

**DESIGN AND ANALYSIS OF MECHANICAL ASSEMBLY VIA
KINEMATIC SCREW THEORY**

A Dissertation

Presented in Partial Fulfillment of the Requirements for
the Degree Doctor of Philosophy in the
Graduate School of The Ohio State University

By

Leonard Priyatna Rusli, M.S.

The Ohio State University

2008

Dissertation Committee:

Dr. Anthony Luscher, Co-adviser

Dr. James Schmiedeler, Co-adviser

Dr. Gary Kinzel

Dr. Krishnaswamy Srinivasan

Approved by

Co-adviser

Co-adviser

Graduate Program in Mechanical Engineering

Copyright by

Leonard Priyatna Rusli

2008

ABSTRACT

The essential function of a mechanical assembly is the removal of degrees of freedom (DOF) and transfer of load between two bodies. Assemblies using integral attachments are composed of unilateral mating surfaces, where quality is greatly affected by the location and orientation of assembly features. Feature-level design is concerned with the dimension, stress, and strain of individual assembly features. This dissertation is concerned with attachment-level design, where design decisions are made on the type, location, and orientation of assembly features.

Previous research in theoretical kinematics, robotic grasping, and fixture design have produced either a binary test for form closure or design optimization for a specific loading condition. There is currently no tool available to: (1) analyze an assembly's quality with a quantitative metric and (2) optimize the design of the assembly constraint configuration (location and orientation of features) to resist motion effectively. Therefore, the objective of this dissertation is to develop an analysis and design tool to address these needs.

The analysis tool models the assembly features as wrench systems. The point, pin, line, and plane constraints in assembly are modeled with equivalent first, second, and

third order wrench systems. The methodology used is based on composing a five-system pivot wrench combination to which a screw motion is reciprocal. The resistance effectiveness of each constraint to these motions is calculated as the ratio of the reaction forces at each resisting constraint to the input wrench magnitude. Based on these individual resistance values, a set of rating metrics is calculated to evaluate an assembly's quality from different perspectives. A design tool based on this analysis methodology is developed to optimize assembly design by constraint modification, constraint reduction, and constraint addition. A set of case studies is used to verify commonly known design principles, explore the design space of attachment, and understand trade-offs in assembly constraint redundancy and resistance quality.

The main contributions of this dissertation are: (1) an analysis tool that is able to model assembly as kinematic constraints and calculate the load amplification ratio in the form of different rating metrics to measure assembly quality, (2) a design tool that is able to explore the design space and find optimal solutions for improving constraint effectiveness and optimize the number of constraints, and (3) an understanding of how constraint modification, reduction, and addition affect the quality of an assembly.

Dedicated to my parents

ACKNOWLEDGMENTS

I would like to thank Dr. Anthony Luscher for his direction, insight, and advice throughout my graduate school. I learned so much about engineering design philosophy, practice, teaching, and research through working together for so many years. I'm also thankful for his personal encouragement, sincere support, and friendship along the path.

I would like to thank Dr. James Schmiedeler for giving substantial knowledge and advice in kinematics and screw theory and for his commitment in reviewing my dissertation.

I would like to thank Dr. Krishnaswamy Srinivasan and Dr. Gary Kinzel for being in my committee, the financial support through my Ph.D. program, and the opportunity to coordinate the capstone design program in the department, which I really enjoy.

I would like to thank my wife for being a faithful friend, for better or for worse, in good and bad times. I would like to thank my parents, brother, and sister for their love, sacrifice, and for giving me the privilege of studying in the United States.

Above all else, I thank Jesus Christ for saving me and giving me the meaning of true and abundant life as he promised, and for all my dear friends and fellow leaders in the Indonesian Christian Fellowship with whom I experience what this life is all about.

VITA

August 17 th , 1978	Born – Jakarta, Indonesia
December, 2000	B.S., Mechanical Engineering, The Ohio State University
March, 2003	M.S., Mechanical Engineering, The Ohio State University
September 2001-present	Graduate Teaching and Research Associate, The Ohio State University

PUBLICATIONS

- Rusli, L., Luscher, A., Sommerich, C., A Study of the Effect of Force and Tactile Feedback to Snap-fit Manual Assembly, Under Review for International Journal of Industrial Ergonomics, 2008.
- Rusli, L., Luscher, A., Evaluation of Rapid Prototyping Methods for Functional Testing in Snap-fits, ANTEC conference paper, 2001.

FIELDS OF STUDY

Major field: Mechanical Engineering

TABLE OF CONTENTS

Abstract	ii
Acknowledgments.....	v
Vita.....	vi
List of Tables	xiii
List of Figures	xv
CHAPTER 1 Introduction to assembly design	1
1.1 Threaded fastener vs. integral attachment features	1
1.2 Feature level design vs. attachment level design	4
1.3 Assembly design and analysis tool	7
1.4 Dissertation overview	7
CHAPTER 2 Literature review.....	10
2.1 Research in theoretical kinematics, screw theory, and extensions of screw theory	11
2.2 Research in analysis and synthesis of robotic grasping	11
2.2.1 Form closure vs. force closure	11
2.2.2 Form and force closure test.....	14
2.2.3 Force distribution analysis	17
2.2.4 Quantitative metric for constraint configuration.....	19
2.3 Research in analysis and synthesis of fixture design	21
2.4 Research in assembly design	31
2.5 Research on constraint analysis in assembly	36
2.6 Summary	37
CHAPTER 3 Overview of screw theory and applications in constraint analysis	40
3.1 Overview of screw theory	40

3.2 Kinematic methods for fixture configuration planning by Bausch and Youcef-Toumi [7]	46
3.3 Constraint analysis of assembly by Bozzo [9]	49
3.4 Two major limitations of Bausch and Bozzo's model.....	52
3.4.1 Modeling of line and planar constraints.....	52
3.4.2 Inconsistent scaling between finite pitch screw motion and pure translation	55
3.5 Other minor issues with Bausch and Bozzo's model.....	57
3.5.1 Inaccurate post-processing of rating matrix.....	58
3.5.2 Failure to calculate ratings for bilateral motion separately.....	59
3.6 Objectives and Scope of Research.....	60
3.6.1 Assembly constraint analysis	61
3.6.2 Assembly Constraint Synthesis / Design	62
CHAPTER 4 Modeling assembly constraints using higher order constraint model	64
4.1 From contacts into constraints	64
4.2 Higher order constraints (HOC) modeling	65
4.2.1 Modeling point contact as a one-system	66
4.2.2 Modeling pin contact as a two-system.....	66
4.2.3 Modeling line contact as a two-system	68
4.2.4 Modeling plane contact as a three-system	69
4.2.5 Modeling other assembly constraints.....	71
4.3 Modeling gaps between screw theory and assembly constraint	73
4.3.1 Bidirectional vs. unidirectional DOF removal.....	73
4.3.2 Infinite span vs. finite span length	74
4.4 Reciprocal screw problem solution.....	75
4.5 Advantages over previous models	76
4.6 Summary	78
CHAPTER 5 Evaluation of motion resistance quality using isolated reaction force model	79
5.1 Virtual penetration model vs. isolated reaction force model	79
5.2 The isolated reaction force model.....	81
5.3 Input wrench magnitude specification	85
5.3.1 Solution for inconsistent units of the input wrench	85

5.3.2 Solution for the pitch discontinuity problem	87
5.4 The resistance value calculation	91
5.5 Additional advantages.....	93
5.6 Reaction wrench composition.....	94
5.7 Summary	95
CHAPTER 6 Assembly performance rating Metrics.....	96
6.1 Rating matrix composition.....	96
6.2 Weakest total resistance (WTR) and the most weakly constrained motion.....	100
6.3 Mean total resistance (MTR)	102
6.4 Mean redundancy ratio (MRR).....	102
6.5 Trade-off ratio (TOR)	104
6.6 Constraint activity and best resistance percentage.....	104
6.7 Summary	105
CHAPTER 7 Constraint analysis tool algorithm and implementation	107
7.1 Main algorithm.....	107
7.2 Pre-processing.....	110
7.2.1 Input file format	110
7.2.2 Input file checker	111
7.2.3 Constraint set transformation to wrench coordinates.....	111
7.2.4 Combination composition of five-system wrench	113
7.3 Main processor.....	114
7.3.1 Pivot wrench set composition and linear independence check	115
7.3.2 Solution for the motion reciprocal to the pivot wrench	116
7.3.3 Separation of forward and backward reciprocal motion.....	117
7.3.4 Removal of duplicate motion.....	118
7.3.5 Input wrench specification	119
7.3.6 Reaction wrench set up	120
7.3.7 Solution to the equilibrium equation.....	126
7.4 Post processing.....	127
7.4.1 Composing the rating matrix	127
CHAPTER 8 Assembly design study	128
8.1 Objectives of the assembly design study	128

8.2 Objective #1: Confirm design principles in assembly using the design tool.....	129
8.3 Objective #2: Trade-off study between assembly resistance quality and redundancy.....	131
8.4 Objective #3: Explore design search space and identify possible design improvements.....	132
8.5 Sensitivity analysis and specified load test	133
8.6 Case study approach	135
8.7 Summary	137
CHAPTER 9 Design study tool algorithm.....	139
9.1 Constraint modification	139
9.1.1 Types of design search space and optimization variable specification.....	140
9.1.2 Factorial search methodology in design space exploration	142
9.1.3 Constraint modification pre-processing.....	143
9.1.4 Incremental constraint modification	145
9.1.5 Recalculation for base motion set.....	145
9.1.6 Recalculation for new motion set	146
9.1.7 Constraint modification post-processing and optimization plots.....	146
9.1.8 Constraint modification flowchart	146
9.2 Constraint reduction.....	147
9.2.1 Constraint reduction algorithm	148
9.3 Constraint addition.....	149
9.3.1 Constraint addition algorithm	150
9.4 Specified loading condition optimization	150
9.5 Sensitivity analysis study	151
9.6 Summary	152
CHAPTER 10 Case study baseline analysis results and model verification	153
10.1 Exactly constrained geometry - Thompson's chair	153
10.1.1 Baseline analysis results	155
10.2 Simple 3D shape geometry - generic cube	159
10.2.1 Baseline analysis results	162
10.2.2 Scalability test case study	163
10.3 Rectilinear geometry - battery cover assembly.....	167
10.3.1 Baseline analysis results	170

10.4 Axisymmetric geometry - end cap assembly	173
10.4.1 Baseline results	176
10.4.2 HOC model vs. non-HOC model.....	177
10.5 Freeform/non-planar geometry - printer housing assembly.....	181
10.5.1 Baseline analysis results	185
10.5.2 Snap-fits vs. threaded fasteners.....	187
CHAPTER 11 Design optimization and trade-off case study results	188
11.1 Exactly constrained geometry - Thompson's chair	189
11.1.1 One-dimensional optimization of the height.....	189
11.1.2 Two-dimensional optimization of the height and trihedral angle.....	190
11.2 Simple 3D shape geometry - generic cube	192
11.2.1 Constraint addition trade-off case study	192
11.2.2 Constraint reduction trade-off case study	195
11.3 Simple rectilinear geometry – planar battery cover.....	201
11.3.1 Design principle test case study – leverage	201
11.3.2 Design principle test case study – symmetry	204
11.3.3 Study type: Design principle test – optimal line-of-action	207
11.4 Axisymmetric geometry - end cap assembly	209
11.4.1 Design optimization case study – constraint addition.....	209
11.4.2 Design optimization case study – snap-fit location	211
11.5 Freeform/non-planar geometry - printer housing assembly.....	214
11.5.1 Design optimization – threaded fastener orientation search	215
11.5.2 Design optimization – threaded fastener location.....	217
11.5.3 Design optimization – snap-fit orientation	220
11.5.4 Design optimization – snap-fit location.....	222
11.5.5 Design optimization – parting line.....	224
11.5.6 Design optimization – line constraint length	227
11.5.7 Synthesis of improved design	230
11.5.8 Design optimization - known loading direction.....	236
11.5.9 Design optimization – constraint reduction	237
11.5.10 Sensitivity analysis.....	240
CHAPTER 12 Conclusions and future work	242

12.1 Research contributions.....	242
12.2 Development of assembly domain-specific knowledge	246
12.2.1 Assembly performance analysis	246
12.2.2 Assembly resistance quality and redundancy trade-off	248
12.2.3 Assembly design optimization.....	250
12.3 Application guideline in using the analysis and design tool in assembly design	252
12.4 Future work to address model and methodology limitation	254
12.4.1 Incorporating stiffness into the analysis model	255
12.4.2 Bifurcation problem in linear dependency.....	257
12.4.3 HOC modeling guidelines.....	258
12.5 Future work to expand the functionality of the analysis and design tool	260
12.5.1 Improvements in the accuracy, efficiency, and capacity of the optimization search	261
12.5.2 Additional features to expand functionality of the design and analysis tool	263
12.6 Future work in understanding assembly design and attachment strategy	264
List of references.....	266
APPENDICES	271
APPENDIX A MATLAB scripts.....	272
APPENDIX B Case study geometries	295

LIST OF TABLES

Table 1.1 Contrast table between threaded fasteners and integral attachments.....	2
Table 2.1 Comparison of Kinematic Design vs. Elastic Averaging Principles	33
Table 3.2. Example of Rating Scheme Inconsistency.....	58
Table 5.1 Constraint Effectiveness Metric.....	80
Table 6.1 Sample Rating Matrix.....	98
Table 7.1 Constraint input file format.....	110
Table 7.2 Wrench transformation formulas for each constraint type	113
Table 7.3 A sample combination that causes duplicate motion.....	119
Table 8.1 List of case studies conducted	135
Table 9.1 Design space mapping for each constraint type.....	140
Table 10.1 Thompson's chair constraint configuration.....	155
Table 10.2 Thompson's chair baseline analysis results.....	155
Table 10.3 Rating matrix for Thompson's Chair (continued)	156
Table 10.4 Cube constraint configuration.....	161
Table 10.5 Cube constraint combination for different number of constraints	161
Table 10.6 Cube baseline analysis result (7 constraints)	162
Table 10.7 Cube scalability test result comparison	163
Table 10.8 Cube scalability test – motion pitch comparison	164
Table 10.9 Battery cover baseline constraint configuration	168
Table 10.10 Battery cover baseline analysis result	170
Table 10.11 End cap constraint configuration	174
Table 10.12 End cap baseline analysis results	176
Table 10.13 End cap assembly constraint configuration (non-HOC model).....	178
Table 10.14 Comparison of HOC vs. non-HOC model analysis results	179
Table 10.15 Sample motion to illustrate pseudo-redundancy.....	180
Table 10.16 Printer housing constraint configuration.....	183
Table 10.17 Printer housing baseline analysis results	185
Table 10.18 Comparison between snap-fit and threaded fasteners.....	187
Table 11.1 Optimization variables and search space for Thompson's chair height	189
Table 11.2 Optimization variables and search space for Thompson's chair height and trihedral angle	191
Table 11.3 Cube constraint combinations for different number of constraints	193
Table 11.4 Overall rating increase as constraints are added	193
Table 11.5 Overall rating change as more constraints are removed	198

Table 11.6 Comparison between baseline and improved 7 constraint cube	200
Table 11.7 Optimization variables and search space for leverage case study	202
Table 11.8 Battery cover symmetry case study point constraint	204
Table 11.9 Optimization variables and search space for symmetry case study	205
Table 11.10 Optimization variables and search space for line-of-action case study	207
Table 11.11 Rating change as the number of snap-fits is increased	210
Table 11.12 Optimization variables and search space for snap-fit location	212
Table 11.13 Optimization variables and search space for fastener orientation	216
Table 11.14 Optimum design for fastener orientation search space	217
Table 11.15 Optimization variables and search space for fastener location.....	218
Table 11.16 Comparison between assembly design with 4 vs. 2 fasteners	220
Table 11.17 Optimization variables and search space for snap-fit orientation.....	220
Table 11.18 Optimum design for snap-fit orientation search space	222
Table 11.19 Optimization variables and search space for snap-fit location	223
Table 11.20 Optimization variables and search space for parting line	225
Table 11.21 Comparison of non-planar parting line vs. planar parting line design.....	227
Table 11.22 Optimization variables and search space for line size (CLIN5, CLIN6)....	228
Table 11.23 Optimization variables and search space for line size (CLIN1-CLIN4)	228
Table 11.24 Summary of design study conclusions.....	231
Table 11.25 Summary of design recommendation for each design revision	232
Table 11.26 Simultaneous search optimum in the design search space	233
Table 11.27 Summary of design alternative ratings	235
Table 11.28 Effect of constraint reduction to assembly rating for printer housing	238
Table 11.29 Comparison of rating between baseline and alternate designs for constraint removal (printer housing)	240
Table 11.30 Sensitivity analysis result for printer housing.....	241

LIST OF FIGURES

Figure 1.1 An example of assembly with various integral attachment features	3
Figure 1.2 An example of assembly with highly redundant integral attachment features..	4
Figure 1.3 Feature level design vs. attachment level design.....	5
Figure 1.4 Topical map of the dissertation	8
Figure 2.1 Examples of passive force closure [60].....	13
Figure 2.2 Flat object grasping example by Salisbury and Roth [42]	16
Figure 2.3 An example of deficiencies in Trinkle's rating scheme [9]	20
Figure 2.4 Example of machining fixture contact by DeMeter [15].....	23
Figure 2.5 Example of modeling machining fixture contact as constraints by DeMeter [15]	24
Figure 2.6 Optimal clamps for part subject to load <i>we2</i>	28
Figure 2.7 A planar example on space of detaching motions [43]	30
Figure 2.8 Poor vs. improved constraint	38
Figure 3.1 The geometry of the velocity field of points in a rigid body [49]	42
Figure 3.2 Graphical representation of the screw Axis [6]	48
Figure 3.3. Motion stop value calculation for contact at C [7]	49
Figure 3.4 An example of representing pin contact as many unilateral contacts	53
Figure 3.5 Relationship between number of motions and number of unilateral constraints	54
Figure 3.6 Screw input magnitude vs. pitch	56
Figure 5.1 Poor vs. good motion resistance	80
Figure 5.2 Passive force closure model	82
Figure 5.3 Isolated reaction force model equilibrium.....	84
Figure 5.4 The input wrench as a force couple.....	86
Figure 5.5 Input wrench magnitude vs. (a) screw pitch and (b) wrench pitch	91
Figure 6.1 Sample total resistance histogram that shows random spread.....	99
Figure 6.2 Sample histogram that shows grouping of motion total resistance	100
Figure 7.1 Main algorithm flowchart.....	109
Figure 7.2 Partial linear dependence example between (a) plane and line constraint and (b) the associated wrench systems	116
Figure 7.3 Pin constraint reaction wrench calculation.....	122
Figure 7.4 Line constraint reaction wrench calculation.....	123
Figure 7.5 Rectangular plane constraint reaction wrench calculation	124
Figure 7.6 Circular plane constraint reaction wrench calculation	126

Figure 8.1 Effect of dimensional accuracy on constraint configuration (example 1).....	134
Figure 8.2 Effect of dimensional accuracy on constraint configuration (example 2).....	134
Figure 9.1 Illustration of the (a) line, (b) curved line, and (c) plane search space	142
Figure 9.2 Simplified flowchart for constraint modification algorithm	147
Figure 10.1 Thompson's chair constraint configuration and WTR motion	154
Figure 10.2 Thompson's chair total resistance histogram.....	159
Figure 10.3 Generic cube constraint configuration.....	160
Figure 10.4 Total resistance histogramn for cube (scale factor = 1)	166
Figure 10.5 Total resistance histogramn for cube (scale factor = 2)	167
Figure 10.6 Battery cover constraint configuration	168
Figure 10.7 Total resistance histogram for battery cover	172
Figure 10.8 Endcap geometry and constraint configuration	173
Figure 10.9 Total resistance histogram for end cap.....	177
Figure 10.10 End cap total resistance histogram (non-HOC model).....	179
Figure 10.11 Printer housing geometry and assembly constraints	182
Figure 10.12 Total resistance histogram for printer housing assembly	186
Figure 11.1 Optimization plot for Thompson's chair height variable	190
Figure 11.2 Optimization plot for Thompson's chair height and CP3 angle variable	192
Figure 11.3 Overall rating increase as constraints are added.....	194
Figure 11.4 Rating change due to constraint removal (one at a time)	196
Figure 11.5 Rating change due to constraint removal (two at a time)	197
Figure 11.6 Overall rating change as constraints are removed	199
Figure 11.7 Optimization variables and search space for leverage case study	202
Figure 11.8 Response surface plot for leverage design principle case study.....	203
Figure 11.9 Optimization variables and search space for symmetry case study	205
Figure 11.10 Response surface plot for symmetry design principle case study	206
Figure 11.11 Optimization variables and search space for line-of-action case study....	207
Figure 11.12 Response surface plot for line-of-action design principle case study	208
Figure 11.13 End cap snap-fit addition configuration	210
Figure 11.14 Rating change as the number of snap-fits is increased.....	211
Figure 11.15 Optimization variables and search space for snap-fit location	212
Figure 11.16 Response surface cplot for end cap snap-fit location optimization	213
Figure 11.17 Worst constraint configuration and its associated WTR motion.....	214
Figure 11.18 Optimization variables and search space for fastener orientation	215
Figure 11.19 Response surface plot for fastener orientation optimization	216
Figure 11.20 Optimization variables and search space for fastener location	218
Figure 11.21 Response surface plot for fastener location optimization.....	219
Figure 11.22 Optimization variables and search space for snap-fit orientation	221
Figure 11.23 Response surface plot for snap-fit orientation optimization.....	222
Figure 11.24 Optimization variables and search space for snap-fit location	223
Figure 11.25 Response surface plot for snap-fit location optimization	224
Figure 11.26 Optimization variables and search space for parting line.....	225
Figure 11.27 Response surface plot for parting line optimization.....	226
Figure 11.28 Optimization variables and search space for line size (CLIN5, CLIN6) ..	228

Figure 11.29 Optimization variables and search space for line size (CLIN1-CLIN4) ...	228
Figure 11.30 Response surface plot for line size optimization (CLIN5, CLIN6).....	229
Figure 11.31 Response surface plot for line size optimization (CLIN1-CLIN4)	230
Figure 11.32 Summary of design alternative ratings	235
Figure 11.33 Response surface plot for snap-fit location optimization (specified loading condition).....	237
Figure 11.34 Effect of constraint reduction to assembly rating for printer housing.....	238
Figure 12.1 Application guidelines in utilizing analysis and design tool	253
Figure B.1 Thompson's chair case study geometry.....	296
Figure B.2 Generic cube case study geometry.....	297
Figure B.3 Battery cover case study geometry	298
Figure B.4 End cap assembly case study geometry	299
Figure B.5 Printer housing assembly case study geometry	300

CHAPTER 1

INTRODUCTION TO ASSEMBLY DESIGN

1.1 Threaded fastener vs. integral attachment features

An unassembled product is without function. Assemblies, not the individual parts, make a product function. Assembly is critical because it brings together the roles of the individual parts and subassemblies. Assembly is essentially the removal of degrees of freedom (DOF) and transfer of loads between two bodies. This DOF removal is traditionally accomplished by fasteners such as nuts and bolts. In mechanical joining of plastic parts, it is very common to attach two mating pieces with what is often called a “snap-fit”. Snap-fits are popular because they provide distinct advantages over traditional mechanical fastening. Among them are reduced assembly time, complexity, and part count. This alternative DOF removal method would be integrated in the part’s overall geometry, and hence is called integral attachment features (IAF). Locking features such as snap-fits, locating pins, and general mating surfaces provide the constraints to remove DOF. Despite the advantages, IAF poses a different challenge to design because it is prone to design flaws in the attachment strategy. The following sections explain the reason for special challenges in designing mechanical part assembly using IAF.

In traditional threaded fasteners, a single feature provides removal of multiple DOF. For example, a single bolted joint removes 5 DOF. Multiple bolted joints tend to be highly redundant in resisting loads and rely heavily on friction and preload to maintain assembly strength. Theoretically, for simple part shapes, 3-4 bolted joints at the edges of the part may accomplish the DOF removal task.

This is generally not true for IAF, which tend to be effective in removing only one or two DOF. For example, a cantilever hook feature is only strong in a single tensile direction. It takes multiple features such as locating pins and other mating surfaces to eliminate all DOF. In addition to removing fewer DOF, most IAF restrain movement by point or surface contacts in one direction only. This is called the unilateral nature of IAF constraints, in contrast to the bilateral nature of kinematic linkages and threaded fasteners. In other words, a unilateral constraint only removes a ‘half’ DOF, as it is typically understood. Because of this, there are in essence a total of 12 unilateral DOF to remove. Table 1.1 summarizes the contrast between threaded fasteners and IAF.

