way, it is used to simulate the positioning and clamping phases). Then, these forces are applied into the second FEA run to simulate the final elastic spring-back.

Calculating the Sensitivity Matrix: First FEA Run

Generally speaking, with respect to the first FEA run, at the i -th station the fol-

Chapter 6. Linear Analysis of Compliant Assemblies

Following KPs may be defined:

- actual fastening points;
- actual fixturing frame; and
- next constraint frame.

The actual fixturing frame is made of ideal and variable (FXV) fixturing tools.

At the next constraint frame belongs:

- next fixturing frame;
- inspection frame;
- next fastening points; and,
- next contact points.

The constraints applied to the next fastening and contact points lock only the DoFs related to the fastening or contact elements (see fastening and contact models).

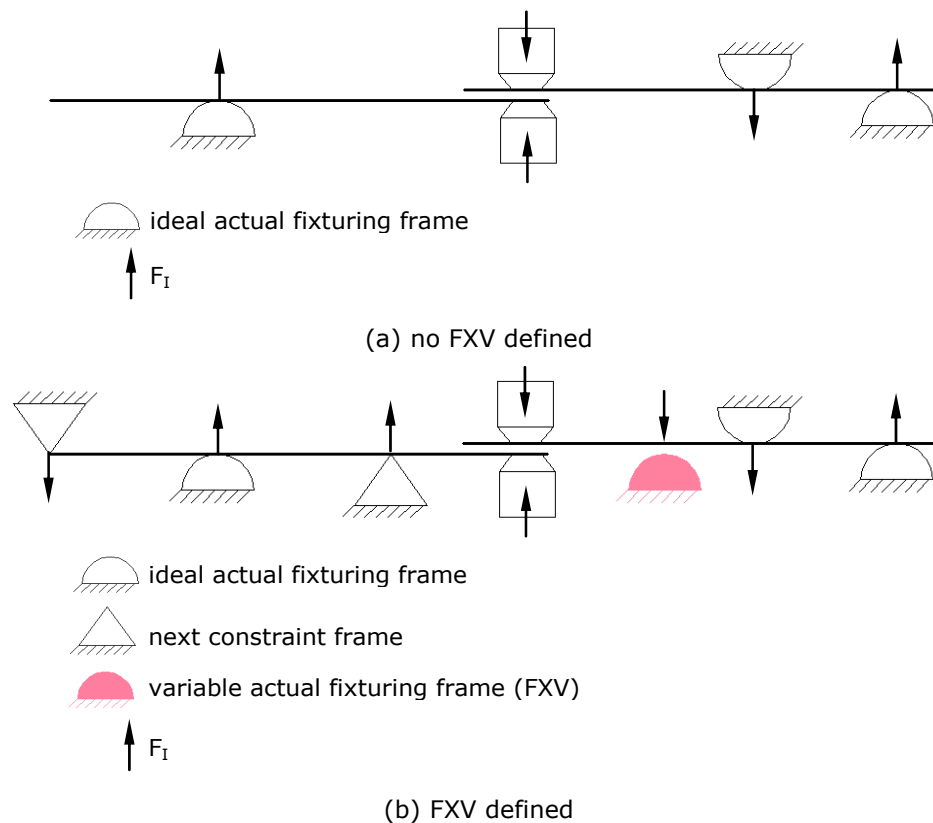


Figure 6.8: constraint configurations for the first FEA run

As depicted in Figure 6.8, when FXVs are defined during the first FEA run also the *next constraint frame* is assigned; otherwise, only the actual fixture frame is set.

By using the MIC, reaction forces are calculated for each KP above defined (F_I forces into Figure 6.8). In addition, if FXVs are defined also node displacements are evaluated. To understand all this, one should think that, under the hypotheses stated into

Chapter 6. Linear Analysis of Compliant Assemblies

Section 6.3.4, fastening guns force parts at their nominal position (the displacement field is then zero), while FXVs may deform parts when they are applied (thus, the displacement field has to be calculated).

In order to perform this calculation by using the same FEA run, sub-cases are used into the main structure of the bdf file (see also Annex G). For example, assuming that only one FSV and one FXV are defined, the bdf file structure looks like this:

```

$- sub-case entry for FSV:
SUBCASE 1
SPC = 1
MPC = 1
LOAD = 1
SPCFORCES(SORT2, REAL) = 1 $- only forces are calculated

$- sub-case entry for FXV:
SUBCASE 3
SPC = 1
MPC = 1
LOAD = 3
SPCFORCES(SORT2, REAL) = 1 $- forces and displacements are here calculated
DISPLACEMENT(SORT2, REAL) = ALL

$- SPC entry for MIC
SPCD, 1, 10, 2, 1.00 $- applying positive unit disp. along Y
SPCD, 2, 10, 2, -1.00 $- applying negative unit disp. along Y
SPCD, 3, 15, 3, 1.00 $- applying positive unit disp. along Z
SPCD, 4, 15, 3, -1.00 $- applying negative unit disp. along Z

```

In this example, FSV and FXV are associated to nodes “10” and “15”, respectively. Moreover, FSV acts along the global **Y** direction, while FXV along the global **Z** direction.

Output data of the first FEA run can be rearranged and stored as follows:

$$\begin{aligned}
 \text{FXV defined} &\rightarrow \begin{cases} \mathbf{S}_i = \mathbf{S}_{i,\text{FXV}} \\ \mathbf{R}_i = [\mathbf{R}_{\text{PSt}}, \mathbf{R}_I]_i \\ \mathbf{R}_{\text{NSt}} = \mathbf{R}_I \end{cases} \\
 \text{no FXV defined} &\rightarrow \mathbf{R}_i = [\mathbf{R}_{\text{PSt}}, \mathbf{R}_I]_i \\
 \mathbf{R}_I &= -\mathbf{F}_I \\
 \forall i &= 1, \dots, N_S
 \end{aligned} \tag{6.12}$$

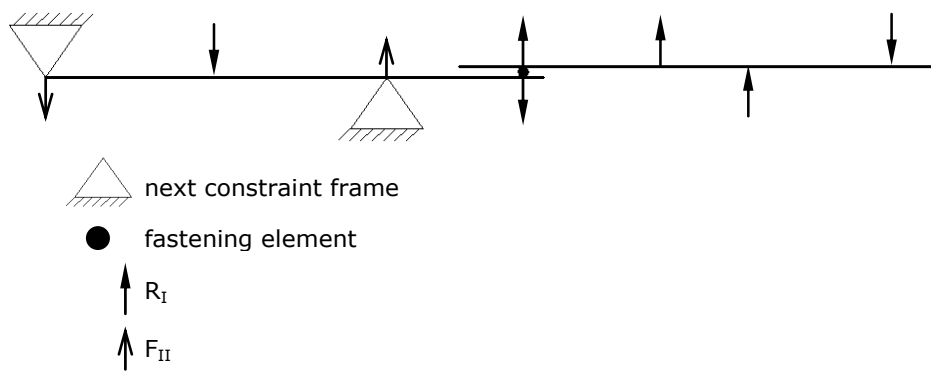
The local sensitivity matrix, \mathbf{S}_i , is updated with the actual sub-assembly displacements (\mathbf{S}_I) due to unit displacements acting at FXVs. Moreover, reaction forces related to all KPs are stored into the local \mathbf{R}_i matrix (one can note that $\mathbf{R}_{\text{PSt},i}$ was already calculated for the previous station).

Reaction forces are also monitored for all those points belonging to the next constraint frame and they are stored into \mathbf{R}_{NSt} .

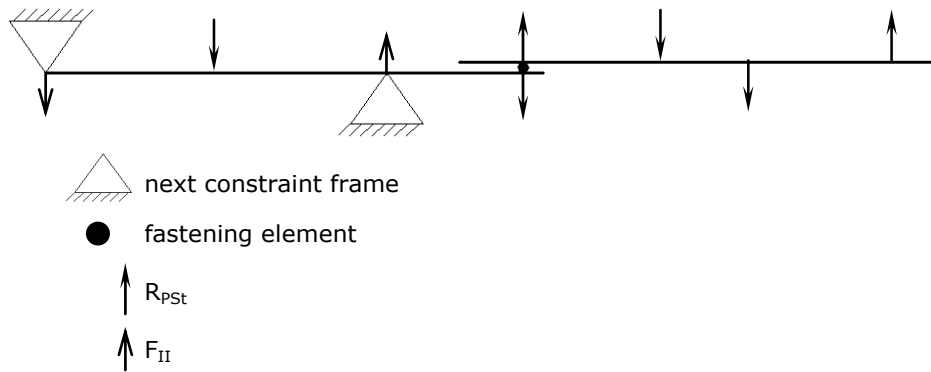
Calculating the Sensitivity Matrix: Second FEA Run

The second FEA run simulates the elastic spring-back. Only the next constraint frame is here assigned. In addition, fastening elements are introduced.

Looking at Figure 6.9, two configurations must be analyzed: in the first one (Figure 6.9.a), the influence of actual variability (by means of \mathbf{R}_I) is evaluated (\mathbf{S}_{II}). Previous variability is accounted (\mathbf{S}_{PSt}) into the second configuration (Figure 6.9.b) through reaction forces \mathbf{R}_{PSt} . Once again, reaction forces at next constraint frame are monitored (\mathbf{F}_{II}).



(a) spring-back due to actual variability (\mathbf{S}_{II})



(b) spring-back due to previous variability (\mathbf{S}_{PSt})

Figure 6.9: constraint configurations for the second FEA run

Output data of the second FEA run can be rearranged and stored as follows:

$$\begin{cases} \mathbf{R}_{NSi} = \mathbf{R}_I + \mathbf{R}_{II} \\ \mathbf{S}_i = [\mathbf{S}_{PSt}, \mathbf{S}_{II} + \mathbf{S}_{I,FXV}] \end{cases} \quad (6.13)$$

$$\mathbf{R}_{II} = -\mathbf{F}_{II}$$

$$\forall i = 1, \dots, N_S$$

It can be pointed out that this mechanism allows to simulate the variability propagation from one station to the next one.

6.5 Data Structure for Tolerance Analysis Software

This Section shows the main structure to manage input data to simulate the whole assembly process.

Figure 6.10 shows the main structure to manage input data needed to assign KPs.

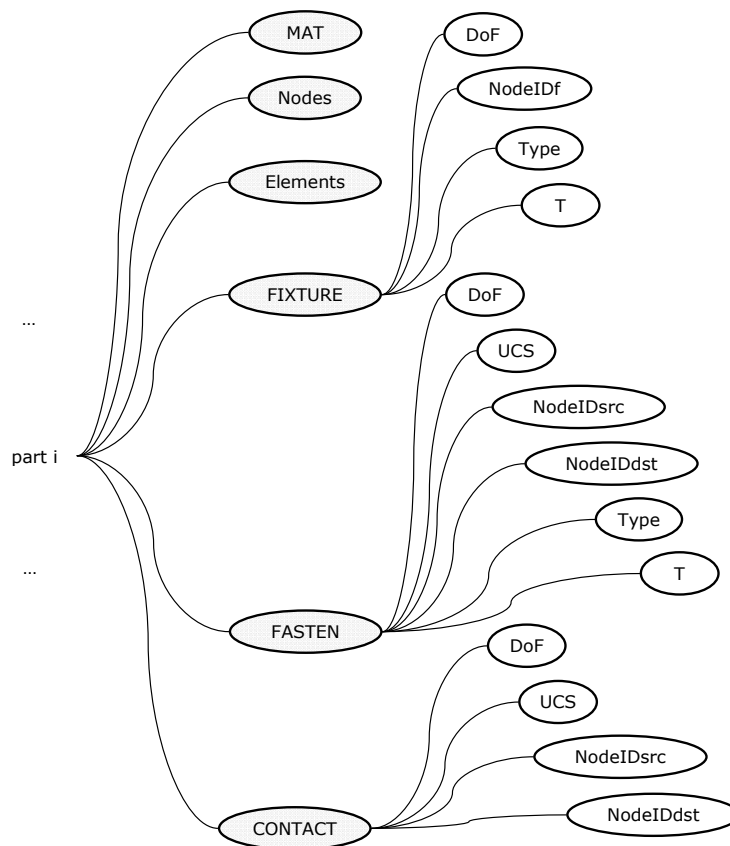


Figure 6.10: data structure for key point definition

For each part (“part i”), six fields are available:

- MAT: material properties and shell thickness;
- Nodes: coordinates of mesh nodes;
- Elements: shell elements;
- FIXTURE/FASTEN/CONTACT: input assignment for fixturing, fastening and contact points.

Chapter 6. Linear Analysis of Compliant Assemblies

With respect to the latter fields, the following sub-fields are available:

- DoF: list of DoFs;
- UCS: local coordinate frame definition;
- NodeIDf: node identification for fixturing point assignment;
- NodeIDsrc / NodeIDdst: source and destination node identification;
- Type: integer value equals to “0” or “1”, meaning ideal or variable assignment;
- T: statistical input value in terms of mean and standard deviation;

6.6 Implementation

The SVA-FEA methodology was implemented into a friendly MatLAB®’s GUI allowing to interactively define input data, assembly process and visualize final results (Figure 6.11).

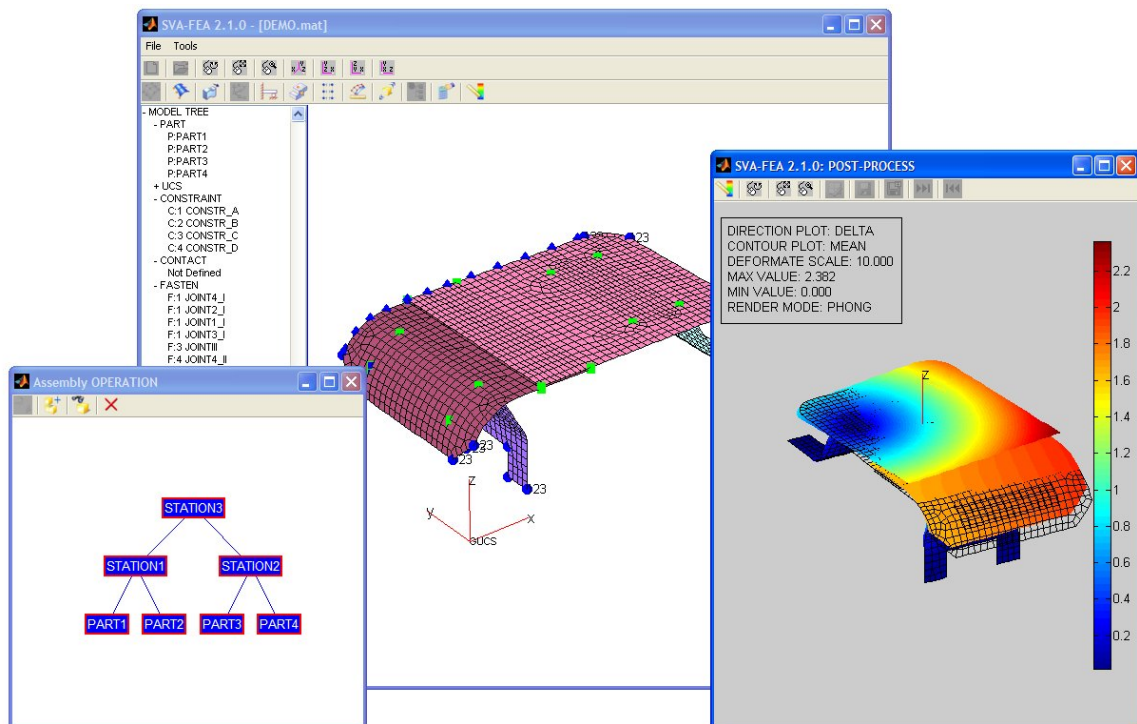


Figure 6.11.a: SVA-FEA Graphical User Interface. Main GUI

Input mesh file is imported in bdf format⁵. Parts are automatically recognized through a growing algorithm.

⁵ The supported input bdf format is the free field format, where data are separated by blanks. The comma-separated format is not available. On the other hand, output bdf file are automatically written into comma-separated format.

Chapter 6. Linear Analysis of Compliant Assemblies

The MatLAB®'s GUI drives the user to define input data, such as fixturing points (*CONSTRAINT* menu), fastening points (*FASTEN* menu) and their variability (*FASTEN* and *CLAMP Variability* menus), contact point (*CONTACT* menu), and to create the assembly process (*Assembly OPERATION* interface).

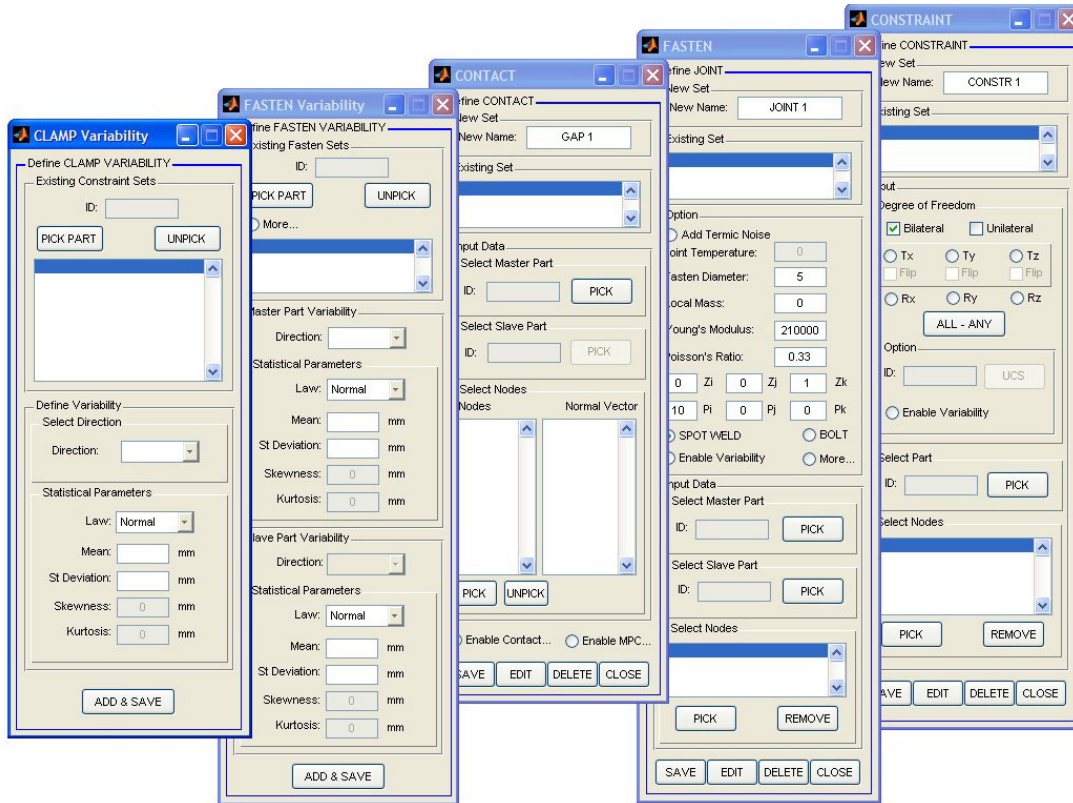


Figure 6.11.b: SVA-FEA Graphical User Interface. Main menus

When fastening or contact points are defined, user must select *source* and *destination* parts⁶ and pick, from the main GUI, the point location, with respect the source part: the destination node is automatically calculated and assigned.

Once all input data have been correctly defined, sensitivity matrices are automatically calculated. In this phase, the MCS NASTRAN® solver is run in background mode. For every FEA run, the related bdf file is automatically generated and parsed in input to the solver. Then, output files, coming from the solver, are post-processed and results are stored. In order to speed-up all these operations, compiled c++ source (mex file) codes have been here adopted (see also Annex C for more details on mex file definition).

Final results can be easily analyzed from the *POST-PROCESS* main GUI: deformed or un-deformed assembly (or sub-assembly) can be visualized; contour plots of

⁶ In this contest, source and destination parts are assumed equivalent to master and slave parts.

Chapter 6. Linear Analysis of Compliant Assemblies

mean or standard deviation values are available. Final results can be exported into EXCEL®⁷ format, to quickly create graphs and diagrams.

⁷ EXCEL® is a trademark of Microsoft Corporation.

6.7 Examples

This Section shows the results of two case studies. In both cases a comparative analysis with the TAA® module of CATIA® CAD system was performed.

6.7.1 Single-station Simulation

The SVA-FEA variation assembly methodology was tested on an assembly of two aluminum sheet-metal parts (Young’s Modulus $E=70.000 \text{ N/mm}^2$; Poisson’s ratio $\nu=0.346$; uniform thickness $T=2.0 \text{ mm}$), shown in Figure 6.12.

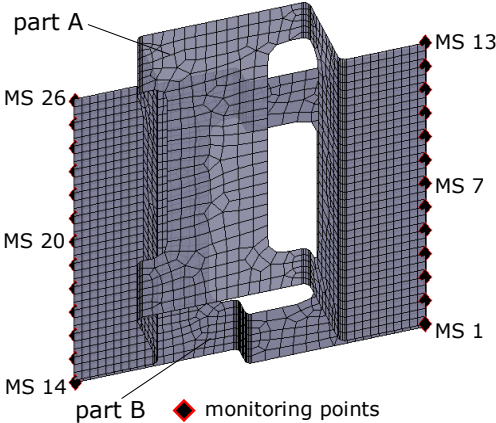


Figure 6.12: two-part assembly and monitoring points

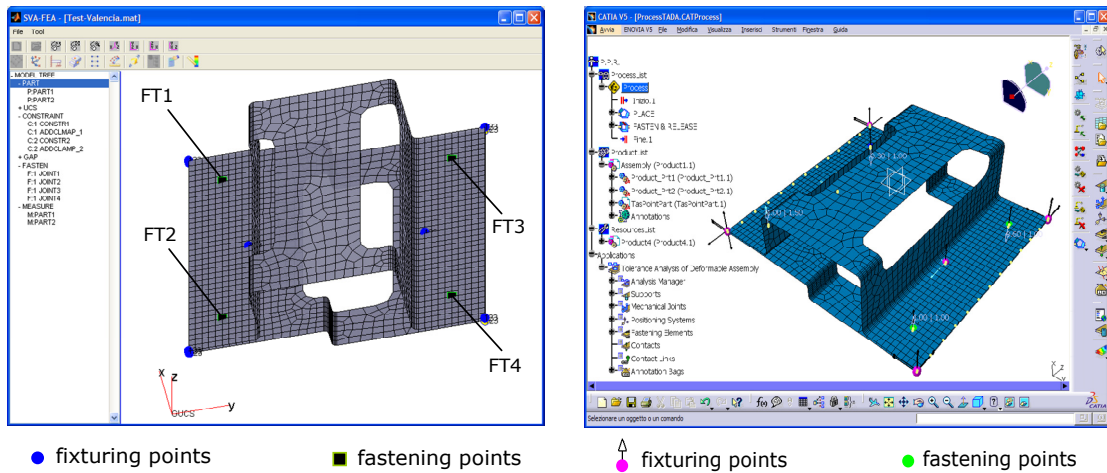
The overall dimensions of the assembly are 200x200x60 mm. Simulation results were performed on the monitoring points shown in Figure 6.12: points 1 to 13 are related to part A, 14 to 26 to part B.

	weld points	Mean X (mm)	Std Dev X (mm)
Part A	initial zero deviation		
Part B	FT1	-0.3	1.0
	FT2	1.0	1.5
	FT3	-0.6	1.0
	FT4	1.0	1.0

Table 6.1: statistical input variability

Mid-surfaces were extracted from the CAD model and then meshed with shell elements. Global mesh size was equal to 5 mm. Figure 6.13 shows the SVA-FEA and TAA models.

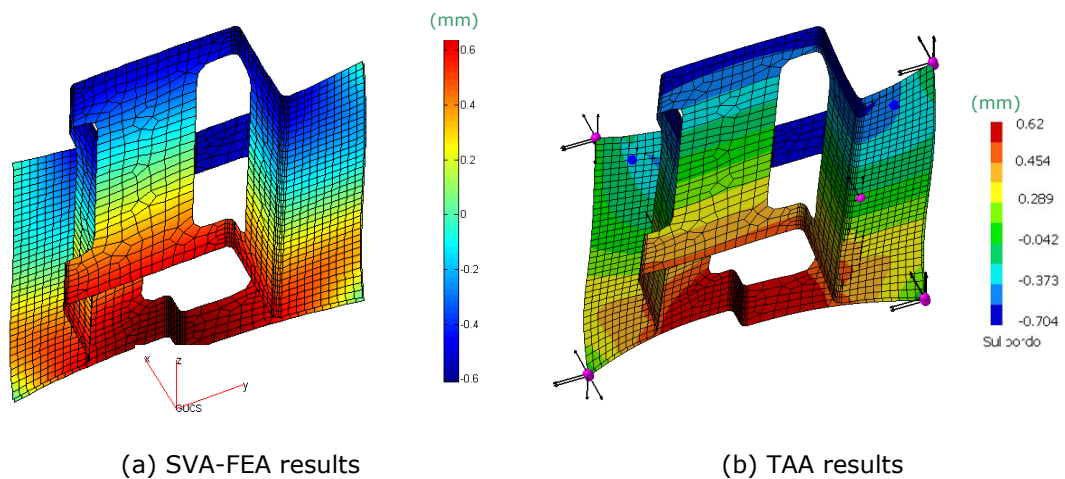
Chapter 6. Linear Analysis of Compliant Assemblies



(a) SVA-FEA model

(b) TAA model

Figure 6.13: SVA-FEA and TAA models



(a) SVA-FEA results

(b) TAA results

Figure 6.14: contour plot for mean displacements along global X axis

Each fixturing point was modeled as a spherical joint: only translations are constrained. Fastening points were modeled as elastic connector elements. Local input variability was applied at fastening points in the global X axis. Table 6.1 shows the adopted statistical values. Part A was considered with an initial variation equal to zero.

The TAA® model was based on the following features:

- mesh elements were imported from SVA-FEA model;
- fastening joints were modeled as spot welding points;
- local input deviations were applied at fastening points; and,
- additional fixturing points were added to fastening points during the positioning phase. In this way, during the positioning phase, fastening and fixturing tools

Chapter 6. Linear Analysis of Compliant Assemblies

were closed to their nominal position.

Figures 6.14 shows the assembly mean deformation evaluated in SVA-FEA and TAA® environments. The assembly configuration is related to the releasing phase of fastening guns.