	Threaded Fasteners	Integral Attachment Features
DOF removal for each feature	3 or more bilateral DOF	1-2 unilateral DOF
Relative redundancy	High	Low
Relies on	Fastener strength, preload and friction	Fastener strength, location and orientation

Table 1.1 Contrast table between threaded fasteners and integral attachments

Figure 1.1 shows a part with multiple feature types in order to eliminate DOF. The pin constraint is only effective in resisting translation in two directions. The tongue-and-groove feature is effective in resisting translation and rotational motion transverse to the line contact. The screw is similar with the pin constraint with additional translational

constraint along the axis of the screw. These different types of features are located and oriented in such a way that together they eliminate all possible DOF without relying on friction.

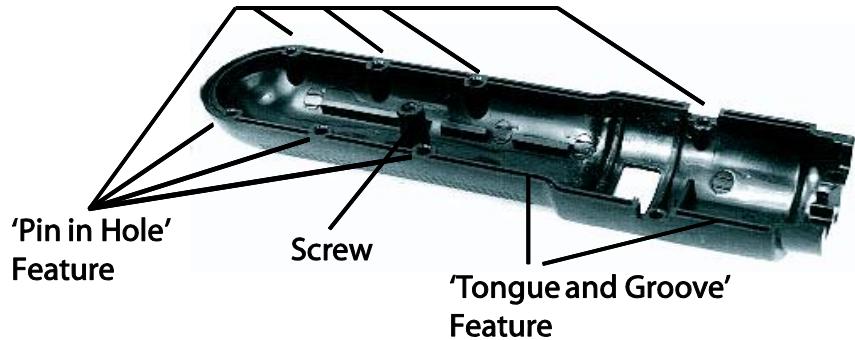


Figure 1.1 An example of assembly with various integral attachment features

Figure 1.2 shows an assembly with many cantilever hook features located in various locations and orientations. The result is a highly redundant design. Redundancy, while offering greater resistance capacity, is subject to locked-in stresses due to manufacturing variation. Redundant designs are often motivated by the fact that IAF's are usually not as stiff as threaded fasteners and can carry less load. IAF assembly can be very weak when a constraint needs to resist motion in its compliant direction. However, properly designed IAF assembly can also resist larger load than expected. This is done by placing an IAF so that it gains leverage and orienting it in its optimal (or stiffer) direction.

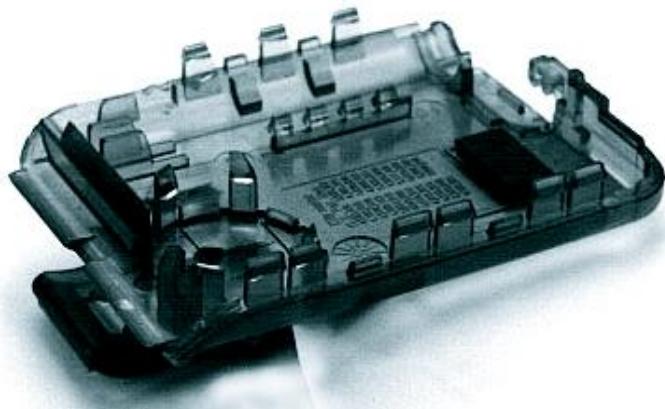


Figure 1.2 An example of assembly with highly redundant integral attachment features

There are then two concerns with IAF assemblies. First, since IAF assembly is less redundant than threaded fasteners, the assembly is prone to significant loss of resistance quality when one or two features fail. That is, there are fewer active constraints to resist a particular motion simultaneously. Second, the assembly quality relies heavily on feature location and orientation. If the location or orientation is ineffective, service loads can result in large reaction forces that cause IAF failure.

1.2 Feature level design vs. attachment level design

There are two levels of decision making in mechanical part assembly design: feature level and attachment level decisions (Figure 1.3). A feature level design goal revolves around determining design equations and optimization approaches for a single feature to meet design requirements such as maximum strength, strain or load capacity. The kinds of questions typically addressed in feature level design are: How strong is this

feature? How much force does it take to assemble? How much stress will this feature experience under a certain load?

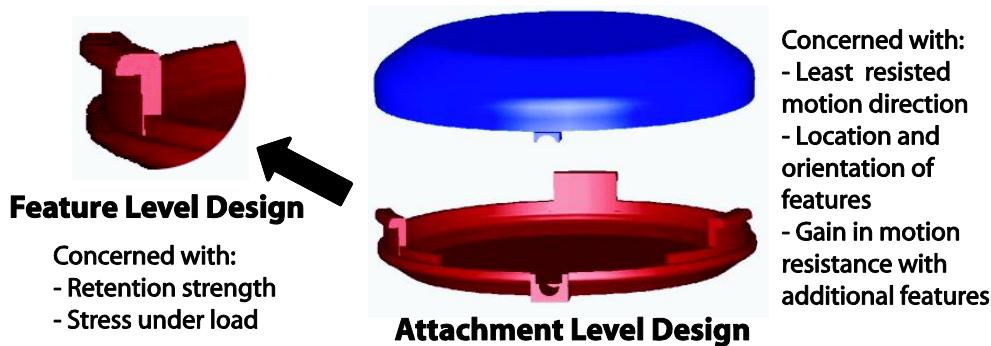


Figure 1.3 Feature level design vs. attachment level design

In contrast, attachment strategy focuses on qualitative assessment of assembly based on the overall constraint strategy. The kinds of questions typically addressed in attachment level design are: In which direction is the majority of the load likely to occur and how will the design maximize resistance to the load? In which direction are loads least resisted? What is gained by adding additional features? Would the features be more effective if they were at a different location or in a different orientation in the part? Most of the time, part assembly is a case of overconstraint. Therefore, the classical analysis of assembly forces and reactions is a statically indeterminate problem and requires a novel method to avoid ad hoc design guidelines.

This dissertation addresses attachment level research, and in particular the analysis of constraint effectiveness in assembly. There is a significant body of knowledge in the areas of bolted joint strength analysis and feature-level design of IAF, but very

little in attachment level design. Since IAF's are effective in resisting motion in one or two directions only, design decisions on feature locations and orientations highly influence the effectiveness of each constraint to resist motion. There is currently no design and analysis tool that can aid designers in optimizing the locations and orientations of features.

Engineering design therefore is a challenge, especially where the shape of the part and the orientation of constraint features are not always rectilinear. This is made even more challenging due to the complex shapes from molding, die casting, and other near net shape processes. DOF that remain unconstrained or weakly constrained might not be readily visible or obvious to the designer. Snap-fit assembly has its benefit in reducing overall manufacturing cost and increasing ease of assembly, but may suffer in constraint reliability. The designer may also overlook possible design improvements that would enhance the quality of the assembly.

In order to accomplish this design task, a designer must be equipped not only with experience, but also with domain-specific knowledge and design tools that provide better analysis and possible design improvements within the design space. There is a need for a design tool to quantify the adequacy of a certain constraint strategy and to make informed design decisions in the process. There is currently no design tool in place to numerically rate a constraint configuration based on its locations and orientations. This research is an attempt to provide better tools in this area of assembly design. Addressing the need in this context, this research is more applicable to assemblies that are constrained by integral

attachment features and other mating surfaces as opposed to traditional threaded fasteners.

1.3 Assembly design and analysis tool

In order to create a design tool that can improve assembly, some sort of evaluation metric of assembly quality must be in place. Therefore, a rating methodology and analysis tool must be in place in order to proceed in searching the design space and optimizing the constraint configuration. In fact, the analysis methodology is at the core of the design optimization tool. The design tool utilizes the analysis methodology and finds optimal solutions based on the assembly rating metric of interest.

In analyzing assembly, the same two-level approach (feature level analysis vs. attachment strategy analysis, Figure 1.3) applies. There is abundant knowledge and research on feature level analysis, even in complex geometries, but analysis methodologies to analyze assembly constraint and give a quantitative metric are scarce. When designers determine the number of constraints used or where they are located and oriented, there is no standard metric to quantify that such a design is adequate or the amount of improvement an alternate design would accomplish.

1.4 Dissertation overview

Figure 1.4 shows the topical map of this dissertation.

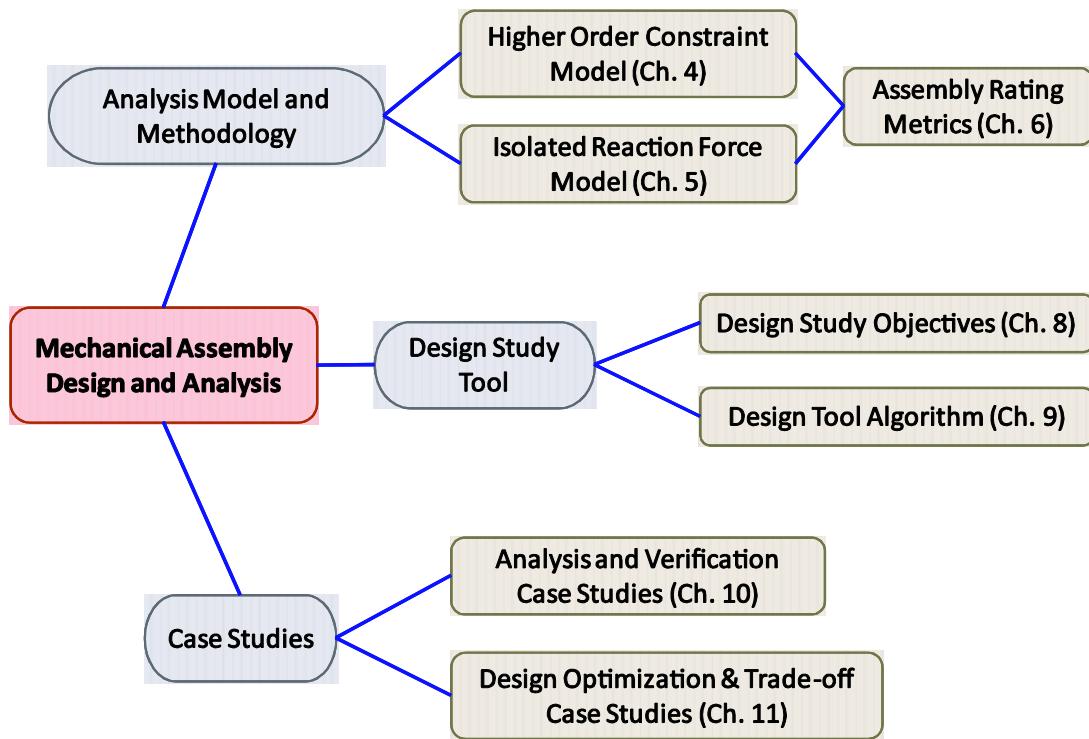


Figure 1.4 Topical map of the dissertation

This dissertation is organized into the following topics:

- Chapters 2 and 3 discuss the literature review. Because of the extensive use of screw theory, an understanding of screw theory and its mathematical implementation is crucial. For this reason, Chapter 2 is dedicated to literature review in general areas such as robotic grasping, fixture design, and kinematics of assembly, while Chapter 3 is dedicated to an overview of screw theory and highly relevant literature that directly contributes to the development of ideas in this dissertation.
- Chapters 4, 5, and 6 discuss the analysis tool methodology, higher order constraint modeling, isolated reaction force calculation model, and rating

metrics of assembly quality. Chapter 7 describes the detailed implementation of these models in MATLAB.

- Chapters 8 and 9 discuss the design objectives and algorithm of the design study.
- Chapters 10 and 11 discuss the case study results to both demonstrate the utilization of the design and analysis tool and generate a domain-specific knowledge of assembly constraint strategy.
- Chapter 12 summarizes the contributions of this dissertation, the knowledge learned about assembly constraint strategy, limitations, and future work in this area of research.

CHAPTER 2

LITERATURE REVIEW

This research utilizes kinematic principles to analyze mechanical part assemblies. There are many similarities between analysis of assembly constraint and analysis of either robotic grasping or fixture design. They all have the same objective, namely to remove DOF by constraining the motion using contact interfaces, locators and clamps. In assembly constraint design, constraints are provided by mating parts instead of fingers or fixture elements. Research literature in robotic grasping and fixture design is of particular interest when it utilizes kinematic principles, screw theory, or an optimization method to synthesize constraints.

For the above reasons, the background reference literature in this dissertation is organized into four main categories:

1. Research in theoretical kinematics, screw theory, and extensions of screw theory
2. Research in analysis and synthesis of robotic grasps
3. Research in analysis and synthesis of fixture design
4. Research in assembly constraint analysis and design

2.1 Research in theoretical kinematics, screw theory, and extensions of screw theory

A brief survey of previous literature in theoretical kinematics is provided here. The earliest studies of kinematic constraint analysis and screw theory were done by Reuleaux [40] and Ball [6]. Ever since then, many studies have developed screw theory further and applied it to kinematic analysis of mechanisms beginning from Waldron [49, 50, 51] and Hunt [18]. Other work that discusses mathematical methods of screw theory in more depth was done by Ohwovoriole and Roth [36], Kerr and Sanger [19, 20], Nguyen et al. [34, 35], and Lakshminarayana [23].

2.2 Research in analysis and synthesis of robotic grasping

Robotic grasping resembles mechanical part assembly more closely compared to research in general kinematics. The reason is that kinematic linkages tend to be dominated by bilateral joints that provide constraint in both directions while robotic grasping deals exclusively with unilateral joints (point, line, or planar contact). A unilateral point constraint prevents the rigid body from moving into the constraint but allows translation away from the constraint, translation across the constraint, and rotation in all three axes around the constraint. This dissertation mainly deals with integral attachment oriented assembly constraints that are unilateral in nature.

2.2.1 Form closure vs. force closure

Rimon and Burdick [41] clarified the differences between form and force closure. Form closure between object A and object B is modeled as a motion constraint from A to

limit the allowable motion of B. The sole purpose of the finger constraint is to limit motion, not to exert force. From a practical perspective, form closure is equivalent to immobilization. Force closure between object A and object B is modeled as A applying a set of contact forces on B. The collection of contact forces and torques W is the collection of all wrenches that can possibly be generated. If W held B in an equilibrium grasp such that all the fingers apply non-zero force, then the grasp is said to be force closure. For the particular definition of force closure by the authors, a proof was provided that a grasp is force closure if and only if it is form closure, although this statement seemingly contradict the traditional or commonly understood definition of force closure.

Yoshikawa [60] introduced the concepts of passive form closure, passive force closure, and active force closure. This further distinguishes the definition of form closure and force closure. A constraint is said to be passive closure if the current position and orientation of the object is maintained, even when an arbitrary external force is applied on the object, without changing the driving force of the constraining mechanism. In passive form closure, the contact forces are provided by constraining locators or preloaded constraints. Passive form closure is equivalent to form closure defined by Reuleaux [40]. Passive force closure implies that the balancing force counteracting the external force is produced by the mechanism itself. Active force closure is a condition wherein the resistance to an external wrench is provided by actively controlled joint actuators.

Figure 2.1 illustrates simple examples of each category. Figure 2.1(a) shows the case where an object is constrained by two constraining limbs with frictionless point

contact locators. This is passive form closure. Figure 2.1(b) is the case where one constraining limb has an active joint making it possible to move the limb in the horizontal direction and to exert an arbitrary force on the object. This is passive force closure. Figure 2.1c is the case where both limbs have active joints. This is active force closure.

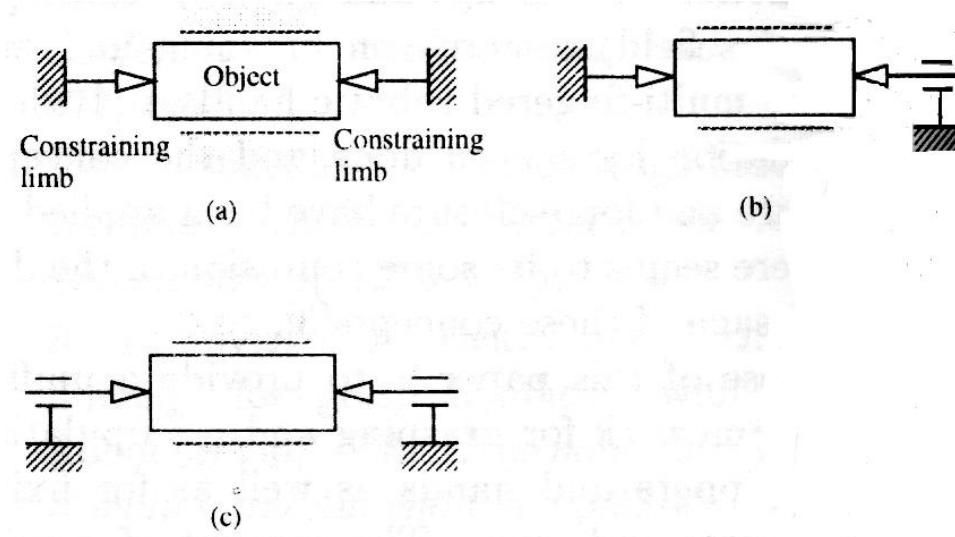


Figure 2.1 Examples of passive force closure [60]

Shapiro et al. [45] extended Yoshikawa's concept of passive force closure to identify the passive stability set, defined as the collection of external wrenches that can be passively resisted by a given grasp. The author goes further to define the various types of contact that provide passive force closure, namely a rigid body contact, a fixed force contact, and a compliant contact. The passive stability set defined in this work is analogous to the space of form/force closure defined by Chen et al. [11].

According to passive force closure definitions, a rigid body contact is a stationary rigid body that passively interacts with the object through a frictional or frictionless contact. A fixed force contact is a frictional point contact that applies a specific force at

the contact point. A compliant contact is a frictional contact that applies force according to a force displacement relationship of the contact point.

Nguyen [34] presented a simple method of constructing wrenches from three different types of contacts, namely frictionless point contact, hard finger contact, and soft finger contact, and determining whether they constitute form closure. The author provided theorems that specify mathematical conditions when the combinations of the contact types listed above constitute form closure. For the case of grasps with seven frictionless point contacts, the requirement given is identical to that of Kerr and Sanger [19], where the vector space must be a set of linearly independent wrenches. The author later improved the model to treat contacts as virtual springs [35].

This literature on passive force closure provides a basis for this dissertation to utilize a more accurate conceptual model in assessing constraint resistance quality as opposed to having to choose between form closure and force closure. Instead, the assembly constraint rating scheme is modeled conceptually as passive force closure. This is explored further in Chapter 5.

2.2.2 Form and force closure test

Asada [4, 5] developed a methodology to analyze kinematic properties (i.e. form closure) of a workpiece grasping constraint. Form closure needs no grasping force to be applied and therefore has distinct advantages over frictional grasping, in which grasping forces must be applied to maintain force closure. The author later applied this to workpart fixturing for flexible assembly, especially in achieving deterministic fixturing. A set of

contacts on the surface of a rigid body produces deterministic location when the part cannot translate or rotate in any direction without breaking contact with at least one of the contacts.

Salisbury and Roth [42] investigated the analysis of articulated mechanical hands using screw theory. The objective was to identify finger configurations that allow complete immobilization of the gripped object but also allow for manipulation of the object by the fingers while maintaining the grasp. This study also applies the analysis to identify internal forces that are acting on the object.

The finger contacts were modeled as a kinematic pair classified into point, line, and plane contact. The model distinguishes the contact type of the finger-object interface into point-on-line, point-on-plane, line-on-plane, and so forth. Both frictionless and frictional models were analyzed. When friction is included, two DOF are added to the wrench system of each type of contact. The model assumes force closure in that although contact may be broken by moving the bodies apart, forces act to maintain contact throughout the motion.

The authors began the analysis by identifying the possible configurations for a number of fingers and reducing the acceptable designs by computing the connectivity of each mechanism that satisfies the objective mentioned above. The evaluation of complete restraint and internal force is accomplished by composing the wrench matrix W of size $6 \times n$. For this matrix, n is the number of wrenches that act on the object with p of which having unidirectional capacity. For the unidirectional constraints, the particular solution must be positive in sign; otherwise a break of contact will occur.

Figure 2.2 shows a planar object grasped with a point contact with friction on the left and a soft finger on the right. w_1 and w_2 are forces normal to the surface and must be kept positive to maintain contact and allow the other five wrenches, which depend on friction, to be active. w_3 thru w_6 are friction forces that depend on the normal forces from w_1 and w_2 . w_7 is a moment due to the soft finger assumption. Since only w_1 and w_2 are unidirectional, $p = 2$.

In general, when more than seven wrenches ($n > 7$) are acting on a body, there is more than one set of independent internal forces. The actual number of independent internal forces can be determined as follows. The dimension of the null space of W is q , where $q \leq n - 6$. Therefore, the homogeneous solution will have up to q free variables that can be chosen to make the first p elements coefficient positive. If this is possible, then total restraint can be achieved.

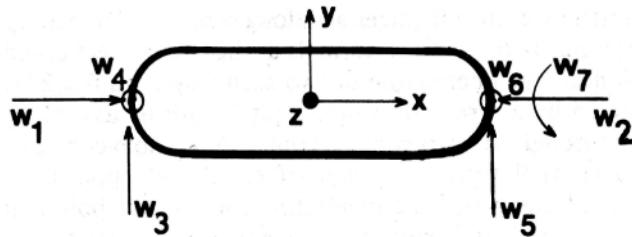


Figure 2.2 Flat object grasping example by Salisbury and Roth [42]

The authors also introduced the grip transformation matrix that relates the external forces measured by sensors at the finger contacts to the internal forces. This can be used in controlling the internal forces on the object based on the measurements. In summary, this work is useful in understanding how to model multiple point contacts with varying DOF and translates them into a combination of constraints. In addition to the capabilities to evaluate total restraint, the model is also capable of calculating internal

force based on the constraint configuration. This literature is one of the first significant efforts in using screw theory to determine total restraint provided by unilateral constraints.

Kerr and Roth [21] followed up this work by discussing methods for determining how hard to squeeze an object in order to ensure a secure grasp, finger-joint motions required to produce desired motion, and the workspace of the hand. The part of this work that is of interest is the task to produce a stable grasp. Stability of grasp is defined here as the ability of the fingers to prevent the object from undergoing unwanted movements. The authors described a method to determine intensity of the wrench matrix to solve the overconstrained problem by examining the flexibility of the object and of the fingers. These statically indeterminate components are called internal grasp forces. The goal then is to optimize the selection of internal grasp forces within the friction constraints and joint torque limits.

2.2.3 Force distribution analysis

Several attempts to analyze force distribution in grasps with point contacts have been done [3, 13, 35, 37, 59]. Nguyen [35] and Yoshikawa [59] attempted to solve the problem using a generalized inverse from which a minimum norm solution to the problem can be obtained. It has been shown that using this approach implies that all the stiffnesses are the same. In practice, stiffnesses can vary widely across grasps. Dai and Kerr [13] analyzed statically indeterminate constraints by combining the elasticity at constraints with geometric compatibility. In this model, each contact point is represented

by contact screws along the normal (frictionless) and tangent (friction) directions. Planar and spatial examples are shown in which force distribution at the constraints can be solved as a function of constraint location, orientation and stiffness ratios. Preload ratio is also obtained based on the ratio of the normal and tangential stiffnesses for cases with friction. This study described an approach to calculate force distribution that might be of interest in analyzing assembly constraints. However, part stiffness values are not as readily and accurately available compared to stiffness in robotic grasping. Obtaining force distribution explicitly based on stiffness values in assembly might be too speculative. In addition, force distribution at constraints in the context of mechanical assembly is due to arbitrary external load as opposed to the constraints being actuated in robotic grasping. These differences limit the application of this methodology significantly.

Chen and Walker [11] proposed a methodology to visualize the capability of a constraint configuration to achieve form and force closure in grasping solid objects. The authors proved that any disturbing force/moment pair at the mass center of an object can be balanced if there exists a particular set of internal forces falling inside the friction cone. These friction forces at the contact points can be increased to bring the forces at each finger into the friction cone. The results of this work can be extended using the methodology proposed by Schimells [43] to provide some visualization tools. The authors' methodology took into account friction. It was not discussed what the space of form/force closure would be without friction. Although this dissertation does not aim to

calculate the space of weakly constrained motion, this is an area that is potentially useful for the designer.

Wang and Pelinescu [52] proposed to predict contact forces in a passive force closure model by using a minimum norm principle, which is a special form of the minimum energy principle. The methodology yields a unique solution for contact forces without intensive numerical procedures. These reactive forces, however, were computed as a reaction to the clamping forces, not opposing an external wrench. Xiong et al. [57] followed up this work with the inclusion of a linear-elastic model and solved the formulation with a nonlinear least squares method.

2.2.4 Quantitative metric for constraint configuration

Markenscoff and Papadimitriou [31] recognized the need to define a measure of the quality of a grip. The metric was defined as the ‘compressional forces required in order to balance the weight of the object’. The smaller the forces, the better the grip. This is inversely analogous to the notion of virtual displacement used by Bausch and Youcef-Toumi [7], as explained later in Chapter 3. The authors developed an analytical function to define the quality of a three-finger grip and optimize the location of the fourth finger grip. It was noted that this is a rather badly behaved function, and there is little hope that a closed-form solution can be found. Furthermore, there were problems with concave vertices and parallel sides for which the author came up with an alternate computational algorithm. This was one of the earliest efforts to address the quantitative analysis of

constraints; unfortunately, the model has only been applied to 2D problems and has many mathematical hindrances toward a practical solution.

Another attempt to develop a quantitative test was done by Trinkle [47, 48]. The author adopted the concept of homogeneous forces developed by Lakshminarayana [23]. This was developed further and implemented as the internal grasp force discussed by Salisbury and Roth [31]. The homogeneous forces are the solutions to the null space due to the redundancy of the constraint configuration. Trinkle's metric evaluates the proximity of a constraint configuration to losing form closure. This is determined by the minimum of the homogeneous force solution. Wolter and Trinkle [56] later used this rating scheme to synthesize fixture points for assemblies. Trinkle's methodology, however, failed to account for the effectiveness of a constraint to resist moments as it gains leverage due to its location with respect to the moment arm. One of the reasons is that the methodology only measures a unique value of internal forces while the quality of constraint configuration will vary as the loading condition changes. Bozzo's [9] thesis discussed this deficiency in more detail with examples such as in Figure 2.3. The four cases in the figure are rated equally. For this reason, the methodology does not apply well in assembly constraint analysis.

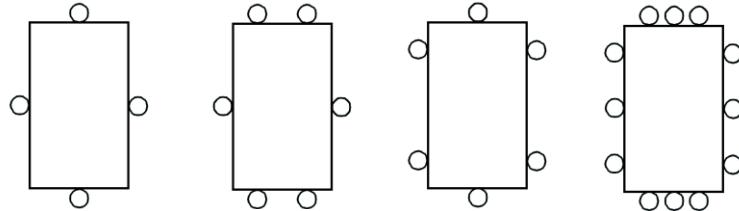


Figure 2.3 An example of deficiencies in Trinkle's rating scheme [9]

2.3 Research in analysis and synthesis of fixture design

One area of research that closely resembles the assembly design problem is in fixture design for machining, precise machine design, and industrial assembly planning. In fixture design research, the design process is naturally divided into two stages. The first consists of developing a scheme for locating the part. The second is concerned with the choice of a set of clamps to completely immobilize the work part when subject to external forces and disturbances. In most part assemblies, assembly constraints such as fasteners and mating surfaces act as both the locators and locks of the rigid body. Therefore, there is no practical need to differentiate them.

De Meter [14] investigated the optimization of locator and clamp location in machining fixtures for the maximization of mechanical leverage. The goal of the optimization was to improve machining accuracy by reducing workpiece deformation at the contact region. To accomplish this, locators and clamps are positioned where their mechanical leverage is maximized. Maximizing mechanical leverage implies minimizing reaction forces at the contact region and therefore reduces deformation.

In the author's model, there are three forces that act on the rigid body, namely the locator constraints, the actuated power clamps, and the machining load. The constraints due to contact regions, locators, and clamps are modeled as point constraints with friction exerting unidirectional reaction forces, represented in wrench motor notation. The direction of these frictional constraints lies in the polyhedral cone approximation normal to the contact plane. Power clamps are single DOF mechanisms that are either pneumatically or hydraulically actuated. While the locators are passive constraints, the

clamps' actuator intensity can be controlled and is limited to the capacity of the actuating system. The machining load is specified by a set of non-linear wrenches acting on the workpiece during the machining operation. The optimization objective function seeks to minimize the maximum normal load across the contact regions while varying the variables defining the location of the locators and intensity of the clamps within the search space defined. The problem is characterized as a nonlinear constrained optimization problem. The author used the feasible direction technique that satisfies the Fritz-John necessary local optimality condition. The main difference between this work and assembly design is that in machining operations, the load wrenches are known and adequately modeled while loads in assembly design are arbitrary. The challenge in assembly optimization is defining these wrenches.

The same author also addressed the challenge of constraint analysis with surface contacts for the cases both without and with friction [15]. Among the types of frictionless contact addressed in the model are point contact, planar contact, spherical contact, and cylindrical contact. The types of frictional contact addressed in the model are hard point contact, soft point contact, hard planar contact, hard spherical contact, and hard cylindrical contact. In general, contact interfaces that are of higher order than point contact are spanned by a finite set of wrenches located at the nonnegative linear hull. Frictional convex cones were approximated by n-sided closed polyhedral cones. The difference between hard and soft point contact is that a hard point contact can only provide reactive force in the plane orthogonal to the normal direction while a soft point contact is capable of providing bidirectional torque in the normal direction in addition to

the frictional forces orthogonal to the normal direction. Mathematical proofs are provided to substantiate the propositions of the model. After modeling the contacts as a finite set of wrenches, total restraint can be determined.

Figure 2.4 and Figure 2.5 show an example of modeling machining fixture contacts as constraints. The fixture consists of a base plate, four external locating pins (R1, R2, R3, and R5), an internal locating pin (R4), and a strap clamp (R6). The workpiece is rectangular with a blind hole from the bottom surface. Planar contact exists at all three regions R1 thru R3. The fourth external locating pin makes line contact (R5) with a planar datum surface. An internal locating pin makes cylindrical contact (R4) with the hole, while the strap clamp makes planar contact (R6) with the top surface.

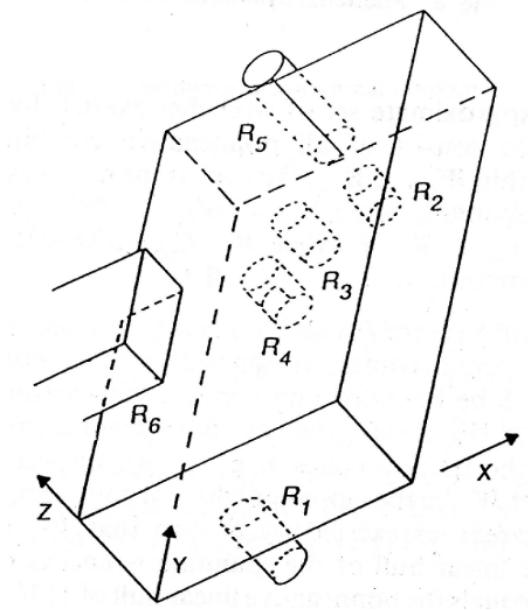


Figure 2.4 Example of machining fixture contact by DeMeter [15]

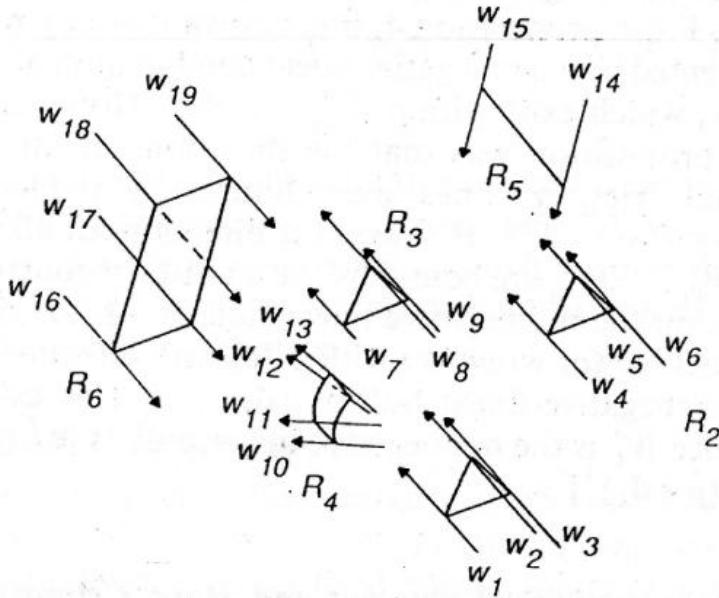


Figure 2.5 Example of modeling machining fixture contact as constraints by DeMeter [15]

This work is one of the few to address modeling of higher order constraints. It presents the wrenches that are linearly dependent to each other within a single higher order constraint. However, the model is still insufficient to be used in modeling assembly constraint in this dissertation. The reason is because the discretization process, similar to what was implemented by Bozzo [9], leads to some fundamental issues explained in Section 3.4.

Most research results have been published on fixture design methods for one machining or stamping operation, but there is no available method that fixture designers can use to synthesize flexible fixture workspaces for a set of different stampings. Lee et al. [24] proposed a new methodology to do this. The methodology involves representing the candidate locator region from the fixture robot workspace and optimizing the

objective function, which is a function of circle size and overlap. The optimization algorithm employed a probabilistic optimal search using a genetic-algorithm-based computer program. Lee and Haynes [25] also attempted to explicitly include part stiffness in optimizing fixture elements using finite element methods. The workpart is modeled as a deformable body based on linear elasticity and a coulomb friction model. The optimization objective is to minimize the total work done on the workpart. A similar effort by Menassa and DeVries [33] that also used finite element methods proposes optimization techniques for minimizing the workpiece deflection at selected points. Xiuwen [58] applied a similar elastic workpiece model with the end goal of improving location accuracy and minimizing error.

An effort aimed at minimizing machining errors in fixture configurations was done by Cai et al. [10]. Instead of using the finite element approach to calculate workpiece deformation, the authors used a variational method. The proposed method provided an analytical approach with closed-form mathematical solutions. This method arguably provided easier physical interpretation as well as insight into the fixture configuration behavior in contrast to the finite element approaches. The following conditions provide the design constraint equations:

- Deterministic locating: If a workpiece under a certain locating scheme cannot make an infinitesimal motion while maintaining contact with all the locators, then the workpiece is deterministically located.

- Total fixturing: If a deterministically located workpiece under the influence of clamping and machining forces maintains contact with all the locators, then the workpiece is totally fixtured.
- Robust fixture configuration: Robust design in this context refers to the minimization of workpiece positional and/or orientation errors (resultant errors) due to workpiece surface and/or fixture set-up errors (source errors) in a deterministic locating environment

Simple 2D and 3D examples are provided to show initial configurations and optimized configurations with reduced error. The above condition, except for the robust fixture configuration, is applicable both in fixture design as well as assembly constraint analysis. The robust fixture optimization is not very applicable in assembly because positional accuracy is of low interest compared to resistance to loading wrenches.

A simple yet systematic analysis of fixture design using screw theory was done by Chou et al. [12]. The authors extended the works of Lakshminarayana [23], Salisbury and Roth [42], and Kerr and Roth [21] in robotic grasping to fixture design analysis. The authors specify equations for:

- Deterministic location: As a workpiece is placed into a fixture, the positioning must be established by a set of locators. Let W be a $6 \times n$ matrix of n normalized locating wrenches, F an $n \times 1$ intensity vector, and w_p the positioning wrench. The equilibrium equation $[W][F] = -w_p$ must have nonnegative solutions for F because locators are unidirectional in their

capability to provide reaction forces. In addition, W must be full rank in order to provide deterministic location.

- Clamping stability: When clamps are applied, workpiece equilibrium must always be maintained; $[W][F] = 0$. It follows that the following inequality must be satisfied: $[F] = \lambda_1 V_1 + \lambda_2 V_2 + \dots + \lambda_q V_q \geq 0$, where V_1 thru V_q are the basis of the null space of W , λ 's are the free variables and $q = n - 6$. λ signifies the actual clamping forces.
- Total restraint: When a workpiece is constrained in a fixture, it must withstand every possible cutting force and torque. For a cutting wrench w_c , there should be a nonnegative F such that $[W][F] = -w_c$.

For $n \geq 7$, the solutions have the form $F = F_p + \lambda F_h$, where F_h is the homogeneous solution. The λ 's are what Salisbury and Roth [42] called internal forces. These equations are solved simultaneously to select and synthesize a fixture configuration that satisfies the inequalities from several candidate locators and clamps. A similar effort by Hirai and Asada [17] also represents these design requirements in the simultaneous inequalities in terms of inner products of two vectors. The problem was then solved using the theory of polyhedral convex cones. This work provides a comprehensive systematic approach by applying previous extensions of screw theory in the fixture design context. These design requirements (form and force closure) are similar in assembly constraint analysis. The implementation is not necessarily applicable in assembly because the optimization procedure is done to bring the fixture configuration to satisfy the design requirements, but not necessarily provide an optimal resistance to

arbitrary motion. In other words, the inequality condition in this literature provides the space of acceptable designs that meet the three conditions stated above, but without a quantitative metric of quality. In assembly, there is no known clamping force applied, so the homogenous solution approach to internal force cannot be applied.

Marin and Ferreira [28, 29, 30] addressed the problem of synthesizing the location of optimal clamping on three-dimensional parts with planar and cylindrical faces, with and without friction. This work solves the optimization problem given a part with pre-defined deterministic locators, a set of polygonal convex regions, and a known set of external disturbing wrenches. These polygonal convex regions are defined as the admissible clamping areas. The objective was to determine the location of the optimal clamping position. The authors investigated the fact that the optimum solution may not be unique, and therefore, they identify lines of constant clamping force, along with clamps that can be moved while maintaining a constant clamping force.

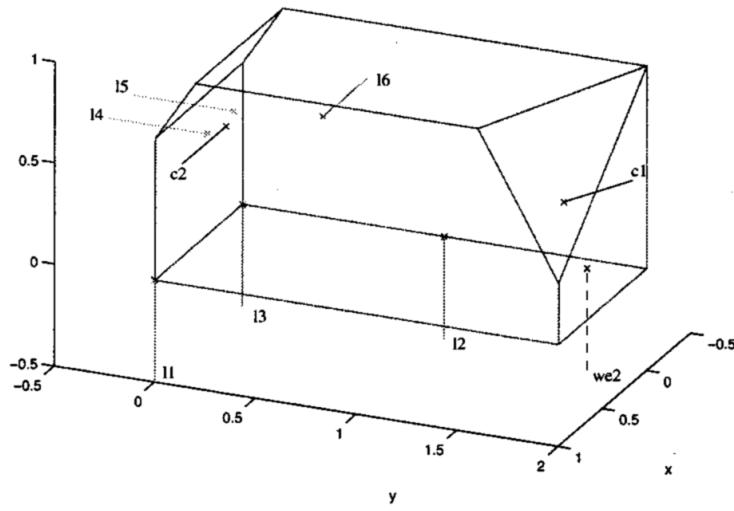


Figure 2.6 Optimal clamps for part subject to load we_2

A graphical example is shown in Figure 2.6. $l1$ through $l6$ are locators of the workpart subject to wrench $we2$. The location of clamps $c1$ and $c2$ are determined to be optimal at the locations shown on the rectangular and triangular planes. The reaction intensities at $c1$ and $c2$ are also calculated.

The optimization method used by this author is not useful for the kinematic analysis of assembly because it requires pre-defined disturbing wrenches. However, in the synthesis part of this dissertation, the problem fits very well with the solution suggested from this author, which is synthesizing the optimum location to resist predefined loading conditions given in an admissible search space to create constraint. The linear optimization method used by this author might be useful to generate candidate constraints on planes and cylindrical surfaces and possibly identify lines of constant constraint location along which constraints can slide without any change in overall assembly rating.

Schimmels [43] developed another model to evaluate the space of allowable motion. Using virtual work, the author defined a linearly force assemblable fixture. The goal was to design a manipulator so that at all possible part misalignments, the contact forces always lead to error-reducing motion in the absence of friction. The author focused on the insertion of a workpiece, not the analysis of the fixture workpiece after it is in place. The author argued that to facilitate proper insertion, a partial fixture is required to be deterministic and strongly detachable. The calculation involved is similar to the model used by Asada and By [4]. A deterministic fixture is one that is properly mated (exactly constrained). A fixture is deterministic if the constraints of a partial fixture provide N

independent wrenches when the workpiece is located in its properly mated position.

Three linearly independent contact wrenches are required for a planar fixture, and six are required for a spatial fixture [18]. In other words, a fixture is deterministic if and only if:

$\text{Rank}(\underline{w}) = N$, where \underline{w} is the matrix composed of the constraint wrenches $\underline{w} = [\underline{w}_1, \underline{w}_2, \dots, \underline{w}_m]$, m is the number of constraints and N is the number of DOF of the workpiece. If the deterministic fixture has no redundant constraints, then $m = N$ (\underline{w} is full rank). A perturbed deterministic fixture will break contact with at least one constraint, while a non-deterministic fixture (underconstrained) has a motion that will allow all constraints to remain in contact.

A detachable fixture is one for which there exists at least one unconstrained trajectory between the desired workpiece location and an outside position [4]. A strongly detachable fixture is one for which there exists a detaching motion that detaches all constraints at the same time. This space of motion D is the space of all motions that are available and are reciprocal or repelling to all constraints. A twist t is a detaching motion if and only if $w^T t \geq 0$.

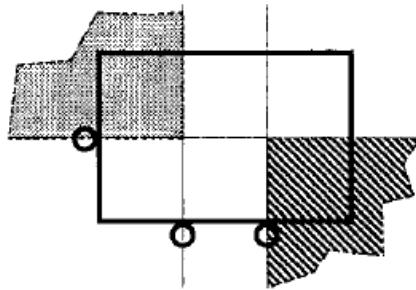


Figure 2.7 A planar example on space of detaching motions [43]

A planar example is shown in Figure 2.7. A detaching motion is reciprocal or repelling to all constraints. Clockwise rotation about any point in the cross-hatched area or counterclockwise rotation about any point in the shaded area is a detaching motion. The locus of instantaneous rotation centers is used to illustrate the space of detaching motions. A rotation about any point strictly within these bounds will cause contact to be broken at all constraints at the same time.

The space of error-reducing velocities contains velocities that maintain contact with the current constraints and cause the workpiece to move toward those constraints not already in contact. In screw theory terms, an error-reducing velocity is a twist that is reciprocal to the currently contacting wrenches and contrary to the currently non-contacting wrenches. If this approach can be extended to 3D objects and space, it might be possible to utilize this method to separate the space of weakly constrained motion for the purpose of assembly constraint analysis.

2.4 Research in assembly design

In Chapter 1, two different approaches in studying assembly design are described. One is feature-level design, where one is concerned mainly with the effectiveness of individual assembly constraint features. The other is attachment-level design, where one is concerned mainly with the removal of DOF by placing constraints in optimum locations and orientations. Since this dissertation addresses the need for a better assembly design tool in attachment-level design, this section discusses literature that approaches the assembly design problem in the same manner.

Design of statically determinate assembly [46, 53] and exact constraint design [8, 22] are well known to practitioners of design of precision instruments and tools.

Designers of fixtures and clamps call it ‘3-2-1’ design. Since a precision instrument’s function is directly dependent on the accuracy with which the component parts achieve their required relationship, principles of assembly design can follow similar principles.

Slocum [46] proposed two important principles for precision structural designs: kinematic design and elastic averaging. The principle of kinematic design states that point contact should be established at the minimum number of points required to constrain a body in the desired position and orientation. This prevents overconstraint, and thus an “exact” mathematically continuous model of the system can be made. Kinematic design creates a very repeatable and precise way to mount or attach parts. However, it is subject to high contact stresses and therefore limited in its load capacity.

Redundant contact points are needed to support larger loads. In these cases, a set of compliant and elastic couplings is used to spread the load. The principal of elastic averaging states that to accurately locate two surfaces and support a large load, there should be a very large number of contact points spread out over a broad region. An example of this would be the windshield wiper blade holders on automobiles. A cascaded system of flexible links is implemented to ensure that the wiper maintains uniform pressure on the windows as the curvature changes beneath the wiper blade as it wipes across the window. The weakness of this principle is that the location of the part becomes indeterminate.

However, from a practical design-for-manufacturing perspective, redundant constraints are inevitably needed, leading to elastic averaging or force closure. The reason for this is that an exactly constrained design is prone to decreased load carrying capability when manufacturing error is introduced. Overconstrained design is undesirable because it is subject to either high preload or loose fit due to manufacturing dimensional variation. In an overconstrained assembly, reaction forces are statically indeterminate.

Each of the two principles - kinematic design and elastic averaging - has its strengths and weakness (Table 2.1). A three-legged stool rests firmly and is exactly constrained. A four-legged stool gives the security and additional load capacity of an extra leg but will not rest firmly unless the legs are elastic enough to deform until all four are in contact.

Kinematic Design	Elastic Averaging
<ul style="list-style-type: none"> • A.K.A. form closure • Deterministic location • Statically determinate • Less reliance on manufacturing • Stiffness and load capacity limited • Internal stresses are kept low 	<ul style="list-style-type: none"> • A.K.A. force closure • Indeterministic location • Statically indeterminate • More reliance on manufacturing • Stiffness and load capacity not limited • Preloaded on elastic support stiffness

Table 2.1 Comparison of Kinematic Design vs. Elastic Averaging Principles

These two principles resemble differences between form closure and force closure. Kinematic design is analogous to form closure because its main concern is

deterministic location of the part. It is generally not designed to resist large service wrenches. Elastic averaging design is analogous to force closure because by redundantly supporting the part, its location is not deterministic. Its main purpose is to support large loads. Most part assembly leans toward utilizing the elastic averaging principle and force closure. Many applications do not require the accuracies offered by kinematic design and are more prone to arbitrary loading in fulfilling their function.

Because exactly constrained design removes the body's DOF one constraint at a time, there is usually a one-to-one relationship of constraints to DOF. Exact constraint design often requires and relies heavily on nesting forces in addition to gravity. Nesting forces keep the body in contact with its locators. Pearce et al. [38] discussed the design of nesting forces and how tolerances affect nesting forces. Nesting forces can be provided with passive features such as a set screw or active features such as a compression or cantilever spring. Active features preload the body in a manner very similar to clamps in fixture design. The advantage of an active feature is that it can absorb variation throughout the life of the design. In a way, nesting forces act as soft constraints. The interaction between nesting forces and tolerance is analyzed by observing vector loops between nesting force locations and constraints. The effect of a nesting force on dimensional tolerance is somewhat analogous to calculating the virtual displacement of the body into the constraints due to the nesting force.

Whitney et al. [55] and his book “Mechanical Assemblies” [54] are among the few works dedicated to addressing the need for assembly design knowledge among

engineers. The authors laid out the design principles for assembly on the attachment strategy level.

The authors presented a theory to design assemblies from a top-down process to deliver Key Characteristics (KC) that achieve top-level customer requirements. The method for designing an assembly starts by defining abstractly a desired constraint arrangement by which parts will be located in space relative to each other. Geometric relationships critical to the KC of the product function are defined. An arrangement called a datum flow chain (DFC) is created and can be checked for certain syntactic errors to ensure a competent assembly [32]. The DFC is then realized by creating links between the parts called assembly features. The DFC represents the constraint plan when the parts and features are of nominal size and location. When considering joints between parts, those that pass constraints deserve special attention and priority and are given the name ‘mates’. Joints that do not pass constraint or that pass incomplete constraint at places where the DFC does not pass are called ‘contacts’. These are susceptible to adjustments. In assembly design, using KC and DFC will enable engineers to identify important connections in a multi-part assembly. Using the tool will provide attachment-level design tools in a bigger picture. Nevertheless, there is little mathematical theory in these references, which consist of the definition of the method and good advice. The constraint analysis tool developed in this dissertation can be used in conjunction with the tool to conduct constraint analysis of two-part assemblies.

Adams et al. [1, 2] used screw theory to determine over-constraint as well as screw reciprocal principles to determine unconstrained motions. In the analysis, the

reciprocal screw equation is used to calculate free twists in a wrench system. The term “motion analysis” is used by the author to determine if an assembly has any underconstrained DOF. The unconstrained motion is found by finding the intersection of different sets of screws. This is the set of screws common to all the sets. The term “constraint analysis” is used by the author to determine if an assembly has any overconstrained degrees of freedom. It does this by finding the intersection of the set of wrenches acting on a part. However, in all this analysis there is no assessment of the assembly design quality. Unfortunately, most mechanical part assembly is overconstrained and already removes all DOF. Therefore, the design tool developed by the author cannot be used to analyze the effectiveness of a constraint configuration, especially a redundant one.