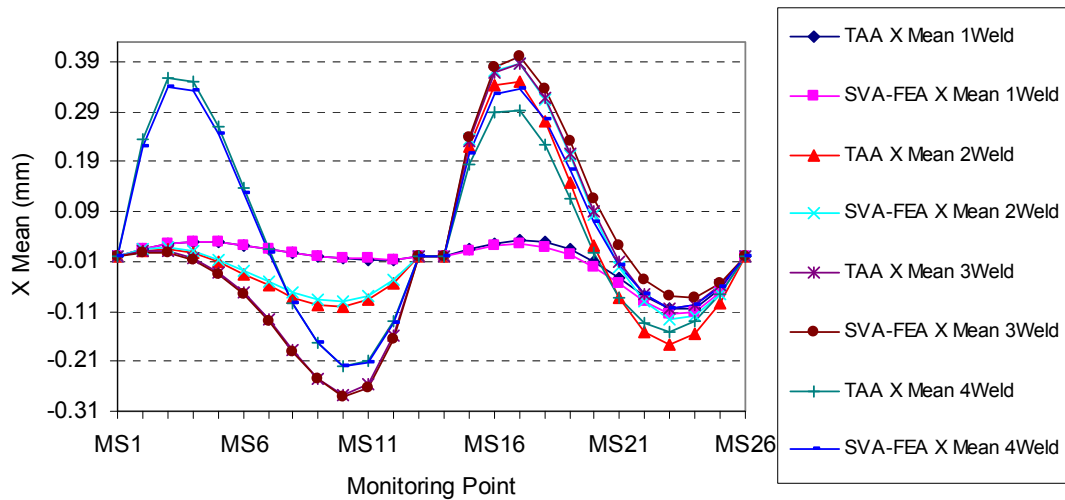


Figure 6.15.a: mean results along global X axis

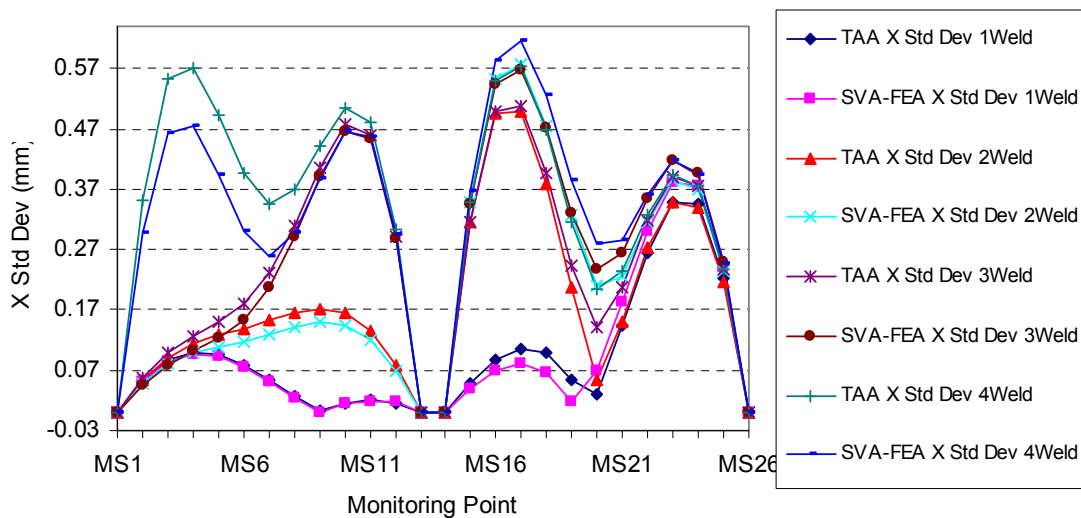


Figure 6.15.b: standard deviation results along global X axis

It is of interest evaluating the influence of fastening points. Therefore, four kinds of configurations were analyzed: fastening point FT1, fastening points (FT1+FT2), fastening points (FT1+FT2+FT3), and fastening points (FT1+FT2+FT3+FT4). Figures 6.15 shows the output deviations, related to monitoring points (see also Figure 6.12), in terms of mean and standard deviation, respectively. The

Chapter 6. Linear Analysis of Compliant Assemblies

results show that final assembly deviations are highly influenced by the number of fastening points. Moreover, SVA-FEA and TAA® results are highly correlated, as shown in Table 6.2.

Configuration	Correlation (Mean)	Correlation (Std Dev)
FT1	0.995	0.988
FT1+FT2	0.989	0.965
FT1+FT2+FT3	0.998	0.979
FT1+FT2+FT3+FT4	0.988	0.952

Table 6.2: correlation indexes

In order to estimate the numerical error between SVA-FEA and TAA® results, an RSS (Root Sum Square) index error was adopted. Table 6.3 shows the RSS index both for mean and standard deviation values. In the configuration (FT1+FT2+FT3+FT4) numerical error, related to standard deviations, is about 5.5%. This is mainly due to the different way to model fastening points in SVA-FEA and TAA®.

Configuration	RSS% (Mean)	RSS% (Std Dev)
FT1	0.464	2.021
FT1+FT2	2.188	4.877
FT1+FT2+FT3	1.221	3.713
FT1+FT2+FT3+FT4	2.826	5.486

Table 6.3: percentage RSS errors

Simulations performed on the assembly allow part-to-part intersection. In order to prevent the penetration between parts, contact points were added both in SVA-FEA and TAA® models.

Figures 6.16 depict final results in terms of mean and standard deviation. The analyses are related to the assembly configuration (FT1+FT2+FT3+FT4).

Configuration	Mean		Std Dev	
	no-contact	contact	no-contact	contact
RSS%	2.826	2.863	5.486	2.590

Table 6.4: percentage RSS error. Contact vs no-contact analysis

The comparison shows that there are significant differences between the simulation with and without contact model in terms of mean and standard deviations. Moreover, numerical error between SVA-FEA and TAA® results decreases with the contact model. Table 6.4 shows the RSS index errors. In particular, the RSS error is less than 3% when the simulations were performed using contact points.

Chapter 6. Linear Analysis of Compliant Assemblies

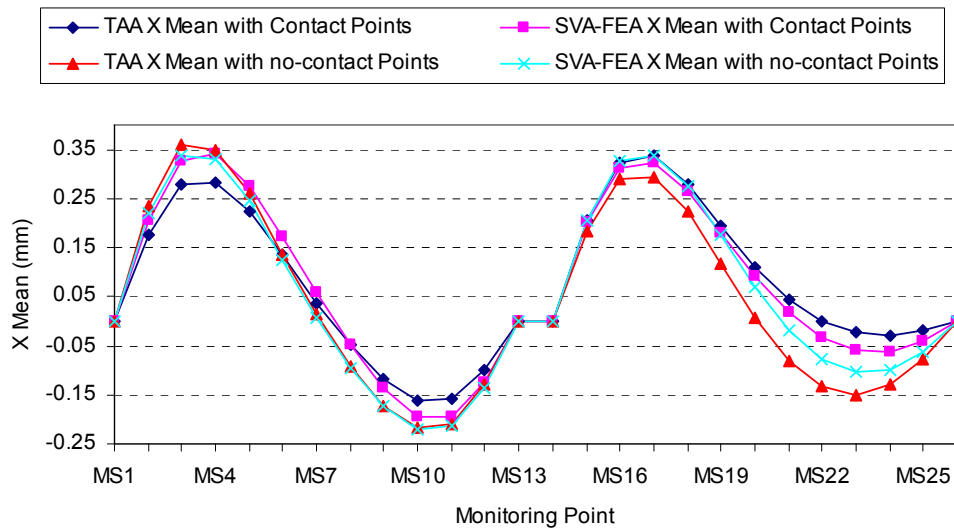


Figure 6.16.a: mean results along global X axis. Contact vs no-contact analysis

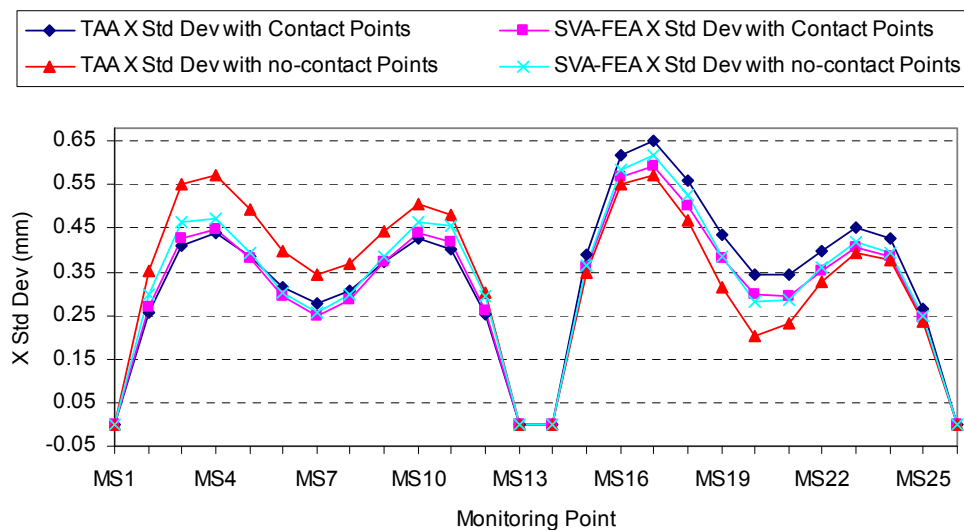


Figure 6.16.b: standard deviation results along global X axis. Contact vs no-contact analysis

6.7.2 Multi-station Simulation

The analyzed assembly is depicted in Figure 6.17. The assembly is made by four aluminum sheet-metal parts (Young's Modulus $E=70.000 \text{ N/mm}^2$; Poisson's ratio $\nu=0.346$, uniform thickness $T=2.0 \text{ mm}$). The overall dimensions of the assembly are $500 \times 250 \times 100 \text{ mm}$. Global mesh size was equal to 10 mm . The mesh was created by using the AMT (Advanced Meshing Tool) module integrated into CATIA® CAD system

Chapter 6. Linear Analysis of Compliant Assemblies

and then imported into SVA-FEA to start with the same mesh model. Simulation results were performed on monitoring points shown in Figure 6.17: points 1 to 7 are related to Part A, 8 to 15 to Part B, 16 to 20 to Part C, and 21 to 26 to Part D.

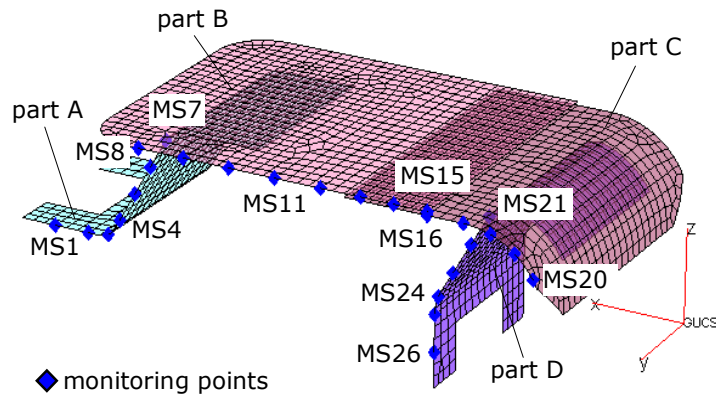


Figure 6.17: four-part assembly and monitoring points

Figure 6.18 shows the models as seen in SVA-FEA and TAA® environments. Each fixturing point was modeled as a spherical joint (translational displacements are constrained). Fastening points were modeled as weld spot points both in SVA-FEA and TAA®.

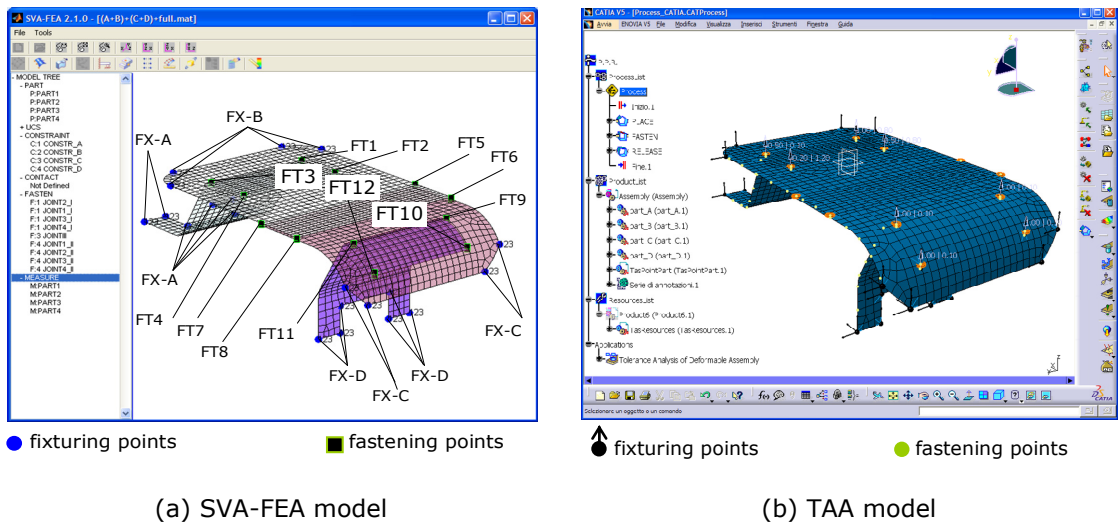


Figure 6.18: SVA-FEA and TAA models

Input variabilities were assigned at fastening points in the global Z axis direction. Table 6.5 shows the statistical values adopted. Part B and C were assumed with an initial zero deviation.

All the following contour plots are related to mean displacements; euclidean norm is adopted and the used deformation scale is equal to 10.

Chapter 6. Linear Analysis of Compliant Assemblies

	weld points	Mean Z (mm)	Std Dev Z (mm)
Part B/C	initial zero deviation		
Part A	FT1	1.0	0.8
	FT2	0.5	0.8
	FT3	-0.5	0.1
	FT4	-0.2	1.2
Part D	FT9	1.0	0.1
	FT10	1.0	0.1
	FT11	1.0	0.1
	FT12	1.0	0.1

Table 6.5: statistical input variability

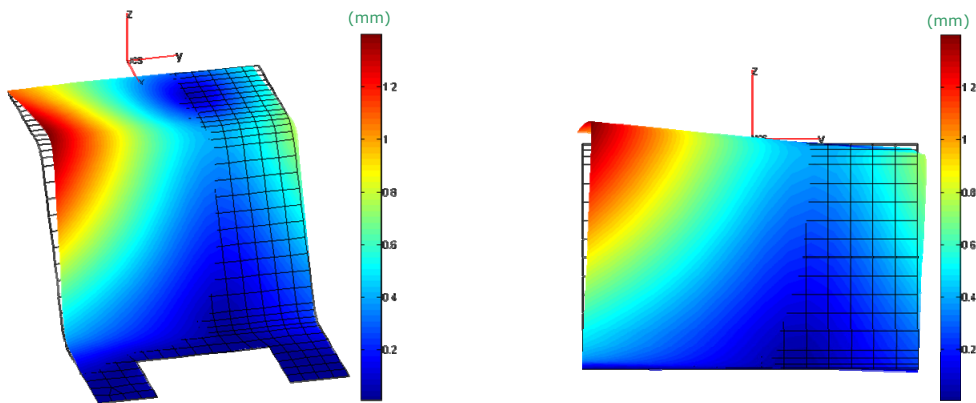


Figure 6.19: initial deviations of part A calculated in SVA-FEA

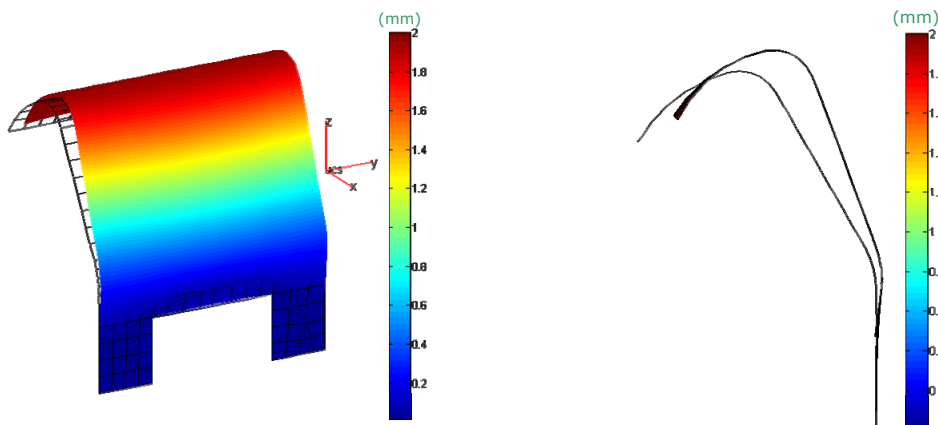


Figure 6.20: initial deviations of part D calculated in SVA-FEA

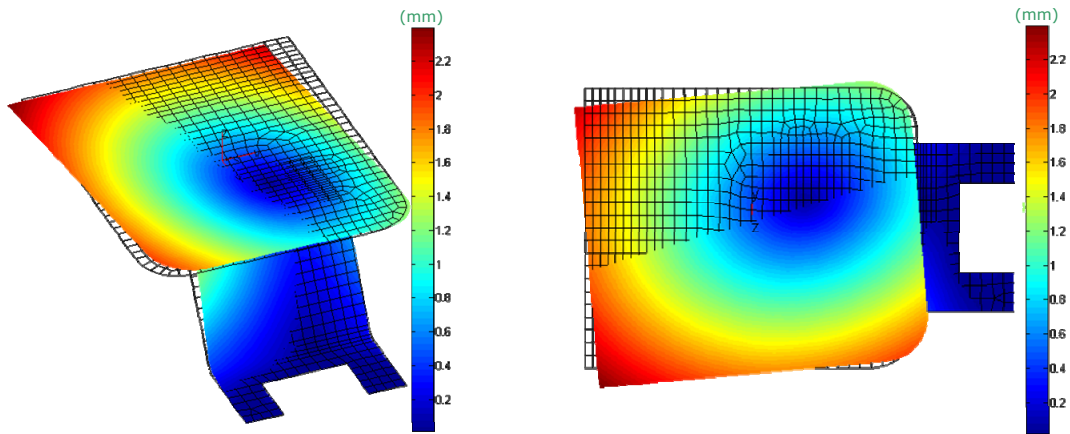
Figures 6.19 and 6.20 show the initial mean deviations of part A and D, respectively, according to statistical deviations in Table 6.5, and with reference to the fixturing frames FX-A and FX-D (see Figure 6.17). No gravity effect and contact points were

Chapter 6. Linear Analysis of Compliant Assemblies

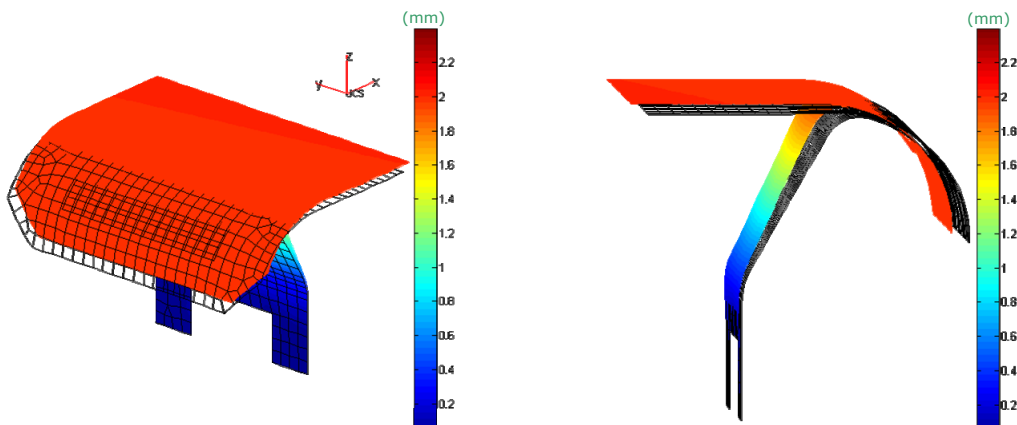
assumed. Two assembly configurations were analyzed (see Table 6.6). In the first configuration (I), part A and B are assembled in the first station (fixturing frame is FX-A+FX-B); part C and D are assembled in the second station (fixturing frame is FX-C+FX-D); finally, sub-assemblies (A+B) and (C+D) are joined in the third station. In the second assembly configuration (II), parts are assembled into a single-station. For both assembly configurations the releasing fixturing frame is FX-A+FX-D.

	Station I	Station II	Station III
(I)	(A+B)	(C+D)	(A+B)+(C+D)
(II)	(A+B+C+D)		

Table 6.6: assembly configurations

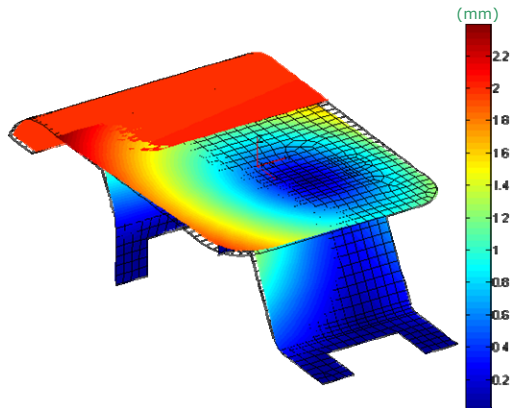


releasing phase in the first station (first assembly configuration)

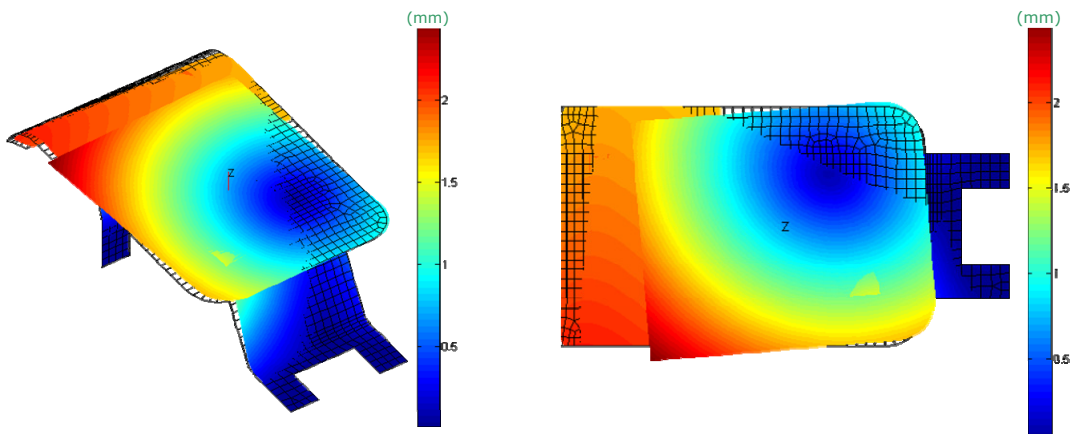


releasing phase in the second station (first assembly configuration)

Chapter 6. Linear Analysis of Compliant Assemblies



positioning phase in the third station (first assembly configuration)



releasing phase in the third station (first assembly configuration)

Figure 6.21: SVA-FEA results in the first assembly configuration

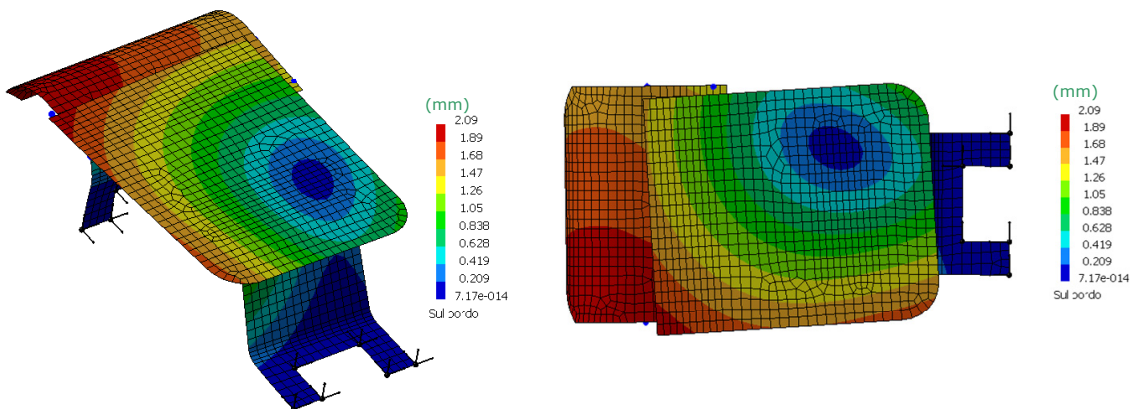


Figure 6.22: TAA results in the first assembly configuration. Releasing phase in the third station

Chapter 6. Linear Analysis of Compliant Assemblies

Figure 6.21 shows the SVA-FEA results, related to the releasing phases of stations I, II and III, with respect to the first assembly configuration. It can be pointed out that in the station I, part B is positioned and then fastened with respect to part A, which presents an initial shape deviation. Therefore, when the sub-assembly (A+B) is repositioned in the third station, and here fastened to sub-assembly (C+D), it preserves previous geometrical deviations: part B and C are not aligned in the final assembly spring-back phase.

Figure 6.22 shows the final assembly deformations as calculated into TAA® with respect to the first assembly configuration.

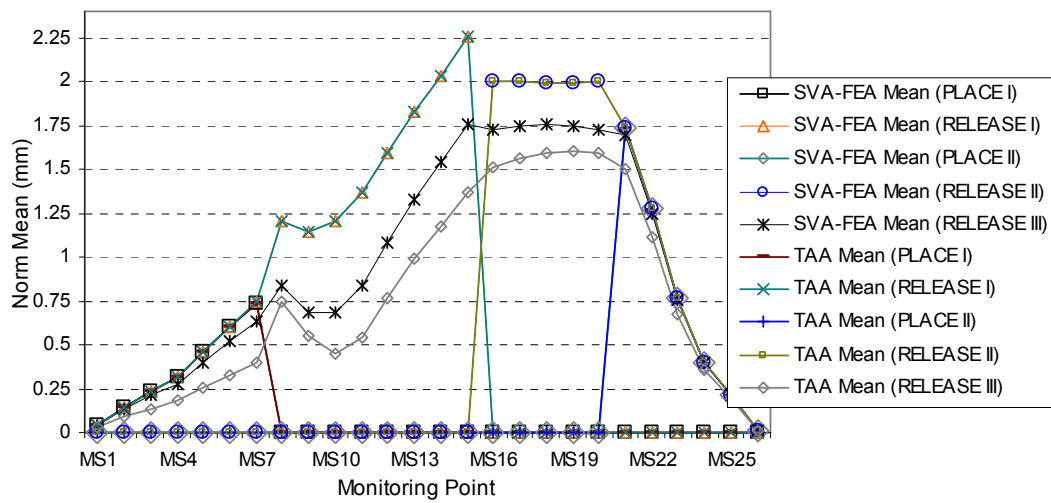


Figure 6.23.a: mean deviations at monitoring points. Euclidean norm is adopted. First assembly configuration. d=5 mm

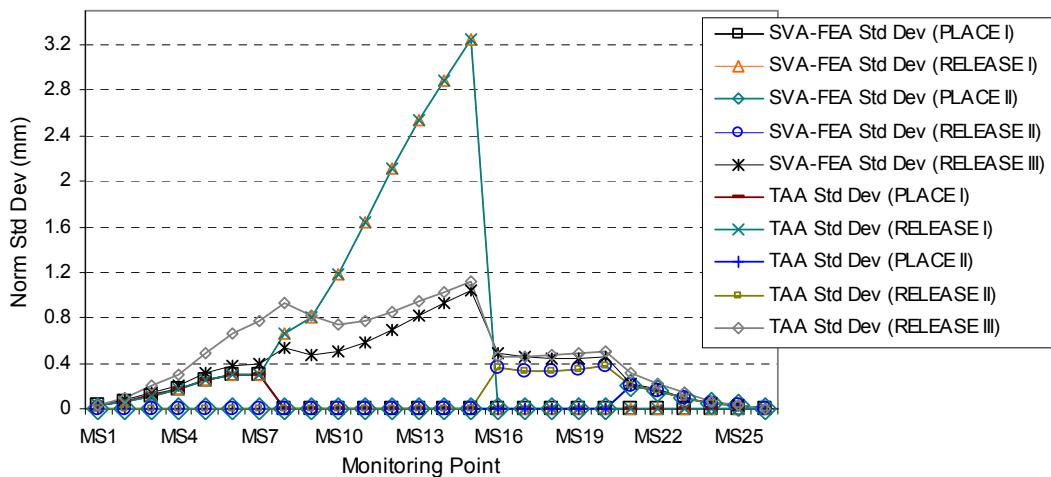


Figure 6.23.b: standard deviations at monitoring points. Euclidean norm is adopted. First assembly configuration. d=5 mm

Chapter 6. Linear Analysis of Compliant Assemblies

Figures 6.23 plots the output deviations, related to monitoring points, in terms of mean and standard deviations.

In order to estimate the numerical error between SVA-FEA and TAA® results, the correlation and the RSS indexes were adopted. Tables 6.7 and 6.8 show those indexes both for mean and standard deviations. It can be highlighted a good correlation among results.

Assembling phase	Correlation (Mean)	Correlation (Std Dev)
PLACE I	0.999983	0.999987
RELEASE I	1.000000	1.000000
PLACE II	0.999997	0.999997
RELEASE II	1.000000	1.000000
RELEASE III	0.988294	0.939633

Table 6.7: correlation indexes.
First assembly configuration

Assembling phase	RSS% (Mean)	RSS% (Std Dev)
PLACE I	0.006078	0.005618
RELEASE I	0.000000	0.000000
PLACE II	0.006835	0.000930
RELEASE II	0.000000	0.000000
RELEASE III	0.998097	0.840581

Table 6.8: RSS errors.
First assembly configuration

However, a relative high RSS error both for the mean and standard deviations, with respect to the final releasing phase (RELEASE III) in the station III, is observed. All this is due to the different ways to model spot weld joints. In SVA-FEA they were assumed as elastic beams, characterized by their beam diameter, d ; instead, rigid kinematic joints (planar, revolute or spherical) are implemented in TAA®.

To better understand this issue, what happens, when the stiffness of the weld spots in SVA-FEA increases (by varying the weld spot diameter), is shown in the Table 6.9.