2.5 Research on constraint analysis in assembly

Bausch and Youcef-Toumi [7] developed a kinematic method to analyze the workholding condition by evaluating the motion stops corresponding to the reciprocal screw motions within a given fixture configuration. Bozzo [9] made adaptations to apply Bausch’s algorithm in IAF assembly constraint analysis. This methodology:

- Is able to evaluate an overconstrained design with more than seven point contacts by taking combinations of five point contacts at a time.
- Calculates an individual rating for each constraint based on a motion stop rating, similar to the virtual work equation. Most previous research is concerned only with the binary analysis of total restraint.

- Generates its motion based on the location of its constraints and therefore is able to identify weak motions that are inherently related to the location and orientation of the constraint configuration.

Since their work in assembly constraint analysis is by far the most appropriate starting point for this dissertation, their methodology will be discussed in detail separately in Chapter 3.

2.6 Summary

In summary, the literature review covers attempts of constraint analysis and synthesis in the area of robotic grasping and fixture design. A number of assembly design contributions to the literature are also surveyed. With the exception of Bausch's and Bozzo's work, it can be observed that there are fundamental gaps between previous research and the need for an assembly constraint analysis and synthesis tool. The literature then can be grouped according to these gaps.

First, almost all of the literature that attempts to evaluate total restraint is incapable of producing a quantitative metric of the constraint configuration – they only analyze whether total restraint is achieved or not achieved. This information is not very useful for the assembly designer. Most assembly is over-constrained and has removed all six DOF under rigid body and rigid constraint assumptions. In real parts, however, the effectiveness of the constraint configuration can vary. This is illustrated in the example below. Figure 2.8 shows two planar cases where the rotational DOF is removed, but both are not of the same effectiveness to resist the loading moment.

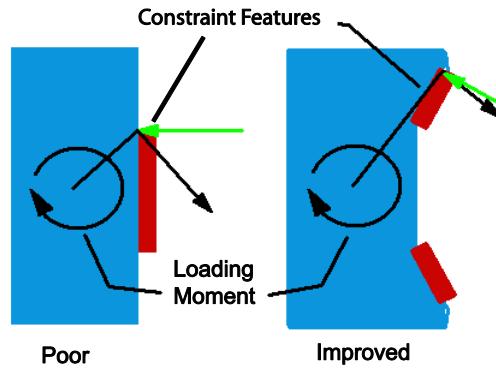


Figure 2.8 Poor vs. improved constraint

Second, when quantitative metrics are assessed for robotic grasping, they are usually measured with regard to grasping force distribution or internal grasp forces. In robotic grasping, forces are applied at the constraints, which are the fingers driven by actuators. The distribution of these forces and the resulting internal grasp force are of interest. In assembly, constraints react to an arbitrary external loading wrench as opposed to being the actuator of the loading wrench. In addition, internal forces are not of interest in understanding the quality of an assembly.

Third, fixture design research has yielded useful tools to optimize locator and clamp placement and orientation for a predetermined loading condition. The loading wrench is usually a known variable in a machining operation. This is not true in assembly design. Many mechanical assemblies need to be optimized with respect to unknown loading conditions. The particular challenge is then to generate a finite set of motions from the infinite possible motions in 3D space to be evaluated.

Chapter 3 discusses the basic concepts and building blocks to create a new methodology for assembly constraint analysis and synthesis, namely screw theory and its application in constraint analysis.

CHAPTER 3

OVERVIEW OF SCREW THEORY AND APPLICATIONS IN CONSTRAINT ANALYSIS

3.1 Overview of screw theory

In this dissertation, the problem of assembly is modeled using physical and geometrical reasoning based on the classical work of Ball [6]. Reuleaux [40] recognized that the minimum number of contact points required for form closure of a planar rigid body is four. Using screw theory, Lakshminarayana [23], Ohwovorile [36], and Salisbury et al. [42] have demonstrated that seven or more frictionless point contacts are required to totally constrain the motion of a body in three-dimensional space and that four or more point contacts are required to totally constrain the planar motion of a body.

An assembly can be achieved by two different ways. One is form closure, defined by Lakshminarayana [23] as purely kinematic, in which the geometries of the contacting rigid parts prevent motion regardless of the magnitude of the applied force. Force closure involves the use of friction to assist in the reduction of the freedom of motion of a kinematically underconstrained object. In force closure, motion along a kinematically

unconstrained direction is prevented as long as the magnitude of the applied force does not exceed the maximum support that can be provided by friction.

Screw theory states that any spatial displacement of a body, except a pure translation, is equivalent to a rotation about a unique line combined with a translation parallel to that line. The line is known as the screw axis. The ratio of the translation parallel to the axis, d , to the rotation about the axis, q , is known as the pitch of the screw, h .

$$h = \frac{d}{\theta}. \quad (3.1)$$

The screw system is a spatial distribution of any instantaneous screw axes (ISA) whose nature is determined by a number called the order of the screw system. The screw system of order zero contains no ISA's. Figure 3.1 shows the geometry of the velocity field around an ISA. The direction of the ISA is that of the angular velocity ω . The translational components of velocity in the direction of the ISA at all points in the body are equal to v_s . The tangential velocity of the vector field can be represented by the expression $\omega \times q$, where q is the normal vector from the screw axis to the point.

In this dissertation, the term one-system is used to denote a 1st order screw system, two-system a 2nd order screw system, etc. A one-system consists of a single, unique ISA. A two-system is a set of ISA's whose axes generate a ruled surface called a cylindroid. A three-system is a spatial distribution of ISA's in which all ISA's having a given pitch have axes that are members of one family of a hyperboloid. Screw systems of order four and five may be related to those of orders two and one by the theory of reciprocal screws.

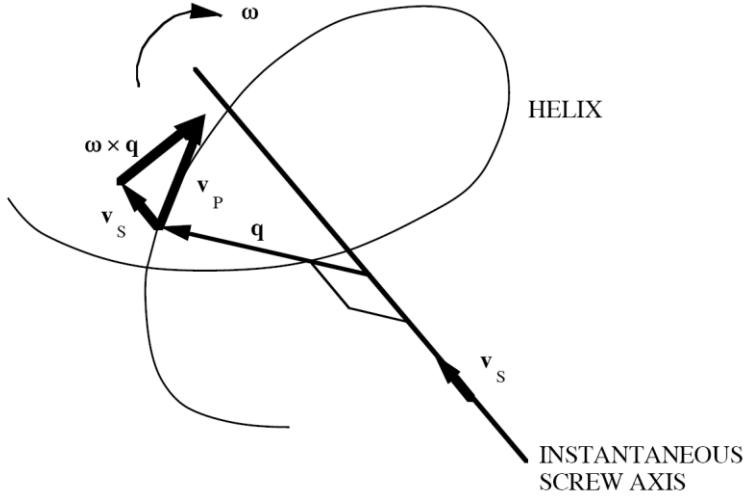


Figure 3.1 The geometry of the velocity field of points in a rigid body [49]

It is helpful to express ISA's in motor notation. A motor about a screw axis is represented by a pair of vectors (ω, μ) . ω is the angular velocity of motion about the screw axis and therefore is parallel to it. The magnitude of the angular velocity is not a property of the screw axis. In this respect, the motor is not a perfect analog of the screw axis. μ is the velocity, produced by motion about the axis with angular velocity ω , of the point in the body that instantaneously coincides with the origin. By the definition of a screw axis, the velocity of a point P on the axis is $h\omega$, where h is the pitch of the axis. If ρ is the position vector of P , the velocity of the point at the origin relative to P is $-\omega \times \rho$. Hence, the velocity of the point at the origin is

$$\mu = h\omega - \omega \times \rho, \quad (3.2)$$

$$h = \frac{\mu \cdot \omega}{\omega \cdot \omega}. \quad (3.3)$$

The velocity of any other point whose position vector is p is given by

$$v = \mu + \omega \times p. \quad (3.4)$$

A pure rotation is a screw with zero pitch. A pure translation can be considered a pure rotation around an axis located at infinity or as translation along a screw axis with infinite pitch.

Hunt [18] and Waldron [50, 51] applied screw theory to analyze the kinematics of machinery. A kinematic joint between two bodies may be represented by a screw system. The dimension and position of the screw system of a joint are uniquely determined by the geometry of the particular constraints of the joint. If all freedom twists allowed by this joint are defined as one screw system, the constraint imposed by this joint will be the reciprocal of the freedom screw system. A wrench is a force or static analogy to a twist. It is a combination of force acting along the axis of a screw and a couple acting about the axis of the screw. In the kinematic assembly model, the constraints represent wrenches, and the set of motions allowed by the respective constraints are twists that are reciprocal to the wrenches. Two screw axes are said to be reciprocal to one another if a wrench applied about one does no work on the other. Two screw systems are mutually reciprocal if every screw of one is reciprocal to every screw of the other [49]. An n^{th} order system is reciprocal to a $(6-n)^{\text{th}}$ order system. The power P produced by the wrench $[R, M]$ about the screw axis of the motor $[\omega, \mu]$ is

$$P = R \cdot \mu + M \cdot \omega. \quad (3.5)$$

It can be identified that an infinitesimal angular rotation about an axis is reciprocal to a static force along its axis. Vice versa, an infinitesimal translation is reciprocal to a couple with an axis coincident to the translational axis [18]. In

infinitesimal kinematics, finite velocities are modeled as virtual displacements and finite work as virtual work.

In this dissertation, motion is referred to as screw motion or twists, while constraints will be referred to as wrenches or a wrench system. There exists a duality between screws and wrenches. A screw with zero pitch is a pure rotation, while a wrench with zero pitch is a pure force. Similarly, a screw with infinite pitch is a pure translation, while a wrench with infinite pitch is a pure torque. This becomes important when the reciprocity relationship is used to generate the motion set to be evaluated.

Ohwovorole and Roth [36] classified all twists t into 3 categories using the principle of virtual work. Virtual work W is defined as

$$W = w^T t. \quad (3.6)$$

For spatial point contacts, w is the wrench associated with the unilateral point contact given by

$$w = \begin{bmatrix} \vec{f} \\ r \times \vec{f} \end{bmatrix} = \begin{bmatrix} f_x \\ f_y \\ f_z \\ m_x \\ m_y \\ m_z \end{bmatrix} t = \begin{bmatrix} \vec{r} \times \vec{\omega} \\ \vec{\omega} \end{bmatrix} = \begin{bmatrix} v_x \\ v_y \\ v_z \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}. \quad (3.7)$$

The 3 categories are:

- Repelling: twists that cause the rigid body to move out of contact with the constraint, for which virtual work is positive. In this case, the angle between the line-of-action of the twist and the constraint normal is less than 90 degrees.

- Reciprocal: twists that cause the rigid body to maintain contact with the constraint (i.e. slide along the constraint), for which the virtual work is zero. In this case, the angle between the line-of-action of the twist and the constraint normal is 90 degrees.
- Contrary: twists that cause the rigid body to penetrate the constraint, for which virtual work is negative. In this case, the angle between the line-of-action of the twist and the constraint normal is more than 90 degrees.

Another extension of screw theory applied in analysis of kinematic restraint by point contacts was done by Kerr and Sanger [19, 20]. The authors used linear algebra theorems of linear independence to determine a total restraint condition. It was pointed out that the essential condition for total restraint is that there are sufficient linearly independent screws to span the column space. In the planar case, three of the four screws must be linearly independent. In the spatial case, seven screws are needed, and six of them must be linearly independent. For screws M_i , it is required that

$$\sum a_i M_i = 0 \text{ with all } a_i > 0. \quad (3.8)$$

The authors went further to note the importance of considering bi-directional screws of kinematic freedom. Any screws that are linearly dependent to the orthogonal set of constraints cannot restrain motion reciprocal to the set. The inner product between the screw motion and the restraining screw will indicate whether restraint is provided in a positive or negative sense. The mathematical method described is then used in the synthesis of point contacts to reduce remaining kinematic freedoms. In this dissertation,

the treatment and manipulation of constraint sets, screw motion, and bidirectional motion follows the general concepts introduced in this work.

3.2 Kinematic methods for fixture configuration planning by Bausch and Youcef-Toumi [7]

From this point forward, the work by Bausch and Youcef-Toumi [7] will be referred to as Bausch's due to its frequent use in this dissertation. As indicated, Bausch's algorithm in fixture constraint analysis and Bozzo's [9] implementation in assembly constraint analysis are the most appropriate starting points for the dissertation. There are infinitely many possible motions in 3D space. In analyzing part assembly, any screw system beyond a one-system results in an infinite linear combination of screw motions. Therefore, the number of motions would be impractical for computation. This set of motions can be discretized, but accuracy would be lost. The core concept in their methodology is to identify or generate the motion set by composing a wrench system from a combination of the constraints and to calculate the reciprocal screw from that system.

The methodology is able to reduce the infinite set of motions to a set of finite motions to evaluate by choosing a five-system and solving for the reciprocal one-system screw. This is true because the screw system reciprocal to an n^{th} order screw system is of order (6-n). Combinations of five unilateral constraints are picked in permutational sequence. These five wrench axes form a five-system. The reciprocal one-system screw motion consists of a unique screw axis in space with a unique pitch. This motion is

resisted by the rest of the constraints that do not belong to the five-system [23]. The resistance of this motion then can be evaluated by first computing the line-of-action vector at each of the resisting constraints. The resistance value is calculated by projecting the virtual displacement of the line-of-action onto the normal direction of the constraint.

In summary, this methodology is based on the following principles:

- A minimum of seven contacts is necessary to satisfy the condition for total restraint.
- Five linearly independent contacts will define a unique screw motion that is not constrained by any one of the five and is the only reciprocal motion possible for the body to maintain a sliding contact with each of the five.
- At least two contacts, one to prevent motion in a positive sense and one to prevent motion in a negative sense, are necessary and sufficient to constrain a unique screw motion.
- A workpart is totally constrained if and only if positive and negative motion stops exist for each and every reciprocal screw motion defined by five linearly independent lines of contact in the fixture.

The author used the reciprocity equation from Waldron [51] to solve for the free screw axis,

$$\mu \cdot \omega_i + \mu_i \cdot \omega = 0 \quad \forall i \in [1,5], \quad (3.9)$$

where μ_i and ω_i define the five system and the solution μ and ω define a single screw motion at the origin of the reference frame that needs to be resisted. Figure 3.2 is a graphical representation of the screw axis.

$$\$ = \rho + u\omega_u. \quad (3.10)$$

where ρ is the position of any point on the screw axis, ω_u is the unit direction vector of the screw axis, and u is a constant.

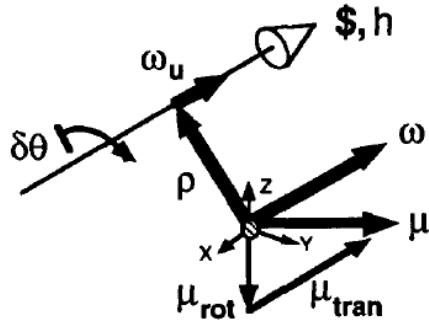


Figure 3.2 Graphical representation of the screw Axis [6]

As a side note, the five wrenches that compose the five-system must not belong to the same linear complex; they must be a linearly independent set. In order to evaluate this linear independence, the wrenches must first be expressed in Plücker line coordinates. Arranging them in matrix form, the rank of this matrix signifies the number of linearly independent wrenches in the set [18]. Given five linearly independent lines, a sixth line is linearly independent of all of them except when it belongs to the linear complex defined by them. Therefore, a condition that six given lines are linearly dependent is that their determinant is zero.

The motion stop computation is defined as a scalar quantity that indicates the effectiveness of a specific point contact in resisting a specific motion. With unilateral point constraints, one can only resist motion in the positive sense along the contact normal direction (N). Let δn be the virtual displacement into the constraint due to a twist $\delta\theta$, then,

$$\delta n = \mu \cdot (-N) = (h\omega_u \delta\theta + (\omega_u \delta\theta \times R)) \cdot (-N), \quad (3.11)$$

the motion stop value M is

$$M = \frac{\delta n}{\delta\theta} = (h\omega_u + (\omega_u \times R)) \cdot (-N), \quad (3.12)$$

$$M_{trans} = \frac{\mu}{|\mu|} \cdot (-N) \text{ for pure translation.} \quad (3.13)$$

The motion stop value calculation is illustrated in Figure 3.3.

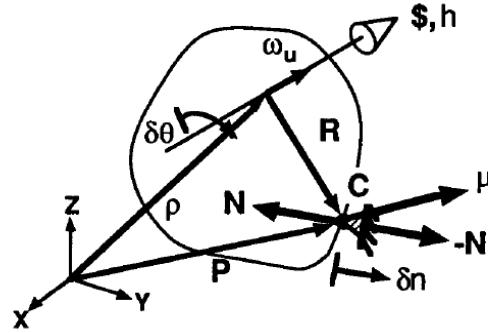


Figure 3.3. Motion stop value calculation for contact at C [7]

3.3 Constraint analysis of assembly by Bozzo [9]

Bozzo [9] applied Bausch's algorithm to create an assembly rating scheme. His work is one of the first to apply screw theory and constraint analysis to quantitative analysis of assembly. Most CAD systems are capable of testing for mating surface and dimensional consistency, but not constraint analysis. Unlike Adams and Whitney's application of screw theory to constraint analysis [266, 270], Bozzo's rating scheme is quantitative. In other words, the algorithm not only checks for unconstrained motion and total restraint, but it evaluates motion stop ratings according to Bausch's motion stop formula, which is similar to Asada's formula for virtual work. The analysis makes several assumptions. They are:

- The workpart and its constraints act as a rigid body.
- Contacts are frictionless.
- Finite area contacts are modeled as unilateral point constraints.
- Constraint positions and orientations in relation to the workpart are explicitly known and do not change throughout their service life.

In applying the constraint rating scheme, the author made several adaptations.

One adaptation is the linear solution simplification to the reciprocal screw problem.

Instead of solving the non-linear reciprocity equation using the Jacobian method proposed by Bausch and Asada, the author linearized the equation by setting $|\omega| = 1$.

Therefore, the equation becomes

$$\mu_1\omega_{1,i} + \mu_2\omega_{2,i} + \mu_3\omega_{3,i} + \mu_{1,i}\omega_1 + \mu_{2,i}\omega_2 + \mu_{3,i}\omega_3 = 0 \text{ for } \forall i \in 1,5, \quad (3.14)$$

$$\text{with } \omega_1^2 + \omega_2^2 + \omega_3^2 = 1. \quad (3.15)$$

Another adaptation is to model planar contact as point contact at the vertices. It is argued that infinite plane contact can be defined by 3 point contacts and finite planes by point contacts at the edge vertices. Thus, by defining the plane from its convex hull, or plane vertices, the planar characteristics of the constraint are captured. This is equivalent to Ohwovoriole's and DeMeter's remarks on planar contact [36, 15].

In handling an overconstrained system, the author commented on several limitations of this rating scheme. One limitation is that the rating scheme tends to put emphasis on individual constraint ratings, while in assembly all the constraints work simultaneously. Another limitation is that the number of motions to evaluate increases dramatically with increasing number of constraints.

The normalization factor proposed by Bausch [7], namely the largest distance between point constraints, is not constant across different constraint configurations. For example, if a constraint located at the farthest edge of a part is added into the constraint set, the largest distance between constraints increases. Therefore, there is a need for a better choice of normalization factor in order to remove this comparison limitation.

The author created a set of case studies designed to test the robustness, accuracy, and repeatability of the algorithm. Among these case studies are:

- Exactly constrained “Thompson’s chair”
- Cube with an unconstrained motion
- Cube with increasing number of constraints
- Sphere with free motion around its centroid
- Rectilinear geometry with typical snap-fit features
- Rectilinear geometry with planar contact
- Curved surface geometry

The author proposed an alternative algorithm because it was claimed that the original algorithm proposed by Bausch was not adequate to assess all possible weakly constrained motions. This alternate algorithm is called single constraint translation and rotation. The basic concept is to generate a set of motions that rotate the object around a single constraint and cause translation away from the constraint. This is an attempt to collect and evaluate a finite set of motions from an infinite number of motions. The reciprocal of a one-system wrench is five-system freedom motion, which consists of ∞^4 motions.

However, the claim for Bausch's algorithm's inadequacy or inaccuracy was based on the inconsistent scaling problem and other inaccuracies. These issues will be discussed in the following sections. It can be shown that this method does not address all motions reciprocal to the one constraint, which is a five-system. In addition, it is sufficient to use the original five point combination algorithm because the weakest motions are a subset of all systems reciprocal to the combinations of the five unilateral points.

3.4 Two major limitations of Bausch and Bozzo's model

Both the modeling of lines and planes and the scaling of screw motion rotation and translation are issues that need to be resolved. They are summarized below and discussed in the following subsections:

1. Lines and planes are modeled as discrete unilateral points. This leads to fictitious motion within a feature and suffers from inaccurate ratings. The solution to this problem is discussed in Chapter 4.
2. There is inconsistent scaling between finite pitch screw motion and pure translation. This is mainly related to the discontinuity between the screw magnitude for large finite pitch and infinite pitch. The solution to this problem is discussed in Chapter 5.

3.4.1 Modeling of line and planar constraints

Constraints in assembly almost always involve higher order constraints (HOC). HOC is defined as any constraint that is not a unilateral point constraint. In this

dissertation, they refer to pin, line, and planar constraints. In Bozzo's work, these HOC are discretized into many point constraints. The first problem occurs when individual point constraints belonging to a single feature are taken apart from the rest of the group. Consider an example (Figure 5) where the pin constraint is discretized into 8 point constraints and only two of them, along with 3 others (from different constraint type) are selected. The motion generated would not be possible in the real part. This is because the two point constraints should not be considered without the rest of the group spanning the pin constraints. This is later illustrated in a case study in Section 10.4.2. In addition, the pin is rated as 8 different numerical values. The way in which these 8 values are processed to yield a single rating for the pin constraint is very critical to the accuracy of the analysis. Furthermore, the motion set generated from the comprehensive combination will be dominated by the "partial" pin constraints

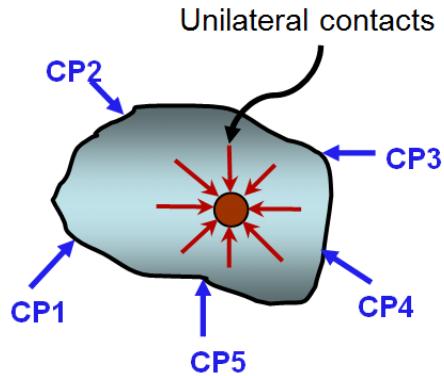


Figure 3.4 An example of representing pin contact as many unilateral contacts

Note that the accuracy of the analysis will increase as the step size of discretization gets finer. The second problem then, occurs as each HOC is discretized so

finely that the total number of point constraints increases significantly. This will lead to an overwhelming number of combinations/iterations to evaluate in the algorithm. A significant proportion of these constraints are from the discretization of higher order constraints alone. The number of combinations generated from n constraints will increase according to

$$C(n, 5) = \frac{n!}{5!(n-5)!}. \quad (3.16)$$

It can be observed that the number of generated motions increases in a factorial manner with increasing number of constraints. With current computing technology, the computation time becomes impractical once the number of constraints goes beyond 44 (approximately 1 million possible combinations, see Figure 3.5). This problem should not be solved simply by creating a more efficient algorithm, but it needs a new methodology to circumvent this problem.

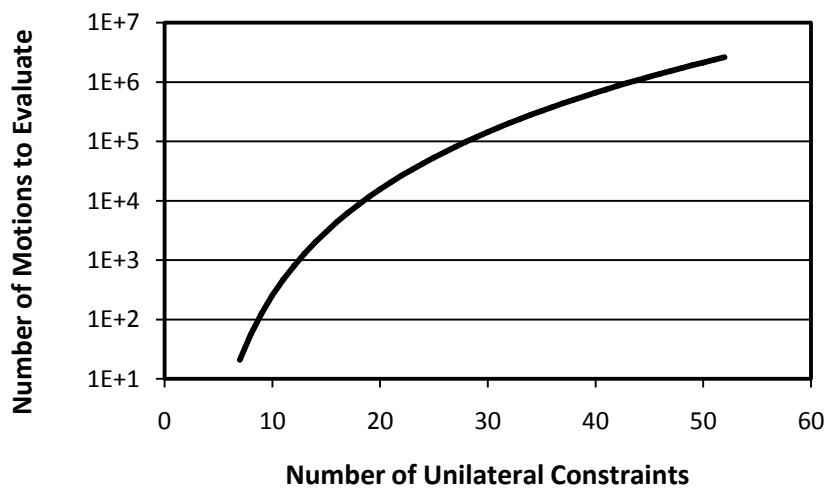


Figure 3.5 Relationship between number of motions and number of unilateral constraints

3.4.2 Inconsistent scaling between finite pitch screw motion and pure translation

The motion stop calculation by Bausch essentially calculates the virtual penetration of the screw motion into the constraints. The virtual penetration model is purely geometric in nature. In order to do this, the instantaneous velocity of the screw motion at the constraint point is calculated. This instantaneous velocity is projected onto the constraint normal direction. The greater the magnitude of the velocity in the normal direction, the more effective resistance the constraint provides. The magnitude of this component of the velocity is the resistance value of the motion provided by the individual constraint.