Weld spot diameter	RSS% (Mean)	RSS% (Std Dev)
$d = 5 \text{ mm}$	0.998097	0.840581
$d = 10 \text{ mm}$	0.774957	0.893082
$d = 20 \text{ mm}$	0.400250	0.664372

Table 6.9: RSS errors.
Influence of spot weld diameter.
First assembly configuration

Chapter 6. Linear Analysis of Compliant Assemblies

When the weld spot diameter increases (the stiffness of the weld spot element increases) the RSS error reduces, accordingly.

Figures 6.24 plots the output deviations, related to monitoring points, in terms of mean and standard deviations, for the second assembly configuration (II).

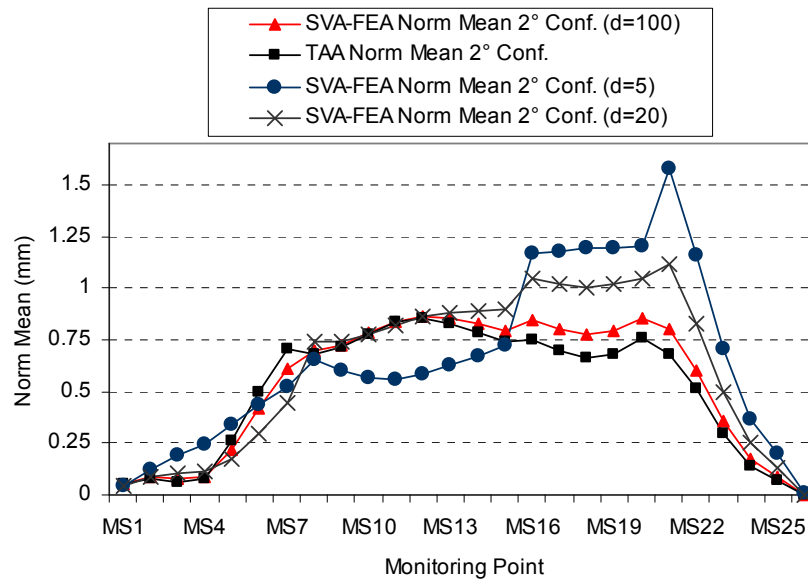


Figure 6.24.a: mean deviations at monitoring points. Euclidean norm is adopted. Second assembly configuration

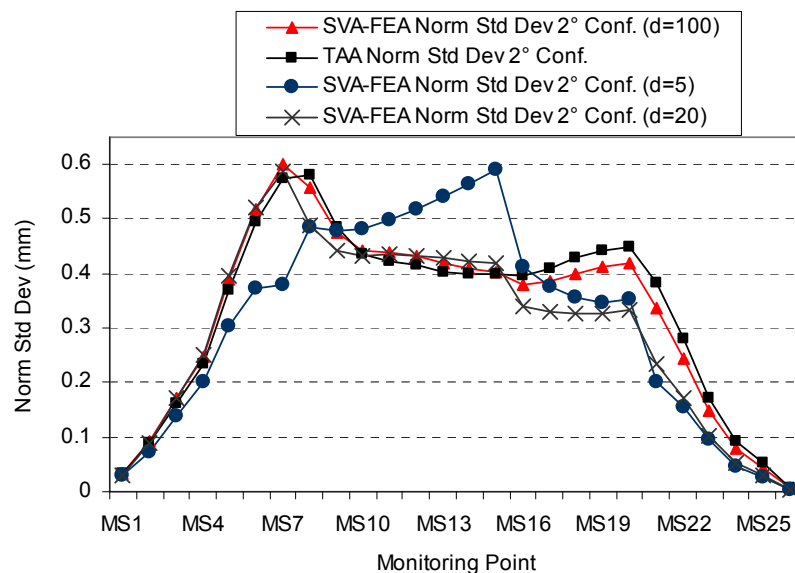


Figure 6.24.b: standard deviations at monitoring points. Euclidean norm is adopted. Second assembly configuration

Chapter 6. Linear Analysis of Compliant Assemblies

Here, in order to show the influence of the spot weld diameter, three values were adopted: 5, 20 and 100 mm. With respect to TAA® results, mean and standard deviations of the monitoring points MS15 and MS16, which correspond to the fastening point FT9, are equal among them. In SVA-FEA, instead, only when the spot weld diameter is 100 mm (the stiffness of the spot weld increases) the difference among deviations of points MS15 and MS16 decreases and the RSS error reduces, accordingly.

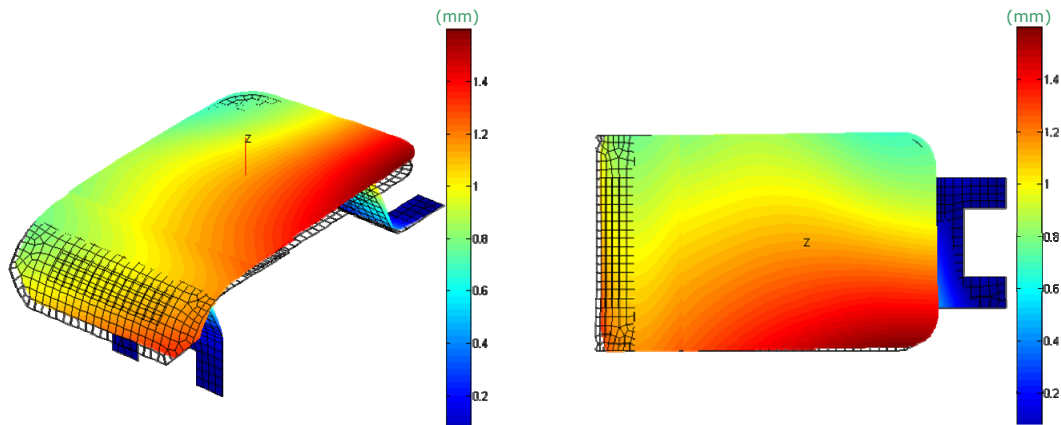


Figure 6.25: SVA-FEA results in the second assembly configuration (final releasing). Spot weld diameter is 100 mm

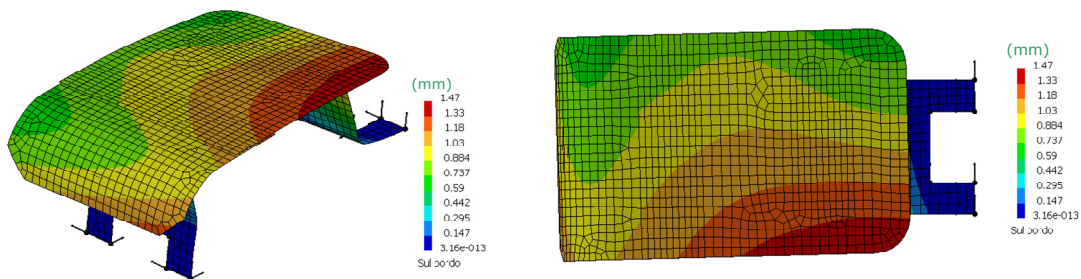


Figure 6.26: TAA results in the second assembly configuration (final releasing). Spot weld diameter is 100 mm

Tables 6.10 and 6.11 show the correlation indexes and the RSS errors for the second assembly configuration, respectively. SVA-FEA and TAA® mean displacements are shown in Figures 6.25 and 6.26, respectively.

Looking at Figures 6.21 and 6.25, 6.22 and 6.26, it can be pointed out that final results, both for SVA-FEA and TAA®, are strong different in the two analyzed assembly configurations. In particular, when the multi-station configuration (I) is adopted, at the final releasing phase, part B and C are not aligned, due to deviations which occur in the previous stations. Instead, in the single-station assembly configuration (II), part B and C are joined in their nominal position, where parts are correctly aligned. The final spring-back is due to the initial deviations of part A and D.

Chapter 6. Linear Analysis of Compliant Assemblies

Assembling phase	Correlation (Mean)	Correlation (Std Dev)
RELEASE	0.985695	0.992584

Table 6.10: correlation indexes.
Second assembly configuration. d=100 mm

Assembling phase	RSS% (Mean)	RSS% (Std Dev)
RELEASE	0.328162	0.106620

Table 6.11: RSS errors.
Second assembly configuration. d=100 mm

6.8 Summary

This Chapter presented a methodology, called SVA-FEA, for the statistical variation analysis of compliant parts in multi-station assembly processes.

The method proposes a global sensitivity matrix, S , in order to simulate multi-station processes and so the re-positioning phase from one station to another one. “Linear contacts” are also included to avoid part-to-part intersecting and their effect on the sensitivity matrix was also described.

Two linear static FEA runs are required for each assembly station in order evaluate the sensitivity matrix. The first run calculates the reaction forces at fixturing and fastening points; the second one evaluates the spring-back effect after releasing fastening tools and re-positioning the assembly onto a new fixture frame.

The proposed method was implemented in a MatLAB®-based user interface that drives the user to define fixturing, contact and fastening points, their variability and the assembly tree. Finally, the SVA-FEA interface allows post-processing results.

SVA-FEA interface were used to analyze two case studies. In both case results were compared with ones coming from the TAA® module integrated into CATIA® CAD system. The first case study was a two-part assembly. Simulation results showed a good agreement between the two simulation software. Moreover, contact points strongly influence final assembly deviations. The second case study was a four-part assembly. Two assembly configurations were analyzed: a three-station assembly compared to a single-station one. In both cases, results showed high numerical correlation. Moreover, the high RSS error between SVA-FEA and TAA®, with respect to the final releasing phase, was due to different ways used to model fastening joints. In SVA-FEA, elastic beams may be used, whereas also rigid joints are implemented in TAA. The influence of the weld spot diameter, in SVA-FEA, was numerically outlined. When the beam element, used in SVA-FEA to model the spot weld joint, increases its stiffness by varying its diameter, the RSS error between SVA-FEA and TAA® results reduces drastically. Moreover, numerical simulations highlighted that the assembly sequence strongly influences final assembly deviations. Selecting the best assembly configuration, which allows to reduce assembly deviations, is not a trivial task. To do it, optimization procedures could be implemented in SVA-FEA.

In the SVA-FEA method all sources of variation are assumed independent each other. This assumption permits to use the method of influence coefficient. However, this hypothesis does not allow to analyze the covariance effect among input statistical data. This limitation will be overcome in Chapter 7, where a morphing mesh approach will be described to simulate free shape errors.

Chapter 7.

NON-LINEAR ANALYSIS OF COMPLIANT ASSEMBLIES¹

As observed in Chapter 2, despite many efforts made into the field of variational modeling, free shape errors are often neglected. However, for certain assembly processes, such as those involving sheet-metal parts, these kinds of errors may strongly affect final assembly deviations.

The present Chapter develops a numerical technique to model and simulate free shape errors to be used into statistical numerical simulations of compliant assemblies.

The aim is to simulate variational shape of parts according to a small number of control points chosen on the part geometry. Starting from the nominal CAD geometry, a FE model is derived. Mesh nodes are moved by applying a morphing mesh procedure. The morphed parts are then used to accomplish the variational assembly analysis following the classical PCFR cycle. In order to achieve statistical results, Monte Carlo simulation is performed: a set of control points driving the *perturbed* parts is generated at each iteration; these parts are then assembled and results are stored. During each a Monte Carlo step, part geometry is updated following the local deformation related to that specific assembly sequence. All this makes the numerical simulation non-linear.

¹ This Chapter is based on: Franciosa P., Gerbino S., Patalano S., *Simulation of Variational Compliant Assemblies with Shape Errors Based on Morphing Mesh Approach*, submitted to Int. Journal of Advanced Manufacturing Technology, 2009.

7.1 Introduction

In Chapter 5, where only rigid part assembled were modeled and simulated, the variational model to generate variational features was presented. In that contest free shape error were neglected. Only rigid translational and rotational parameters were accounted.

In Chapter 6 a linear methodology to do tolerance analysis of compliant assemblies was presented. This methodology was based on the main assumption of independent input variations. This means that free shape errors were not considered into the numerical simulation. The present Chapter overcomes this limit by introducing a morphing mesh procedure to generate free shape features to be used into a non-linear simulation involving compliant assemblies. In this way, input variations now become correlated each other.

Figure 7.1 shows the general work-flow proposed to perform the variational analysis of compliant assemblies with shape errors.

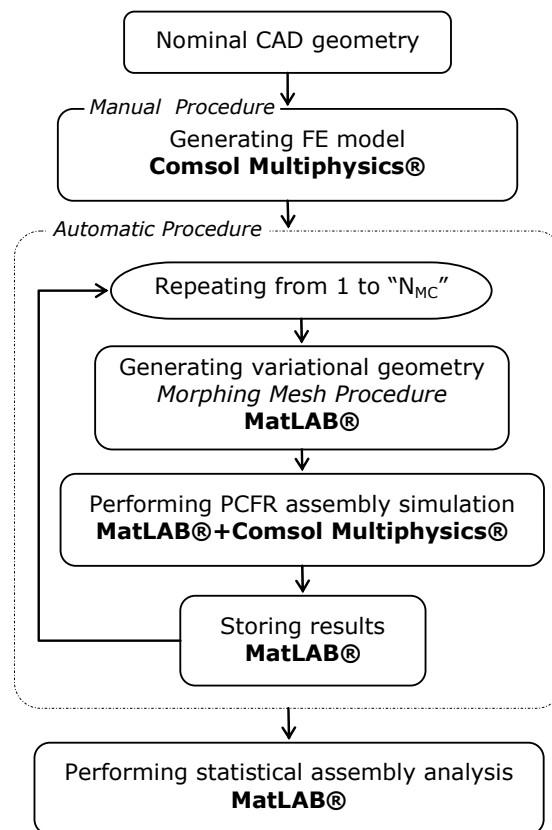


Figure 7.1: general work-flow of the methodology

The software architecture is based on MatLAB® environment, that controls the whole analysis, running in background mode all FEA simulations by using Cmsol

Chapter 7. Non-Linear Analysis of Compliant Assemblies

Multiphysics®² routines. Comsol Multiphysics® is a very powerful tool to solve for Partial Differential Equation (PDE) with a FEM approach. All this allows to easily manage expressions and functions, asking for numerically evaluating integrals or derivatives defined on geometrical boundaries or sub-domains.

FE model is generated from the nominal CAD geometry. Boundary conditions, contact and identity pairs (as surface-to-surface type) and material properties are defined, accordingly.

Once FE model is defined, it is imported into MatLAB® where the PCFR cycle is simulated. Typically, during assembly processes parts are positioned on fixturing frames. Then, they are clamped and fastened. Finally, fixturing frames are removed and assembled parts are released reaching their final configuration. In order to obtain statistical results, Monte Carlo approach is adopted (N_{MC} is the total number of Monte Carlo simulations). For each step, the nominal geometry is *perturbed* according to deviations occurring at control points (Morphing Mesh Procedure). Then, the PCRF cycle is performed and the final assembly configuration is stored. Statistical results are calculated by taking into account all assembly configurations.

7.2 Free Shape Feature Modeling

In Chapter 5 variational features were modeled by using 4x4 transformation matrices. Small rotational and translational parameters were introduced to simulate rigid motions of the feature within the tolerance range. Free shape errors were there neglected. This Section focuses on how to model and simulate such kind of errors. Weighted functions, following a morphing mesh approach, are introduced in order to correlate points inside the feature. Then, a reject method (similar to that one introduced in Chapter 5) is adopted to statistically sample free form features.

7.2.1 Morphing Mesh Procedure

Morphing mesh is a well-know technique used in computer graphic applications as a powerful tool for shape modeling and designing (Borrel, 1994; Raffin, 2000). The user defines a set of control points, by giving a desired displacement and the relative radius of influence for each point. Control points directly influence the final shape of the *morphed* objects, and this shape can be fine-tuned by adjusting the influence radius or the position of each control point.

² Comsol Multiphysics® is a trademark of Comsol AB.

Chapter 7. Non-Linear Analysis of Compliant Assemblies

Variational parts, to be used in the PCFR assembly simulation, are here generated through a Morphing Mesh Procedure (MMP). MMP works on a large amount of points describing the whole geometry. These points are directly derived from the FE model, used to perform computer simulation of the assembly process.

Generally speaking, a FE model is made of nodes connected by elements. The number of nodes depends on the shape function of each element (Zienkiewicz, 2005). Let \mathbf{V} and \mathbf{F} be the matrix of nodes to be moved and the matrix of elements, respectively; \mathbf{V} is a $N_n \times 3$ rectangular matrix, where N_n is the total number of nodes; whereas, \mathbf{F} is a $N_e \times 3$ rectangular matrix, where N_e is the total number of elements. Moreover, let \mathbf{P}_c be the matrix of control points; \mathbf{P}_c is an $N_{cp} \times 3$ rectangular matrix, where N_{cp} is the total number of control points (see Figure 7.2). R_i is the influence radius related to the i -th control point. The region embedded within the influence radius is here called *influence hull*.

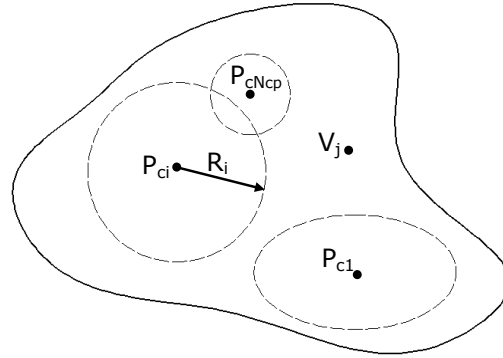


Figure 7.2: definition of control points and relative influence hulls

The displacement, $\Delta \mathbf{V}_j$, of any mesh node \mathbf{V}_j , is calculated as a weighted mean of all control points \mathbf{P}_c , as stated in equation (7.1).

$$\begin{cases} \Delta V_{j,x} = \sum_{i=1}^{N_{cp}} W_i \cdot M_{i,x} \\ \Delta V_{j,y} = \sum_{i=1}^{N_{cp}} W_i \cdot M_{i,y} \\ \Delta V_{j,z} = \sum_{i=1}^{N_{cp}} W_i \cdot M_{i,z} \\ \dots \\ W_i = f(\mathbf{V}_j, \mathbf{P}_{ci}, R_i) \forall i = 1, 2, \dots, N_{cp} \end{cases} \quad (7.1)$$

where \mathbf{W} and \mathbf{M} are the weighted and the morphing matrix, respectively. \mathbf{W} is a $1 \times N_{cp}$ row vector, while \mathbf{M} is a $N_{cp} \times 3$ matrix.

The i -th weighted element depends on: (I) the point \mathbf{V}_j ; (II) the control point \mathbf{P}_{ci} , and (III) its influence radius R_i , as expressed into the second equation in (7.1) The influ-

ence radius defines the 3D region within which any node is influenced by the related control point.

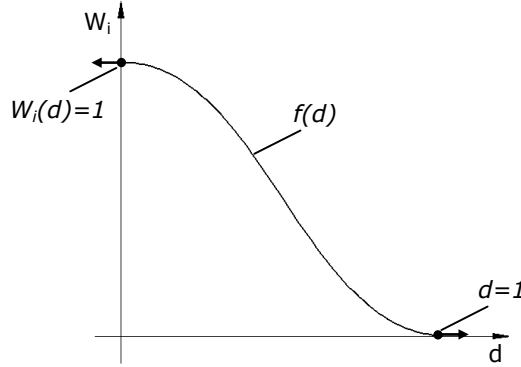


Figure 7.3: definition of the basic function, f

f is the so-called basic function. f may be assumed as a piecewise Bezier curve or a B-spline based function as proposed in (Raffin, 1998). However, in this dissertation, in order to easily handle this function, a third degree polynomial function is adopted. Therefore, one can write:

$$W_i = f\left(\frac{\|\mathbf{V}_j - \mathbf{P}_{ci}\|}{R_i}\right) = f(d) = 1 - 3 \cdot d^2 + 2 \cdot d^3, \forall i = 1, 2, \dots, N_{cp} \quad (7.2)$$

Relation (7.2) states that the function f is equal to 1 when \mathbf{V}_j and \mathbf{P}_{ci} are coincident, and tends toward zero for points \mathbf{V}_j whose distance from \mathbf{P}_{ci} is greater than zero. To assure smoothing shape, function f has zero slopes at the two end points (Figure 7.3). In equation (7.2) d is the a -dimensional distance from point \mathbf{V}_j to the control point \mathbf{P}_{ci} .

In order to evaluate the morphing matrix, \mathbf{M} , equation (7.1) is specified with respect to all control points \mathbf{P}_c . Then, it has:

$$\begin{cases} \Delta \mathbf{P}_{c1} = \mathbf{W}(\mathbf{P}_{c1}) \cdot \mathbf{M} \\ \Delta \mathbf{P}_{c2} = \mathbf{W}(\mathbf{P}_{c2}) \cdot \mathbf{M} \\ \dots \\ \Delta \mathbf{P}_{cN_{cp}} = \mathbf{W}(\mathbf{P}_{cN_{cp}}) \cdot \mathbf{M} \end{cases} \rightarrow \Delta \mathbf{P}_c = \mathbf{W}(\mathbf{P}_c) \cdot \mathbf{M} \quad (7.3)$$

where $\Delta \mathbf{P}_c$ is the $N_{cp} \times 3$ matrix of displacements related to control points, whereas $\mathbf{W}(\mathbf{P}_c)$ is a $N_{cp} \times N_{cp}$ square matrix. Thus, morphing matrix \mathbf{M} can be calculated by solving for the following linear system:

$$\mathbf{M} = \mathbf{W}(\mathbf{P}_c)^{-1} \cdot \Delta \mathbf{P}_c \quad (7.4)$$

It should be noted that matrix $\mathbf{W}(\mathbf{P}_c)$ is singular when two control points are coincident. When this happens the system (7.4) has not a unique solution and morphing matrix \mathbf{M} is un-determined. To solve this issue, the linear system in (7.4) may be solved with a least-squares approach. To do it, the pseudo-inverse of the matrix $\mathbf{W}(\mathbf{P}_c)$ must be evaluated (Strang, 2009). Commercial routines are based on the Singular Value Decomposition (SVD). In this dissertation, the MatLAB®'s "pinv" function has been adopted (see also Annex B).

Once morphing matrix \mathbf{M} is known, displacements of any mesh node \mathbf{V}_j can be calculated by applying the relationship (7.1).

In order to get a more powerful control on the deformation of the nominal geometry, different influence hulls can be adopted. If the radius of influence is constant, the influence hull becomes a sphere. Instead, an ellipsoid can be derived by defining three principal radiuses (defined along the principal directions of the ellipsoid). More general influence hulls can be defined as proposed in (Raffin, 1998). However, in this dissertation only spheres and ellipsoids are considered since they offer a more intuitive control.

For a sphere hull, the a-dimensional distance, d_s , is:

$$d_s = \frac{\sqrt{(V_{j,x} - P_{ci,x})^2 + (V_{j,y} - P_{ci,y})^2 + (V_{j,z} - P_{ci,z})^2}}{R_s} \quad (7.5)$$

whereas, for an ellipsoid hull it becomes:

$$d_e = \sqrt{\frac{(V_{j,x} - P_{ci,x})^2}{R_{e,1}^2} + \frac{(V_{j,y} - P_{ci,y})^2}{R_{e,2}^2} + \frac{(V_{j,z} - P_{ci,z})^2}{R_{e,3}^2}} \quad (7.6)$$

where R_s is the radius of the sphere, while $R_{e,1}$, $R_{e,2}$ and $R_{e,3}$ are the principal radiuses of the ellipsoid.

7.2.2 Statistical Simulation

Equation (7.1) allows to *morph* any point of the feature, once the morphing matrix, \mathbf{M} , is known. The aim is to statistically sample free shape features for a given tolerance range, T (supposed symmetrical). Generally speaking, taking into account equation (7.1), the following condition has to be satisfied:

$$\begin{aligned} |\mathbf{N}_j \cdot \Delta \mathbf{V}_j| &\leq \frac{T}{2} \rightarrow |\mathbf{N}_j \cdot [\mathbf{W}(\mathbf{V}_j) \cdot \mathbf{M}]| \leq \frac{T}{2} \\ \forall j &= 1, 2, \dots, N_n \end{aligned} \quad (7.7)$$

Chapter 7. Non-Linear Analysis of Compliant Assemblies

where \mathbf{N}_j is the unit vector normal to the feature into point \mathbf{V}_j . Equation (7.7) states that any point belonging to the morphed feature lies inside the tolerance range.

Since the morphing matrix \mathbf{M} may be expressed as into equation (7.4), relationship (7.7) may be rearranged as into (7.8).

$$\left| \mathbf{N}_j \cdot \left[\mathbf{W}(\mathbf{V}_j) \cdot \mathbf{W}(\mathbf{P}_c)^{-1} \cdot \Delta \mathbf{P}_{c\eta} \right] \right| \leq \frac{T}{2} \quad (7.8)$$

$$\forall j = 1, 2, \dots, N_n$$

The matrix of displacements related to control points, $\Delta \mathbf{P}_{c\eta}$, is randomly generated. Therefore, one can write:

$$\Delta \mathbf{P}_{c\eta i} = \Delta \mathbf{P}_{ci} \cdot \eta_{Pci} \quad (7.9)$$

$$\forall i = 1, 2, \dots, N_{cp}$$

where $\boldsymbol{\eta}_{Pc} = [\eta_{Pc1}, \eta_{Pc2}, \dots, \eta_{PcN_{cp}}]$ is the vector of independent random values.

Finally, having constraints defined in (7.8), features are sampled by using a Monte Carlo approach. Depending on the adopted statistical law, each Monte Carlo step calculates a specific matrix $\Delta \mathbf{P}_c$ (see equation 7.9). If all constraints in (7.8) are satisfied then the result is stored. This procedure is stopped when the number of sampled features becomes equal to the number of Monte Carlo simulations.

MMP may now generate variational parts. User has to set the position of control points, their influence radius and relative displacements.

$$\Delta \mathbf{V}_j = f(\mathbf{V}_j, \mathbf{P}_{ci}, R_i) \rightarrow \begin{cases} U_m = f(\mathbf{V}_j, \mathbf{P}_{ci}, R_i) \\ V_m = f(\mathbf{V}_j, \mathbf{P}_{ci}, R_i) \\ W_m = f(\mathbf{V}_j, \mathbf{P}_{ci}, R_i) \end{cases} \quad (7.10)$$

Then, with respect to the *Global Coordinate Frame* (GCF), directly derived from the CAD model and here called $(\mathbf{X}_0, \mathbf{Y}_0, \mathbf{Z}_0)$, any geometrical point \mathbf{V}_j with components (x, y, z) is transformed by means of the MMP as in equation (7.10). (U_m, V_m, W_m) are the displacements of the point \mathbf{V}_j with respect to the GCF.

7.3 PCFR Simulation

A typical PCFR cycle starts with placing parts on a fixturing frame, followed by clamping parts in the fixture, then joining them together, and finally releasing fixtures and clamps.

A critical aspect is related to the joining and the releasing phases. In fact, fasten-

ing tools force mating surfaces together at the joining area. Compliant parts tend to deform locally during these operations, and this changes contact conditions: surfaces initially in contact may move apart producing gaps or overlappings. Moreover, depending on the joint method, multiphysics phenomena may occur in the contact zones, such as material plastic deformation, thermo-structure interaction, thermo-electrical-structure interaction, friction phenomena. A non-linear methodology, aiming to numerically simulate the whole PCFR cycle, taking into account contact area among parts being assembled, is here proposed. To do it a FEM approach is adopted. Figure 7.4 shows the general work-flow used to perform the PCFR cycle.

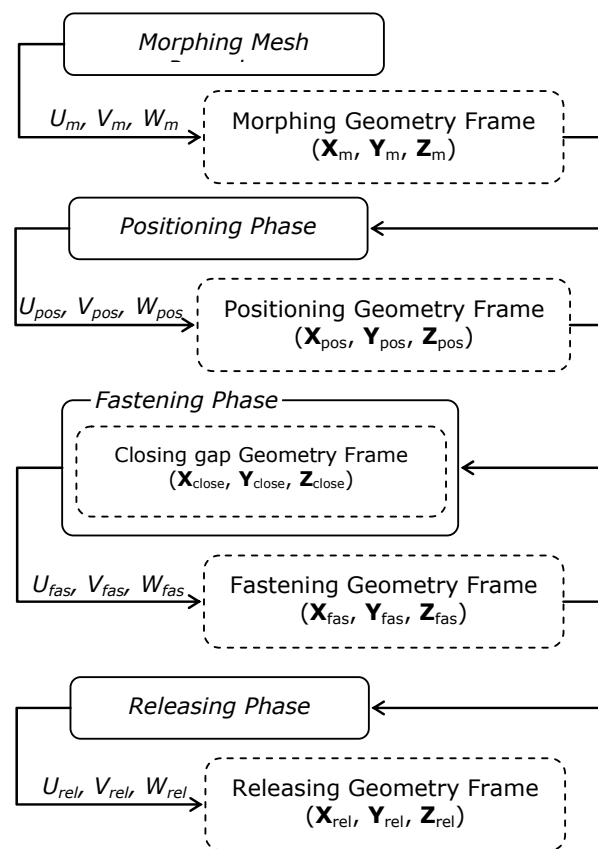


Figure 7.4: work-flow of the non-linear methodology proposed for PCFR cycle simulation

Generally speaking, three consecutive FEA runs are required to perform the whole PCFR simulation. For instance, the first FEA run solves the positioning and clamping phase (phase I), considering initial variational parts, derived from the MMP. Fastening phase (phase II) is achieved into the second FEA simulation, taking into account the deformed geometry derived from the first FEA run. Finally, the releasing phase (phase III) is solved in the third FEA run, by applying the residual stresses (initial stresses) generated during previous phases.

$$\text{DGF}_i : \begin{cases} X_i = X_0 + \dots + U_i \\ Y_i = Y_0 + \dots + V_i \\ Z_i = Z_0 + \dots + W_i \end{cases} \quad (7.11)$$

After the i -th phase ($i=I, II, III$), the geometrical domain is updated according to the displacement field, (U_i, V_i, W_i) . In this way, an updated geometry frame, here called *Deformed Geometry Frame* (DGF) is calculated. For the i -th phase, the DGF_i is defined as in (7.11). For example, the “fastening phase” is performed with respect to the *Positioning Geometry Frame*. This frame is defined as:

$$\text{DGF}_{\text{pos}} : \begin{cases} X_{\text{pos}} = X_0 + U_m + U_{\text{pos}} \\ Y_{\text{pos}} = Y_0 + V_m + V_{\text{pos}} \\ Z_{\text{pos}} = Z_0 + W_m + W_{\text{pos}} \end{cases} \quad (7.12)$$

In particular, DGF_0 corresponds to the nominal CAD geometry.

PCFR cycle calculation has to be iterated for each Monte Carlo step. For each step, non-linear equations have to be solved for. Convergence problems may additionally arise from iterative solution of non-linear tasks. Furthermore, all this may become very difficult to automate³. Due to the above mentioned reasons, an original and efficient methodology to solve for the contact problem between parts being joined will be showed in the following.

In addition, the proposed methodology assumes the following hypotheses:

- friction among parts is neglected;
- only spot weld joints are considered; and,
- local thermal effects are not accounted.

7.3.1 Positioning/Clamping Phase

First FEA run aims to simulate the location of parts onto fixturing frames. Generally speaking, fixturing frames are designed to determine location (mate or locator constraint) of a part and to keep the effect of the location (contact or effector constraint) (Whitney, 2004). In the contest of the present dissertation, the positioning phase determines location of parts, while the clamping phase keeps such constraints. Changing the order in the positioning and clamping phases influences final results.

³ In order to make the numerical model as much as possible closer to real assembly processes, fastening tools, such as welding guns, should be modeled. However, all this requires the definition of other contact pairs at the interface between parts being joined and fastening tools. Such model does not assure the convergence of the solution, especially when an automatic approach - as in this dissertation - is required.

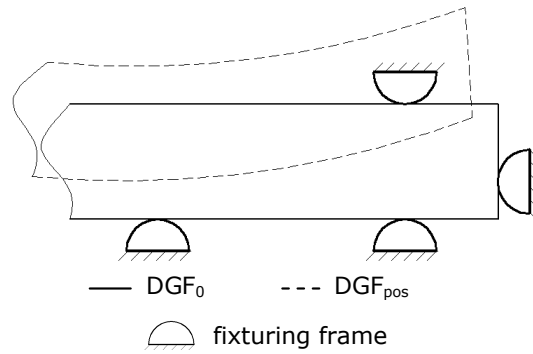


Figure 7.5: positioning modeling

However, positioning and clamping fixturing frames are assumed acting simultaneously. Then, in the next the positioning/clamping phase will be simply named positioning phase briefly.

Let DGF_0 and DGF_m be the nominal geometry frame and the morphing geometry frame, respectively. Looking at Figure 7.5, it can be highlighted that parts, after applying MMP, may not fit boundaries corresponding to fixturing frames (defined on DGF_0).

First FEA run is performed on the DGF_m . Contact pairs are introduced to avoid part-to-part penetration. Furthermore, boundary constraints are added in order to fit the actual boundary on the fixturing frame. To do this, Boundary Conditions (BCs) are defined as follows:

$$\begin{cases} BC_{Xm} = -U_{m,bnd,fixt} + \varepsilon_{x,fixt} \\ BC_{Ym} = -V_{m,bnd,fixt} + \varepsilon_{y,fixt} \\ BC_{Zm} = -W_{m,bnd,fixt} + \varepsilon_{z,fixt} \end{cases} \quad (7.13)$$

where $(U_{m,bnd,fixt}, V_{m,bnd,fixt}, W_{m,bnd,fixt})$ are the displacement fields calculated with MMP and related to boundaries to be fit on the fixturing frame. Fixturing errors may be accounted, when necessary, as $(\varepsilon_{x,fixt}, \varepsilon_{y,fixt}, \varepsilon_{z,fixt})$. The parameter ε_{fixt} may be assumed as an input statistical variable and generated during each Monte Carlo step.

7.3.2 Fastening Phase

Fastening phase is performed with respect to the DGF_{pos} . When two parts are fastened with weld joints, firstly, tools force mating surfaces to close their gap, and then parts are joined together. Fastening phase is made of two consecutive sub-steps: closing gap phase and joining phase.

It can be pointed out that in the DGF_{pos} mating surfaces do not penetrate each

other. In fact, during the positioning phase, penetrations among parts have been already avoided, by assigning contact pairs.

Fastening Phase: Closing Gap

A spot weld is characterized by its radius, R_{weld} , the nominal position of its centre, \mathbf{P}_{weld} , and the direction, \mathbf{N}_{weld} , along which fastening tool acts. Moreover, the source boundary (bnd_s , in Figure 7.6) and the destination boundary (bnd_d , in Figure 7.6) are defined. In addition, a local coordinate frame (\mathbf{X}_{weld} , \mathbf{Y}_{weld} , \mathbf{Z}_{weld}) is assigned having the local \mathbf{Z}_{weld} axis directed along the \mathbf{N}_{weld} vector.

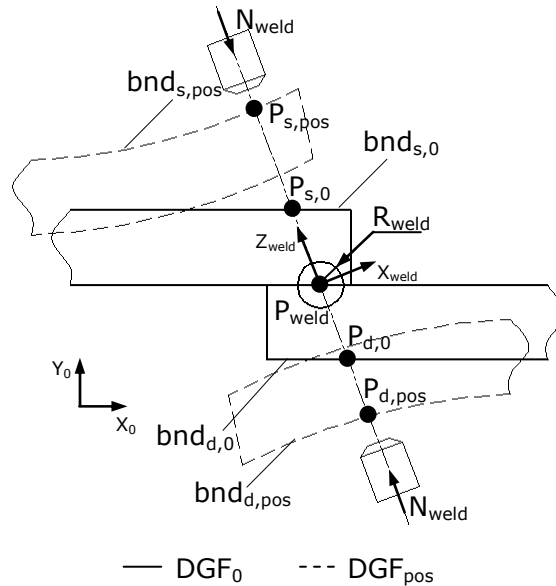


Figure 7.6: closing gap phase

Looking at Figure 7.6, the aim is to close the gap, defined as:

$$\begin{cases} \mathbf{g}_s = \mathbf{P}_{s,\text{pos}} - \mathbf{P}_{s,0} \\ \mathbf{g}_d = \mathbf{P}_{d,\text{pos}} - \mathbf{P}_{d,0} \end{cases} \quad (7.14)$$

Pairs $(\mathbf{P}_{s,\text{pos}}, \mathbf{P}_{s,0})$ and $(\mathbf{P}_{d,\text{pos}}, \mathbf{P}_{d,0})$ depend on the initial deformation field related to the DGF_{pos} . Point $\mathbf{P}_{s,\text{pos}}$ is calculated as intersection of the line, having direction \mathbf{N}_{weld} and passing through point \mathbf{P}_{weld} , with the mesh belonging to the boundary $\text{bnd}_{s,\text{pos}}$. The same procedure may be adopted for the opposite point. In the following only triangular mesh are considered. However, the same approach may be easily extended to quadrilateral or mixed meshes. As described above, the mesh is characterized by means of nodes and elements.

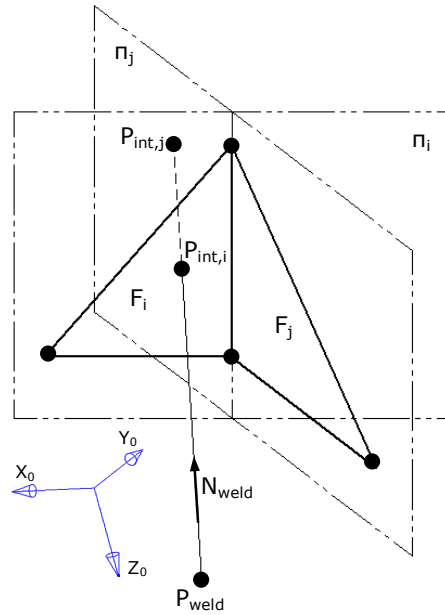


Figure 7.7: line-mesh intersection procedure

Here, the matrix of nodes is updated according to the DGF_{pos} . Therefore, point $\mathbf{P}_{s,pos}$ is calculated by evaluating, on the so-updated mesh, point \mathbf{P}_{int} , which is the closest to point \mathbf{P}_{weld} (see also Figure 7.7). To do it, for the i -th triangular element, the intersection point $\mathbf{P}_{int,i}$ between the line $(\mathbf{N}_{weld}, \mathbf{P}_{weld})$ and the plane π_i , whose element F_i belongs to, is evaluated. Then, if point $\mathbf{P}_{int,i}$ belongs to the planar face F_i (“inpolygon” function available in MatLAB® has been used to check this condition) it can be written:

$$\mathbf{P}_{s,pos} \equiv \mathbf{P}_{int} = \min_i(d_i), \forall i = 1, \dots, N_e \quad (7.15)$$

$$d_i = \|\mathbf{P}_{int,i} - \mathbf{P}_{weld}\|$$

Once gaps \mathbf{g}_s and \mathbf{g}_d have been evaluated, BCs may be assigned with respect to the local coordinate frame $(\mathbf{X}_{weld}, \mathbf{Y}_{weld}, \mathbf{Z}_{weld})$. Then, it can be written:

$$BC_{s,Zweld} = \begin{cases} (-\mathbf{g}_s + \boldsymbol{\varepsilon}_{fast}), & \text{if } \Delta_s \leq R_{weld} \\ \text{free displacement,} & \text{otherwise} \end{cases} \quad (7.16)$$

$$BC_{d,Zweld} = \begin{cases} (-\mathbf{g}_d + \boldsymbol{\varepsilon}_{fast}), & \text{if } \Delta_d \leq R_{weld} \\ \text{free displacement,} & \text{otherwise} \end{cases}$$

$$\Delta_i = \|\mathbf{P}_{pos} - \mathbf{P}_{i,pos}\|, \forall i = s, d$$

where \mathbf{P}_{pos} is any point defined into the DGF_{pos} . The parameter $\boldsymbol{\varepsilon}_{fast}$ has been introduced in (7.16) to take into account also an error, if any, in the nominal closure point of the

Chapter 7. Non-Linear Analysis of Compliant Assemblies

weld guns. It should be considered that in the local coordinate frame, (\mathbf{X}_{weld} , \mathbf{Y}_{weld} , \mathbf{Z}_{weld}), BCs are not assigned along local \mathbf{X}_{weld} and \mathbf{Y}_{weld} directions: displacements are free along those directions.

Finally, in the second FEA run a constrained displacement as in equation (7.16) is assigned at any point \mathbf{P}_{pos} , belonging to the sphere of radius R_{weld} and centre $\mathbf{P}_{i,\text{pos}}$ and belonging to boundary bnd_i . Equations stated in (7.16) may be easily implemented in Comsol Multiphysics® by using boundary expressions and boundary constraints.

Fastening Phase: Joining Parts

During joining phase, bnd_s and bnd_d are constrained to move together (coupling constraint condition). Now, the geometry frame has to be updated taking into account the displacement field derived from the closing gap phase ($\text{DGF}_{\text{close}}$).

A Multi Point Constraint (MPC) condition is defined in order to couple bnd_s and bnd_d . Let u_s be the displacement field of bnd_s , available during joining phase. Coupling constraint condition may be written as in relationship (7.17), where μ_{us} is the mean displacement related to points belonging to the spot weld sphere and belonging to the boundary bnd_s ; Ω_s is the volume of spot weld sphere (see Annex F for more mathematical details).

$$\begin{aligned} \mu_{us} &= \frac{1}{\Omega_s} \int_{\Omega_s} \Delta_{\Omega} \cdot u_s \cdot d\Omega_s \\ \text{BC}_d &= \begin{cases} \mu_{us}, & \text{if } \Delta_d \leq R_{\text{weld}} \\ \text{free displacement}, & \text{otherwise} \end{cases} \\ &\quad \text{---} \\ \Delta_{\Omega} &= \begin{cases} 1, & \text{if } \Delta_s \leq R_{\text{weld}} \\ 0, & \text{otherwise} \end{cases} \\ \Omega_s &= \int_{\Omega_s} \Delta_{\Omega} \cdot d\Omega_s \\ \Delta_i &= \|\mathbf{P}_{\text{close}} - \mathbf{P}_{i,0}\|, \forall i = s, d \end{aligned} \tag{7.17}$$

Relationship (7.17) states that any point of bnd_d belonging to the sphere of radius R_{weld} and centre $\mathbf{P}_{d,0}$ (see Figure 7.6), is constrained with a displacement equal to the mean displacement, μ_{us} , related to bnd_s . In equation (7.17) all integrals are surface integrals, where $d\Omega$ is the differential operator.

7.3.3 Releasing Phase

Third FEA run aims to simulate the final elastic spring-back taking into account MPC conditions as defined in the previous Section.

In this phase, the geometry frame is updated with respect to the displacement field derived from fastening phase (DGF_{fas} - note that DGF_{fas} corresponds to DGF_{close}). Final displacements, $(U_{rel}, V_{rel}, W_{rel})$, are calculated by applying to any part of the assembly the initial stress field calculated during previous steps (positioning and fastening phases). Then, it can be written:

$$\sigma_{rel,i} = \sigma_{pos,i} + \sigma_{fast,i}, \forall i = 1, \dots, N_p \tag{7.18}$$

where N_p is the total number of assembled parts, while σ_{pos} and σ_{fast} are the stress fields related to the positioning and fastening phases, respectively, and calculated in the previous runs.

7.4 Implementation

The proposed methodology was implemented into a GUI developed in MATLAB® environment, linked to Comsol Multiphysics® working in background mode (Figure 7.8). FE model can be imported as structured array data directly from Comsol Multiphysics®.

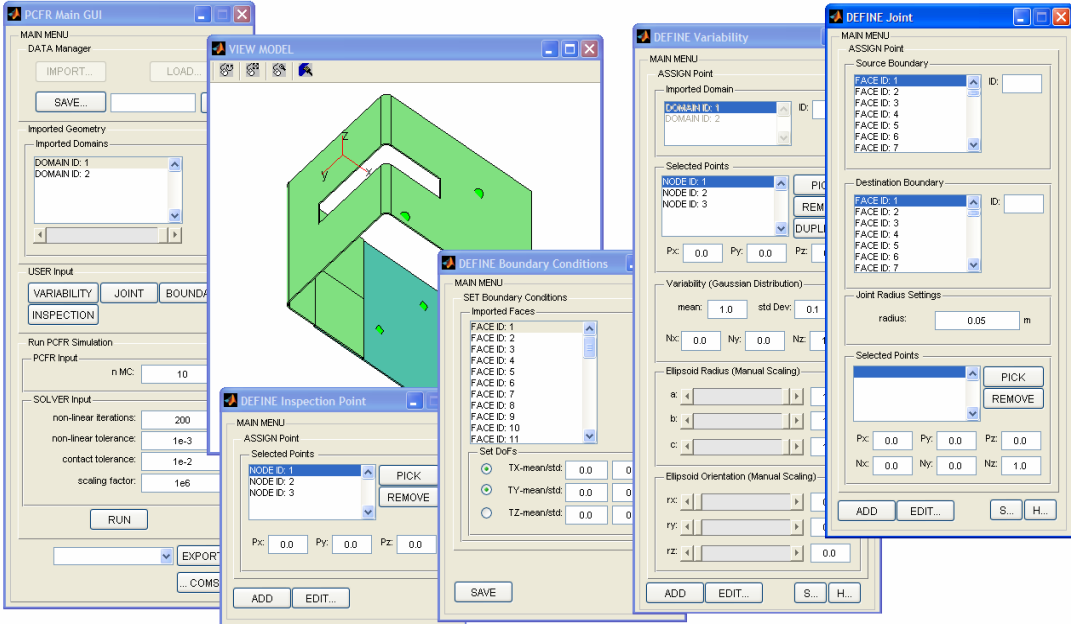


Figure 7.8: Graphical User Interface

Chapter 7. Non-Linear Analysis of Compliant Assemblies

MMP may be controlled from the *VARIABILITY* menu, where user can set control points and their variability, in terms of mean and standard deviation (statistical normal distribution is here assumed). Control points may be easily set by picking any point from the *VIEW MODEL* interface, or giving in input its coordinates in terms (x, y, z). In addition, the sphere or ellipsoid influence hull may be interactively defined by setting its orientation and its size.

JOINT menu allows user to define welding joints, setting radius (R_{weld}), direction (\mathbf{N}_{weld}), nominal position (\mathbf{P}_{weld}), source and destination boundaries.

Boundary constraints with their variability (ϵ_{fixt}) are set from the *BOUNDARY* menu.

Final assembly shape, in terms of mean and standard deviation displacements, can be shown directly in the Comsol Multiphysics® GUI, where user can easily set post-processing options. To understand what happens at point level through the whole assembly simulation, from *INSPECTION* menu, monitoring points can be selected and their displacements, in terms of mean and standard deviations, may be exported into EXCEL® format.

Comsol Multiphysics® solves contact problems by using an augmented lagrangian method. This means that the software solves for equations in a segregated way (augmentation components are introduced for the contact pressure). A good alternative to augmented lagrangian method is the penalty method. However, the latter is more stable but less accurate than the augmented lagrangian method (Wriggers, 2002). To implement penalty method within Comsol Multiphysics®, a contact normal penalty factor expression has to be defined as k/h , where k is a stiffness constant with the same order of the material stiffness, while h is the mean dimension of the mesh.

The proposed methodology requires that different assembly configuration simulations (depending on the number of Monte Carlo simulations) are solved in automatic way (user sets only initial solver parameters). In order to achieve this, a stable non-linear solution is required. Therefore, in this dissertation, the penalty method was adopted, despite it is less accurate than augmented lagrangian method.

7.5 Examples

7.5.1 Free Shape Feature Modeling

This example shows how the proposed procedure works, to statistically generate free shape features. Figure 7.9.a draws the nominal CAD geometry. A profile surface tolerance error of 1.0 mm is assigned. The derived FE model, made of 447 nodes and 432 elements, is drawn into Figure 7.9.b.

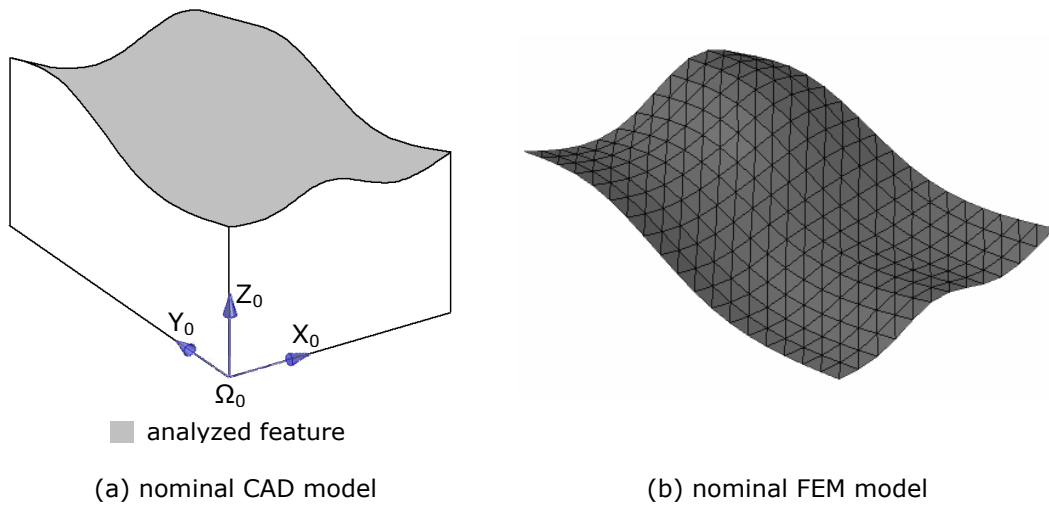


Figure 7.9: free shape feature modeling. Nominal model and influence hulls

In addition, in Figure 7.10 the adopted spherical influence hulls are depicted, related to three control points.

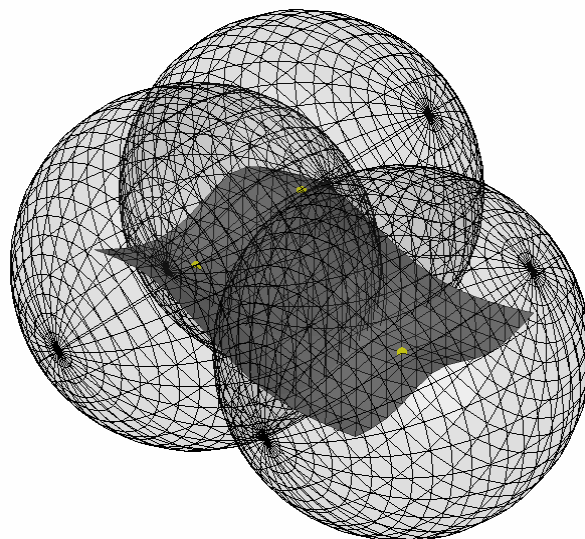


Figure 7.10: free shape feature modeling. Influence hulls

Figure 7.11 shows six sampled features inside the tolerance range (natural tolerance range equal to 3). It should be noted that any point of the sampled features belongs to the 3D tolerance zone (shaded features).

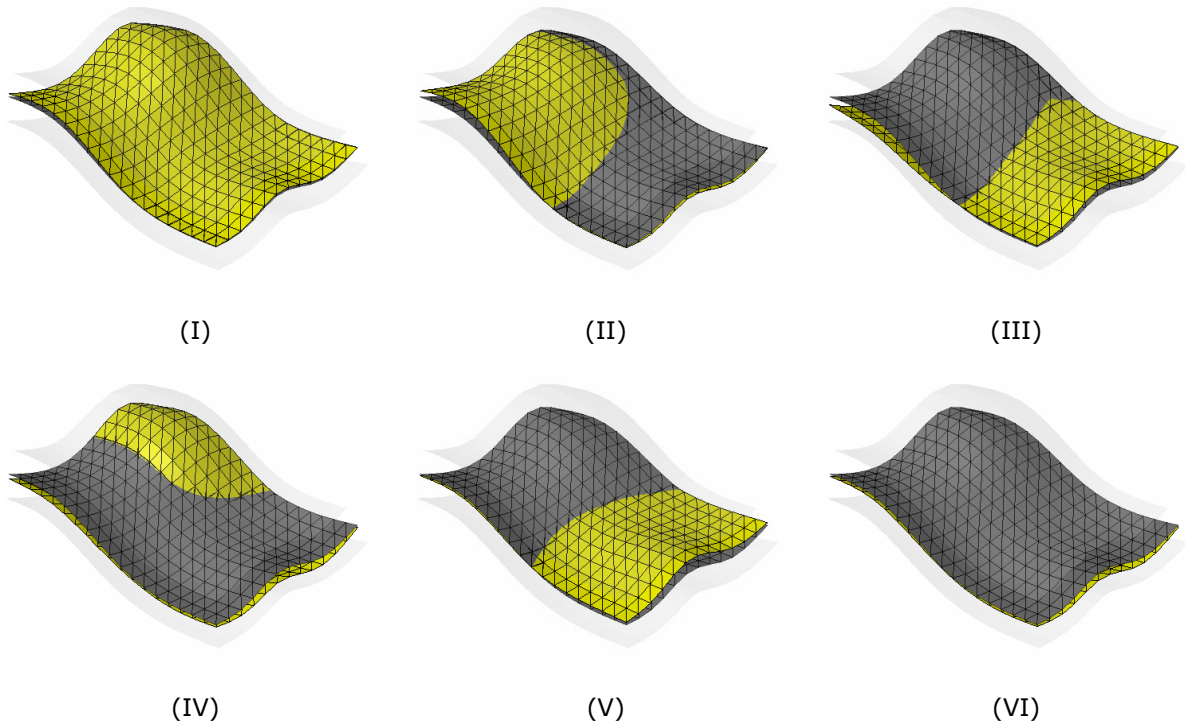


Figure 7.11: free shape feature modeling. Six sampled features.
Variation scale factor = 10

7.5.2 Free Shape Part Modeling

This example shows how MMP may be applied to generate also free shape parts. FE model was made of 3059 nodes with tetrahedral mesh elements⁴.

Statistical input values (gaussian statistical distribution assumed) were assigned at control points, \mathbf{P}_c , along the global \mathbf{Y}_0 axis (Figure 7.12).

Two kinds of MMPs were performed. In the first one (MMP_I), each control point influences the whole geometry (global influence). This means that each influence radius is equal to the maximum distance between any control point and the points of the part. A local influence, instead, was assigned into the second simulation (MMP_{II}). In both cases, 1000 Monte Carlo sampling were generated.

⁴ 3D elements are here required, even for a model usually described with shell elements, as in the actual release, Comsol Multiphysics® does not allow to define contact pairs for shell models. That is why all FEA simulations were performed with tetrahedral elements.

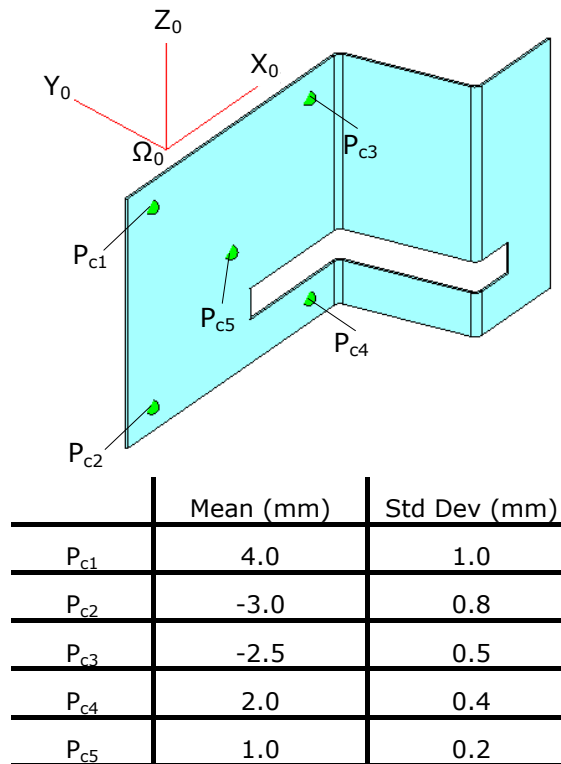


Figure 7.12: free shape part modeling. Input statistical values

Figures 7.13.a and 7.13.b show contour plots related to mean and standard deviations, respectively (in the following the euclidean norm is adopted). Looking at Figure 7.13.a, a twist effect due to deviations of control points P_{c1} and P_{c2} , which are opposite each other, can be pointed out.