Since the set of five wrenches cannot do any virtual work to resist the motion, the motion stop value for these five constraint points is zero. For any evaluated motion, there must be at least five-constraint points that are rated as zero, namely the constraint points that belong to the five-system and any other wrenches that are linearly dependent to this set.

In Bausch's model, the motion stop value is calculated by the formula specified in Equation 3.12, depending whether it's a finite pitch screw or a pure translation. Pure translation is treated differently with the auxiliary Equation 3.13. The value M is the ratio between the virtual penetration due to an input angle θ around the screw axis.

One of issues Bozzo identified with this rating scheme is that it does not have a normalization factor taking the pitch into account. Bozzo deals with this absence of normalization factor by dividing the moment M by a parameter λ

$$M_{norm} = \frac{M}{\lambda}, \quad (3.17)$$

where λ is the longest distance between any two points on the workpart.

The main problem with both rating schemes is that they do not have a continuous scale between the main equation for a regular twist and the auxiliary equation for a pure translation. For example, compare and contrast the two cases where h is very large and h is equal to infinity

$$\text{Case 1: Let } h = 10000, M = (10000\omega_u + (\omega_u \times R)) \cdot (-N). \quad (3.18)$$

$$\text{Case 2: Let } h = \infty, M = \mu_u \cdot (-N). \quad (3.19)$$

ω_u , μ_u , and N have unit magnitude. In case 1, a twist with a very large pitch (close to a pure translation, but still finite pitch) has a magnitude of 1 unit for the rotational component and 10000 units for the translational component, yielding very high virtual displacement magnitude and hence very high rating. In case 2, a pure translation with infinite pitch is modeled as a vector with magnitude of 1 unit for the translational component. As a consequence, in case 1, M has a magnitude on the order of 10^4 , while in case 2, M is on the order of 10^0 (between 0 and 1). As h approaches infinity, the two equations have a discontinuity in scaling due to the different treatment. Therefore, the rating for the pure translation has a totally different metric and scaling value than M . Figure 3.6 shows the magnitudes of the rotational and translational components as the screw pitch increases.

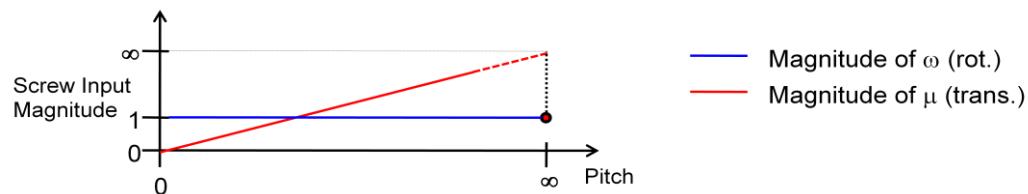


Figure 3.6 Screw input magnitude vs. pitch

The two equations also yield inconsistent units. The input unit for case 1 is length/angle, and the input unit for case 2 is length/length. In addition, this problem makes the rating scheme not scalable. As the part size grows, the maximum value of M grows proportional to the part size as the moment arm increases, while the maximum value of M in the pure translation case remains capped at 1. This is addressed somewhat in Bozzo's scheme, but it is still subject to different scaling as the units chosen are changed (for example, when using metric instead of English units). The inconsistency between the units and scaling makes comparison between zero-to-finite pitch screw motion and pure translation (infinite pitch) impossible. This prevents the proper identification of the most weakly constrained motion in the part assembly.

There are other variations of this method by Asada [4]. The alternative is to calculate the virtual work of the reaction force to resist a unit twist. Although it has slightly better physical significance, this method suffers from exactly the same problems as the previous method. The core of the problem lies in the fact that screw theory is not designed to consider screw magnitude as a variable in the computation; it is generally set to be unity in magnitude.

3.5 Other minor issues with Bausch and Bozzo's model

There are other issues with the work of Bausch and Bozzo that are less fundamental in nature. While the main issues need a completely different way to model assembly constraints and to calculate resistance quality, the issues discussed in the following subsection revolve around the computational method. Nevertheless, these

issues yield false ratings and incorrect weakly constrained motion identification. Hence, their impact is not negligible.

3.5.1 Inaccurate post-processing of rating matrix

In the final rating matrix, the rows correspond to screw motions being evaluated and the columns correspond to contact points. The extreme rating scheme is defined to be the lowest absolute value of the whole rating matrix. In addition to the overall extreme rating of the whole matrix, the individual point ratings are calculated by observing the lowest absolute values in each column. The problem with this rating scheme can be illustrated in Table 3.2

	CP1	CP2	CP3	CP4	CP5	CP6	CP7	CP8	CP9
Motion 1	0.113	0	0.802	0.675	0.779	0	0	0	0
Motion 2	0.259	0	0.214	0	0.374	0	0	0.231	0

Table 3.2. Example of Rating Scheme Inconsistency

In the rating matrix, the lowest absolute value for motion 1 is 0.113, which refers to CP1's resistance value, while the lowest absolute value for motion 2 is 0.214, which refers to CP3. According to the “extreme” rating scheme proposed, motion 1 would be weakly constrained compared to motion 2. However, an observation of the rest of the values in the first row shows that motion 1 is relatively well constrained by CP3 (0.802), CP4 (0.675), and CP5 (0.779). Motion 1 is not necessarily weakly constrained because CP3 has a very good resistance value. On the other hand, an observation of the rest of the values in the second row shows that for motion 2, the best resistance is offered by CP5 (0.374). This is low compared to the best resistance to motion 1 (0.802). Therefore,

motion 2 is actually the more weakly constrained of the two. The rating should be based on the best resistance instead of the lowest absolute value.

Another rating metric used by the authors is judging the individual constraint's effectiveness by taking the lowest absolute value of the respective constraint for all motions (lowest value for each column). This is not accurate for reasons similar to the previous case. The fact that the lowest absolute value is small for a particular constraint does not mean that it is not giving a good resistance for the rest of the screw motions (the rest of the values in the same column). In fact, if one particular contact point has a very low absolute value in one motion while it is giving very good and critical resistance in other motions, this constraint is very critical.

3.5.2 Failure to calculate ratings for bilateral motion separately

In many mechanisms, such as four-bar and slider-crank mechanisms, all joints are bilateral as opposed to unilateral. The constraints prevent motion in two directions. Therefore, each constraint's screw representation does not need a sign to determine its direction. In typical assembly with unilateral constraints, the direction of motion matters. The freedom motion twist from a five-system is bidirectional. The rigid body can move forward or backward along the twist path.

In Bozzo's work, the ratings are evaluated by summing the negative ratings separately from the positive ratings. The lowest absolute value of the ratings is taken to be what is called the "extreme" type rating. Since the forward and backward motion is

evaluated simultaneously, there is no distinction between the two ratings. In addition, the exclusive summation of positive and negative values is computationally costly.

3.6 Objectives and Scope of Research

In summary, research in kinematic analysis using screw theory has produced useful tools in analyzing robotic grasps, fixture design, and total restraint. However, many of the principles used in these methods are not applied in mechanical part assemblies. Determination of total restraint by Adams and Whitney [2] is not useful in practical assembly design because it is not capable of analyzing redundant constraints typical in real part assembly. Among the design tools developed to determine whether an assembly has form closure, force closure, or total restraint, none of the literature creates a quantitative metric of assembly quality.

In recent years, Bausch [7] and Bozzo [9] attempted to address quantitative assembly constraint analysis in mechanical part assembly, but this rating tool has major issues of inconsistency and needs much modeling improvement. This research is an extension of these recent attempts to analyze assembly strategies. It aims to develop a design and analysis tool to address the need for a scientific approach to attachment strategy. This objective can be divided into two parts, namely assembly constraint analysis and assembly constraint synthesis or design.

3.6.1 Assembly constraint analysis

The objective in the analysis part of this research is to develop an analysis tool that addresses the main issues by producing an accurate and consistent model. The model and rating metric will provide designers an analysis tool that calculates a physically meaningful, accurate, and scalable rating scheme. This analysis tool identifies the motions reciprocal to the constraint configuration and evaluates the resistance to these motions. Different rating metrics are developed to give different ways to measure assembly quality. Each rating metric has different applications in design decisions. The additional analysis tool of sensitivity analysis due to perturbation in location and orientation is also provided to assess assembly robustness toward dimensional variance in manufacturing.

As a starting point, this dissertation adopts the methodology from Bausch in its basic form, which is to:

1. Pick a combination of constraints in order to form a five-system wrench, called the *pivot constraint* combination.
2. The motion reciprocal to all possible five-system wrenches is the evaluated motion set.
3. The effectiveness of the remaining constraints, called the *reaction constraints*, to restrain this reciprocal motion set is evaluated.

The following assumptions are made:

- The part and its constraints act as a rigid body with no deformation.

Therefore, constraint locations and orientations do not change during motion resistance.

- Contact interfaces are frictionless.

Chapter 4 discusses the methodology used to model assembly constraints using kinematic screw theory and to generate the set of reciprocal motions to be evaluated. This model provides the solution to the major limitation of modeling higher order constraints.

Chapter 5 discusses the methodology used to calculate the resistance quality of the assembly constraint configuration to resist the set of motions generated. This model provides the solution to the inconsistency in Bausch's and Bozzo's methodology. Chapter 6 discusses the post-processing of the resistance values to yield the overall rating metrics of assembly quality. Chapter 7 discusses in detail the implementation and calculations involved in the methodologies discussed in Chapters 4 to 6.

3.6.2 Assembly Constraint Synthesis / Design

This dissertation also aims to develop a design tool that utilizes the analysis tool to explore possibilities in improving assembly design. This objective is approached by first defining possible design spaces in assembly attachment strategy. This is composed of various location and orientation search spaces. The search space is to be specified by the user based on the geometry of the part.

The design tool is used in a series of case studies that verify commonly known design principles and study the trade-off in constraint redundancy and assembly quality.

In doing so, this dissertation also produces domain-specific knowledge in assembly design. Finally, the case studies also demonstrate that the design tool can be used with different objectives depending on the design context such as modifying, adding, or reducing constraints in the configuration. The ultimate goal of the design tool is to be able to find optimum solutions for assembly constraint design within the design space.

CHAPTER 4

MODELING ASSEMBLY CONSTRAINTS USING HIGHER ORDER CONSTRAINT MODEL

4.1 From contacts into constraints

As mentioned in Chapter 3, the basic methodology by Bausch [7] and Bozzo [9] is the most appropriate starting point for this research. The methodology is useful for analyzing assembly due to the following reasoning. In order to evaluate the quality of an assembly or constraint configuration to resist arbitrary loads, a finite set of motions to be evaluated must be identified. Since there are ∞^5 motions in 3D space, there is a need for an algorithm that reduces this infinite set to a finite set.

In utilizing screw theory for evaluating assembly, assembly features that remove DOF need to be modeled as wrenches. In assemblies dominated by integral attachment features, most DOF removal is provided by contact interfaces. Contact interfaces can exist between locking features and mating parts or between mating surfaces that are intentionally designed for load transfer or aesthetic purposes. Contact interfaces that remove DOF in integral attachment features can be point, pin, line, or planar contacts.

One needs to note that there is no clear distinction between these different categories. For example, when the area of planar contact is small enough, it can be considered a point contact, hence modeled as a single unilateral constraint. The pin, line, and planar constraints are considered higher order constraints (HOC) because they remove more than one DOF.

All types of assembly contact interfaces are modeled as higher kinematic pairs. It is important to note, however, that all of these contact interfaces, with the exception of pin contact, are unidirectional or unilateral in their ability to restrain motion. In a sense, there are 12 DOF because of this fact. This chapter deals with the modeling transition from assembly contact interface to mathematical wrench constraint representation.

4.2 Higher order constraints (HOC) modeling

Bausch's methodology was applied to fixture configuration evaluation in which all the fixtures and clamps are adequately modeled as unilateral point constraints. It did not address higher order contact interfaces that remove more than one DOF per feature. In attempting to adopt Bausch's methodology to integral attachment assembly, Bozzo modeled pins, lines, and planes as an array of discrete point constraints. With the exceptions of short lines and small planes, modeling pin, line and planar constraints with discrete point constraints leads to several problems explained in Section 3.4.1. Therefore, there is a need for better modeling of higher order constraints in assembly. To address this need, higher order constraints (HOC) are modeled using higher order wrench

systems. The following subsections identify the wrench equivalent of four types of assembly constraints addressed in this thesis.

4.2.1 Modeling point contact as a one-system

A point contact is defined as a constraint that can exert a one-way reaction force along its normal direction in compression, but not in tension. A point constraint removes one DOF, namely translation in the normal direction. The equivalent wrench system is a single zero-pitch wrench coincident with the normal vector of the constraint (Figure 4.1). This is a one-system. Short line contact and small planar contact areas can also fall into this category and be modeled as point constraints at the mid-point. Equation 4.1 describes the wrench system in twist notation.

$$\vec{S}_\alpha = \begin{bmatrix} 0 \\ \hat{\omega}_\alpha \end{bmatrix}. \quad (4.1)$$

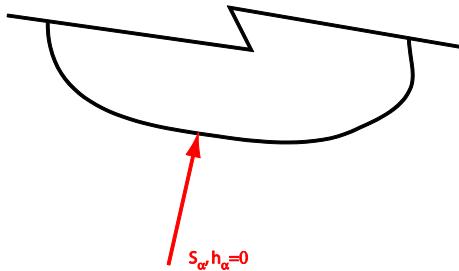


Figure 4.1 Point Contact

4.2.2 Modeling pin contact as a two-system

A pin contact is defined as a constraint that can exert a two-way reaction force in the radial direction. These reaction forces lie in the plane normal to the pin axis. A pin

contact removes two DOF, namely translation along the two principal axes perpendicular to the pin axis. The equivalent wrench system is composed of ∞ zero-pitch wrenches that are co-planar and intersecting at a point. This is the first-special two-system. It is composed of a planar pencil of wrenches in the radial direction. The principal wrenches of this system are two perpendicular wrenches of zero pitch in the planar pencil (Figure 4.2). Equation 4.2 describes the wrench system in twist notation.

$$\vec{S}_\alpha = \begin{bmatrix} 0 \\ \hat{i} \end{bmatrix}, \vec{S}_\beta = \begin{bmatrix} 0 \\ \hat{j} \end{bmatrix}. \quad (4.2)$$

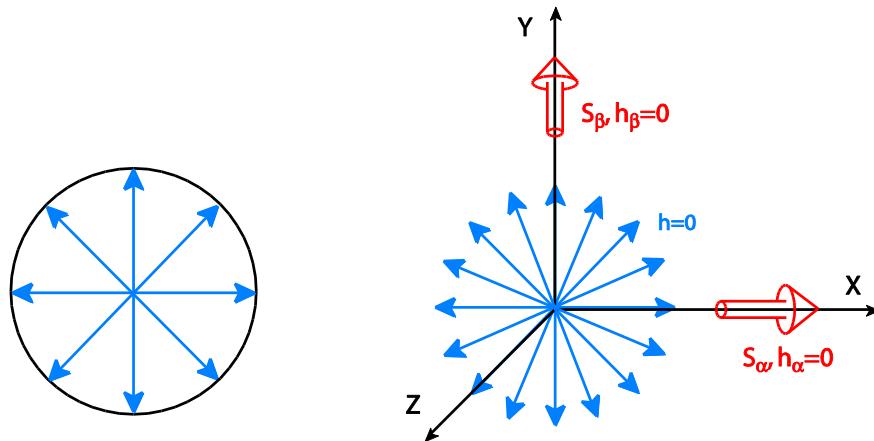


Figure 4.2 Pin Contact Wrench Representation

Pin contact interfaces in assembly tend to be short in depth and are not designed to take moment loads. Therefore, it is appropriate to mathematically model the pin constraint with zero thickness. It cannot exert reaction moments transverse to the pin axis. Hence, a pin constraint is different from a cylindrical constraint. The cylindrical constraint can exert reaction moments to restrain rotation transverse to the pin axis in addition to the two translational constraints. Since pin constraints only constrain

translation in the radial direction, the constraint effectiveness is invariant with respect to radius.

4.2.3 Modeling line contact as a two-system

A line contact can exert a one-way reaction force in the normal direction along the line. A line constraint can be represented as a uniformly distributed unilateral constraint along a finite line, oriented normal to the mating surface. The equivalent wrench system is composed of ∞ parallel and coplanar zero-pitch wrenches (Figure 4.3). This is the second-special two-system. The general form of this special system is defined by a finite pitch wrench coincident with the local x-axis and an infinite pitch wrench oriented at an angle θ from the x-axis and lying in the XY plane. Equation 4.3 describes this special form.

$$\vec{S}_\alpha = \begin{bmatrix} h_\alpha \omega_\alpha \hat{i} \\ \omega_\alpha \hat{i} \end{bmatrix}, \vec{S}_\beta = \begin{bmatrix} \mu_\beta (\cos\theta \hat{i} + \sin\theta \hat{j}) \\ 0 \end{bmatrix}. \quad (4.3)$$

$$h = h_\alpha + \rho \cot\theta. \quad (4.4)$$

The line constraint is a special case when $h_\alpha = 0$ and $\theta = \pi/2$. The first principal wrench for this system is a zero-pitch wrench perpendicular to the line in the normal contact direction. The second principal wrench is an ∞ -pitch wrench perpendicular to and intersecting the first principal wrench. Equation 4.5 describes the special case for the wrench system in twist notation.

$$\vec{S}_\alpha = \begin{bmatrix} 0 \\ \hat{i} \end{bmatrix}, \vec{S}_\beta = \begin{bmatrix} \hat{j} \\ 0 \end{bmatrix}. \quad (4.5)$$

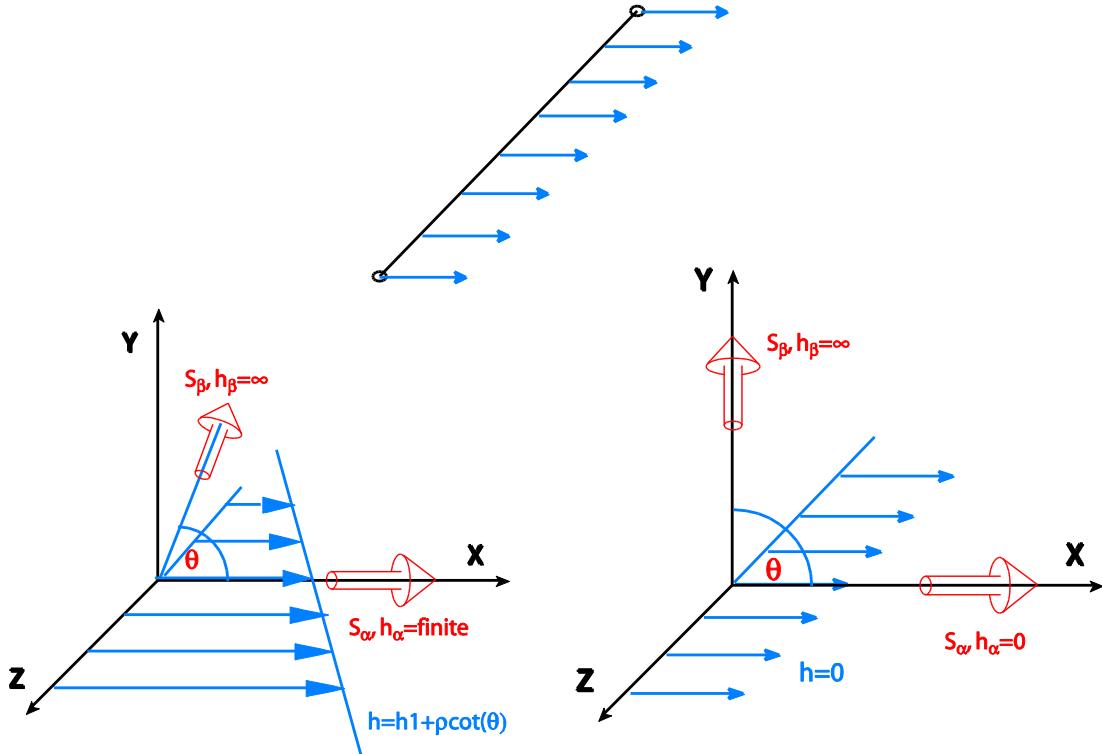


Figure 4.3 Line Contact Wrench Representation

4.2.4 Modeling plane contact as a three-system

A planar contact can exert a one-way reaction force in the direction normal to the plane. A plane constraint can be illustrated as a uniformly distributed unilateral constraint normal to a finite plane. The equivalent wrench system is composed of ∞^2 parallel zero-pitch wrenches (Figure 4.4). This is the seventh-special three-system. This system is defined by a finite pitch wrench coincident with the local x-axis, an infinite pitch wrench oriented at an angle θ from the x-axis and lying in the XY-plane, and another infinite pitch wrench oriented at an angle α from the x-axis and lying in the XZ-plane. This special form is described in Equations 4.6 and 4.7.

$$\vec{S}_\alpha = \begin{bmatrix} h_\alpha \omega_\alpha \hat{i} \\ \omega_\alpha \hat{i} \end{bmatrix}, \vec{S}_\beta = \begin{bmatrix} \mu_\beta (\cos\theta \hat{i} + \sin\theta \hat{j}) \\ 0 \end{bmatrix}, \vec{S}_\gamma = \begin{bmatrix} \mu_\gamma (\cos\alpha \hat{i} + \sin\alpha \hat{k}) \\ 0 \end{bmatrix}. \quad (4.6)$$

$$h = h_\alpha + \frac{\mu_\beta}{\omega_\alpha} \cot\theta + \frac{\mu_\gamma}{\omega_\alpha} \cot\alpha. \quad (4.7)$$

The plane constraint is a special case when $h_\alpha = 0$ and $\theta = \alpha = \pi/2$. The first principal wrench for this system is a zero-pitch wrench normal to the plane. The second and third principal wrenches are ∞ -pitch wrenches perpendicular to the first principal wrench and to each other. The three principal wrenches create an orthogonal set of principal wrenches.

A curved line contact or a circumferential line contact can also be modeled as a plane contact because it spans a set of parallel wrenches but not coplanar. Therefore, it is not equivalent to straight line contact. Equation 4.8 describes the wrench system in twist notation.

$$\vec{S}_\alpha = \begin{bmatrix} 0 \\ \hat{i} \end{bmatrix}, \vec{S}_\beta = \begin{bmatrix} \hat{j} \\ 0 \end{bmatrix}, \vec{S}_\gamma = \begin{bmatrix} \hat{k} \\ 0 \end{bmatrix}. \quad (4.8)$$

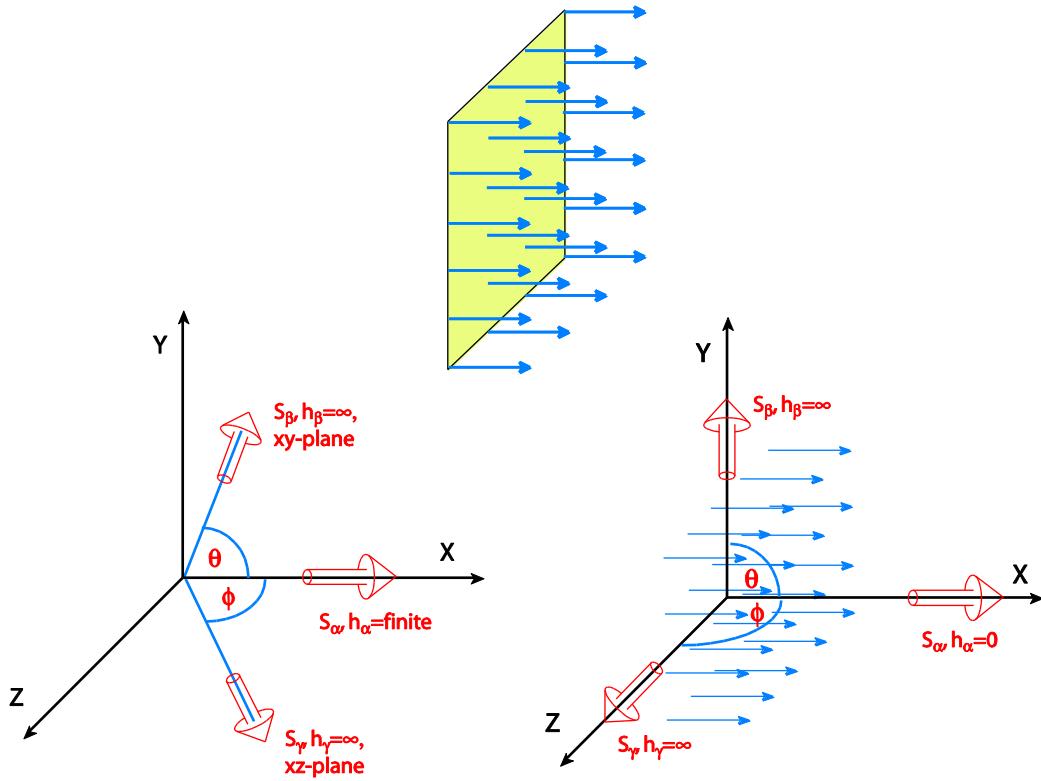


Figure 4.4 Plane Contact Wrench Representation

4.2.5 Modeling other assembly constraints

There are two additional assembly constraints that can be modeled as HOC, namely the threaded fastener and a long pin constraint. They are identified in this section but not implemented in the code. The reason is that integral attachment oriented assembly is not dominated by these contact types. In addition, these additional contact types add much complexity to the code and beyond the scope of this dissertation.

The first is the threaded fastener constraint. The word ‘threaded fastener’ is used to prevent ambiguity with the term ‘screw’ in screw theory nomenclature. A threaded fastener constraint in assembly can be modeled as a pin constraint with two additional

opposite point constraints along the axis. The remaining DOF's are rotation around the three principle axes. The rotation about the fastener axis is free because of the frictionless assumption. The rotation transverse to the fastener axis is free because the bolt head diameter is small enough and the wall thickness is thin enough. Hence, the fastener does not effectively resist the moments about the two transverse axes. The equivalent wrench system is composed of ∞^2 infinite-pitch wrenches in all direction. This is the fifth-special three-system. This system is defined by three orthogonal infinite pitch wrenches.

Mathematically, it is equivalent to a ball-in-socket joint. Equation 4.9 describes the wrench system in twist notation.

$$\vec{S}_\alpha = \begin{bmatrix} \hat{i} \\ 0 \end{bmatrix}, \vec{S}_\beta = \begin{bmatrix} \hat{j} \\ 0 \end{bmatrix}, \vec{S}_\gamma = \begin{bmatrix} \hat{k} \\ 0 \end{bmatrix}. \quad (4.9)$$

The second is a long pin contact found in hinges or dowel pins. This is modeled as a cylindrical constraint. The cylindrical constraint removes four DOF and allows only two-DOF motion, which is rotation around the pin axis and translation along the axis. The equivalent wrench system is the third-special four-system. A four-system is specified by its reciprocal two-system. The reciprocal of this four-system is a screw axis coincident with the pin axis with any pitch. This reciprocal screw is the third special two-system (Equation 4.10).

$$\vec{S}'_\alpha = \begin{bmatrix} 0 \\ \hat{i} \end{bmatrix}, \vec{S}'_\beta = \begin{bmatrix} \hat{i} \\ 0 \end{bmatrix}. \quad (4.10)$$

4.3 Modeling gaps between screw theory and assembly constraint

The wrench equivalents for each type of assembly contact interface have been identified, but there are two distinct differences between wrenches in screw theory and assembly contacts in general. These are especially applicable in line and plane constraints. The two distinctions are discussed in the subsections below:

1. Wrenches in screw theory provide bidirectional DOF removal while assembly contact interfaces provide unidirectional DOF removal.
2. Wrenches in screw theory always have infinite span length while assembly contact interfaces are always finite in length and width.

4.3.1 Bidirectional vs. unidirectional DOF removal

Assembly contact interfaces only resist motion in the compressive direction, not in the tension direction away from the contact. By the definition of reciprocity, the pivot wrench set cannot do work with respect to the reciprocal screw motion. This means that the motion “slides” along the pivot constraints’ normal. The motion will not break contact from the wrenches because mathematically any motion that is not in the null space of the pivot wrench set will produce virtual work, positive or negative. Although an assembly contact interface physically allows motion away from the constraint, it is assumed that the constraint configuration exerts reaction forces that always act to maintain contact throughout the infinitesimal motion. This is known as force closure [60]. This assumption essentially restricts the motion to only “slide” along the assembly

contact interfaces. Therefore, modeling the one-way assembly contact interfaces as a two-way wrench system arrives at the identical reciprocal screw motion.

Furthermore, the motions are separated into forward and reverse screw motions when the individual resistance quality is calculated. All constraints are modeled in the capacity of exerting one-way reaction forces. The bidirectional motion calculation is discussed in Chapter 6, and the resistance value calculation is explained in Chapter 5.

4.3.2 Infinite span vs. finite span length

It can be argued that the wrench system equivalents, the line and planar contacts in particular, span infinite lengths, while the physical constraints are finite in length. Both the finite line and the infinite line belong to the same wrench system. This can be illustrated using an example. Consider two finite line contacts, one has a length twice the other, but the two are coincident in space. Let two linearly independent wrenches be picked from each line and used to calculate the reciprocal motion allowed by the lines. This motion will be identical because they span identical wrench systems. As a result, when the pivot wrench set contains at least two wrenches from the two-system spanned by the line contact, the reciprocal screw solution is invariant with respect to the line length.

There remains a need for a modeling guideline to determine the transition point where a line is short enough to be modeled as a single point constraint or long enough to be considered a two-system wrench. The guideline needs to evaluate when the line is too short to assume that rotation transverse to the line should be evaluated in the set. This

guideline is specified by the ratio between the length of the line and the longest distance between constraint points in the analyzed part. The above explanation can also be extended to apply to the three-system planar constraint.

Furthermore, during the resistance value calculation, all lines and planes are indeed modeled as finite in length. The resistance value calculation model is discussed in Chapter 5. In that chapter, all constraint features are treated and modeled in their true length and will have their ratings proportionally affected by the actual lengths of the particular features.

4.4 Reciprocal screw problem solution

In Bozzo's work, the set of 5 linear equations is augmented with a sixth equation specifying the magnitude of either ω_1 , ω_2 , or ω_3 . The linear simplification of the reciprocal screw problem is stated in $Ax=b$ form, plus a non-linear equation forcing the magnitude of ω to be one. In the infinitesimal kinematics model, the magnitude is not relevant to the analysis. This equation is then solved by calculating an explicit inverse in general, or using a linear programming optimization algorithm for pure translation.

A new and simpler solution method is introduced in this dissertation by calculating the null space for $Ax=0$ shown in Equation 4.5.

$$\begin{bmatrix} \omega_{1x} & \omega_{1y} & \omega_{1z} & \mu_{1x} & \mu_{1y} & \mu_{1z} \\ \omega_{2x} & \omega_{2y} & \omega_{2z} & \mu_{2x} & \mu_{2y} & \mu_{2z} \\ \omega_{3x} & \omega_{3y} & \omega_{3z} & \mu_{3x} & \mu_{3y} & \mu_{3z} \\ \omega_{4x} & \omega_{4y} & \omega_{4z} & \mu_{4x} & \mu_{4y} & \mu_{4z} \\ \omega_{5x} & \omega_{5y} & \omega_{5z} & \mu_{5x} & \mu_{5y} & \mu_{5z} \end{bmatrix} \begin{bmatrix} \mu_{\mu_{6x}} \\ \mu_{\mu_{6y}} \\ \mu_{\mu_{6z}} \\ \omega_{\omega_{6x}} \\ \omega_{\omega_{6y}} \\ \omega_{\omega_{6z}} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \quad (4.11)$$

The reciprocal screw motion is in the null space of the wrench system. The reciprocal screw motion is then obtained by solving for the null space of the pivot wrench set. The dimension of the null space is $6-m$ for m linearly independent wrenches. m is always equal to 5 because the pivot wrenches must be a linearly independent set. In other words, the rank of the column space is always 5. Therefore the dimension of the null space is one, namely the unique screw motion reciprocal to the wrench system.

4.5 Advantages over previous models

The model used in this dissertation provides several advantages to the kinematic analysis accuracy and efficiency. Representing higher order continuous contact as a higher degree wrench system enables the motion generation routine to be executed in a much more accurate and efficient manner, as quantified subsequently in this section.

This model eliminated two problems:

1. The problem of the growing number of iterations due to false internal combinations previously discussed in Section 3.4.1. There are two kinds of unnecessary and inaccurate solutions/iterations. One occurs when linearly dependent sets are picked. This occurs when a single wrench from an HOC is processed apart from its set. Using a set of discretized unilateral constraints to represent the higher order constraints creates many combinations consisting of linearly dependent sets of screws. The other occurs when the iteration selects multiple screws that are linearly independent, but creates a screw system that has been used before. This is because a single constraint set (i.e. lines or

planes) can only be represented by one unique screw system. In either case, the unnecessary iterations will either fail to solve the reciprocal equation or generate a duplicate motion which takes significantly more computation time.

The following is an example to quantify the computation time savings.

Consider a part assembly having a few of the constraints as HOC. The assembly is composed of 8 point constraints, 1 pin constraints, 2 line constraint, and 1 plane constraints. In Bozzo's model, HOC are discretized into point constraints. These constraints then are discretized into 8 (point) + 4 (to represent pins) + 8 (to represent lines) + 4 (to represent plane) = 24 point constraints. The discretization results in 42504 iterations. This is a very conservative approach because each HOC is only discretized into 2-4 points per HOC. By using the current model, there are 12 constraints of various types. This results in 792 iterations to address. In this example, the number of iterations is reduced by 1/50 or 98%. By representing these constraints using HOC, the number of total iterations can be reduced up to two orders of magnitude.

2. The pseudo-redundancy effect to the rating metrics in higher order screw systems previously discussed in Section 3.4.2. Because HOC are modeled as many discrete point constraints, the rating metrics are subject to multiple resistance values that actually belong to the same physical constraints. Rating metrics that sum these values then convey the false conclusions that an assembly is very strong and/or redundant in its resistance to motion. This

model also has additional advantages that result in truly higher ratings because it eliminates the pseudo-redundancy effect. This is manifested in their reaction force capacity being higher than point constraints.

4.6 Summary

In summary, this model allows most integral attachment oriented assemblies to be abstracted into point, pin, line, and plane constraints. These constraints are then translated into the equivalent wrench system identified in this chapter. The details of how this model is implemented in the code are discussed in Chapter 7.

CHAPTER 5

EVALUATION OF MOTION RESISTANCE QUALITY USING ISOLATED REACTION FORCE MODEL

5.1 Virtual penetration model vs. isolated reaction force model

The effectiveness of each reaction constraint to resist a particular motion can be measured in two ways. One way is to calculate the virtual penetration of the motion line-of-action onto the reactive constraint in the normal direction. The greater this value, the more effective the constraint is in resisting the motion. The other way is to calculate the reaction force of the constraint. A constraint that resists motion with less reaction force is more effective. The smaller this value, the more effective the constraint is in resisting the motion. These two effectiveness metrics of a constraint provide the same information, but are inverted in scale.

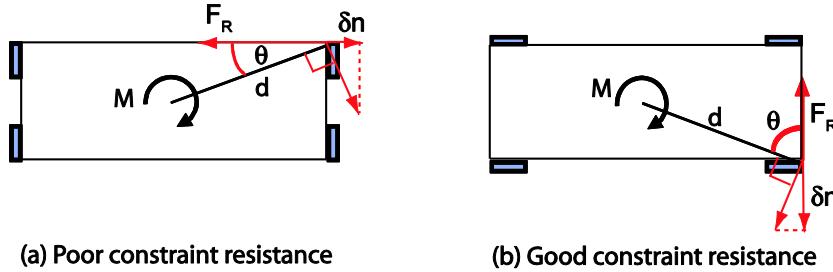


Figure 5.1 Poor vs. good motion resistance

Figure 5.1 shows two cases of motion resistance. Case (a) shows poor resistance quality to motion, and case (b) shows good resistance quality for the same motion. In both cases, the location of the constraint is identical, so the moment arm length is equal (d is equal for both cases). The difference in resistance quality is therefore affected by the orientation of each constraint, which is the angle between the line-of-action and the constraint normal.

For identical constraint locations, the virtual penetration is proportional to the cosine of the angle between the moment arm and the constraint normal, while the reaction force is proportional to the sine of the angle. In Figure 5.1, it can be observed that the virtual penetration δn is proportional to $d * \cos(\theta)$. Therefore, case (a) has less virtual penetration than case (b). The reaction force F_R is proportional to $\frac{\sin(\theta)}{d}$. Therefore, the reaction force in case (a) is greater than case (b). The relationship between the evaluation metric and constraints' effectiveness is summarized in Table 5.1.

	Virtual penetration	Reaction Force
Good constraint effectiveness	Large	Small
Poor constraint effectiveness	Small	Large

Table 5.1 Constraint Effectiveness Metric

It has been discussed in Section 3.4 that the main issues of inconsistency in units and scaling are prevalent in the virtual penetration method used by Bausch [7] and Bozzo [9]. Since both methods deliver the same information content, using reaction force as an evaluation method for constraint effectiveness is a potential alternative to the virtual penetration method.

There are further modifications needed to implement the reaction force evaluation method for producing a consistent and realistic rating scheme. The first involves modeling the input torque as a force couple. The second is the scaling of the input wrench magnitude based on the magnitude of the pitch. The rest of this chapter will discuss the implementation of the isolated reaction force model in more detail.

5.2 The isolated reaction force model

The method used to evaluate constraint effectiveness is called the isolated reaction force model. Reaction forces at the constraints are directly proportional to the input load. In this model, the resistance value is calculated as the ratio between the isolated reaction force magnitude at the constraint point and the input wrench magnitude exerted along the screw motion path. The lesser this ratio, the more effective the constraint is in resisting motion.

Conceptually, the motion resistance is modeled after Yoshikawa's passive force closure model [60]. Yoshikawa stated that "A constraint is said to be passive closure if the current position and orientation of the object is maintained, even when an arbitrary external force is applied on the object, without changing the joint driving force of the

constraining mechanism ... passive closure implies that the balancing force counteracting the applied external force is produced by the mechanism itself of the constraining mechanism.”

Figure 5.2 shows a generic example of passive force closure for a 2D case. There are two types of constraints in this model. The first type are a rigid constraints (C_1 and C_2) responsible for maintaining the location of the object. This is usually referred to as locators in fixture design. In this dissertation, these are called pivot constraints/wrenches, referring to the constraints that form the five-system to which the motion is reciprocal. The second type is an active constraint that provides the balancing force counteracting the applied external force and therefore possibly restrains motion depending on the motion direction (C_3 and C_4). C_3 in this case is inactive in providing restraint to the motion prescribed. In this dissertation, these are called reaction constraints/wrenches, referring to the rest of the constraints that provides restraint to the motion.

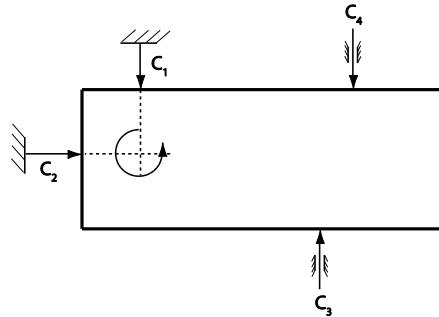


Figure 5.2 Passive force closure model

The reaction forces are calculated by setting up a static equilibrium between the input wrench \vec{L} and the constraining wrenches \vec{C}_i . The equilibrium is set up with a point

at the motion screw axis ρ as the origin. Taking ρ as the origin simplifies the calculation for the input wrench along the motion screw

$$\vec{L} + \vec{C} = \vec{0}. \quad (5.1)$$

The constraining wrench \vec{C} is composed of the pivot wrench $\vec{C}_{P1,P5}$ (five-system) and the reaction wrench \vec{C}_R (one-system). Substituting this into the equilibrium equation,

$$\vec{L} + [\vec{C}_{P1,P5}, \vec{C}_R] = \vec{0}, \quad (5.2)$$

$$\begin{bmatrix} \vec{\tau}_L \\ \vec{f}_L \end{bmatrix} + \begin{bmatrix} \hat{f}_{p1,5} \times \vec{r}_{p1,5} & \hat{f}_R \times \vec{r}_R \\ \hat{f}_{p1,5} & \hat{f}_R \end{bmatrix} \begin{bmatrix} \lambda_{1,5} \\ \lambda_6 \end{bmatrix} = \vec{0}. \quad (5.3)$$

The process of setting the magnitude of the input wrench \vec{L} will be discussed in Section 5.3. Note that the pivot wrench $\vec{C}_{P1,P5}$ is not necessarily composed of five constraints, but will always have a set of five linearly independent wrenches (rank = 5). When an HOC is part of the pivot wrench, less than 5 constraints may be involved. When the reaction wrench is a point constraint, it is a one-system. When the reaction wrench is an HOC, a number of reaction wrenches are evaluated simultaneously. This will be explained in Section 5.6. The system of linear equations is deterministic, and the intensity of the reaction wrenches λ_1 through λ_6 can be found with a unique solution. Figure 5.3 shows the equilibrium of a rigid body under the loading wrench and constraining wrenches.

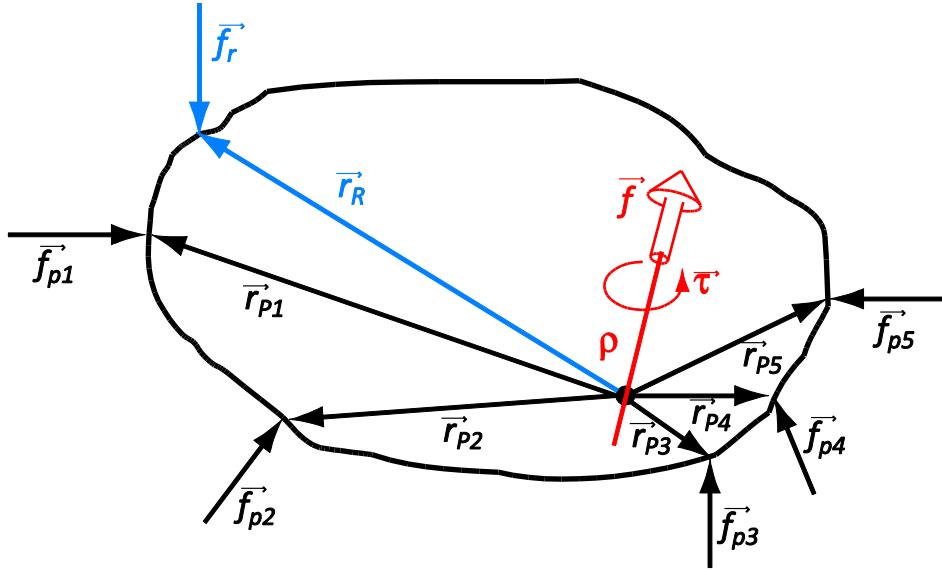


Figure 5.3 Isolated reaction force model equilibrium

It is important to mention here that the input wrench in particular is applied along the motion screw axis, not reciprocal to it. This is important to distinguish because wrenches that are associated with screws are commonly understood in terms of the duality principle. For example, a zero pitch screw is a pure rotation (as a screw) *or* a pure force (as a wrench). Conversely, an infinite pitch screw is a pure translation (as a screw) *or* a pure torque (as a wrench). In this isolated reaction force model, the load is applied coincident to the motion path of the screw, not as its reciprocal. In order to maintain the standard mathematical convention of screw theory, the pitch of the input wrench needs to be adjusted as

$$h_w = \frac{1}{h_s}, \quad (5.4)$$

where h_w is the pitch of the input wrench applied coincident to the motion screw with pitch h_s .

5.3 Input wrench magnitude specification

It was shown in Section 3.4.1 that the magnitude of the input wrench is critical to the inconsistency problem in the calculation of the resistance value. This difference in input units and discontinuity in the scaling caused rotation and translation to not be in the same scale and hence incomparable in their ratings.

5.3.1 Solution for inconsistent units of the input wrench

The first inherent problem with the virtual penetration model is that the input units for rotation are radians, revolutions, or angular units in general, while the input units for pure translation are inches, mm or length units in general. The solution to this problem is to model the rotational component of the input wrench as a force couple instead of a concentrated moment (Figure 5.4). The moment load induced by applying an input force couple λ_i at a distance d from the screw axis is $\lambda_i * d$. The moment arm length d is determined by measuring the longest perpendicular distance between the screw axis and the farthest constraint location. By using the longest distance as the moment arm d , the calculation results will be on the conservative side. Although the load can be applied anywhere on the part, this is the worst case where $\lambda_i * d$ is the largest load possible that the constraint configuration must resist.

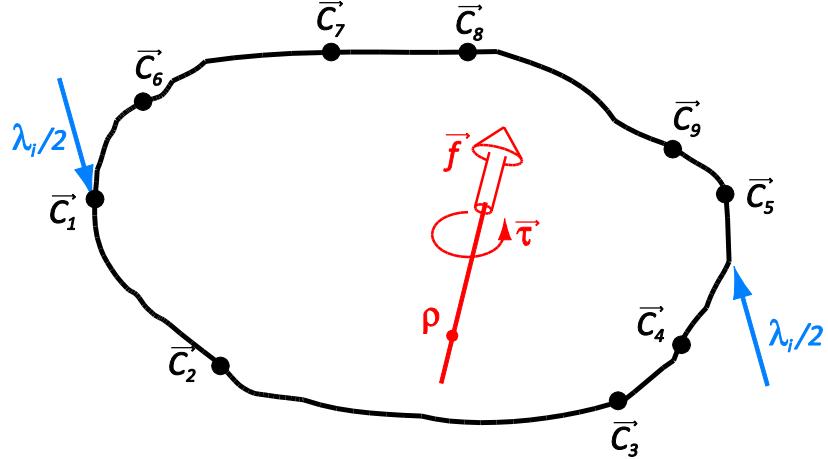


Figure 5.4 The input wrench as a force couple

In the context of real parts, any moment load applied on the part is always in the form of a force or a force couple and never as a concentrated moment. Because of this, letting the rotational component of the input wrench be dependent on the moment arm length is realistic. Possible application points for the force couple include the constraint locations and/or anywhere on the part surface. In this dissertation, the actual calculation only accounts for the constraint locations and not the part shape vertices because part geometry information is not currently included in the input parameters of the constraint analysis computation tool. This is acceptable because transfer of load is often applied through the constraints. In the future work section, it is mentioned that resolving moment arm distances using part overall shape may improve the accuracy of the calculation.

For any given constraint configuration and screw axis, d has a fixed value that can be determined by measuring the perpendicular distances between the screw axis and the constraint location points. The input force magnitude λ_i is not fixed. The resistance value will be the ratio between the reaction force magnitude and the input force magnitude.

By modeling the input wrench as a force couple, the inherent problem of input units is resolved. The translational component of the input wrench is also a force. The units for pure rotation, for combination of rotation and translation, as well as for pure translation are consistent.

5.3.2 Solution for the pitch discontinuity problem

The second inherent problem in the virtual penetration model is the discontinuity of the input magnitude scale between finite pitch screw motion and infinite pitch screw motion (pure translation). In the virtual penetration method, the input magnitude is set to one for the rotational component. As previously explained, the magnitude of the translational component of the input wrench approaches infinity as the pitch increases to infinity. But when pitch is equal to infinity (i.e. pure translation), the magnitude for the translational component is one.

One way to eliminate this discontinuity is by setting the magnitude of the translational component to one and letting the rotational component be inversely proportional to the pitch. This eliminates the discontinuity at the transition from near-infinite pitch to infinite-pitch motion screws, but creates an identical problem at the opposite end, namely at the transition from near-zero pitch to zero-pitch motion screws. Merging the two approaches can solve the discontinuity problem. This is done by creating a critical pitch transition point h_c at which the input magnitude (for both rotational and translational components) is equal.

In this approach then, there are two cases to consider:

- **Case 1:** For motion with pitch

$$0 \leq h_s \leq h_c, \quad (5.5)$$

or

$$\frac{1}{h_c} \leq h_w \leq \infty. \quad (5.6)$$

This screw motion has a relatively small pitch compared to the critical pitch.