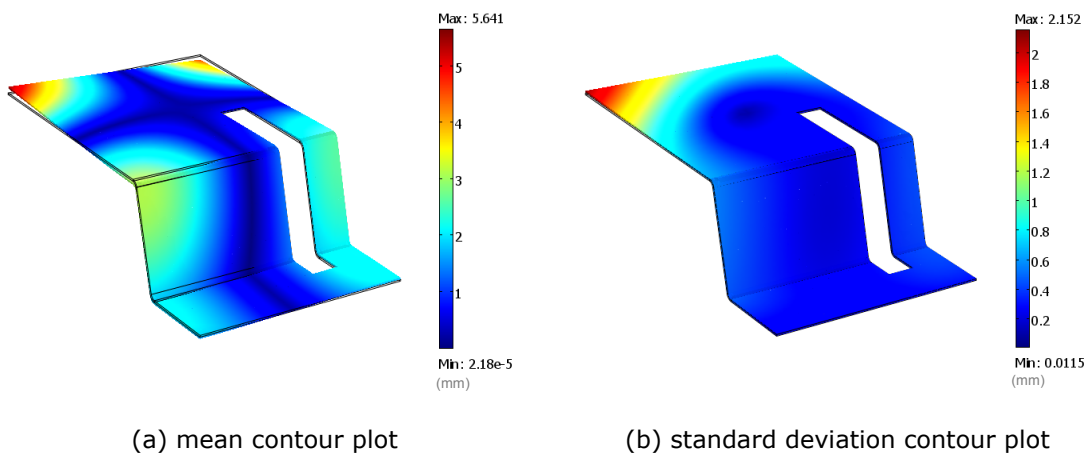


Figure 7.13: free shape part modeling. MMP_I results

Chapter 7. Non-Linear Analysis of Compliant Assemblies

Figure 7.14 depicts influence hulls adopted in the MMP_{II} simulation. Results, in terms of mean and standard deviations are, instead, shown in Figures 15.

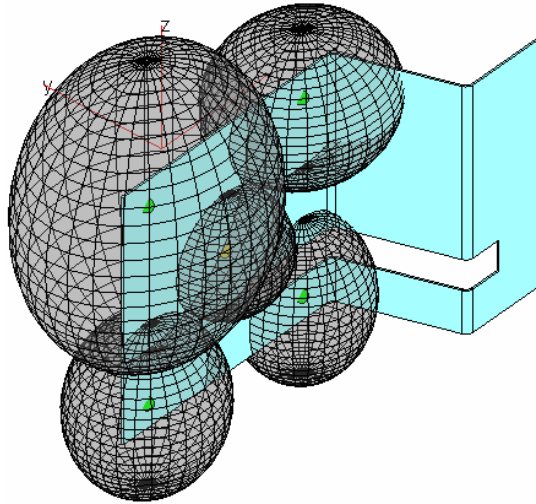
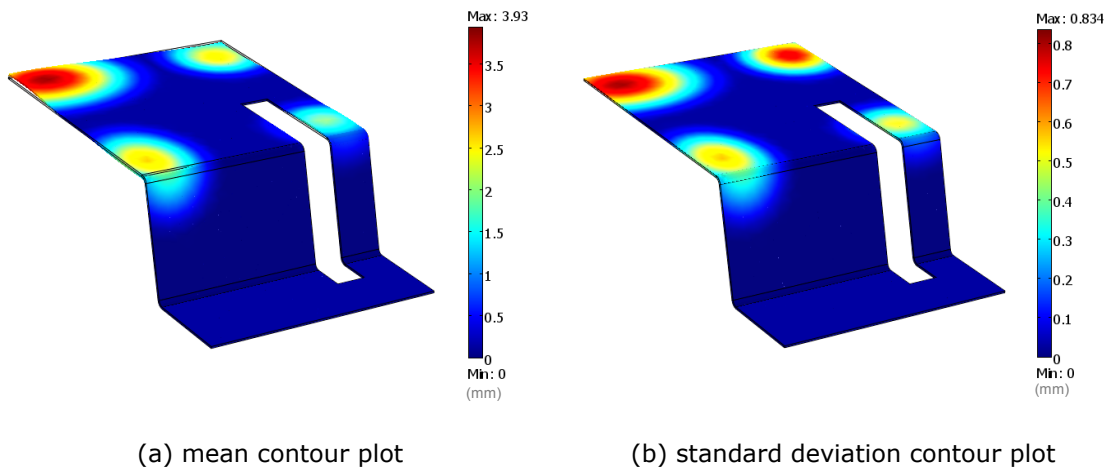


Figure 7.14: free shape part modeling. MMP_{II} - influence hulls assigned at control points for the local influence of the geometry

Looking at Figure 7.15, it can be pointed out a local deformation of the geometry according to the adopted influence hulls.



(a) mean contour plot

(b) standard deviation contour plot

Figure 7.15: free shape part modeling. MMP_{II} results

7.5.3 PCFR Cycle Simulation

The proposed PCFR assembly methodology was applied to a two-part assembly (Young's Modulus $E=210.000 \text{ N/mm}^2$; Poisson's ratio $\nu=0.30$; uniform thickness $T=2.0$

Chapter 7. Non-Linear Analysis of Compliant Assemblies

mm), shown in Figure 7.16. Overall dimensions of the assembly are about 800x600x500 mm.

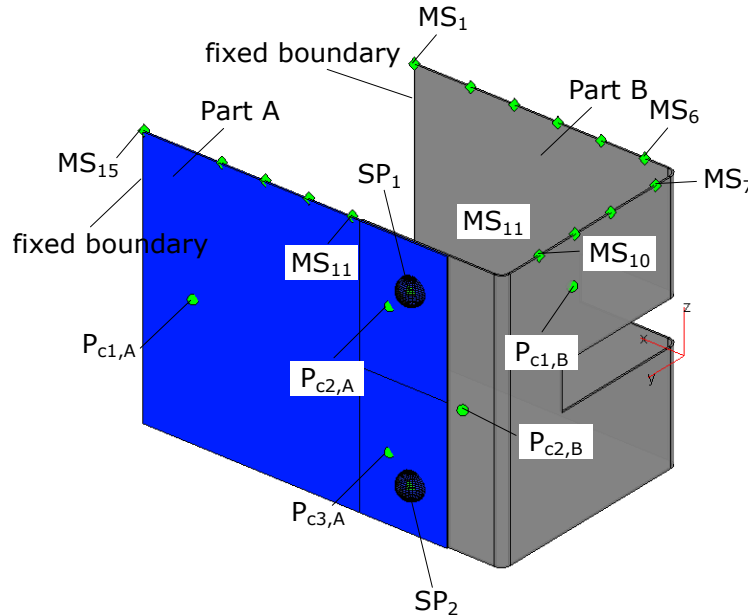


Figure 7.16: PCFR case study. Nminal geometry, control points ($P_{c,i,j}$), spot welds (SP_i) and monitoring points (MS_i)

FE model was made of 2889 nodes with tetrahedral mesh elements. The total number of unknowns in the simulation was 43947. Two spot welds were assigned (SP_1 and SP_2 in Figure 7.16), with 20 mm radius. A fixed fixturing frame was assigned for both parts (“fixed boundary” in Figure 7.16). Contact pairs were defined at boundary interface between “Part A” and “Part B”.

Table 7.1 shows statistical values assigned at control points. $N_{MC} = 500$ Monte Carlo simulations were performed in about two hours on a notebook with Core Duo 1.83 GHz, 2 GB RAM, Win XP 32 bit.

	P_c	Mean (mm)	Std Dev (mm)
Part A	P_{c1} (Y)	0.3	0.1
	P_{c2} (Y)	2.0	0.3
	P_{c3} (Y)	1.0	0.6
Part B	P_{c1} (X)	2.0	0.5
	P_{c2} (X)	-1.0	0.1
	P_{c2} (Z)	-1.0	0.1

Table 7.1: statistical values assigned at control points. Gaussian distribution assumed

Figure 7.17 depicts mean contour plots for each phase of the PCFR cycle. The initial gap existing between “Part A” and “Part B” is then closed in the fastening phase.

To understand what happens when contact pairs are not assigned, a new simula-

Chapter 7. Non-Linear Analysis of Compliant Assemblies

tion was performed (run-time about 20 minutes). Related results are shown into Figures 7.18 with respect to monitoring points (MS in Figure 7.16), in terms of mean and standard deviations.

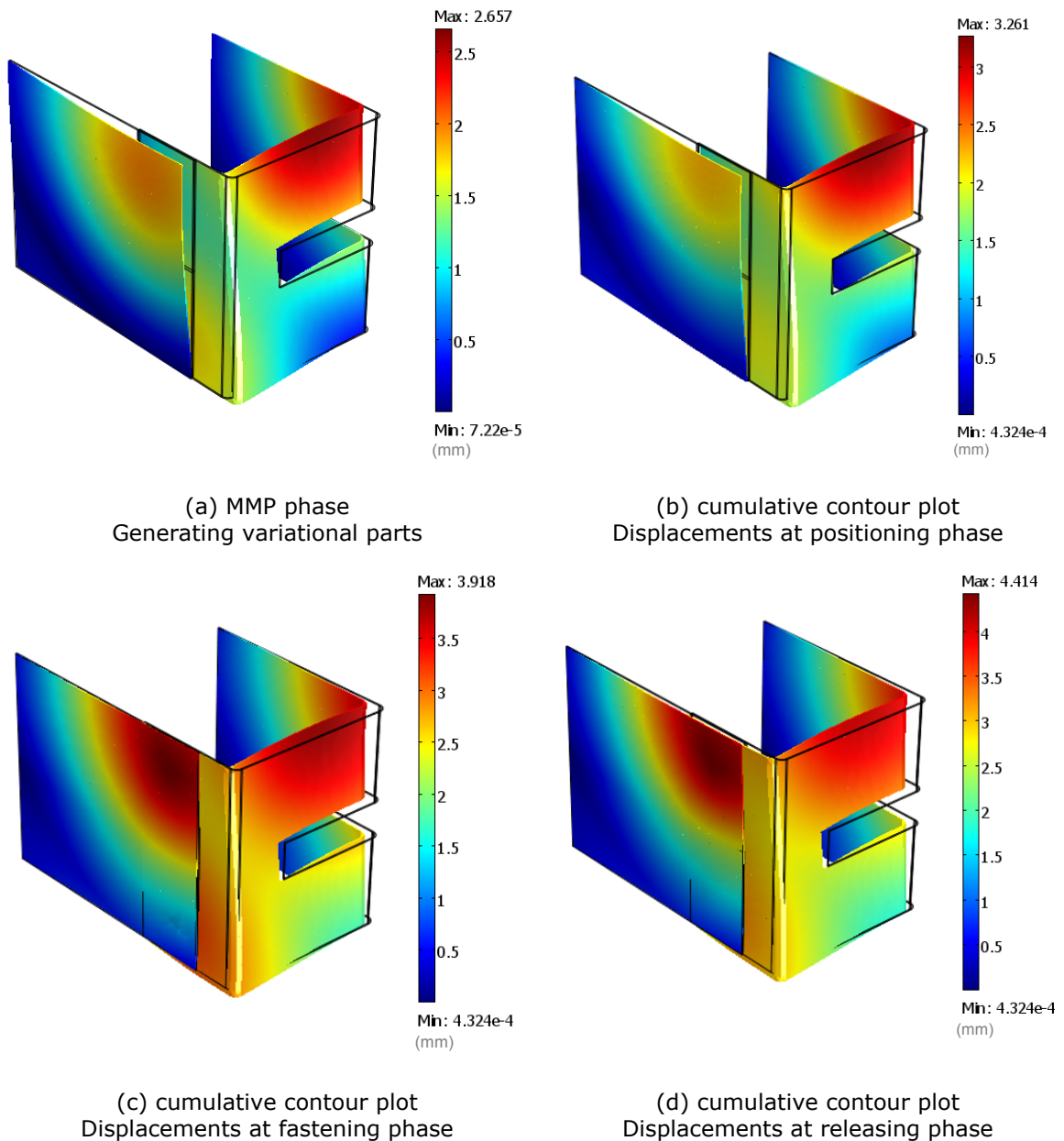
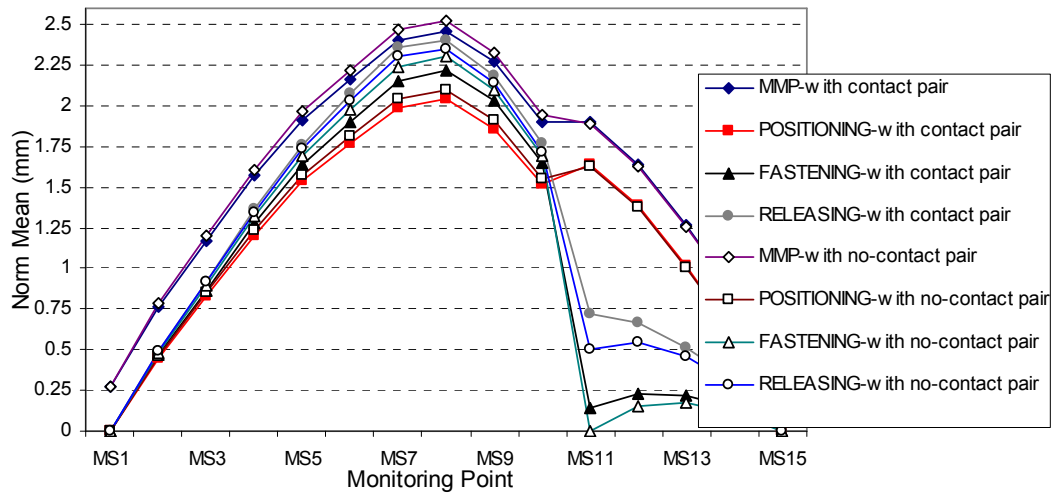


Figure 7.17: PCFR case study. results

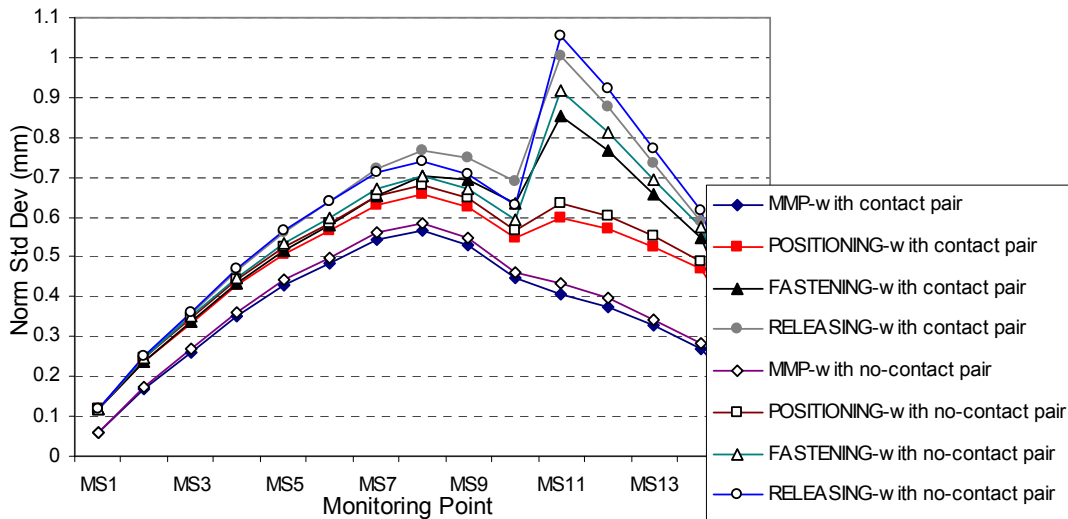
In order to evaluate the error between the two assembly simulations (with and without contact pairs), RSS (Root Sum Square) index was calculated. Results are shown in Table 7.2. RRS index is quite low for “MMP” and “positioning” phases (if the number of Monte Carlo simulations increases it can be expected an RSS index toward to

Chapter 7. Non-Linear Analysis of Compliant Assemblies

zero). Moreover, RSS index becomes higher for “fastening” and “releasing” phases. This is due to the high influence that contact pairs assume both in the closing gap phase and in the final releasing phase.



(a) mean deviation



(b) standard deviation

Figure 7.18: mean and standard deviation results

Chapter 7. Non-Linear Analysis of Compliant Assemblies

Phase	RRS (Mean)	RSS (Std Dev)
MMP	0.1460	0.0619
POSITIONING	0.1232	0.0819
FASTENING	0.2445	0.1091
RELEASING	0.2844	0.1132

Table 7.2: contact pair vs no-contact pair

7.6 Summary

In this Chapter, a methodology to perform statistical variational analysis of compliant parts, mainly based on a non-linear FEA approach, was proposed. Statistical evaluation is performed via Monte Carlo simulations to generate variational parts to be assembled and then released.

The variational shape of parts being assembled was generated by using a morphing based-mesh approach, borrowed by computer graphics field. User can manage and control the input shape by handling a small set of control points, assigning their position and the influence radius.

Variational parts are, then, assembled following the classical PCFR cycle. The geometry domain is updated for each step of the PCFR cycle. The attention focused on spot weld joints. To improve the solution convergence, an efficient method, aiming to simulate fastening and releasing phases, was proposed. To do it, scalar and integral expressions, defined at boundary level, were defined.

The proposed methodology was implemented into a friendly MatLAB®'s GUI, linked to Comsol Multiphysics® run into background mode. The GUI drives user loading the initial FE model, assigning input parameters, running Monte Carlo simulations and, finally, exporting result data.

The proposed morphing based-mesh procedure was applied to a single sheet-metal part, highlighting how, by managing control points and their influence hulls, different shape can be generated.

Finally, a two-part assembly was simulated following the PCFR cycle. Results pointed out that contact pairs highly influence final results. Contacts allow to reach more realistic results but, on the other hand, a very time consuming simulation is required. Selecting the right balance between simulation time and goodness of numerical results is not a trivial task. Solving this issue requires more investigation. More attention should be addressed also to different fastening joints such as rivets and bolts, and to multi-station assembly processes.

Chapter 8.

CONCLUSIONS AND FUTURE WORKS

In this Chapter the main contributions and the overall findings derived from the present research are outlined. Furthermore, future perspectives of research in the tolerance field are discussed.

8.1 Conclusions

The research in tolerance analysis was motivated by the actual need to analyze how variations propagate in real manufacturing systems and what is their impact on delivering specific key characteristics. Often, in real industrial applications, failures of assembly processes are related to dimensional or geometrical variations caused by weakness of technology or knowledge for accurate prediction of process variation during the early design stage.

Since its origin, tolerance analysis has covered several key topics, focusing on specific methodologies and procedures to predict the amount of variation due to part or process variability. Literature on tolerance analysis provides many and many methods about specification, modeling and analysis of tolerances, and they mainly treat two categories of assemblies: ideal-rigid assemblies and compliant assemblies. In the first category parts being assembled are assumed rigid, so no variation occurs for part deformation; in the second case, instead, parts are assumed flexible and the additional effect due to part deformation during the assembly process is considered in the simulation.

This dissertation covered the following two main topics: tolerance analysis of rigid and compliant assemblies. The main goal was to develop automatic algorithms

Chapter 8. Conclusions and Future Works

able to numerically simulate real assembly processes involving both rigid and compliant parts. Some specific mathematical tools, such as Graph Theory and Screw Theory, were widely adopted in the present dissertation to model functional key characteristics and kinematic constraints among parts. In particular, Graph Theory was used since it provides a very efficient way to handle mechanical assembly from a functional point of view. Thus, automatic procedures to detect the global consistency can be easily implemented starting from the edge or incidence matrix. Screw Theory is a well-known methodology in Robotics and mechanical fields to analyze the motion and the constraint state for a given set of kinematic joints. Screw Theory was adopted in this dissertation to automatically calculate the degrees of freedom of parts being assembled during the assembly process simulation.

In summary, the present dissertation gave a contribution to the following features:

Variational Feature Modeling

The objective of variational feature modeling is to provide a mathematical representation of tolerance specification according to International Standards. Features are randomly generated inside every specific tolerance zone.

When free shape errors may be neglected, variational features were modeled through rigid small rotational and translational parameters. To assure that any point, belonging to the analyzed feature, is inside the tolerance zone, parameters constraints were calculated. This dissertation gave a solution for two particular features: planar and cylindrical. In particular, planar feature was modeled through a vector and a point, the length and the width of the same feature. An axis, instead, was associated to cylindrical features. Once the statistical law is assigned, variational features are generated by using a Monte Carlo approach: a reject method was applied to sample only those features falling inside the specified tolerance zone. This dissertation did not focus on the efficiency of the reject method, which may be expensive when many Monte Carlo simulations are required.

Variational features were then used to simulate assembly processes involving rigid part assembly, as reported in Chapter 5.

This dissertation provided also an innovative methodology to model free shape errors, accounting a small set of control points, defined on the nominal geometry. Starting from a FE model, created by using the nominal CAD model, mesh nodes are moved. To assure that each node is inside the specified tolerance zone, a morphing mesh approach, combined with a constrained deformation, was adopted. In particular, the designer defines a set of control points, by giving a desired displacement and the relative

Chapter 8. Conclusions and Future Works

radius of influence for each point. Control points directly influence the final shape of the morphed feature, and this shape can be fine-tuned by adjusting the influence radius or the position of each control point. The reject method was then adopted to statistically sample free shape feature inside the specified tolerance zone.

Case studies in Chapter 7 showed that the morphing mesh approach can be successfully used to simulate free shape features: only few control points, with their radius of influence, and the related weighted function are given in input. Results pointed out that the position and radius of influence can be managed in order to generate different free shape features. Free shape features were then adopted to simulate assembly processes involving compliant parts.

Tolerance Analysis of Rigid Part Assembly

The objective of this research was to develop a general methodology able to do tolerance analysis of rigid part assemblies. The proposed methodology, called SVA-TOL (Statistical Variation Analysis for Tolerancing), aims to simulate 3D assembly joints acting between variational features. The focusing was on mating joints between planar and cylindrical features. Two solvers were proposed: the sequential solver and the least squares solver.

The sequential solver solves one-by-one assembly joints. All this allows to simulate different assembly sequences. Homogeneous transformations, in the small displacement hypotheses, were adopted to model assembly constraints. The aim was to move an object part in order to fit target part constraints: small rotational and translational motion parameters were then calculated. With respect to mating joints among planar features, an optimization algorithm was adopted: target and object planes are best aligned when the angle between the related unit normal vectors becomes minimum. In addition, a linear contact procedure was implemented in order to prevent plane-to-plane intersection. Fit joints among cylindrical features were solved by minimizing the relative angle between the two cylindrical axes. When a new constraint joint is added, the list of DoFs is automatically updated following a Screw Theory approach. DoFs are calculated with respect to a local coordinate frame attached to the previous joint. Therefore, this approach allows to keep the effect of all previous joints, into a sequential fashion.

The least squares solver, instead, solves all assembly constraints simultaneously. Feature-to-feature intersections are not avoided in this approach. However, this solver is able to simulate also over-constraint configurations, in which the number of constraint equations is greater than the number of unknown assembly parameters.

Chapter 8. Conclusions and Future Works

Case studied illustrated in Chapter 5 highlighted the applicability of the proposed approach to simulate both sequential assembly processes (the classical 3-2-1 assembly scheme) and simultaneously fitting conditions. Moreover, in order get more realistic results, flexibility of part should be considered. All this is especially true when over-constraint configurations are considered. Chapter 6 focused just on this last topic.

Tolerance Analysis of Compliant Part Assembly

When parts have a wide flexibility the main hypothesis of ideal-rigid assembly is inadequate. In some specific industrial contests, involving highly deformable parts, dedicated methodologies are then needed to predict the amount of variation on the final assembly product.

This dissertation focused on the modeling and simulation of sheet-metal assembly processes, performed both in single- and multi-station configurations. The methodology was based on linear assumptions. All assembly phases were assumed linear: small displacement hypotheses, linear materials, friction among parts neglected. Local plastic deformations, which may happen locally at fastening area, were not taken into account.

The assembly process was based on the Place, Clamp, Fasten and Release (PCFR) cycle. Four sets of key points were introduced: fixturing, fastening, contact and inspection points. Fixturing points can be used to model fixturing frames to locate and hold part during the assembly process. Real fastening operations, such as welding, bolting, riveting and gluing, are modeled through fastening points. In order to avoid part-to-part penetration “linear contact points” are also available. The final elastic spring-back of the assembly is then reached on a specific fixturing frame, the so-called inspection frame, used in real assembly process to do measurement and inspection.

The general methodology, able to do *linear* tolerance analysis of single- and multi-station assembly and named SVA-FEA (Statistical Variation Analysis & Finite Element Analysis), was based on a linear relationship linking input deviations at part (fastening variability) or process (fixturing variability) level to assembly deviations. The *Global Sensitivity Matrix* was then introduced. Once statistical data are available, statistical output data can be easily obtained by managing the global sensitivity matrix. In this contest, all input sources of variation were assumed as random independent variables. In this way no statistical covariance effect was considered.

In order to numerically evaluate the global sensitivity matrix, two linear static FEA runs were required for each sub-station assembly. The first run calculates the reaction forces at fixturing and fastening points; the second one evaluates the spring-back effect after releasing fastening tools and re-positioning the assembly onto a new fixture frame.

Chapter 8. Conclusions and Future Works

The SVA-FEA methodology was implemented into a MatLAB®'s GUI, linked into background mode to MSC NASTRAN®, which solves FE analyses. GUI drives the user to import the input FE model into MSC NASTRAN® format (bdf format). Fixturing, fastening and contact points can be easily set by using the friendly GUI. A graph representation is also available to define the whole assembly process (assembly tree) and to set, for each sub-station, fixturing frames and fastening tools. Finally, SVA-FEA interface allows post-processing results in terms of mean and standard deviations.

In the SVA-FEA method all sources of variation are assumed independent each other. This means that each deviation may influence just itself. No coupling effect may be taken into account. Therefore, in SVA-FEA approach the covariance effect among input statistical data is neglected. To overcome this limitation, in Chapter 7 free shape errors were modeled by means of the morphing mesh approach. In this way the geometric covariance among neighborhood points was introduced. Starting from this, a non-linear assembly process methodology was then presented. Statistical input geometries were generated through a Monte Carlo simulation. Each simulation generates variational parts, which are then assembled following the PCFR cycle. During each step of the assembly process, the geometry domain is updated, accordingly to the local deformation of parts.

This new methodology allows to take into account also non-linear effect such as surface-to-surface contact pairs or plastic deformation phenomena. However, a more expensive simulation is needed with respect to SVA-FEA linear approach.

In order to easily perform simulation analyses, a MatLAB®'s GUI, similar to SVA-FEA but with less interactivity, was developed. The interface allows to generate free shape geometry and to simulate the non-linear assembly process. FE analyses are performed in Comsol Multiphysics®, run into background mode.

8.2 Future Works

This dissertation described a general framework to do tolerance analysis of rigid- and compliant-part assembly. However, starting from the current research, many topics need to be improved and investigated. They may be summarized as follows:

Variational Feature Modeling

Tolerance analysis process begins with the definition and the assignment of variational features. This dissertation focuses only on planar and cylindrical features. Much more features, as proposed into the TTRS Theory, should be integrated into the

Chapter 8. Conclusions and Future Works

general framework proposed in this dissertation.

With respect to rigid assembly, variational features were modeled with small rigid translational and rotational displacements: free shape errors were there neglected. The morphing mesh procedure, adopted to generate free shape features and already used to simulate compliant assembly, may be integrated into the general framework for rigid part assembly.

Methodology Integration

In the last decade, rigid and compliant assembly methodologies have been developed in parallel. However, real assembly systems involve both rigid and compliant parts. This need is not covered by the actual technical literature. More investigations are then needed to develop an integrated methodology able to take into account both phenomena simultaneously.

Tolerance Synthesis

One actual need in real industrial contests is the optimum allocation, typically based on cost criteria, of tolerance specifications. In the field of rigid-part assembly many and many methods have been suggested. Linear and non-linear programming techniques are successfully used to calculate the amount of variation on each source of variation, for a given key characteristic assignment. In the field of compliant-part assembly this task may be much more complicated, especially when non-linear effects are considered. The technical literature does not provide a valid solution to this urgent need. More investigations are so welcome. By using the linear approach proposed in this dissertation, based on the global sensitivity matrix, an optimization framework could be developed.