The motion is called rotation-dominant motion. In this case, the input wrench magnitude is defined by the rotational component, namely the torque. The input wrench is generically defined by

$$\begin{bmatrix} \vec{\tau} \\ \vec{f} \end{bmatrix} = \begin{bmatrix} h_w \hat{f} + (\vec{\rho} \times \hat{f}) \\ \hat{f} \end{bmatrix}, \quad (5.7)$$

where h_w is the pitch of the wrench and $\vec{\rho}$ is a point that lies on the wrench axis. In order to set the magnitude of the rotational component proportional to $\lambda_i d$, the translational component is multiplied by $\frac{\lambda_i d}{h_w}$ to give

$$\begin{bmatrix} \vec{\tau} \\ \vec{f} \end{bmatrix} = \begin{bmatrix} h_w \left(\frac{\lambda_i d}{h_w} \hat{f} \right) + \left(\vec{\rho} \times \left(\frac{\lambda_i d}{h_w} \hat{f} \right) \right) \\ \frac{\lambda_i d}{h_w} \hat{f} \end{bmatrix}. \quad (5.8)$$

Simplifying, the input wrench magnitude is

$$\begin{bmatrix} \vec{\tau} \\ \vec{f} \end{bmatrix} = \lambda_i \begin{bmatrix} d \hat{f} + \frac{d}{h_w} (\vec{\rho} \times \hat{f}) \\ \frac{d}{h_w} \hat{f} \end{bmatrix}, \quad (5.9)$$

or in terms of the motion screw pitch h_s ,

$$\begin{bmatrix} \vec{\tau} \\ \vec{f} \end{bmatrix} = \lambda_i \begin{bmatrix} d \hat{f} + h_s d (\vec{\rho} \times \hat{f}) \\ h_s d \hat{f} \end{bmatrix}. \quad (5.10)$$

It can be observed that when $h_s = 0$ or $h_w = \infty$,

$$\begin{bmatrix} \vec{\tau} \\ \vec{f} \end{bmatrix} = \begin{bmatrix} \lambda_i d \hat{f} \\ 0 \end{bmatrix}. \quad (5.11)$$

This is a pure rotation screw. Therefore, the input wrench is a pure torque with magnitude $\lambda_i d$.

- **Case 2:** For motion with a pitch

$$h_c \leq h_s \leq \infty, \quad (5.12)$$

or

$$0 \leq h_w \leq \frac{1}{h_c}. \quad (5.13)$$

This screw motion has a relatively large pitch compared to the critical pitch.

The motion is called translation-dominant motion. In this case, the input wrench magnitude is defined by the translational component, namely the force. The input wrench is generically defined by

$$\begin{bmatrix} \vec{\tau} \\ \vec{f} \end{bmatrix} = \begin{bmatrix} h_w \vec{f} + (\vec{\rho} \times \vec{f}) \\ \vec{f} \end{bmatrix}. \quad (5.14)$$

In order to set the magnitude of the translational component to λ_i , the equation is simply multiplied by λ_i to give

$$\begin{bmatrix} \vec{\tau} \\ \vec{f} \end{bmatrix} = \lambda_i \begin{bmatrix} h_w \hat{f} + (\vec{\rho} \times \hat{f}) \\ \hat{f} \end{bmatrix}, \quad (5.15)$$

or in terms of the motion screw pitch h_s ,

$$\begin{bmatrix} \vec{\tau} \\ \vec{f} \end{bmatrix} = \lambda_i \begin{bmatrix} \frac{1}{h_s} \hat{f} + (\vec{\rho} \times \hat{f}) \\ \hat{f} \end{bmatrix}. \quad (5.16)$$

It can be observed that when $h_s = \infty$ or $h_w = 0$,

$$\begin{bmatrix} \vec{\tau} \\ \hat{f} \end{bmatrix} = \lambda_i \begin{bmatrix} (\vec{\rho} \times \hat{f}) \\ d\hat{f} \end{bmatrix}. \quad (5.17)$$

This is a pure translation screw. Therefore, the input wrench is a pure force with magnitude λ_i .

Based on the magnitude formulation above, the critical pitch transition point is determined to be $h_c = \frac{1}{d}$. At $h_s = h_c$, the input wrench in both cases is equal to

$$\begin{bmatrix} \vec{\tau} \\ \hat{f} \end{bmatrix} = \lambda_i \begin{bmatrix} d\hat{f} + (\vec{\rho} \times \hat{f}) \\ \hat{f} \end{bmatrix}. \quad (5.18)$$

Substituting the input wrench specification into Equation 5.3, the equilibrium equations, for cases 1 and 2 respectively, become

$$\lambda_i \begin{bmatrix} d\hat{f} + h_s d(\vec{\rho} \times \hat{f}) \\ h_s d\hat{f} \end{bmatrix} + \begin{bmatrix} \hat{f}_{p1,5} \times \vec{r}_{p1,5} & \hat{f}_R \times \vec{r}_R \\ \hat{f}_{p1,5} & \hat{f}_R \end{bmatrix} \begin{bmatrix} \lambda_{1,5} \\ \lambda_6 \end{bmatrix} = \vec{0}, \quad (5.19)$$

$$\lambda_i \begin{bmatrix} \frac{1}{h_s} \hat{f} + (\vec{\rho} \times \hat{f}) \\ \hat{f} \end{bmatrix} + \begin{bmatrix} \hat{f}_{p1,5} \times \vec{r}_{p1,5} & \hat{f}_R \times \vec{r}_R \\ \hat{f}_{p1,5} & \hat{f}_R \end{bmatrix} \begin{bmatrix} \lambda_{1,5} \\ \lambda_6 \end{bmatrix} = \vec{0}. \quad (5.20)$$

For simplicity, the equilibrium is set up with ρ as the origin, and hence $\vec{\rho} = \vec{0}$.

Simplifying further, the equilibrium equations for cases 1 and 2 become

$$\lambda_i \begin{bmatrix} d\hat{f} \\ h_s d\hat{f} \end{bmatrix} + \begin{bmatrix} \hat{f}_{p1,5} \times \vec{r}_{p1,5} & \hat{f}_R \times \vec{r}_R \\ \hat{f}_{p1,5} & \hat{f}_R \end{bmatrix} \begin{bmatrix} \lambda_{1,5} \\ \lambda_6 \end{bmatrix} = \vec{0}, \quad (5.21)$$

$$\lambda_i \begin{bmatrix} \frac{1}{h_s} \hat{f} \\ \hat{f} \end{bmatrix} + \begin{bmatrix} \hat{f}_{p1,5} \times \vec{r}_{p1,5} & \hat{f}_R \times \vec{r}_R \\ \hat{f}_{p1,5} & \hat{f}_R \end{bmatrix} \begin{bmatrix} \lambda_{1,5} \\ \lambda_6 \end{bmatrix} = \vec{0}. \quad (5.22)$$

Figure 5.5a shows the input wrench magnitude as related to the motion screw pitch h_s . The plot where the pitch is less than the critical pitch refers to case 1 and where the pitch is greater than the critical pitch refers to case 2. Figure 5.5(b) simply show the

reverse of this plot in terms of wrench pitch h_w . It can be observed that the input wrench magnitude is equal at the critical pitch and hence the discontinuity problem is eliminated by using this approach.

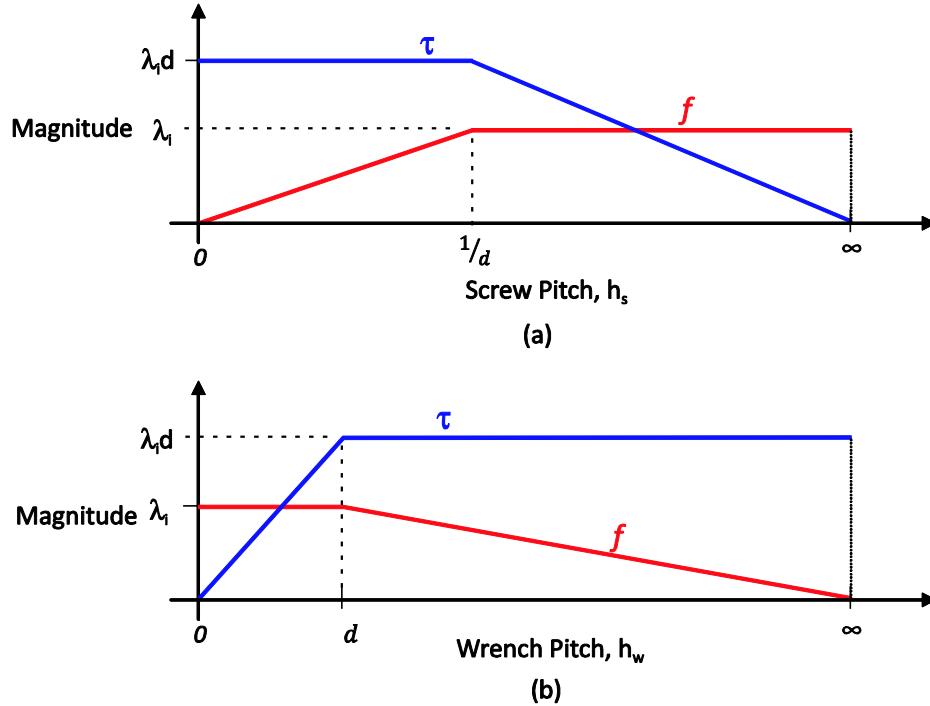


Figure 5.5 Input wrench magnitude vs. (a) screw pitch and (b) wrench pitch

5.4 The resistance value calculation

The ratio between the magnitude of the reaction wrench λ_6 and the magnitude of the input wrench λ_i will be taken as the individual rating of the effectiveness of the respective constraint with regard to the respective motion. The rating is called the resistance value, RV , defined as

$$RV = \frac{\lambda_6}{\lambda_i}. \quad (5.23)$$

Because it is a ratio, the resistance value is dimensionless. For simplicity in the mathematical implementation, λ_i is specified to be of magnitude 1. The value RV then can be interpreted as the required reaction force magnitude to resist a unit input wrench (for rotational motion, applied at the farthest constraint from the screw axis). So a resistance value of 2 means that the balancing reaction force magnitude needed to counter the input force is equivalent to two times the input force. A resistance value of 0.5 means that the balancing reaction force magnitude needed to counter the input force is equivalent to half the input force. In the context of real parts, smaller reaction force intensities imply that constraints are placed and oriented in an effective and efficient manner. A motion that is close to being unconstrained will have a reaction force close to infinity, while a motion that is very effectively resisted will have a small reaction force.

The values λ_1 through λ_5 , which indicate the magnitudes of the reaction forces at the pivot wrenches, are to be ignored. This is the reason the model is called the *isolated* reaction force. As previously explained in Section 3.2, the motion screw reciprocal to the pivot wrench system is by definition impossible to be resisted by any linear combination of the pivot wrenches because it lies in the null space of the five-system. The pivot wrench is only active as a balancing force as the motion along the screw is resisted by the reaction wrench \vec{C}_R . In the absence of this reaction wrench, the motion is left unconstrained, and hence the magnitudes of the pivot wrenches λ_1 through λ_5 are infinity. In fact, this is the case when the reaction wrench happens to be linearly dependent to the pivot wrenches. When this is true, the reaction force magnitude λ_6 is calculated as infinity.

5.5 Additional advantages

There are two additional advantages of the isolated reaction force method compared to the virtual penetration method. The first one is that the isolated reaction force method is more intuitive than virtual penetration. For example, the notion that a large reaction force is undesirable is easier to interpret compared to the notion of small virtual penetration. In addition, the value of the resistance value has direct and practical design application. Interpreting the resistance value as load amplification ratio (explain in Section 6.2), the reaction force magnitude can be cross-checked with the constraints' load carrying capabilities for design acceptability. There is no basis for scaling the input magnitude in the virtual penetration model, but the scaling of the input magnitude using the force couple is based on real parts. This manipulation of the input magnitude provides a solution to the inconsistency problem described above.

The second advantage worth noting is that λ_6 is inversely proportional to its moment arm distance to the screw axis

$$\lambda_6 \sim \frac{1}{|\vec{r}_R|}. \quad (5.24)$$

On one hand, it can be observed that a very short moment arm distance increases the reaction force to approach infinity. This happens as the constraint is located very close to the screw axis. On the other hand, the increase of moment arm distance has a 'diminishing return' in giving the reaction wrench a mechanical advantage. This happens as the constraint is located very far from the screw axis. Both of these facts accurately describe the advantage of having constraints located far from the rotational load in assembly design. This desired non-linear behavior was not present in the virtual

penetration method, where the rating is directly (and linearly) proportional to its distance from the screw axis.

5.6 Reaction wrench composition

The reaction wrench \vec{C}_R is composed differently depending on each type of constraint. There is no further process for the point constraint case because there is only one possible reaction wrench to represent it, namely a zero pitch wrench with the normal direction as the wrench axis. In Section 4.3.2, it was explained that although a line or plane constraint is modeled as having infinite length and width, its resistance quality evaluation would take into account its size. In the context of real parts, a line or plane contact constraint's effectiveness is directly proportional to its size.

As a general rule, HOC reaction wrenches will be modeled as point reaction forces at their optimum resistance locations or orientations. This is accurate because the reaction force distribution in an HOC always moves to the location and orients itself toward the direction where it can resist with the least force because it is at the lowest state of energy. This fact can be illustrated with a simple example. Consider a box sitting on a perfectly flat planar surface with friction be pushed at its top edge until it tips over. As the box is pushed with increasing force, the normal force on the planar surface moves from being uniformly distributed to a concentrated force at the edge farthest from the force application point. It needs to counter the moment induced by the force, and this is accomplished most efficiently at the edge. Section 7.3.6 describes in detail the process to calculate this optimal resistance location or orientation for each HOC type.

5.7 Summary

The isolated reaction force model is presented as the method to evaluate for each constraint the quality of individual resistance to the motion generated by the methodology in Chapters 3 and 4.

This method is able to solve two types of problems associated with the virtual penetration model. It was shown that the isolated reaction force model provided the identical, but inverted metric of constraint effectiveness. However, the isolated reaction force model has features that can solve the inconsistency problems. The inconsistency in the unit input magnitude between rotation and translation is solved by modeling the rotational component as a force couple. In this way, the rotational and translational input have the same units, namely those of force. The discontinuity in magnitude scaling as pitch approaches infinity is solved by categorizing the motion into two cases: rotation-dominant motion and translation-dominant motion. By specifying the input wrench magnitude differently for the two cases, a continuous scaling can be achieved. It was also shown that the isolated reaction force is superior in its physical interpretation and realistic scaling.

Each individual resistance value for each constraint and each motion needs to be organized and processed into meaningful overall assembly rating metrics. Chapter 6 describes the post-processing of these values and the overall metrics used in this dissertation.

CHAPTER 6

ASSEMBLY PERFORMANCE RATING METRICS

Chapter 5 discussed the isolated reaction force model used to evaluate the effectiveness of individual constraints to resist motion. This quantity is called the resistance value. This chapter discusses the post-processing involved in calculating the final overall rating metric for the assembly constraint configuration.

6.1 Rating matrix composition

The post-processing of these resistance values begins with arranging them in a matrix. The rows correspond to the motions, and the columns correspond to the constraints. For m number of evaluated motions and n number of evaluated constraints, the size of the rating matrix is $m \times n$. For each motion (row), there should always be a few constraints with infinite reaction forces. These represent either the set of pivot constraints to which the motion is reciprocal or any constraint that is linearly dependent to the set. They have infinite reaction forces because these constraints cannot do any

work to resist the motion by the definition of reciprocity. Any constraint that is linearly independent from the pivot wrenches resists the motion and has a finite reaction force.

In over-constrained cases, more than one constraint is involved in resisting motion. In order to consider the overall constraint configuration resistance to a motion, the resistance values of all the constraints for the specific motion must be considered. This can be done by summing the resistance values. By summing the resistance values, larger sums imply more resistance. However, this is contradictory to the scale of an individual resistance value. For each individual resistance value, a smaller value signifies stronger resistance, while a larger value signifies weaker resistance. Recall that the physical meaning of each resistance value, as explained in Section 5.4, is the ratio between the magnitude of the reaction force and the input load.

To resolve the contradictory scale between the individual resistance values and the sum of the resistance values, the scale of the individual resistance value is inverted by taking a reciprocal transformation ($1/x$). This transformation reverses the scale so that the individual resistance value is on the same scale as the sum of these values. Note that at this point, the infinite reaction forces from the pivot wrenches become zero instead of infinity. The interpretation of the resistance value is also reversed. Instead of the magnitude of the reaction force for a unit input wrench, it is interpreted as the required input wrench magnitude to yield a unit reaction force. A smaller value signifies a weak assembly because it only requires a small input wrench to generate a large reaction force at the reaction constraints. This transformation is done prior to further post-processing described in the following sections.

Table 6.1 shows a sample rating matrix with 8 constraints (C_1 through C_8). The zero values are written as “x”. It can be observed that in each row, the constraints that have zero resistance values refer to either the pivot constraints or any other constraint that is linearly dependent to the pivot constraints. The non-zero resistance values are the actively resisting constraints for the respective motion.

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	Total Resistance
Motion 1	x	x	x	x	x	0.802	1.031	1.018	2.851
Motion 2	0.369	x	x	1.495	x	x	0.716	x	2.580
Motion 3	x	x	0.775	x	0.796	x	x	x	1.571
Motion 4	x	0.946	x	0.057	x	0.798	x	x	1.801
Motion 5	x	1.120	x	x	x	x	1.296	1.200	3.616
Motion 6	1.159	x	0.903	x	x	x	x	0.590	2.652
Motion

Table 6.1 Sample Rating Matrix

In all known previous work, constraints are always modeled as a series of unilateral point constraints. In the case of assemblies with HOC, this leads to redundant ratings that can be deceptively high. This pseudo-redundancy effect becomes worse as HOC's are discretized into more dense point constraints spaced closer to each other. Because of this pseudo-redundancy effect, summing the resistance values of all constraints as a rating metric is misleading. At the same time, the resistance to motion from all constraints needs to be considered. This problem is solved by modeling the HOC with a higher order screw system as discussed in Chapter 4. With this model, it is accurate to sum all of the resistance values from all of the constraints that resist a particular motion. Although this metric contains redundancy because the motion is

resisted by multiple constraints, the redundancy effect is real because each constraint is now a separate feature providing resistance to the motion.

For each motion, the total resistance is calculated by summing each row. This measures how well resisted each motion is. In order to observe the ratings for the whole motion set, a histogram is created for the total resistance ratings. A sample histogram is shown in Figure 6.1.

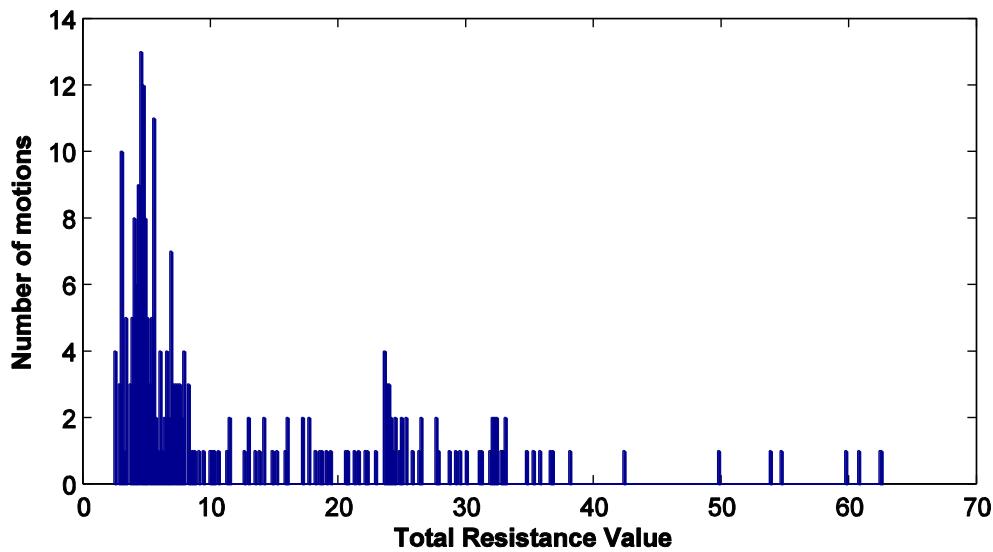


Figure 6.1 Sample total resistance histogram that shows random spread

The above histogram is taken from an assembly with many non-rectilinear and non-symmetrical constraints. In such cases, there is usually a single most weakly constrained motion. A typical assembly with many symmetrical and rectilinear constraints has a much simpler histogram such as the one shown in Figure 6.2. The total resistance histogram is useful in observing whether there is a collection of multiple weakly constrained motions or only a single weakly constrained motion. It is also useful

to see the total resistance value of the most weakly constrained motion in light of how the rest of the motion is rated. Figure 6.2 shows a sample histogram where there are 2 motions rated as 1 – the lowest, another group rated approximately 1.7, and another group rated approximately 2.6. The significance of this chart is to show the groupings of motions that are rated equally.

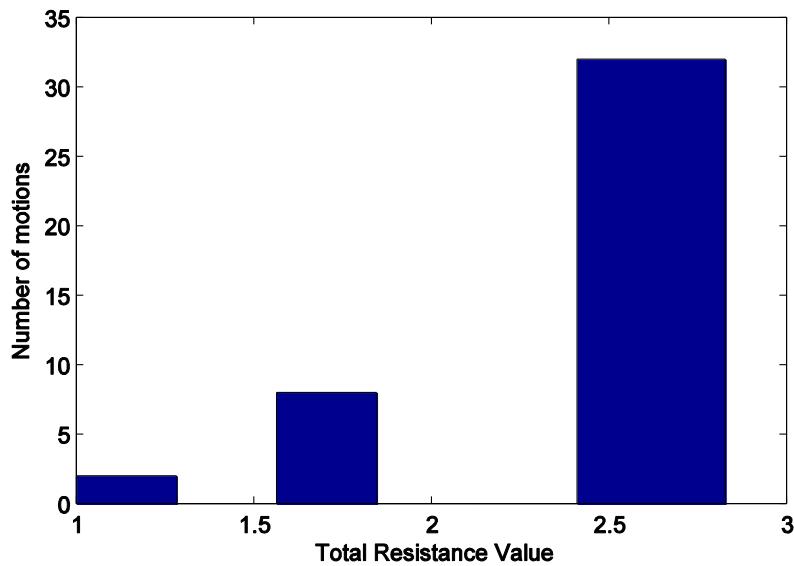


Figure 6.2 Sample histogram that shows grouping of motion total resistance

6.2 Weakest total resistance (WTR) and the most weakly constrained motion

Since an assembly is as weak as the most weakly constrained motion (weakest link in the chain principle), it is appropriate to measure assembly resistance quality based on its ability to restrain the most weakly constrained motion. This rating is the appropriate metric for assessing the quality of an assembly configuration to remove the critical DOF(s). The sum of ratings for the most weakly constrained motion is called the weakest total resistance (WTR) rating. The rating is calculated by

$$WTR = mi n(\sum_1^n M_j), \quad (6.1)$$

where M is the rating matrix with i rows and j columns of size $m \times n$. In the sample rating matrix displayed in Table 6.1, the WTR is identified for motion 3 with a WTR value equal to 1.571.

Identifying the most weakly constrained motion that is associated with this rating is crucial in understanding the quality of the assembly. This is reported in the analysis report file. The most weakly constrained motion is specified in terms of the screw axis direction vector ω , a coincident point that lies on the motion screw axis ρ , and the pitch h , as

$$\omega = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}, \rho = \begin{bmatrix} \rho_x \\ \rho_y \\ \rho_z \end{bmatrix}, h. \quad (6.2)$$

When there are multiple motions that are equally rated as the most weakly constrained motion, all of them are identified. A tolerance of 10% of the WTR rating is applied to capture similarly rated motions.

As an aid for physical interpretation, the WTR rating is also presented as the load amplification ratio (LAR). The load amplification ratio is simply the reciprocal of the WTR rating. This needs to be distinguished from the reciprocal transformation explained in Section 6.1. The previous reciprocal transformation is done to each resistance value in the rating matrix, while the load amplification ratio of the WTR is the reciprocal of the final WTR rating quantity. The LAR is shown exclusively as an aid to interpret the WTR rating. For example, if the WTR rating of an assembly is 0.25. Presenting this in terms of load amplification ratio is $1/0.25 = 4$. This means that the input wrench along the WTR

motion is ‘amplified’ 4 times on each reaction constraints on average. Higher load amplification ratios yield undesirable large reaction forces on the constraints.