Linear vs Non-Linear Simulation

As widely pointed out in this dissertation, in order to reach much more accurate results, a non-linear approach is needed when compliant parts are considered. However, whether on one hand closer results to real assembly processes can be reached when non-linear contact pairs or plastic phenomena are considered, on the other hand, a very time consuming simulation is required. In addition, thousand and thousand Monte Carlo simulations are needed to get statistical output. The technical literature has been addressing the attention to non-linear simulation of compliant-part assembly. However, no

Chapter 8. Conclusions and Future Works

comparative study with the linear approach has been proposed. Therefore, a comprehensive comparative analysis is necessary in order to explore the weaknesses of the linear and non-linear approaches and to understand the limit of their applicability.

Annex A.

4X4 TRANSFORMATION MATRIX

The aim of this Annex is to provide a general point of view on 4x4 transformation matrix, showing how to easily calculate it, starting from geometrical data available from any CAD modeler.

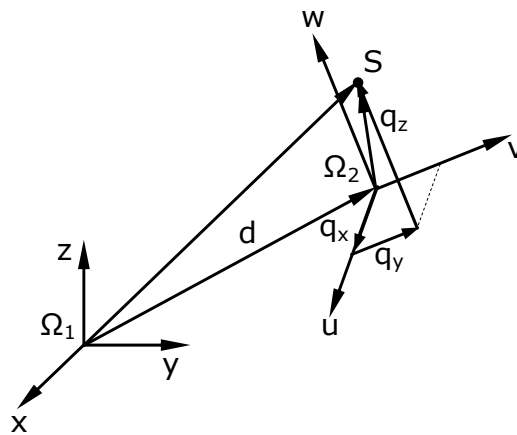


Figure A.1: frame definition

A.1 Transformation Matrix

Looking at Figure A.1, frame Ω_1 , (x, y, z) , and frame Ω_2 , (u, v, w) are assigned. Let \mathbf{d} and $(\mathbf{u}, \mathbf{v}, \mathbf{w})$ be the position and the unit vectors of the frame Ω_2 with respect the frame Ω_1 , respectively. This geometrical information may be derived (by scripting or manually) from any CAD modeler.

Annex A. 4x4 Transformation Matrix

Let \mathbf{P} be the vector defining the point \mathbf{S} into the frame Ω_1 :

$$\mathbf{P} = \begin{Bmatrix} p_x \\ p_y \\ p_z \end{Bmatrix} \quad (\text{A.1})$$

while \mathbf{Q} is the same point defined into frame Ω_2 .

$$\mathbf{Q} = \begin{Bmatrix} q_x \\ q_y \\ q_z \end{Bmatrix} \quad (\text{A.2})$$

Based on geometrical consideration, one can write the following vectorial equation:

$$\mathbf{P} = \mathbf{d} + \mathbf{u} \cdot q_x + \mathbf{v} \cdot q_y + \mathbf{w} \cdot q_z \quad (\text{A.3})$$

which may be arranged as into equation (A.4):

$$\begin{aligned} \mathbf{P} &= \mathbf{d} + \mathbf{R} \cdot \mathbf{Q} \\ \mathbf{R} &= [\mathbf{u}, \mathbf{v}, \mathbf{w}] \end{aligned} \quad (\text{A.4})$$

where \mathbf{R} is the 3x3 rotation matrix defining the orientation of Ω_2 with respect to Ω_1 one. As it is, each column of \mathbf{R} has the cosine directions of the axes of the frame Ω_2 with respect to Ω_1 one (see also (Asada, 1986)).

Therefore, equation (A.4) stated that to transform a point \mathbf{Q} , defined into frame Ω_2 , to Ω_1 , one must firstly apply a rotation, \mathbf{R} , and then set a translation, \mathbf{d} . Equation (A.4) may be then re-written by using the 4x4 homogeneous notation:

$$\begin{Bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{Bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & d_x \\ r_{21} & r_{22} & r_{23} & d_y \\ r_{31} & r_{32} & r_{33} & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{Bmatrix} q_x \\ q_y \\ q_z \\ 1 \end{Bmatrix} \quad (\text{A.5})$$

or:

$$\mathbf{P} = \begin{bmatrix} \mathbf{R} & \mathbf{d} \\ \mathbf{0} & 1 \end{bmatrix} \cdot \mathbf{Q} \rightarrow \mathbf{P} = \mathbf{T}_{1,2} \cdot \mathbf{Q} \quad (\text{A.6})$$

$\mathbf{T}_{1,2}$ is finally the 4x4 transformation matrix which expresses frame Ω_2 within Ω_1 one. May be of interest evaluating the inverse transformation of $\mathbf{T}_{1,2}$. To do it, looking at equation (A.4), one can write:

Annex A. 4x4 Transformation Matrix

$$\begin{aligned} \mathbf{R}^T \cdot \mathbf{P} &= \mathbf{R}^T \cdot \mathbf{d} + \mathbf{R}^T \cdot \mathbf{R} \cdot \mathbf{Q} \rightarrow \dots \\ \dots \rightarrow \mathbf{Q} &= -\mathbf{R}^T \cdot \mathbf{d} + \mathbf{R}^T \cdot \mathbf{P} \end{aligned} \quad (\text{A.7})$$

thus, by using 4x4 notation:

$$\mathbf{Q} = \begin{bmatrix} \mathbf{R}^T & -\mathbf{R}^T \cdot \mathbf{d} \\ \mathbf{0} & 1 \end{bmatrix} \cdot \mathbf{P} \rightarrow \mathbf{Q} = \mathbf{T}_{2,1} \cdot \mathbf{P} \quad (\text{A.8})$$

$\mathbf{T}_{2,1}$ is the inverse transformation, representing the frame Ω_1 into Ω_2 one.

When the frame Ω_2 has the same origin of Ω_1 one, then, one can write the basic rotation matrices, with respect to the reference axes, \mathbf{x} , \mathbf{y} , \mathbf{z} , as into equation (A.9).

$$\begin{aligned} \mathbf{R}_x(\alpha) &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) & 0 \\ 0 & \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ \mathbf{R}_y(\beta) &= \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ \mathbf{R}_z(\gamma) &= \begin{bmatrix} \cos(\gamma) & -\sin(\gamma) & 0 & 0 \\ \sin(\gamma) & \cos(\gamma) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \quad (\text{A.9})$$

A.2 Composition of Transformations

The main use of 4x4 transformation matrix is to chain together a series of frames (typically, into Robotics applications, one is aimed to describe the position of the end-effector of the robot with respect to a global reference frame, taking into account all joint transformations).

Looking at Figure A.2, the scope is evaluating the transformation $\mathbf{T}_{1,3}$. Thinking the point \mathbf{S} as initially defined into the frame Ω_3 , one can write:

$$\begin{aligned} \mathbf{S}_2 &= \mathbf{T}_{2,3} \cdot \mathbf{S}_3 \\ \mathbf{S}_1 &= \mathbf{T}_{1,2} \cdot \mathbf{S}_2 = \mathbf{T}_{1,2} \cdot \mathbf{T}_{2,3} \cdot \mathbf{S}_3 \end{aligned} \quad (\text{A.10})$$

Annex A. 4x4 Transformation Matrix

Thus:

$$\mathbf{T}_{1,3} = \mathbf{T}_{1,2} \cdot \mathbf{T}_{2,3} \quad (\text{A.11})$$

Equation (A.11) may be extended to a chain of n frame as into equation (A.12).

$$\mathbf{T}_{1,n} = \mathbf{T}_{1,2} \cdot \mathbf{T}_{2,3} \cdot \dots \cdot \mathbf{T}_{i-1,i} \cdot \dots \cdot \mathbf{T}_{n-1,n} \quad (\text{A.12})$$

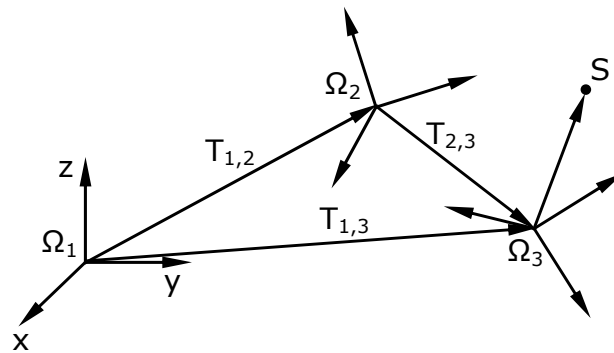


Figure A.2: composition of transformations

Looking at equation (A.12), the order of transformations is preserved into the subscripting. To easily understand this result, one should read the equation from the right side toward the left one. For example, into equation (A.11) there is a first transformation from frame Ω_3 to frame Ω_2 , and then a transformation from frame Ω_2 to frame Ω_1 .

Annex B.

LINEAR ALGEBRA OVERVIEW

This Annex provides an overview on the key features of linear algebra, used overall in the dissertation. First of all, system of linear equations are introduced, highlighting some critical aspects to be accounted when solving such as problems. Then, the eigenvalue and the singular value decomposition are drawn.

Nowadays, many valid source codes and commercial tools to perform linear numerical analyses are available. In this Annex, the built-in MatLAB® functions are adopted as reference.

B.1 System of Linear Equations

System of Linear Equations (SLE) are the milestone for any numerical analysis, from the mechanical field to the chemical or fluid dynamic applications.

The general form of a SLE is:

$$\begin{cases} a_{11} \cdot x_1 + a_{12} \cdot x_2 + \dots + a_{1i} \cdot x_i + \dots + a_{1n} \cdot x_n = b_1 \\ \dots \\ a_{j1} \cdot x_1 + a_{j2} \cdot x_2 + \dots + a_{ji} \cdot x_i + \dots + a_{jn} \cdot x_n = b_j \\ \dots \\ a_{m1} \cdot x_1 + a_{m2} \cdot x_2 + \dots + a_{mi} \cdot x_i + \dots + a_{mn} \cdot x_n = b_m \end{cases} \quad (\text{B.1})$$

or in a more compact form:

$$\mathbf{A} \cdot \mathbf{x} = \mathbf{b} \quad (\text{B.2})$$

Annex B. Linear Algebra Overview

where \mathbf{A} is the $m \times n$ matrix of coefficient, \mathbf{b} is the vector of constants, while \mathbf{x} is the vector of unknowns.

The numerical solution to equation (B.2) depends on how the \mathbf{A} matrix appears. The following cases are possible:

- \mathbf{b} is not null ($\mathbf{A} \cdot \mathbf{x} = \mathbf{b}$)
 - \mathbf{A} is a $n \times n$ square matrix
 - \mathbf{A} is a $m \times n$ rectangular matrix
- \mathbf{b} is null ($\mathbf{A} \cdot \mathbf{x} = \mathbf{0}$)
 - \mathbf{A} is a $n \times n$ square matrix
 - \mathbf{A} is a $m \times n$ rectangular matrix

Next, each case will be analyzed in detail.

\mathbf{b} is not null - \mathbf{A} is a $n \times n$ square matrix

If \mathbf{A} is a $n \times n$ square matrix and its rank equals n , then, the general equation (B.2) has a unique solution, given by:

$$\mathbf{x} = \mathbf{A}^{-1} \cdot \mathbf{b} \quad (\text{B.3})$$

From a numerical point of view, calculating the inverse of a matrix is a very huge task. In literature many methods are available. The general idea is to find out an equivalent SLE and then solve that one. The LU (Lower-Upper) decomposition method is a well-known numerical method which decomposes the matrix \mathbf{A} into a product of two matrices, \mathbf{L} and \mathbf{U} . The first one is an n by n lower triangular matrix, while \mathbf{U} is an n by n upper triangular matrix. So, one can write:

$$\mathbf{L}_{n,n} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ l_{21} & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ l_{n1} & l_{n2} & \dots & 1 \end{bmatrix}$$

$$\mathbf{U}_{n,n} = \begin{bmatrix} u_{11} & u_{12} & \dots & u_{1n} \\ 0 & u_{22} & \dots & u_{2n} \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & u_{nn} \end{bmatrix} \quad (\text{B.4})$$

$$\mathbf{A} = \mathbf{L} \cdot \mathbf{U}$$

By using the decomposition into (B.4), the solution to the SLA in (B.2) is achieved into two consecutive steps. Assuming $\mathbf{U} \cdot \mathbf{x} = \mathbf{c}$, one can write:

Annex B. Linear Algebra Overview

$$\begin{cases} \mathbf{L} \cdot \mathbf{c} = \mathbf{b} \\ \mathbf{U} \cdot \mathbf{x} = \mathbf{c} \end{cases} \quad (\text{B.5})$$

Once \mathbf{c} is derived from the first SLE by forward elimination, then \mathbf{x} (which is the solution vector of the original SLE) is calculated from the second SLE by back substitution.

The MatLAB® routine to perform the LU decomposition is the “lu” function.

b is not null - A is a mxn rectangular matrix

When the \mathbf{A} matrix is rectangular (its range may be different from m), then, from a mathematical point of view, the SLE has no solution. This means that the number of unknowns is different from the number of equations. A numerical solution may be achieved by imposing that the error ε , stated into equation (B.6), is minimum (least-squares approximation).

$$\min(\varepsilon) \rightarrow \min(\|\mathbf{b} - \mathbf{A} \cdot \mathbf{x}\|^2) \quad (\text{B.6})$$

It could be shown that the condition in (B.6) is satisfied when:

$$\mathbf{A}^T \cdot \mathbf{A} \cdot \mathbf{x} = \mathbf{A}^T \cdot \mathbf{b} \rightarrow \mathbf{x} = (\mathbf{A}^T \cdot \mathbf{A})^{-1} \cdot \mathbf{A}^T \mathbf{b} = \mathbf{A}_{LS} \cdot \mathbf{b} \quad (\text{B.7})$$

The $n \times m$ matrix, \mathbf{A}_{LS} , is called pseudo-inverse of \mathbf{A} . As for the inverse matrix, stated into equation (B.2), also the pseudo-inverse is not usually calculated through its definition. An efficient way is based on the Singular Value Decomposition (SVD) technique (see Section B.3 for details).

The MatLAB® routine to numerically evaluate the pseudo-inverse \mathbf{A}_{LS} is “pinv”. Moreover, an other efficient way to perform this calculation is the backslash “/” operator.

b is null - A is a nxn square matrix

In this case, the SLE reduces to:

$$\mathbf{A} \cdot \mathbf{x} = \mathbf{0} \quad (\text{B.8})$$

If the rank of \mathbf{A} matrix equals n , then the SLE in (B.8) has only the trivial solution $\mathbf{x}=\mathbf{0}$.

b is null - A is a mxn rectangular matrix

When \mathbf{A} is a rectangular matrix, the solution to the homogeneous equation in (B.8) is not unique, but it is described by a subspace of \mathbb{R}^n , called *nullspace* of \mathbf{A} , $N(\mathbf{A})$. The *nullspace* contains any solution \mathbf{x} of the SLE in (B.8). The size of the *nullspace* is equal to the number of columns of \mathbf{A} (the number of unknowns) minus its rank¹. Then, for example, if the rank of \mathbf{A} is r , then the solution space is described by an $n \times (n-r)$ matrix. It should be noted that columns of $N(\mathbf{A})$ are orthogonal among them (for instance, they define the subspace $N(\mathbf{A})$).

Obviously, the *nullspace* is empty only when the rank of \mathbf{A} equals the number of unknowns. In that case, the SLE has only the trivial solution $\mathbf{x}=\mathbf{0}$.

From a numerical point of view the *nullspace* is solved for by elimination on \mathbf{A} matrix. Typically, numerical algorithms try to reduce \mathbf{A} to an upper triangular matrix. The MatLAB® routine to evaluate the *nullspace* of an homogeneous SLE is the “null” function.

B.2 Eigenvalues and Eigenvectors

Let \mathbf{E} be an $n \times n$ square matrix. In addition, let λ and \mathbf{v} be a scalar value and a vector of \mathbb{R}^n . λ is an eigenvalue of \mathbf{E} if and only if:

$$\mathbf{E} \cdot \mathbf{v} = \lambda \cdot \mathbf{v} \tag{B.9}$$

In addition, if $\mathbf{v} \neq \mathbf{0}$ then it is the eigenvector associated to the eigenvalue λ .

It may be shown that eigenvectors corresponding to different eigenvalues are linearly independent. Let \mathbf{V} and $\mathbf{\Omega}$ be the matrix of linearly independent eigenvectors and the diagonal matrix of eigenvalues, respectively, defined into equation (B.10):

$$\mathbf{V} = \begin{bmatrix} v_{11} & \dots & v_{1n} \\ \dots & \dots & \dots \\ v_{n1} & \dots & v_{nn} \end{bmatrix} = [\mathbf{v}_1 \quad \dots \quad \mathbf{v}_n] \tag{B.10}$$

$$\mathbf{\Omega} = \begin{bmatrix} \lambda_1 & \dots & 0 \\ \dots & \dots & \dots \\ 0 & \dots & \lambda_n \end{bmatrix}$$

equation (B.9) may be generalized as in (B.11).

$$\mathbf{E} \cdot \mathbf{V} = \mathbf{V} \cdot \mathbf{\Omega} \rightarrow \mathbf{E} = \mathbf{V} \cdot \mathbf{\Omega} \cdot \mathbf{V}^T \tag{B.11}$$

¹ If \mathbf{A} is a $m \times n$ matrix, the rank of \mathbf{A} is less or equal to $\min(m, n)$.

Annex B. Linear Algebra Overview

Equation (B.11) is the general formulation of the eigenvector decomposition. It should be noted that the \mathbf{V} matrix is orthogonal.

The MatLAB® routine to accomplish the eigenvector decomposition of a square matrix is the “eig” function.

B.3 Singular Value Decomposition

SVD aims to decompose a rectangular $m \times n$ matrix, \mathbf{S} , into a product of three matrices, two orthogonal and one diagonal. The general form of the SVD may be written as into equation (B.12):

$$\mathbf{S} = \mathbf{P} \cdot \mathbf{\Delta} \cdot \mathbf{Q}^T \quad (\text{B.12})$$

where \mathbf{P} and \mathbf{Q} are the orthogonal matrices of singular vectors, while $\mathbf{\Delta}$ is the diagonal matrix of singular values.

Assuming \mathbf{E} a square matrix, defined as $\mathbf{E} = \mathbf{S}^T \cdot \mathbf{S}$ (such as matrix is usually called semi-definite matrix), then, from equation (B.12), it has:

$$\mathbf{S}^T \cdot \mathbf{S} = \mathbf{Q} \cdot \mathbf{\Delta}^T \cdot \mathbf{P} \cdot \mathbf{P} \cdot \mathbf{\Delta} \cdot \mathbf{Q}^T \rightarrow \mathbf{E} = \mathbf{Q} \cdot \mathbf{\Delta}^2 \cdot \mathbf{Q}^T \quad (\text{B.13})$$

which is formally equal to the eigenvector decomposition in (B.11). Finally, one can say that the SVD decomposition for a semi-definite matrix corresponds to the eigenvector decomposition. Moreover, the eigenvalues correspond to the squared singular values. It should be pointed out that SVD is widely used into numerical algebra analysis since it exists for any matrix.

The MatLAB® routine to perform the SVD on a general rectangular matrix is the “svd” function.

Annex C.

MEX FILE IN MATLAB®

All numerical procedures and algorithms proposed in this dissertation have been tested and implemented by using the MatLAB® computer programming language. However, when large data sets are allocated and accessed within loops, MatLAB® is not too much efficient. One manner to optimize and speed-up MatLAB® accessing data is by using compiled source codes, written in fortran or c++ language.

This Annex describes how to write and compile a MEX *dll* function, written in c++ language, for MatLAB®.

C.1 Introduction

Assume to create an array¹, **A**, whose entries are all integer from 1 to 10^7 . From MatLAB® script one can write:

```
%- create sample array
N=10^7;

%- initialize array
N=zeros(1,N);

%- start loop
for I=1:N
    A(1,I)=int(I); %-allocate "int" array
end
```

¹ Obviously, such as array may be easily defined as "x=1:10⁷". This example should show how loops are not efficient into MatLAB®.

Annex C. MEX File in MatLAB®

The run-time on a Core Duo 1.83 GHz, 2 GB RAM, Win Xp SP2 is 0.10650 s. The same array will be created by using a MEX compiled function, as described in the next Section.

C.2 Write and Compile MEX File

The source code for a MEX file consists of two main distinct parts:

- *computational routine*: it contains the code performing the needed computations.
- *gateway routine*: it is the main function which links with MatLAB®.

The general form of a source MEX file is shown below:

```
// include mex header
#include "mex.h"

// COMPUTATIONAL ROUTINE SECTION
void userfnc#1(...)
{
    //... routine code...
}

double userfnc#2(...)
{
    //... routine code...
}

// GATEWAY ROUTINE SECTION
void mexFunction(int nlhs, mxArray *plhs[], int nrhs,
                  const mxArray *prhs[])
// nlhs = number of output items
// nrhs = number of input items
// plhs = output pointer
// prhs = input pointer
{
    // get pointer from MatLAB® input
    A = mxGetPr(plhs[...]);

    // create MatLAB® variable
    plhs[...] = mxCreateDoubleMatrix(...);

    // allocate output variable
    userfnc(...);
}
```

The name of the gateway function is always “*mexFunction*”. This function parses all MatLAB® inputs into pointers variables (“*mxGetPr*”) and create the output MatLAB® variables (“*mxCreateDoubleMatrix*”). Assuming *testmex.c* is the source code file, the following line should be written to compile from MatLAB®:

```
%- link and compile mex source code
mex testmex.c
```

Therefore, one can now write a MEX file which creates the **A** array. The source

Annex C. MEX File in MatLAB®

c++ code is like this:

```
#include "mex.h"

// computational routine
void allocateArray(double A[], int nr)
{
    int i; // locale variable
    for (i=0; i<nr; i++){
        A[i]=i+1; // fill array
    }
}

// gateway routine
void mexFunction(int nlhs, mxArray *plhs[], int nrhs,
                 const mxArray *prhs[])
{
    int *A; // pointer to "int" type
    int nr; // define "int" variable

    // get "int" value from the pointer
    nr = (int)*mxGetPr(prhs0);

    // allocate MatLAB® double matrix
    plhs[0] = mxCreateDoubleMatrix(nr,1, mxREAL);
    // get the pointer
    A = mxGetPr(plhs[0]);
    // allocate the output array by using the computational routine
    allocateArray(A,nr);
}
```

The compiled function may be easily called from MatLAB®, typing:

```
%- use compiled function
A=testmex(N);
```

The elapsed run-time is 0.00010 s. Then, by using compiled MEX function the MatLAB® code speeds-up of a factor equals to 10^3 .

Finally, when large data set are managed or nested loops are required to perform any calculation, using MEX compiled functions is a good manner to speed-up MatLAB® execution.

Annex D.

INTRODUCTION TO GRAPH THEORY

Graphs are a very powerful tool used in many fields, from electrical applications to image processing and mesh data analysis. Graphs are widely used in this dissertation as a support to model and analyze mechanical assembly, perform motion and over-constraint analysis and to manipulate mesh data for computer graphics purposes.

This Annex provides an overview on the main concept of Graph Theory, highlighting the key features needed for the implementation of numerical algorithms proposed in the present dissertation.

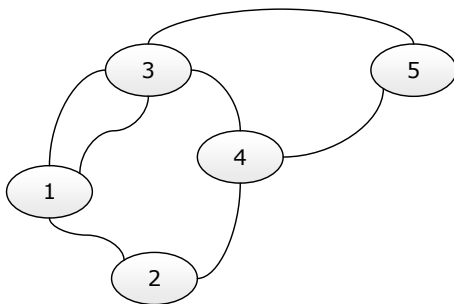


Figure D.1.a: graph representation.
Non-oriented graph

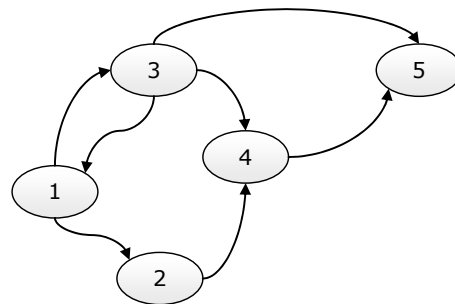


Figure D.1.b: graph representation.
Oriented graph

D.1 Edge Matrix and Incidence Matrix

A graph G is usually defined by means of the vector list of *vertices*, \mathbf{V} , and the *edge* matrix, \mathbf{E} . \mathbf{V} is a vector of integer ranging from 1 to N_v , where N_v is the total number of vertices. \mathbf{E} is an $N_e \times 2$ matrix, in which the i -th row has the indices of vertices

Annex D. Introduction to Graph Theory

connected by that edge. Let (i, j) be the couple of entries on the i -th row. For non-oriented graphs (see Figure D.1.a) the couple (i, j) corresponds to (j, i) . Instead, for oriented graphs, the couple (i, j) does not match (j, i) one. Moreover, the same couple of vertices may be connected with more than one edge (in Figure D.1, vertices 1 and 3 are connected with 2 edges)¹. Looking at Figure D.1, edge matrices, \mathbf{E}_{NO} and \mathbf{E}_{O} , for non-oriented and oriented graphs, respectively, are:

$$\mathbf{E}_{\text{NO}} = \begin{bmatrix} 1 & 2 \\ 1 & 3 \\ 3 & 1 \\ 2 & 4 \\ 3 & 4 \\ 3 & 5 \\ 4 & 5 \end{bmatrix} \equiv \begin{bmatrix} 2 & 1 \\ 3 & 1 \\ 1 & 3 \\ 4 & 2 \\ 4 & 3 \\ 5 & 3 \\ 5 & 4 \end{bmatrix}, \quad \mathbf{E}_{\text{O}} = \begin{bmatrix} 1 & 2 \\ 1 & 3 \\ 3 & 1 \\ 2 & 4 \\ 3 & 4 \\ 3 & 5 \\ 4 & 5 \end{bmatrix}$$

An useful representation of graphs, based on the edge matrix, may be achieved with the *incidence* matrix, \mathbf{I} . This matrix is a rectangular $N_v \times N_e$ matrix and it is defined as follows:

$$\begin{cases} \mathbf{I}(\mathbf{E}(i,1),i) = 1 \\ \mathbf{I}(\mathbf{E}(i,2),i) = \begin{cases} -1 \rightarrow \text{oriented} \\ 1 \rightarrow \text{non - oriented} \end{cases} \\ 0 \rightarrow \text{otherwise} \end{cases} \quad \forall i = 1,2,\dots,N_e \quad (\text{D.1})$$

$$\mathbf{I}_{\text{NO}} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

$$\mathbf{I}_{\text{O}} = \begin{bmatrix} 1 & 1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & -1 & -1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & -1 & -1 \end{bmatrix}$$

¹ Multi-edge graphs are typically used to handle mesh or tessellated format into computer graphics or CAD applications.

Annex D. Introduction to Graph Theory

For example, the incidence matrices, \mathbf{I}_{NO} and \mathbf{I}_O , for non-oriented and oriented graphs, respectively, in Figure D.1, are listed above.

D.2 Graph Matrix

Similar to incidence matrix is the *graph* matrix, \mathbf{G} . This matrix is symmetric and squared. The number of rows or columns corresponds to the number of vertices, N_v . It is defined as follows:

$$\begin{cases} \mathbf{G}(i, j) = \sum_{k=1}^{N_e} \text{edge}_k, i \neq j \\ 0 \rightarrow \text{otherwise} \end{cases} \quad \forall i, j = 1, 2, \dots, N_v \quad (\text{D.2})$$

The entry (i, j) in \mathbf{G} counts all edges connecting the vertex V_i to V_j . For example, looking at Figure D.1.a, it has:

$$\mathbf{G} = \begin{bmatrix} 0 & 1 & 2 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 2 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

This kind of matrix is often used into computer graphics and reverse engineering applications to easily manage mesh data format.

Annex E.

TOLERANCE ZONES

This Annex provides a mathematical representation of tolerance zones, related to planar and cylindrical features. Parameter constraints are derived for each tolerance zone and a graphical representation of the corresponding hyper-domain is provided.

E.1 Introduction

Into TTRS Theory, features are classified according to their respective degree of freedom, also called Degree of Invariance, referring to the fact that their geometry and position remain mathematically unchanged when subjected to displacement along or around these directions. Basically, seven main features are identified: planar feature, cylindrical feature, revolution feature, spherical feature, prismatic feature, helicoidal feature and complex feature (Desrochers, 1999).

Each main feature is described by means of a combination of elementary geometrical entities, point, line and plane. These elements define the so-called Minimal Geometrical Reference Element (MGRE).

This dissertation focuses on planar and cylindrical features. The MGRE associated to a planar feature is the plane itself, while the MGRE associated to the cylindrical feature is its axis¹. TTRS Theory proposes five tolerance zones for the planar and the cylindrical features: one related to the plane and four related to the axis. It should note that this classification accounts all ISO or ANSI tolerance specification rules, such as datum precedence, multi-tolerance representation, application of material modifiers.

¹ In the next cylindrical features and axis will be confused, since they have the same meaning.

Annex E. Tolerance Zones

As stated into Chapter 5, for each feature an attached local frame is built. In the next, the five tolerance zones will be analyzed, taking into account the specific coordinate frame.

Generally speaking, the displacement of any point P_j , belonging to the derived feature is calculated by applying the relationship (5.11), which is here re-written for brevity.

$$\mathbf{D}_{P_j} = \Delta_F + \Theta_F \wedge P_j \quad (5.11)$$

E.2 Tolerance Zone Representation

Planar Feature: Two Parallel Planes

For planar features, translational and rotational small displacement parameters become $\Delta_F = (\Delta x_F, 0, 0)$ and $\Theta_F = (0, \beta_F, \gamma_F)$, respectively.

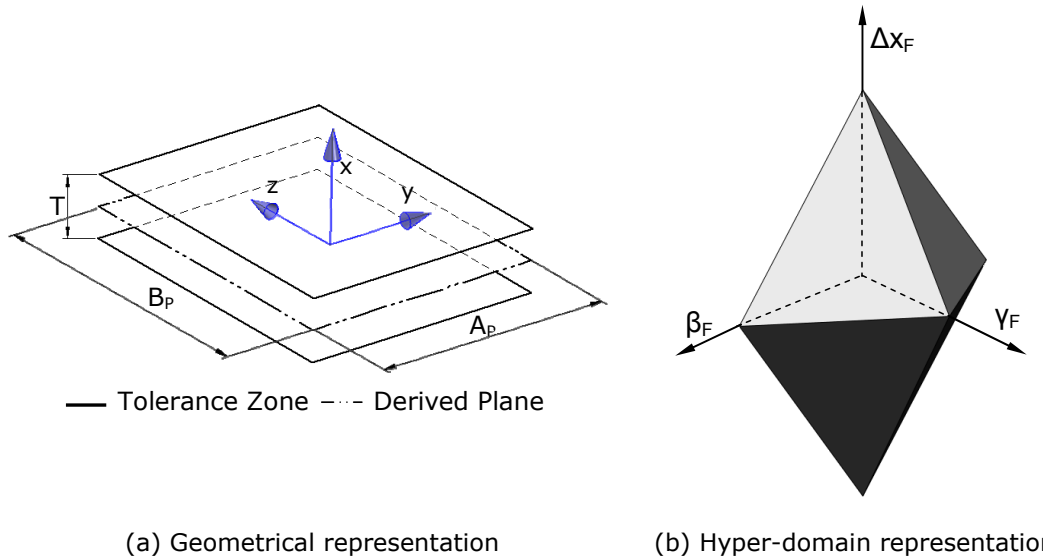


Figure E.1: tolerance zone. Planar feature-two parallel planes

After writing equation (5.11) for the four vertices of the rectangular planar geometry, variational parameter constraints are:

$$|\Delta x_F \pm \gamma_F \cdot A_P / 2 \mp \beta_F \cdot B_P / 2| \leq \frac{T}{2} \quad (E.1)$$

Inequalities (E.1) assure that any point belonging to the derived feature is inside the specified tolerance zone, T , supposed symmetrical.

Variational parameter intervals may be evaluated by calculating the minimum

Annex E. Tolerance Zones

and maximum extends along the parameter space axes, as stated into relationships (E.2).

$$\begin{cases} \Delta x_F \in \left[-\frac{T}{2}; \frac{T}{2} \right] \\ \beta_F \in \left[-\frac{T}{B_p}; \frac{T}{B_p} \right]; \gamma_F \in \left[-\frac{T}{A_p}; \frac{T}{A_p} \right] \end{cases} \quad (E.2)$$

Figure E.1.b shows the hyper-domain corresponding the inequalities (E.1). It can be note that each face of the hyper-domain is a triangular planar face.

Cylindrical Feature: Cylindrical Zone

For cylindrical features, translational and rotational small displacement parameters become $\Delta_F=(0, \Delta y_F, \Delta z_F)$ and $\Theta_F=(0, \beta_F, \gamma_F)$, respectively.

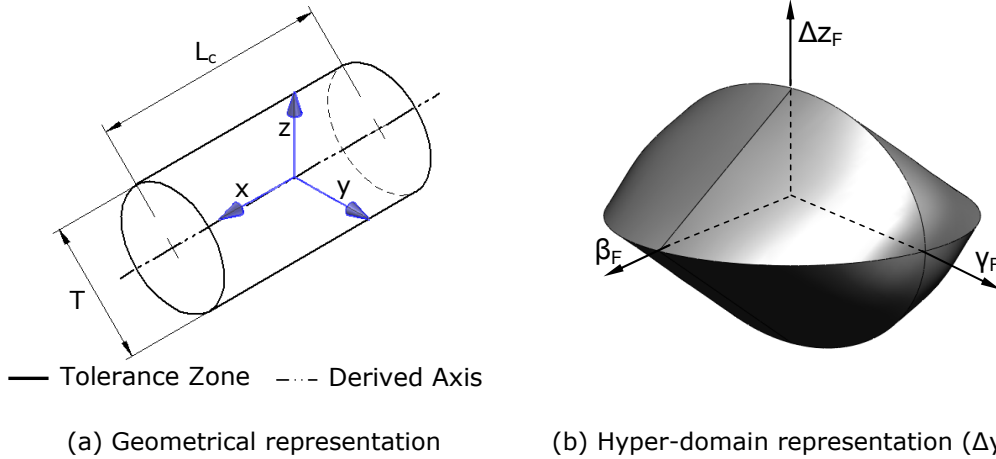


Figure E.2: tolerance zone. Cylindrical feature-cylindrical zone

In this case, equation (5.11) must be written for the two end points of the cylindrical feature's axis, as into equation (E.3). Variational parameter intervals are stated into relationship (E.4).

$$\sqrt{(\Delta y_F \pm \gamma_F \cdot L_c/2)^2 + (\Delta z_F \mp \beta_F \cdot L_c/2)^2} \leq \frac{T}{2} \quad (E.3)$$

$$\begin{cases} \Delta y_F \in \left[-\frac{T}{2}; \frac{T}{2} \right]; \Delta z_F \in \left[-\frac{T}{2}; \frac{T}{2} \right] \\ \beta_F \in \left[-\frac{T}{L_c}; \frac{T}{L_c} \right]; \gamma_F \in \left[-\frac{T}{L_c}; \frac{T}{L_c} \right] \end{cases} \quad (E.4)$$

Figure E.2.b shows the hyper-domain corresponding the inequalities (E.3). It

Annex E. Tolerance Zones

can be noted that the hyper-domain (assuming $\Delta y_F=0$ - the same may be said when $\Delta z_F=0$) becomes an ellipse into the plane $(\Delta z_F, \gamma_F)$, a circle into the plane (β_F, γ_F) , and a rhombus into the plane $(\beta_F, \Delta z_F)$.

Cylindrical Feature: Two Coaxial Cylinders

Looking at Figure E.3.a and relationship (E.3), hyper-domain inequalities may be written as follows:

$$\frac{T_1}{2} \leq \sqrt{(\Delta y_F \pm \gamma_F \cdot L_c/2)^2 + (\Delta z_F \mp \beta_F \cdot L_c/2)^2} \leq \frac{T_2}{2} \quad (E.5)$$

This means that any point of the cylindrical feature's axis lies inside the geometrical space obtained as difference between the cylinder of diameter T_2 and the cylinder of diameter T_1 .

Variational parameter intervals become, accordingly:

$$\begin{cases} \Delta y_F \in \left\{ \begin{array}{l} [T_1/2; T_2/2] \\ [-T_2/2; -T_1/2] \end{array} \right\}; \Delta z_F \in \left\{ \begin{array}{l} [T_1/2; T_2/2] \\ [-T_2/2; -T_1/2] \end{array} \right\} \\ \beta_F \in \left\{ \begin{array}{l} [T_1/L_c; T_2/L_c] \\ [-T_2/L_c; -T_1/L_c] \end{array} \right\}; \gamma_F \in \left\{ \begin{array}{l} [T_1/L_c; T_2/L_c] \\ [-T_2/L_c; -T_1/L_c] \end{array} \right\} \end{cases} \quad (E.6)$$

Figure E.3.b shows the hyper-domain corresponding to the inequalities (E.5).

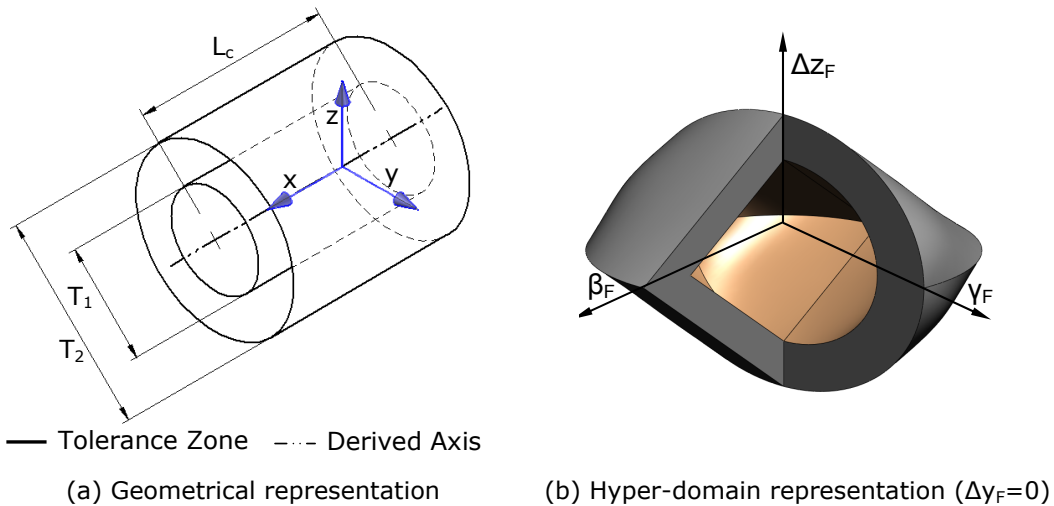


Figure E.3: tolerance zone. Cylindrical feature-two coaxial cylinders

Cylindrical Feature: Two Parallel Planes

Looking at Figure E.4.a, translational and rotational small displacement parameters become $\Delta_F=(0, \Delta y_F, 0)$ and $\Theta_F=(0, 0, \gamma_F)$, respectively.

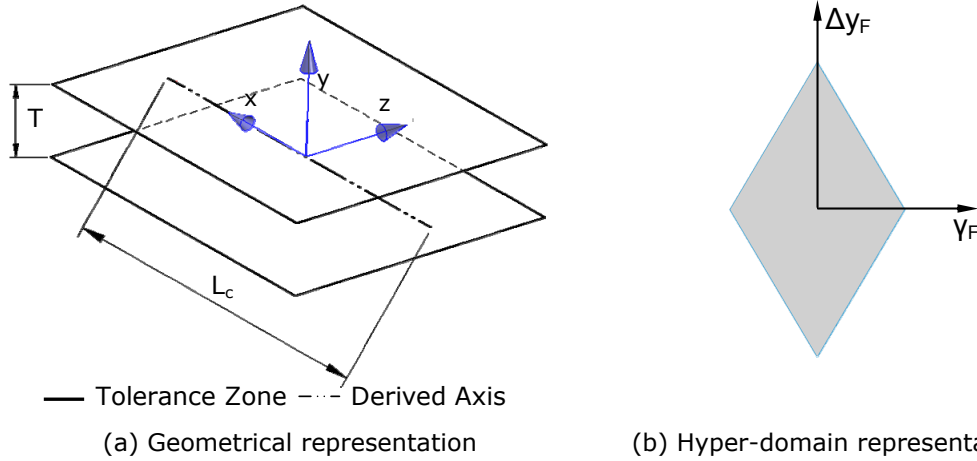


Figure E.4: tolerance zone. Cylindrical feature-two parallel planes

Writing equation (5.11) for the two end points of the cylindrical feature's axis, and imposing that the relative displacement lies within the tolerance range, T, it has:

$$|\Delta y_F \pm \gamma_F \cdot L_c/2| \leq \frac{T}{2} \quad (E.7)$$

Variational parameter intervals become, accordingly:

$$\begin{cases} \Delta y_F \in \left[-\frac{T}{2}; \frac{T}{2} \right]; \\ \gamma_F \in \left[-\frac{T}{L_c}; \frac{T}{L_c} \right] \end{cases} \quad (E.8)$$

It can be noted that the hyper-domain reduces to a rhombus belonging to the plane $(\Delta y_F, \gamma_F)$.

Cylindrical Feature: Parallelepiped Zone

Looking at Figure E.5.a, translational and rotational small displacement parameters become $\Delta_F=(0, \Delta y_F, \Delta z_F)$ and $\Theta_F=(0, \beta_F, \gamma_F)$, respectively.

Moreover, translations along feature coordinate frame y and z are independent each other. Thus, one can write:

Annex E. Tolerance Zones

$$\begin{cases} |\Delta y_F \pm \gamma_F \cdot L_c / 2| \leq \frac{T_2}{2} \\ |\Delta z_F \pm \beta_F \cdot L_c / 2| \leq \frac{T_1}{2} \end{cases} \quad (E.9)$$

This means that the couples $(\Delta z_F, \beta_F)$ and $(\Delta y_F, \gamma_F)$ are independent each-other.

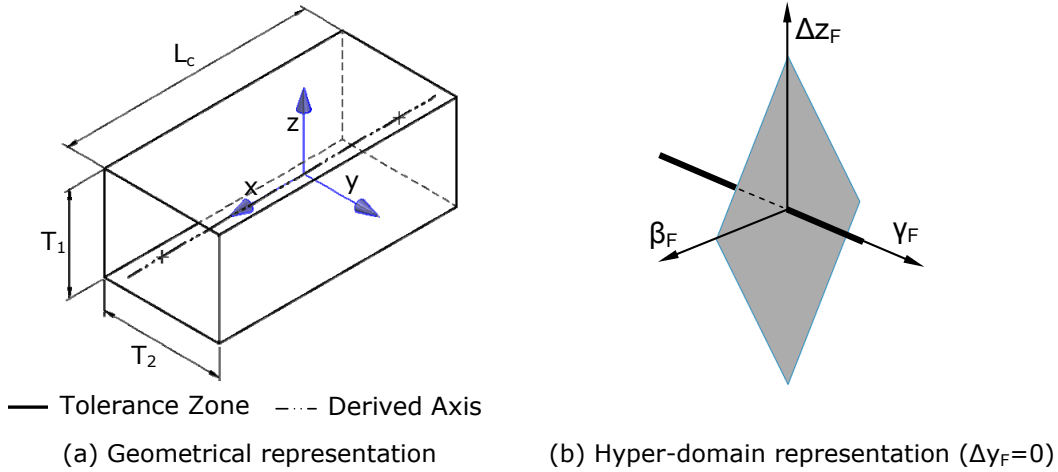


Figure E.5: tolerance zone. Cylindrical feature-parallelepiped zone

Variational parameter intervals become, accordingly:

$$\begin{cases} \Delta y_F \in \left[-\frac{T_2}{2}, \frac{T_2}{2} \right]; \Delta z_F \in \left[-\frac{T_1}{2}, \frac{T_1}{2} \right] \\ \beta_F \in \left[-\frac{T_1}{L_c}, \frac{T_1}{L_c} \right]; \gamma_F \in \left[-\frac{T_2}{L_c}, \frac{T_2}{L_c} \right] \end{cases} \quad (E.10)$$

Figure E.5.b shows the relative hyper-domain. It can be noted that in parametric space $(\beta_F, \gamma_F, \Delta z_F)$ the hyper-domain reduces to the union of a rhombus and a line: the line is normal to the plane $(\beta_F, \Delta z_F)$.

E.3 Summary

Table E.1 depicts the interval ranges, related to each analyzed tolerance zone.

Tolerance Zone	Interval Range
<p>planar feature two parallel planes</p> $ \Delta x_F \pm Y_F \cdot A_p/2 \mp \beta_F \cdot B_p/2 \leq \frac{T}{2}$	$\left\{ \begin{array}{l} \Delta x_F \in \left[-\frac{T}{2}; \frac{T}{2} \right] \\ \beta_F \in \left[-\frac{T}{B_p}; \frac{T}{B_p} \right]; Y_F \in \left[-\frac{T}{A_p}; \frac{T}{A_p} \right] \end{array} \right.$
<p>cylindrical feature cylindrical zone</p> $\sqrt{(\Delta y_F \pm Y_F \cdot L_c/2)^2 + (\Delta z_F \mp \beta_F \cdot L_c/2)^2} \leq \frac{T}{2}$	$\left\{ \begin{array}{l} \Delta y_F \in \left[-\frac{T}{2}; \frac{T}{2} \right]; \Delta z_F \in \left[-\frac{T}{2}; \frac{T}{2} \right] \\ \beta_F \in \left[-\frac{T}{L_c}; \frac{T}{L_c} \right]; Y_F \in \left[-\frac{T}{L_c}; \frac{T}{L_c} \right] \end{array} \right.$
<p>cylindrical feature two coaxial cylinders</p> $\frac{T_1}{2} \leq \sqrt{(\Delta y_F \pm Y_F \cdot L_c/2)^2 + (\Delta z_F \mp \beta_F \cdot L_c/2)^2} \leq \frac{T_2}{2}$	$\left\{ \begin{array}{l} \Delta y_F \in \left\{ \begin{array}{l} [T_1/2; T_2/2] \\ [-T_2/2; -T_1/2] \end{array} \right\}; \Delta z_F \in \left\{ \begin{array}{l} [T_1/2; T_2/2] \\ [-T_2/2; -T_1/2] \end{array} \right\} \\ \beta_F \in \left\{ \begin{array}{l} [T_1/L_c; T_2/L_c] \\ [-T_2/L_c; -T_1/L_c] \end{array} \right\}; Y_F \in \left\{ \begin{array}{l} [T_1/L_c; T_2/L_c] \\ [-T_2/L_c; -T_1/L_c] \end{array} \right\} \end{array} \right.$
<p>cylindrical feature two parallel planes</p> $ \Delta y_F \pm Y_F \cdot L_c/2 \leq \frac{T}{2}$	$\left\{ \begin{array}{l} \Delta y_F \in \left[-\frac{T}{2}; \frac{T}{2} \right]; \Delta z_F = 0 \\ Y_F \in \left[-\frac{T}{L_c}; \frac{T}{L_c} \right]; \beta_F = 0 \end{array} \right.$
<p>cylindrical feature parallelepiped zone</p> $\left\{ \begin{array}{l} \Delta y_F \pm Y_F \cdot L_c/2 \leq \frac{T_2}{2} \\ \Delta z_F \pm \beta_F \cdot L_c/2 \leq \frac{T_1}{2} \end{array} \right.$	$\left\{ \begin{array}{l} \Delta y_F \in \left[-\frac{T_2}{2}; \frac{T_2}{2} \right]; \Delta z_F \in \left[-\frac{T_1}{2}; \frac{T_1}{2} \right] \\ \beta_F \in \left[-\frac{T_1}{L_c}; \frac{T_1}{L_c} \right]; Y_F \in \left[-\frac{T_2}{L_c}; \frac{T_2}{L_c} \right] \end{array} \right.$

Table E.1: tolerance zones-summary

Annex F.

BOUNDARY INTEGRAL DEFINITION

F.1 Integral Definition

This Annex gives more mathematical details to the integral expression defined in equation (7.17).

Let Ω be a domain. The aim is to define an integral on a sub-domain, Ω_s . Let Ω_s be a spherical sub-domain. A point \mathbf{P} belongs to Ω_s if the following condition is satisfied:

$$\sqrt{(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2} \leq R \rightarrow \|\mathbf{P} - \mathbf{P}_0\| \leq R \quad (\text{F.1})$$

where \mathbf{P}_0 is the center of the sphere, while R is its radius. A step-wise function, Δ_Ω , may be defined to account all points belonging to domain Ω . Then, it has:

$$\Delta_\Omega = \begin{cases} 1, & \text{if } \|\mathbf{P} - \mathbf{P}_0\| \leq R \\ 0, & \text{otherwise} \end{cases} \quad (\text{F.2})$$

Let now λ be a physical variable aiming to be integrated on the sub-domain Ω_s . One can write:

$$\lambda_{\text{int}} = \int_{\Omega} \Delta_\Omega \cdot \lambda \cdot d\Omega \quad (\text{F.3})$$

Taking in account equation (A.2), λ_{int} becomes:

$$\lambda_{\text{int}} = \int_{\Omega_s} 1 \cdot \lambda \cdot d\Omega_s + \int_{\Omega} 0 \cdot \lambda \cdot d\Omega = \int_{\Omega_s} \Delta_\Omega \cdot \lambda \cdot d\Omega_s \quad (\text{F.4})$$

Annex F. Boundary Integral Definition

The average value of the physical variable λ may be achieved by applying the definition of mean for a continuous function. Finally, one can write:

$$\lambda_{\mu} = \frac{1}{\Omega_s} \int_{\Omega_s} \Delta_{\Omega} \cdot \lambda \cdot d\Omega_s \quad (\text{F.5})$$

Annex G.

BULK DATA FILE

This Annex provides a systematic overview of the MSC NASTRAN®'s Bulk Data File (bdf), widely used in this dissertation to implement the SVA-FEA software.

G.1 Introduction

The bdf file is an ASCII formatted file containing all input information needed to do the numerical analysis in MSC NASTRAN®. The main structure is made of three sub-sections:

- *executive control statement*: includes solver options and diagnosis operations;
- *case control section*: includes sub-case entries and output queries; and,
- *bulk data section*: includes the FE model (nodes, elements and boundary conditions).

The general structure of a bdf file is listed below.

```
$- executive control statement
SOL 101 $- solver option
CEND

$- case control section
SUBCASE 1 $- sub-case entries
...

$- beginning bulk data section
BEGIN BULK
...
CQUAD4, 104, 1, 122, 123, 134, 133
...
GRID, 1, , 0.00, 0.0, 0.0, 5
```

Annex G. Bulk Data File

```
...
SPC, 1, 176, 1, 0.00
...
PWELD, 1, 3, 5.00
...
CWELD, 1, 1, , ALIGN, 125, 9
...
FORCE, 1, 9, 2, 10.54, 0.00, 0.00, 1.00
...
CORD2R, 1, , 80.00, 0.00, 4.00, 80.00, 0.00, 5.00, 79.00, 0.00, 4.00
...
ENDDATA
$- ending bulk data file
```

The solver adopted in this dissertation is the linear static solver. The related call into the executive control statement is SOL 101.

In the following, some specific entries of the bdf file will be analyzed in detail.

G.2 BDF Structure

This dissertation focuses on shell elements with four (CQUAD4) or three (CTRIA3) nodes. Therefore, in following only these elements will be accounted.

Material and Shell Property Entry

The bdf syntax to define an isotropic material is:

MAT1, IDMAT, E, ν , ρ

where MAT1 is the bdf statement, IDMAT is the identification integer, E, ν and ρ are the Young's modulus, the Poisson's ratio and the density, respectively. An example is listed below.

```
$- creating an isotropic material
MAT1, 1, 210000.0, , 0.30, 7900.0
```

Once the material is correctly defined, the shell property entry is:

PSHELL, IDSH, IDMAT, TH

where PSHELL is the bdf statement, IDSH is the identification integer, TH is the input thickness. A bdf example, using the material "1" and setting the thickness equal to 2.4, is listed below.

```
$- creating a pshell property
PSHELL, 1, 1, 2.4
```

Element Entry

CQUAD4 and CTRIA3 elements may be defined with the following syntax:

CQUAD4, IDEL, IDSH, IDG1, IDG2, IDG3, IDG4

CTRIA3, IDEL, IDSH, IDG1, IDG2, IDG3

where QUAD4 and CTRIA are the bdf statements, IDEL is the identification integer of the element, while IDGi is the identification integer for the node i-th, connected by the element. An example is listed below.

```
$- creating a CQUAD4 element
CQUAD4, 48, 1, 1, 2, 3, 4
$- creating a CTRIA3 element
CTRIA3, 23, 1, 3, 4, 5
```

Coordinate Frame Entry

Cartesian coordinate frames may be easily assigned by defining three not aligned points, P_1 , P_2 and P_3 . The syntax is:

CORD2R, IDUCS, IDUCS0 , X/Y/Z-P1, X/Y/Z-P2, X/Y/Z-P3

where CORD2R is the bdf statement, IDUCS is the identification integer, while IDUCS0 is the coordinate frame in which the three points are defined. An example is listed below.

```
$- creating a cartesian coordinate frame
CORD2R, 1 , , 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 1.0, 0.0
$- using frame "1" to define a new coordinate frame
CORD2R, 4, 1, 80.00, 0.00, 4.00, 80.00, 0.00, 5.00, 79.00, 0.00, 4.00
```

Node Entry

Nodes may be defined with the following syntax:

GRID, IDG, IDUCSg, X, Y, Z, IDUCSdof

where GRID is the bdf statement, IDG is the identification integer, X, Y and Z are the coordinates defined into frame IDUCSg. IDUCSdof is the coordinate frame used to define the DoFs and the output results for that node.

An example is listed below. In should be noted that the geometric frame and the DoF frame are different among them.

```
$- creating a GRID entry
GRID, 33, 1, 10.4, 8.4, 0.0, 4
```

Single Point Constraint (SPC) Entry

SPC may be defined with the following syntax:

SPC, IDSPC, IDG , DoF, SPCvalue

where SPC is the bdf statement, IDSPC is the identification integer, DoF is the list of constrained DoFs and SPCvalue is the assigned value. An example is listed below.

```
$- creating an SPC
SPC, 3, 1, 123456, -5.00 $- ("123456" all 6 DoFs are constrained and are con-
strained to "-5.00")
SPC, 4, 1, 16, 0.00 $- ("16" only X translation and Z rotation are constrained
to "0.00")
```

Multi Point Constraint (MPC) Entry

MPCs define linear coupling conditions among DoFs of different nodes (master and slave nodes). The general mathematical formulation is:

$$\sum_j R_j \cdot u_j = 0 \quad (G.1)$$

where R_j is a constant, u_j is the j -th DoF. MPC may be defined with the following syntax:

MPC, IDMPC, IDGms, DoFms, R_{jms} , IDGsl, DoFsl, R_{jsl}

where MPC is the bdf statement, IDMPC is the identification integer, IDGms and IDGsl are the identification integers for the master and the slave nodes, whereas DoFms and the DoFsl are the related DoFs. An example, in which the translation along X, Y and Z and the rotation around Y for node "232" and "116" are set equal each-other, is listed below.

```
$- creating MPC
MPC, 1, 232, 1, 1.00, 116, 1, -1.00
MPC, 1, 232, 2, 1.00, 116, 2, -1.00
MPC, 1, 232, 3, 1.00, 116, 3, -1.00
MPC, 1, 232, 5, 1.00, 116, 5, -1.00
```

Linear Contact Entry

In MSC NASTRAN® contacts among surfaces may be analyzed also in a linear static (SOL 101) solution. The general MPC equation is:

$$\mathbf{S} - \mathbf{u}_{ms} + \mathbf{u}_{sl} - \mathbf{u}_g = \mathbf{0} \quad (G.2)$$

Annex G. Bulk Data File

which corresponds to the canonical form of the MPC condition (see equation (G.1)).

An example is listed below.

```
$- creating a linear gap between nodes "4" and "5" along the Z direction
$- definition of scalar points (SPOINT)
SPOINT, 243, 244
$- setting the variable "S"
SUPPORT, 243, 0
$- setting the initial gap equal to 3 "ug=3"
SPC, 1, 244, 0, 3.00
$- MPC entry
MPC, 1, 4, 3, 1.00, 5, 3, -1.00, , , 244, 0, 1.00, 243, 0, -1.00
```

PWELD/CWELD Entry

Connectors elements may be defined in three different way: (I) point-to-point, (II) point-to-patch, (III) patch-to-patch. This dissertation focuses on the first one (CWELD). CWELDs may be defined as follows:

PWELD, IDMATpw, IDMAT, DIA

CWELD, IDCw, IDMATpw, ALIGN, IDGms, IDGsl

where PWELD and CWELD are the bdf statements. In particular, PWELD defines the material property for the connector: the diameter DIA has to be assigned. The flag ALIGN declares nodes IDGms and IDGsl are connected.

An example is listed below. PWELD uses the material "3" and sets the diameter equal to "5.0". CWELD connects nodes "125" and "9".

```
$- pweld entry
PWELD, 1, 3, 5.0
$- cweld entry
CWELD, 1, 1, , ALIGN, 125, 9
```

FORCE/MOMENT Entry

Forces/moments may be defined by assigning a unit vector, the magnitude and the coordinate frame, in which forces/moments are declared. The syntax is:

FORCE/MOMENT, IDFM, IDGfm, IDUCSfm, FMI, Xfm, Yfm, Zfm

where FORCE/MOMENT are the bdf statements, IDFM is the identification integer, FMI defines the magnitude, and Xfm, Yfm and Zfm define the unit vector.

An example is listed below. A force of "13.45" is applied to node "9" along the Z direction, with respect to frame "3". Moreover, a moment of "4.22" is set to node "10" around the X direction, with respect to the frame "4".

Annex G. Bulk Data File

```
$- force entry into z direction  
FORCE,1,9,3,13.45,0.00,0.00,1.00  
$- moment entry into x direction  
MOMENT,1,10,4,4.22,1.00,0.00,0.00
```

Annex G. Bulk Data File

G.3 Examples

In order to clarify all above bdf entries, this Section shows a complete bdf code to define a SOL 101 analysis (this code may be copied and pasted into any word editor and run into MSC NASTRAN®).

```
$- solver input
SOL 101
CEND
$- sub-case entry
SUBCASE 1
SPC = 1
LOAD = 2
DISPLACEMENT(SORT1,REAL)=ALL $- store in output only node displacement

$- beginning bulk section
BEGIN BULK
PARAM, POST, 0 $- set output file (post-processing file)
$- pshell entry
PSHELL, 1, 1, 2.0000, 1, , 1
$- material entry
MAT1, 1, 210000.00, , 0.30
$- elements entry
CQUAD4, 1, 1, 1, 2, 3, 4
CTRIA3, 2, 1, 3, 4, 5
$- grids entry (DoFs are defined into frame "1")
GRID, 1, , , 0.0, 0.0, 0.0, 1
GRID, 2, , , 0.0, 100.0, 0.0, 1
GRID, 3, , , 100.0, 100.0, 0.0, 1
GRID, 4, , , 100.0, 0.0, 0.0, 1
GRID, 5, , , 150.0, 50.0, 0.0, 1
$- spc entry
SPC, 1, 1, 3, -1.00
SPC, 1, 2, 123456, 0.00
$- coordinate frame entry
CORD2R, 1, , , 100.00, 0.00, 0.00, 150.00, 50.00, 0.00, 100.00, 100.00, 0.00
$- force entry (force is defined into frame "1")
FORCE, 2, 5, 1, 1000.00, 0.00, 1.00, 0.00
ENDDATA
$- ending bulk section
```


References

References

(Asada, 1986), Adams, H., Slotine, J. J. E., 1986, *Robot Analysis and Control*, Wiley-Interscience, ISBN: 978-0471830290.

(ASME Y14.5M, 1994), ASME Y14.5M, Dimensioning and Tolerancing, ASME Y14.5M-1994, American Society of Mechanical Engineers, New York.

(ASME Y14.5.1M, 2004), ASME Y14.5.1M, Mathematical definition of dimensioning and Tolerancing principles, American Society of Mechanical Engineers, New York, re-affirmed 2004.

(Adams, 1998), Adams, J. D., 1998, *Feature Based Analysis of Selective Limit Motion in Assemblies*, MS Thesis, MIT, Cambridge, MA.

(Adams, 1999a), Adams, J. D., Whitney, D. E., 1999, *Application of Screw Theory to Constraint Analysis of Assemblies of Rigid Parts*, Proc. of IEEE International Symposium on Assembly and Task Planning, Porto, Portugal, pp. 69-74.

(Adams, 1999b), Adams, J. D., Gerbino, S., Whitney, D. E., 1999, *Application of Screw Theory to Motion Analysis of Assemblies of Rigid Parts*, Proc. of IEEE International Symposium on Assembly and Task Planning, Porto, Portugal, pp. 75-80.

(Adams, 2001), Adams, J. D., Whitney, D. E., 2001, *Application of Screw Theory to Constraint Analysis of Mechanical Assemblies Joined by Features*, ASME Journal of Mechanical Design, Vol. 123, pp. 26-32.

(Baker, 1980), Baker J. E., 1980, *On Relative Freedom between Links in Kinematic Chains with Cross-Jointing*, Mechanism and Machine Theory, Vol. 15, pp. 397-413.

(Ballu, 1999), Ballu, A., Mathieu, L., 1999, *Choice of Functional Specifications using Graphs within the Framework of Education*, Proc. of the 6th CIRP Int. seminar on Computer-Aided Tolerancing, Ensched, The Netherlands, pp.197-206.

References

(Bihlmaier, 1999), Bihlmaier, B. F., 1999, *Tolerance Analysis of Flexible Assemblies using Finite Element and Spectral Analysis*, MS Thesis, Brigham Young University, Utah.

(Borrel, 1994), Borrel, P., Rappoport, A., 1994, *Simple Constrained Deformations for Geometric Modelling and Interactive Design*, ACM Trans Graph 13, pp. 137-155.

(Bourdet, 1988), Bourdet, P., Clément, A., 1988, *A Study of Optimal Criteria Identification based on Small Displacement Screw Model*, CIRP Annals, Manufacturing Technology, Vol. 37, pp. 503-506.

(Cai, 1996), Cai, W., Hu, S. J., Yuan, J. X., 1996, *Deformable Sheet Metal Fixturing: Principles, Algorithms and Simulations*, ASME Journal of Manufacturing Science and Engineering, Vol. 118, pp. 318-324.

(Cai, 2006), Cai, W., Hsieh, C. C., Long, Y., Marin, S. P., Oh, K. P., 2006, *Digital Panel Assembly Methodologies and Applications for Compliant Sheet Components*, ASME Journal of Manufacturing Science and Engineering, Vol. 128, pp. 270-279.

(Camelio, 2002), Camelio, J. A., 2002, *Modeling and Diagnosis of Dimensional Variation for Assembly Systems with Compliant Parts*, PHD Dissertation, University of Wisconsin-Madison, USA.

(Camelio, 2003), Camelio, J., Hu, S. J. and Ceglarek, D., 2003, *Modeling Variation Propagation in Multi-Station Assembly Systems with Compliant Parts*, ASME Journal of Mechanical Design, Vol. 125, pp. 673-681.

(Camelio, 2004a), Camelio, J., Hu, S.J. and Marin, S.P., 2004, *Compliant Assembly Variation Analysis Using Component Geometric Covariance*, ASME Journal of Manufacturing Design and Engineering, Vol. 126, pp. 655-660.

(Camelio, 2004b), Camelio, J. A., Hu, S. J., Ceglarek, D., 2004, *Impact of Fixture Design on Sheet Metal Assembly Variation*, Journal of Manufacturing Systems, Vol. 23, pp. 182-193.

(Caputo, 2006), Caputo, F., Gerbino, S., 2006, *CAT-FEA Methods for Tolerance analysis of Flexible Part Assemblies: A Critical Overview*, Int. Conference on Accuracy in Forming Technology, ICAFT, Stadthalle Chemnitz, Germany, November 13-15.

(Ceglarek, 2009), Ceglarek, D., Huang, W., Zhou, S., Ding, Y., Kumar, R., Zhou, Y., 2009, *Time-Based Competition in Multistage Manufacturing: Stream-of-Variation Analysis (SOVA) Methodology - Review*, Journal of Flexible Manufacturing Systems, Vol. 16, pp. 11-44.

(Chang, 1997), Chang, M., Gossard, D. C., 1997, *Modeling the Assembly of Compliant, non-Ideal Parts*, Computer-Aided Design, Vol. 29, p. 701-708.

References

(Chase, 1991), Chase, K. W., Parkinson, A. R., 1997, *A Survey of Research in the Application of Tolerance Analysis to the Design of Mechanical Assemblies*, Research in Engineering Design, Vol. 3, pp. 23-37.

(Chase, 1995a), Chase, K. W., Magleby, S. P., Glancy, G., 1995, *A Comprehensive System for Computer-Aided Tolerance Analysis of 2-D and 3-D Mechanical Assembly*, in Geometric Design Tolerancing: Theories, Standards and Applications, ElMaraghy, H. A. Editor, Chapman and Hall, London, pp. 294-307.

(Chase, 1995b), Chase, K. W., Gao, J., Magleby, S. P., 1995, *General 2-D Tolerance Analysis of Mechanical Assemblies with Small Kinematic Adjustments*, ADCATS report.

(Chase, 1996a), Chase, K. W., Gao, J., Magleby, S. P., 1996, *Generalized 3-D Tolerance Analysis of Mechanical Assemblies with Small Kinematic Adjustments*, ADCATS report.

(Chase, 1996b), Chase, K. W., Gao, J., Magleby, S. P. Sorensen, C. D., 1996, *Including Geometric Feature Variations in Tolerance Analysis of Mechanical Assemblies*, ADCATS report.

(Chase, 1997), Chase, K. W., Magleby, S. P., 1997, *A Comprehensive System for Computer-Aided Tolerance Analysis of 2-D and 3-D Mechanical Assemblies*, Proc. of the 5th Int. Seminar on Computer-Aided Tolerancing Toronto, Canada, April 27-29.

(Choley, 2007), Choley, J. , Riviere, A., Clement, A., Bourdet, P., 2007, *A new Variational Association Process for the Verification of Tolerance Specification*, ASME Journal of Computing and Information Science in Engineering, Vol. 7, pp. 66-71.

(Chiabert, 2004), Chiabert, P. , Orlando, M., 2004, *About a CAT Model Consistent with ISO/TC 213 Last Issues*, Journal of Material Processing Technology, Vol. 157, pp. 61-66.

(Clément, 1998), Clément, A., Rivière, A., Serré, P., Valade C., 1998. *The TTRSs: 13 Constraints for Dimensioning and Tolerancing*, in Geometric Design Tolerancing: Theories, Standard and Application, Chapman and Hall, New York.

(Clément, 1999), Clément, A., Rivière, A., Serré, P., 1999, *Global Consistency of Dimensioning and Tolerancing*, Proc. of the 6th CIRP International Seminar on Computer-Aided Tolerancing pp. 1-26.

(Desrochers, 1999), Desrochers, A., 1999, *Modeling Three Dimensional Tolerance Zones Using Screw Parameters*, ASME, CD-ROM Proc. of 25th Design Automatic Conference, Las Vegas (USA), pp. 615-622.

(Dahlström, 2007), Dahlström, S., Lindkvist, L., 2007, *Variation Simulation of Sheet Metal Assemblies Using the Method of Influence Coefficients with Contact Modelling*,

References

ASME Journal of Manufacturing Science and Engineering, Vol. 129, pp. 615-622.

(Davies, 1971), Davies T. H. and Primrose E. J. F., 1971, *An Algebra for the Screw Systems of Pairs of Bodies in a Kinematic Chain*, Proc. 3rd World Cong. for the Theory of Machines and Mechanisms, Kupari (Yugoslavia), pp.199-212.

(Davies, 1983a), Davies, T. H., 1983, *Mechanical Networks - I: Passivity and Redundancy*, Mechanism and Machine Theory, Vol. 18, n. 2, pp. 95-101.

(Davies, 1983b), Davies, T. H., 1983, *Mechanical Networks - II: Formula for the Degree of Mobility and Redundancy*, Mechanism and Machine Theory, Vol. 18, no. 2, pp. 103-106.

(Davies, 1983c), Davies, T. H., 1983, *Mechanical Networks - III: Wrenches on Circuit Screws*, Mechanism and Machine Theory, Vol. 18, n. 2, pp. 107-112.

(Ding, 2005), Ding, Y., Jin, J., Ceglarek, D., Shi, S., 2005, *Process-oriented Tolerancing for Multi-Station Assembly Systems*, Trans. of IIIE on Design and Manufacturing, Vol. 37, pp. 493-508.

(Falgarone, 2006), Falgarone, U., Chevassus, N., 2006, *Structural and Functional Analysis for Assemblies*, Springer, Advances in Design, DOI: 10.1007/1-84628-210-1.

(Giordano, 2005a), Giordano, M., Samper, S., Petit J. P., 2005, *Tolerance Analysis and Synthesis by means of Deviation Domains, Axi-Symmetric Cases*, Proc. of 9th CIRP Int. Seminar on Computer-Aided Tolerancing, Tempe, Arizona, USA, April 10-12.

(Giordano, 2005b), Giordano, M., Pairel, E., Hernandez, P., 2005, *Complex Mechanical Structure Tolerancing by means of Hyper-graphs*, Proc. of 9th CIRP Int. Seminar on Computer-Aided Tolerancing, Tempe, Arizona, USA, April 10-12.

(Giordano, 2007), Germain, F., Giordano, M., 2007, *A New Approach for three-Dimensional Statistical Tolerancing*, Proc. of 10th CIRP Int. Seminar on Computer-Aided Tolerancing, Erlangen, Germany, March 21-23.

(Gogu, 2005), Gogu, G., 2005, *Mobility of Mechanisms: a Critical Review*, Mechanism and Machine Theory Vol. 40, pp. 1068-1097.

(Gupta, 1991), Gupta, S., Turner J, U, 1991, *Variational Solid Modeling for Tolerance Analysis*, Trans. of IEEE Computer Graphics & Applications, pp. 64-74.

(Hu, 1998), Hu, S. J., 1998, *Stream-of-Variation Theory for Automotive Body Assembly*, CIRP Annals, Vol. 46, pp. 1-6.

(Hu, 2001), Hu, M., Lin, Z., Lai, X., Ni, J., 2001, *Simulation and Analysis of Assembly Processes Considering Compliant, Non-Ideal Parts and Tooling Variations*, Journal of Machine Tools and Manufacture, Vol. 41, pp. 2233-2243.

References

- (Hong, 2002), Hong, Y. S., Chang, T. C., 2002, *A Comprehensive Review of Tolerancing Research*, Journal of Production Research, Vol. 40, pp. 2425-2459.
- (Huang, 2002), Huang, W., Ceglarek, D., 2002, *Decomposition Mode-based off Share Form Error by Discrete Cosine Transformation with Implementation to Assembly and Stamping Systemes with Compliant Shares*, Annals of CIRP, Vol. 51, pp. 21-26.
- (Huang, 2004a), Huang, W., Ceglarek, D., Zhou, Z., 2004, *Tolerance Analysis for Design of Multistage Manufacturing Processes Using Number-Theoretical Net Method (NT-net)*, Journal of Flexible Manufacturing Systems, Vol. 16, pp. 65-90.
- (Huang, 2004b), Huang, W., 2004, *Methodology for Modeling and Analysis of Stream of Variation in Compliant and Rigid Assemblies*, PHD Dissertation, University of Wisconsin-Madison, USA.
- (Huang, 2007a), Huang W., Lin, J., Bezdecny, M., Kong Z., Ceglarek, D., 2007, *Stream-of-Variation Modeling – Part I: A Generic 3D Variation Model for Rigid Body Assembly in Single Station Assembly Processes*, ASME Journal of Manufacturing Science Engineering, Vol. 129, pp. 821-831.
- (Huang, 2007b), Huang W., Lin, J., Kong Z., Ceglarek, D., 2007, *Stream-of-Variation Modeling – Part II: A Generic 3D Variation Model for Rigid Body Assembly in Multi Station Assembly Processes*, ASME Journal of Manufacturing Science Engineering, Vol. 129, pp. 832-842.
- (Huang, 2009), Huang, W., Kong, Z., 2009, *Simulation and Integration of Geometric and Rigid Body Kinematics Errors for Assembly Variation Analysis*, Journal of Manufacturing Systems, Vol. 27, pp. 36-44.
- (Kandikjan, 2003), Kandikjan, T., Shah J. J., Davidson J. K., 2003, *A Mechanism for Validating Dimensioning and Tolerancing Schemes in CAD Systems*, Computer-Aided Design, Elsevier, Vol. 33, pp. 721-737.
- (Kim, 2004), Kim, J. S., Kim K. M., Lee, J., Jung, H. B., 2005, *Solving 3D Geometric Constraints for Closed-Loop Assemblies*, ASME Journal on Advanced Manufacturing Technologies, Vol. 23, pp. 755-761.
- (Kim, 2005a), Kim, J. S., Kim, K. M., Choi, K., Lee, J., 2005, *Solving 3D Geometric Constraints for Assembly Modelling*, ASME Journal on Advanced Manufacturing Technologies, Vol. 16, pp. 843-849.
- (Kim, 2005b), Kim, J. S., Kim K. M., Lee, J., Jeong, J., 2005. *Generation of Assembly Models from Kinematic Constraints*, ASME Journal Advanced Manufacturing Technologies, Vol. 16, pp. 131-137.
- (Kong, 2009), Kong, Z., Huang, W., Oztekin, A., 2009, *Variation Propagation Analysis for Multi-station Assembly Process With Consideration of GD&T Factors*,

References

ASME Journal of Manufacturing Science and Engineering, Vol. 131, pp. 1-10.

(Kramer, 1992), Kramer, G. A., 1992, *Solving Geometric Constraint System: A Case Study in Kinematics*, MIT Press.

(ISO 1101, 2004), ISO 1101, Geometrical Product Specifications (GPS) - Geometrical Tolerancing - Tolerances of form, orientation, location and run-out.

(ISO 16792, 2006), ISO 16792, Technical Product Documentation - Digital Product Definition Data Practices.

(Jian, 2005), Jian, A. D., Ameta, G., Davidson, J. K., Shah, J. J., 2005, *Tolerance Analysis and Allocation using Tolerance-Maps for a Power Saw Assembly*, Proc. of 9th CIRP Int. Seminar on Computer-Aided Tolerancing, Tempe, Arizona, USA, April 10-12.

(Jin, 1999), Jin, J., Shi, J., 1999, *State Space Modeling of Sheet Metal Assembly for Dimensional Control*, ASME Journal of Manufacturing Science and Engineering, Vol. 121, pp. 756-762.

(Lengyel, 2003), Lengyel, E., 2003, *Mathematics for 3D Game Programming and Computer Graphics*, Charles River Media, 2nd Edition.

(Liao, 2007), Liao, X., Wang, G. G., 2007, *Non-linear Dimensional Variation Analysis for Sheet Metal Assemblies by Contact Modeling*, Finite Elements in Analysis and Design, Vol. 44, pp. 34-44.

(Liu, 1997), Liu, C. S., Hu, J. S., 1997, *Variation Simulation for Deformable Sheet Metal Assemblies Using Finite Element Methods*, ASME Journal of Manufacturing Science and Engineering, Vol. 119, pp. 368-374.

(Long, 1998), Long, Y., Hu, S. J., 1998, *A Unified Model for Variation Simulation of Sheet Metal Assemblies*, Geometric Design Tolerancing: Theories, Standards and Applications, Dr. Hoda A. ElMaraghy, Chapman and Hall.

(Loose, 2009), Loose, J. P., Zhou, Q., Zhou, S., Ceglarek, D., 2009, *Integrating GD&T into Dimensional Variation Models for Multistage Machining Processes*, Journal of Production Research, Vol. 1, pp. 1-21.

(Matripragada, 1998a), Matripragada, R., Whitney, D.E., 1998, *The Datum Flow Chain: A Systematic Approach to Assembly Design and Modeling*, Research in Engineering Design, Vol. 10, pp. 150-165.

(Matripragada, 1998b), Matripragada, R., 1998, *Assembly Oriented Design: Concepts, Algorithms and Computational Tools*, PHD Dissertation, MIT, Cambridge, MA.

(Matripragada, 1999), Matripragada, R., Whitney, D. E., 1999, *Modeling and Controlling Variation Propagation in Mechanical Assemblies using State Transition Mod-*

References

els, Trans. of IEEE on Robotics and Automation, Vol. 15:1, pp. 124-140.

(Meadows, 1997), Meadows, J. D., 1997, *Geometric Dimensioning and Tolerancing, Applications and Techniques for Use in Design, Manufacturing, and Inspection*, ISBN 0-8247-9309-9.

(Merkley, 1998), Merkley, K., 1998, *Tolerance Analysis of Compliant Assemblies*, PHD Dissertation, Brigham Young University, Utah.

(Mortensen, 2002), Mortensen, A. J., 2002, *An Integrated Methodology for Statistical Tolerance Analysis of Flexible Assemblies*, MS Thesis, Brigham Young University, Utah.

(Pasupathy, 2003), Pasupathy, T. M. K., Morse, E. P., Wilhelm, R. G., 2003, *A Survey of Mathematical Methods for the Construction of Geometric Tolerance Zones*, ASME Journal of Computing and Information Science in Engineering, Vol. 3, pp. 64-75.

(Raffin, 1998), Raffin, R., Neveu, M., Derdouri, B., 1998, *Constrained Deformation for Geometric Modelling and Object Reconstruction*, Proc. Winter School Computer Graphics, pp. 299-306.

(Raffin, 2000), Raffin, R., Neveu, M., Jaar, F., 2000, *Curvilinear Displacement of Free-Form-based Deformation*, The Visual Computer, Vol. 16, pp. 38-46.

(Requicha, 1986), Requicha, A. A. G., Chan, S. C., 1986, *Representation of Geometric Features, Tolerances and Attributes in Solid Modelers Based on Constructive Geometry*, Trans. of IEEE Robotic Automation, Vol.2, pp. 156-166.

(Rotondi, 2005), Rotondi, A., Pedroni, P., Pievatolo, A., 2005, *Probability, Statistics and Simulation*, 2nd Edition, ISBN 978-88-470-0262-3 (in Italian).

(Samper, 2007), Samper, S., Formosa, F., 2007, *Form Defect Tolerancing by Natural Modes Analysis*, ASME Journal of Computing and Information Science in Engineering, Vol. 7, pp. 44-51.

(Sederberg, 1986), Sederberg, T. W., Berry, S. R., 1986, *Free Form Deformation of Solid Geometric Models*, Trans. of ACM, Vol. 20, pp. 18-22.

(Sellem, 1998), Sellem, E., Rivière, A., 1998, *Tolerance Analysis of Deformable Assemblies*, Proc. of DETC98, ASME Design Engineering Technical Conference, Atlanta GA.

(Sellem, 1999), Sellem, E., Rivière, A., Hillerin, C. A. D., Clement, A., 1999, *Validation of the Tolerance Analysis of Compliant Assemblies*, ASME Design Engineering Technical Conference, Las Vegas, Nevada, September 12-15.

(Sellem, 2001), Sellem, E., Sellakh, R., and Rivière, A., 2001, *Testing of Tolerance*

References

Analysis Module for Industrial Interest, Proc. of 7th CIRP Int. Seminar on Computer Aided Tolerancing, ENS de Cachan, France, April 24-25.

(Shah, 2007), Shah, J. J., Ameta, G., Shen, Z., Davidson, J. K., 2007, *Navigating the Tolerance Analysis Maze*, Computer-Aided Design & Applications, Vol. 4, pp. 705-718.

(Shen, 2003), Shen, Z., 2003, *Tolerance Analysis with EDS/VisVSA*, ASME Journal of Computing and Information Science in Engineering, Vol. 5, pp. 95-99.

(Shen, 2005), Shen, Z., Ameta, G., Shah, J. J., Davidson J. K., 2005, *A Comparative Study of Tolerance Analysis Methods*, ASME Journal of Computing and Information Science in Engineering, Vol. 5, pp. 247-256.

(Shiu, 1996), Shiu, B. W., Ceglarek, D., Shi, J., 1996, *Multi-Station Sheet Metal Assembly Modeling and Diagnostic*, Trans. of NAMRI/SME, Vol. 24, pp. 199-204.

(Shiu, 1997), Shiu, B. W., Ceglarek, D., Shi, J., 1997, *Flexible Beam-based Modeling of Sheet Metal Assembly for Dimensional Control*, Trans. of NAMRI, Vol. 25, pp. 49-54.

(Shukla, 2001a), Shukla, G., Whitney, D.E., 2001, *Systematic evaluation of constraint properties of Datum Flow Chain*, Trans. of IEEE ISATP, Fukuoka, Japan.

(Shukla, 2001b), Shukla, G., 2001, *Augmenting Datum Flow Chain Method to Support the Top-Down Design Process for Mechanical Assemblies*, MS Thesis, MIT Cambridge MA.

(Soderberg, 2006), Soderberg, R., Lindkvist, L., Dahlström, S., 2006, *Computer-Aided Robustness Analysis for Compliant Assemblies*, Journal of Engineering Design, Vol. 17, pp. 411-428.

(Soderberg, 2008), Soderberg, R., Wickman, C., Lindkvist, L., 2008, *Improving Decision Making by Simulating and Visualizing Geometrical Variation in non-Rigid Assemblies*, CIRP Annals, Manufacturing Technology, Vol. 57, pp. 175-178.

(Srinivasan, 1997), Srinivasan, R. S., Wood, K. L., 1997, *A Form Tolerancing Theory using Fractals and Waves*, ASME Journal of Mechanical Design, Vol. 119, pp. 185-193.

(Stewart, 2004), Stewart, L. M., 2004, *Variation Simulation of Fixtured Assembly Processes for Compliant Structures using Piecewise-Linear Analysis*, PHD Dissertation, Brigham Young University, Utah.

(Strang, 2009), Strang, G., 2009, *An Introduction to Linear Algebra*, Wellesley Cambridge Press, 4th edition, ISBN 978-09802327-14.

(Tanaka, 2001), Tanaka, F., Murai, M., Kishinami, T., Tokunaga, H., 2001, *Con-*

References

straint Reduction based on a Lie Algebra for Kinematic Analysis of Assembly, Trans. of IEEE on Robotics and Automation, pp. 741-750.

(Tonks, 2002), Tonks, M., 2002, *A Robust Geometric Covariance Method for Flexible Assembly Tolerance Analysis*, MS Thesis, Brigham Young University, Utah.

(Turner, 1992), Turner, J. U., Subramaniam, S., Gupta, S., 1992, *Constraint Representation and Reduction in Assembly Modelling and Analysis*, Trans. of IEEE on Robotics and Automation, Vol. 8, pp. 741-750.

(Ungemach, 2009), Ungemach, G., Mantwill, F., 2009, *Efficient Consideration of Contact in Compliant Assembly Variation Analysis*, ASME Journal of Manufacturing Science and Engineering, Vol. 131, pp. 1-9.

(Waldron, 1966), Waldron, K. J., 1966, *The Constraint Analysis of Mechanisms*, Journal of Mechanisms, Vol. 1, pp. 101-114.

(Wang, 2009), Wang, H., Ceglarek, D. 2009, *Variation Propagation Modeling and Analysis at Preliminary Design Phase of Multi-Station Assembly Systems*, Assembly Automation, Vol. 29, pp. 154-166.

(West, 2001), West, D. B., 2001, *An Introduction to Graph Theory*, 3rd edition, Prentice Hall.

(Whitney, 1994), Whitney, D. E., Gilbert, O. L., Jastrzebski, M., 1994, *Representation of Geometric Variations Using Matrix Transformation for Statistical Tolerance Analysis in Assembly*, Research in Engineering Design, Vol. 6, pp. 191-210.

(Whitney, 1999), Whitney, D. E., Mantripragada, R., Adams, J. D., Rhee, S. J., 1999, *Designing Assemblies*, Research in Engineering Design, Vol. 11, pp. 229-253.

(Whitney, 2004), Whitney D. E., 2004, *Mechanical Assemblies: their Design, Manufacture, and Role in Product Development*, Oxford University Press, New York.

(Whitney, 2008), Whitney, D. E., 2008, *Proper Constraint as a Design Principle of Assemblies*, Assembly Automation, Vol. 28, pp. 120-125.

(Wriggers, 2002), Wriggers, P., 2002, *Computational Contact Mechanics*, Wiley, New York.

(Xie, 2007), Xie, K., Wells L., Camelio J. A., Youn, B. D., 2007, *Variation Propagation Analysis on Compliant Assemblies Considering Contact Interaction*, ASME Journal of Manufacturing Science and Engineering, Vol. 129, pp. 934-942

(Zienkiewicz, 2005), Zienkiewicz, O. C., Taylor R. L., Zhu J. Z., 2005, *The Finite Element Method, its Basis and Fundamentals*, Elsevier, 6th Edition.