6.3 Mean total resistance (MTR)

Two constraint configurations may have identical WTR ratings, but their overall effectiveness may not be identical. This is affected by how the rest of the motions (except the most weakly constrained motion) are resisted. Though WTR is a crucial metric of assembly performance in general, there are cases where the most weakly constrained motion can be ignored depending on the context of the design task. This total resistance for all motion is also averaged to yield a measure of how well the overall assembly configuration resists all motion. This is called the mean total resistance (MTR) rating. It is calculated by

$$MTR = \frac{\sum_1^n M_j}{m}. \quad (6.3)$$

Similar to the WTR rating, the MTR rating is also presented in terms of load amplification ratio in order to aid physical interpretation.

6.4 Mean redundancy ratio (MRR)

The redundancy metric offers a way to quantify the level of redundancy in a constraint configuration. The redundancy ratio for each motion is the ratio between the total resistance value (sum of a row) and the peak resistance value (maximum of a row). This ratio has a minimum value of one. When a redundancy ratio is equal to one, it means that for that motion, the peak resistance value is equal to the total resistance value. In this

case, there is only a single constraint resisting the particular motion. As the redundancy ratio increases to a large value, the peak resistance value is much smaller than the total resistance value. In this case, the best resisting constraint is only one among the many redundant constraints resisting the motion. When constraints are assumed as rigid contacts with no deformation, which is the case throughout this dissertation, only a single constraint is adequate to resist a motion. Therefore, a ratio of 3 means that the total resistance provided is three times what the best resisting constraint already provides for the particular motion. The redundancy ratio RR for each row i is calculated by

$$RR_i = \frac{\sum_j M_j}{\max(M_j)}. \quad (6.4)$$

The overall rating metric to measure the redundancy for the overall assembly configuration is the average of all redundancy ratios for all motion. It is called the mean redundancy ratio (MRR) rating, expressed as

$$MRR = \frac{\sum_{i=1}^m RR_i}{m}. \quad (6.5)$$

When the MRR is equal to one, it means that each motion is resisted by a single constraint. This is a deterministic design, also known as an exactly constrained design. The actual reaction force is statically determinate for all motions. The larger the value of this rating, the more redundantly resisted is the motion in the assembly. The closer the value is to one, the constraint set is in general less redundant and more reliant on the peak resistance constraint.

6.5 Trade-off ratio (TOR)

Because assembly design decisions, especially on the number of constraints and type of constraints are often a trade-off between resistance quality and redundancy, this rating metric is important. While assembly resistance quality is desirable for obvious reasons, redundancy beyond a certain point is undesirable because it increases internal stresses and preload, which may cause long-term creep in plastic snap-fit applications. By observing the change in TOR as the design space is searched, one can select design that yield the most resistance quality with minimal redundancy. The TOR value is to be used as a comparative metric during design optimization search because its value is only significant in comparison and not in absolute scale. The TOR is calculated by

$$TOR = \frac{MTR}{MRR}. \quad (6.6)$$

6.6 Constraint activity and best resistance percentage

This rating metric is identical to the one from Bozzo [9]. A constraint is inactive when its resistance value is zero for a particular motion. There are two cases when a constraint can have zero resistance value. The first case is when the constraint belongs to the pivot wrench set that is reciprocal to the motion that is being evaluated. The second case is when the motion results in movement away from the unilateral nature of the constraint (infinitesimal loss of contact). Obviously in this case the constraint cannot resist the motion. This is similar to the repelling screws described by Asada and By [4]. Mathematically, this is equivalent to having a negative reaction force as the solution to the equilibrium Equations 5.21 and 5.22 of the isolated reaction force model.

In the rating matrix, the percentage of the motion for which a particular constraint is active. This is called the constraint active percentage (CA%). It is calculated by

$$CA\%_j = \frac{\# \text{ of } (M_j \neq 0)}{m}. \quad (6.7)$$

This value is proportional to the number of times a constraint acts as a reaction wrench that is linearly independent to the rest of the assembly constraints. This rating metric gives a measurement of the uniqueness of the DOF removal provided by an individual constraint. This value tends to be low in symmetrical assembly constraint design and high in non-orthogonally oriented constraint configurations.

Similar to the active percentage, the best resistance percentage is the frequency of a constraint not only actively resisting motion, but also being the one providing the best resistance to the motion. The constraint best resistance percentage (CBR%) is calculated by

$$CBR\%_j = \frac{\# \text{ of } (\max(M_j))}{m}. \quad (6.8)$$

6.7 Summary

The rating metrics for the overall assembly performance are developed in this chapter. Because assembly design is complex and context sensitive, the designer needs different perspectives on the performance of an assembly. The reason why multiple metrics are developed is to understand the various aspects of performance of an assembly. In general, the WTR and MTR ratings are the measurements of the assembly's resistance quality to the motions. In many overconstrained cases, both the WTR and MTR ratings increase as constraints are added. This means that the assembly is more

redundant and farther away from an exactly constrained design, which is desirable. This is where the MRR rating measures the level of redundancy in the assembly configuration. Many times, redundancy cannot be avoided in assembly design. There is a tradeoff between WTR, MTR, and MRR as constraints are added. This will be studied further in Chapter 8. The CA% and CBR% are useful in understanding the importance of an individual constraint feature relative to the rest of the constraints. Chapter 8 also discusses the usage of these different rating metrics in various design contexts.

CHAPTER 7

CONSTRAINT ANALYSIS TOOL ALGORITHM AND IMPLEMENTATION

This chapter describes the details of the implementation of the analysis methodology described in Chapters 3 through 6. The kinematic constraint analysis tool is implemented in MATLAB version 7.4. All of the functions used in the MATLAB script are built into the standard version of MATLAB and do not need any additions from the MATLAB toolbox. The MATLAB script code can be found in Appendix A with additional detailed comments attached within the lines of code.

7.1 Main algorithm

The overall analysis algorithm is divided into three parts:

- 1. Pre-processing**

The preprocessing involves taking the input file containing the constraint coordinates and orientations and transforming them into standard wrench notation. This is where the HOC modeling discussed in Chapter 4 is implemented. After all the constraints are represented with their wrench

equivalents, a combination script creates all possible combinations of constraints to compose the pivot wrenches. The scripts involved in this process are the *input_file.m*, *input_filecheck.m*, *input_preproc*, and *combo_preproc.m*.

2. Main processor

The main processor is composed of a for-loop with the number of iterations equal to the total number of possible combinations of constraints. Before each combination is processed, it is tested for linear independence. When a pivot wrench set is found to be of rank 5, the null space is solved to generate the screw motion reciprocal to the five-system. Before the motion resistance is calculated, the script checks for any duplicate motion already processed. The isolated reaction force method discussed in Chapter 5 is implemented to calculate the motion resistance value for each constraint for each motion. The scripts involved in this process are *main_loop.m* and any applicable sub-functions called within the script.

3. Post-processing

The post-processor organizes the resistance values into a rating matrix and calculates the overall rating metrics such as the total resistance values, WTR, MTR, MRR, and other metrics. A report file is generated in html format. At the end of the analysis, a question asking whether to conduct further design space exploration is posed to the user. The scripts involved in this process are *rating.m* and *report.m*.

The high-level flowchart of the algorithm is shown in Figure 7.1. Most of each process box is a subroutine in itself.

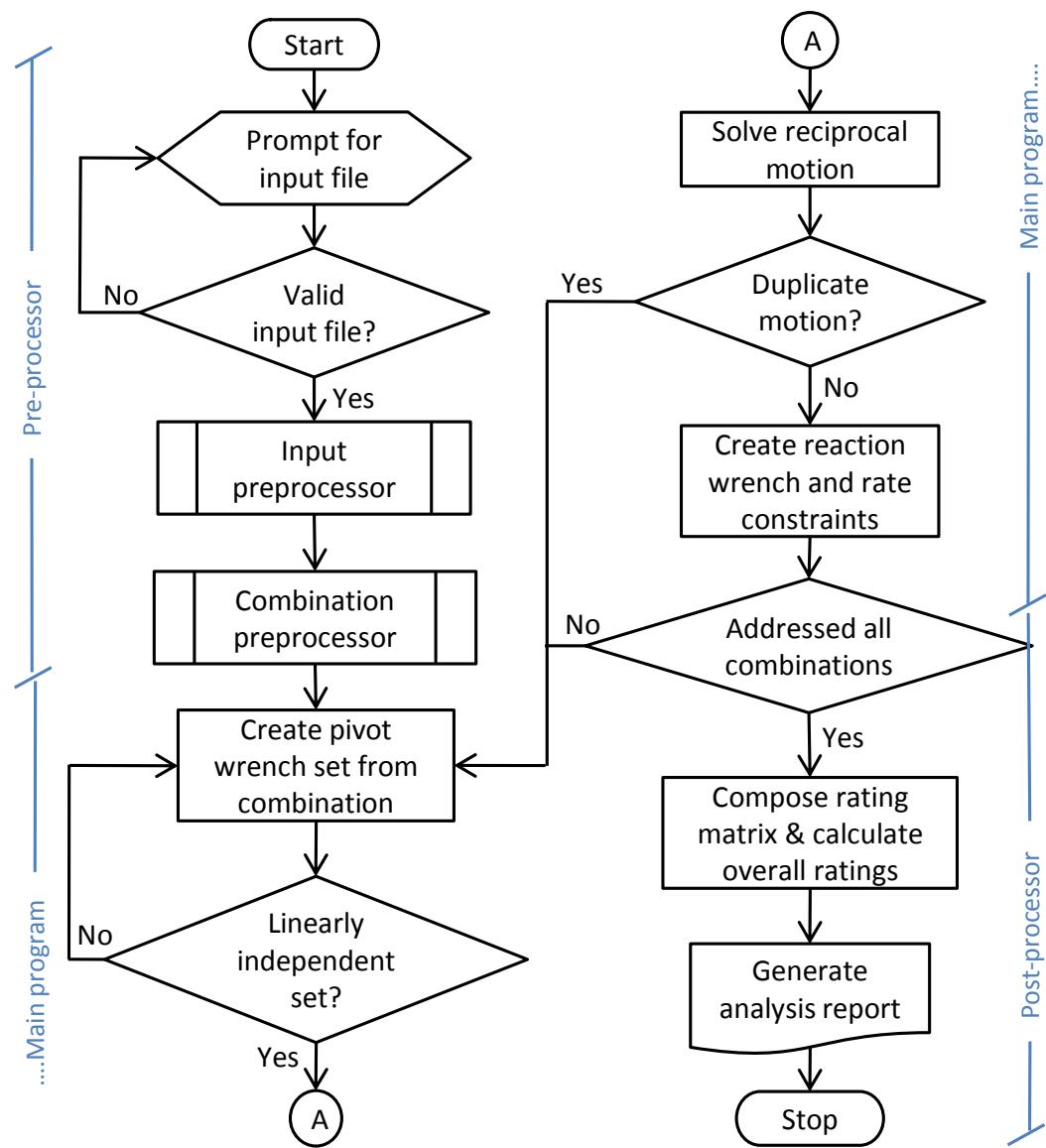


Figure 7.1 Main algorithm flowchart

7.2 Pre-processing

7.2.1 Input file format

The input file has the following format:

Point constraint	$cp = [P_x \ P_y \ P_z \ N_x \ N_y \ N_z]$
Pin constraint	$cpin = [P_x \ P_y \ P_z \ w_x \ w_y \ w_z]$
Line constraint	$clin = [P_x \ P_y \ P_z \ l_x \ l_y \ l_z \ N_x \ N_y \ N_z \ L]$
Plane constraint	$cpln = [P_x \ P_y \ P_z \ N_x \ N_y \ N_z \ Pln_type]$ $cpln_prop = [u_x \ u_y \ u_z \ L_1 \ v_x \ v_y \ v_z \ L_2]$ or $cpln_prop = [r]$

Table 7.1 Constraint input file format

Where:

P : Location coordinates of the center point of the constraint

N : Normal direction vector of constraint orientation (not applicable for pin constraints)

w : Pin constraint axis direction vector

l : Line constraint axis direction vector

Pln_type : The type of plane constraint (1 for rectangular, 2 for circular)

u : Plane constraint local x-axis direction vector

v : Plane constraint local y-axis direction vector

L_1 : Rectangular plane constraint x-axis length

L_2 : Rectangular plane constraint y-axis length

r : Circular plane constraint radius

7.2.2 Input file checker

The input file checker script *input_check.m* validates the input file by checking the following conditions:

- The total degree of constraints must be greater than or equal to seven, without taking linear dependence into account. The total degree of constraints can be calculated by doing a weighted sum of the number of constraints. The weights signify the number of DOF removed by each constraint (1 for point constraints, 2 for pin and line constraints, 3 for planar constraints). If the total degree of constraints is less than seven, it can be concluded that there will be one or more DOF that is unconstrained.
- Each constraint is properly defined by the expected number of parameters depending on their type (6 for point and pin constraints, 10 for line constraints, 6 for planar constraints and an additional 8 for associates planar constraint properties).

7.2.3 Constraint set transformation to wrench coordinates

The script *input_prepoc.m* transforms constraints into wrenches. This section is the continuation of Section 4.2 in the HOC model implementation. A generic wrench in motor notation, $[\vec{F} \quad \vec{M}]$, is composed of the force axis direction \vec{F} and the moment \vec{M} around the origin. The wrench is calculated by

$$wr = [\vec{F} \quad \vec{\rho} \times \vec{F}], \text{ for } h = \text{finite pitch.} \quad (7.1)$$

or

$$wr = [\vec{0} \quad \vec{F}], \text{ for } h = \infty. \quad (7.2)$$

where $\vec{\rho}$ is a point that lies on the wrench axis. The force axis \vec{F} and the pitch h are different for each constraint type, but \vec{M} is always calculated using Equation 7.1.

- For a point constraint, \vec{F} is defined by the normal direction \vec{N} . Its pitch is zero.
- For a pin constraint, \vec{F} is defined by the two principal axes perpendicular to the pin axis direction \vec{w} . They define the planar pencil of the pin constraint. The basis of the planar pencil lies in the null space of \vec{w} ; its dimension is two. The pitches of both principal axes are zero.
- For a line constraint, the first principal axis is defined by the normal resistance direction \vec{N} with zero pitch. The second principal axis is perpendicular to both the line direction vector \vec{l} and the normal direction vector \vec{N} with infinite pitch. It is calculated by taking the cross product $\vec{l} \times \vec{N}$.
- For a plane constraint, the first principal axis is defined by the plane normal direction \vec{N} with zero pitch. The second and third principal axes are perpendicular to the plane normal direction \vec{N} . The pitches of both the second and third principal axes are infinite. The second and third principal axis directions can be calculated by taking the null space $null(\vec{N})$.

Table 7.2 shows the wrench transformation formulas for each constraint type.

Constraint type	Wrench axis	Pitch
Point constraint	$wr_{cp} = [\vec{N} \quad \vec{P} \times \vec{N}]$	$h_{cp} = 0$
Pin constraint	$wr1_{cpin} = [null(\vec{w})_1 \quad \vec{P} \times null(\vec{w})_1]$ $wr2_{cpin} = [null(\vec{w})_2 \quad \vec{P} \times null(\vec{w})_2]$	$h1_{cpin} = 0$ $h2_{cpin} = 0$
Line constraint	$wr1_{clin} = [\vec{N} \quad \vec{P} \times \vec{N}]$ $wr2_{clin} = [\vec{0} \quad \vec{l} \times \vec{N}]$	$h1_{clin} = 0$ $h2_{clin} = \infty$
Plane constraint	$wr1_{cpln} = [\vec{N} \quad \vec{P} \times \vec{N}]$ $wr2_{cpln} = [\vec{0} \quad null(\vec{N})_1]$ $wr3_{cpln} = [\vec{0} \quad null(\vec{N})_2]$	$h1_{cpln} = 0$ $h2_{cpln} = \infty$ $h3_{cpln} = \infty$

Table 7.2 Wrench transformation formulas for each constraint type

The preprocessor also collects all of the position vectors of the constraints in order to calculate the longest possible distance between constraints in order to limit the calculation of the moment arm distance used in Section 7.3.5.

7.2.4 Combination composition of five-system wrench

The script *combo_prepoc.m* creates an array that contains all possible combinations of constraints to compose the pivot wrench set. In order to simplify the coding implementation of the combination scheme, all constraints are combined regardless of their type and degree of constraint. Degree of constraint is the number of DOF removed by a constraint. If all types of constraints are considered, the combination scheme becomes very complicated. For example, a five-system can be a combination of:

- 5 point constraints

- 1 point constraint + 2 pin constraints
- 2 point constraints + 1 plane constraint
- 1 point constraint + 1 pin constraint + 1 line constraint
- 3 point constraints + 1 line constraint

The level of complication is significant because partial linear dependence is possible with combinations of higher degree constraints (see Section 7.3.1). For example, a combination of 3 pin constraints (total degree of constraint is $3 \times 2 = 6$) can be reduced to five linearly independent wrenches if only one axis of the first pin is linearly dependent to the axes of the second pin.

Since the largest possible degree of constraint provided by a single constraint (in the context of this dissertation) is 3, namely the plane constraint, the smallest number of constraints that can be a five-system set is 2. Similarly, since the smallest possible degree of constraint provided by a single constraint is 1 (point constraint), the largest number of constraints that is able to compose a five-system is 5. Therefore, the combination scheme creates pivot wrench combination sets containing two to five constraints at a time.

7.3 Main processor

The main processor is a for-loop subroutine that iterates a set of processes for each combination of constraints. The core functions in the for-loop solve for the reciprocal motions and calculate the resistance values for the motions. The following subsections describe the procedures within the for-loop in further detail.

7.3.1 Pivot wrench set composition and linear independence check

The first step in the for-loop is to compose the pivot wrench set $\vec{C}_{P1,P5}$ (Equation 5.2) from each constraint listed in the constraint combination set produced by the combination preprocessor *combo_prepoc.m*. The wrenches from each constraint are stored in the array *wr_all*. Based on the number of constraints listed in the particular constraint combination, two, three, four, or five constraints are merged into a wrench matrix.

Because the number of wrenches for constraints that are higher order (pins, lines, and planes) may be more than one, the pivot wrench set might have more than five wrenches. However, a linear independence test is done to process only pivot wrench sets that are exactly five-systems. This test is done by calculating the rank of the wrench set. The rank indicates the number of linearly independent wrenches in the wrench set. A wrench set containing m number of wrenches or rows can only have a rank of m or less. Hence, pivot wrench sets that are composed of more than five wrenches must have linearly dependent sets within them in order to be a five-system.

It is important to note that linear dependency can occur on one of the principal axes between higher order constraints. Because not all of the principal axes are linearly dependent, this is called partial linear dependence between constraints. For example, consider a plane constraint defined by w_1 , w_2 , and w_3 and a line constraint defined by w_4 and w_5 shown in Figure 7.2. These two constraints are partially linearly dependent. It can be observed that the infinite pitch wrench w_5 can be expressed as a linear combination of w_2 and w_3 .

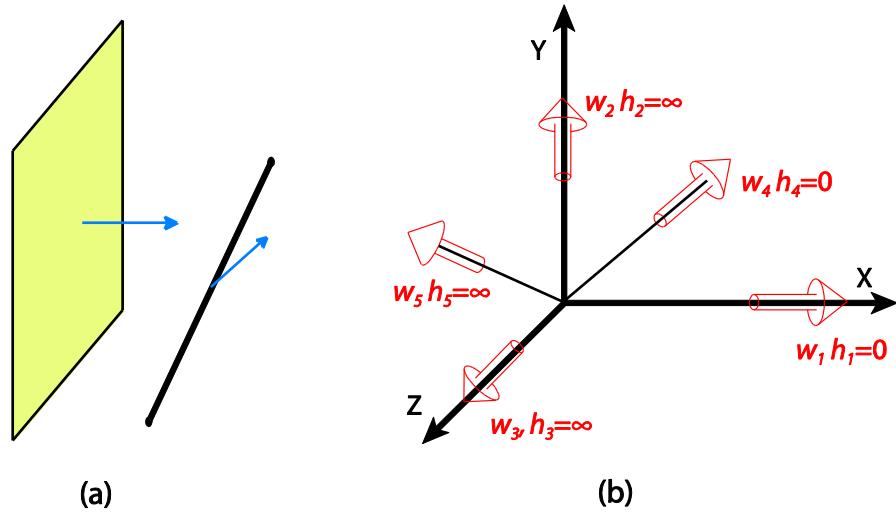


Figure 7.2 Partial linear dependence example between (a) plane and line constraint and (b) the associated wrench systems

If the rank of the pivot wrench set is less than five, the reciprocal motion DOF will be a two-system or greater, and as a result, there are an infinite number of motions that need to be evaluated. If the rank is more than five (maximum of six), then there is no motion allowed. A set of five linearly independent wrenches will allow exactly one single motion. This is why unless there are exactly five linearly independent wrenches in the set, the set is not processed, and the algorithm continues to the next combination in the list.

7.3.2 Solution for the motion reciprocal to the pivot wrench

The script *rec_mot.m* solves for the motion reciprocal to the pivot wrenches. The single DOF motion reciprocal to the pivot wrench is called the reciprocal motion. This motion is calculated by solving the equation

$$\begin{bmatrix} \omega_{1x} & \omega_{1y} & \omega_{1z} & \mu_{1x} & \mu_{1y} & \mu_{1z} \\ \omega_{2x} & \omega_{2y} & \omega_{2z} & \mu_{2x} & \mu_{2y} & \mu_{2z} \\ \omega_{3x} & \omega_{3y} & \omega_{3z} & \mu_{3x} & \mu_{3y} & \mu_{3z} \\ \omega_{4x} & \omega_{4y} & \omega_{4z} & \mu_{4x} & \mu_{4y} & \mu_{4z} \\ \omega_{5x} & \omega_{5y} & \omega_{5z} & \mu_{5x} & \mu_{5y} & \mu_{5z} \end{bmatrix} \begin{bmatrix} \vec{\mu} \\ \vec{\omega} \end{bmatrix} = \vec{0}. \quad (7.3)$$

$\vec{\mu}$ and $\vec{\omega}$ in the equation above comprise the motion screw axis of interest. $\vec{\mu}$ and $\vec{\omega}$ by definition form the null space of the pivot wrench. To minimize rounding errors, the vector of the null space is rounded to the fourth decimal point. Note that this freedom motion is also specified in terms of screw coordinates. Once $\vec{\omega}$ and $\vec{\mu}$ are calculated, the pitch h and the coordinate of the point ρ that lies on the screw axis can be calculated by

$$h = \frac{\vec{\omega} \cdot \vec{\mu}}{\vec{\omega} \cdot \vec{\omega}}, \quad (7.4)$$

$$\vec{\rho} = \frac{\vec{\omega} \times \vec{\mu}}{\vec{\omega} \cdot \vec{\omega}}. \quad (7.5)$$

The computation time using this solution method (i.e. calculating the null space) was shown to be an order of magnitude more efficient compared to the linearized optimization method used by Bozzo [9].

7.3.3 Separation of forward and backward reciprocal motion

The motion reciprocal to the pivot wrench set described in the previous section is unconstrained by the pivot wrenches in both forward and backward directions. The resistance quality of these two directions can be very different due to the unilateral nature of the constraints because they only restrain one direction. With the exception of the pin constraint, none of the constraints can resist both directions at the same time. It is important therefore to separate the motion into forward and backward screw motion. In

order to do this, each reciprocal motion is first duplicated. To reverse the direction of the motion, the screw axis direction $\vec{\omega}$ and the velocity at the origin $\vec{\mu}$ are multiplied by -1.

In order to prevent doubling the computational time due to considering both directions of motion, the resistance value toward the backward motion screw is calculated simultaneously with the forward motion screw. This is done by taking the negative reaction forces and considering the value as the resistance value for the backward motion. It was explained in Section 6.1 that any constraint that has a negative reaction force is considered loss of contact, or zero resistance value, and hence does not restrain the motion. However, the same constraint will resist the backward motion with the reaction force of exactly equal magnitude, but positive. This is verified to be true in the script. To put it differently, the positive reaction forces resist the forward direction motion, and negative reaction forces resist the backward direction motion. In this manner, the backward motion can be evaluated simultaneously with the forward motion without any additional computation time.

7.3.4 Removal of duplicate motion

After further investigation, many of the motions generated from the reciprocal screw equation are identical. This is especially true in assembly constraint configurations where many constraints are located and oriented symmetrically. The number of duplicate motions in these cases can account for 50-80% of the total possible motions. In non-symmetrical and non-rectilinear constraint configurations, the number of identical motions can account for 20-40% of the total combinations. Using the MATLAB function

unique.m, duplicate motions can be found and anticipated before the computation is done further. Once a duplicate motion is detected, the for-loop immediately advances to the next combination in the list.

Duplicate motions exist mainly due to linear dependence between constraints. This occurs very often in symmetrical constraint configurations. Although the test for linear independence already excludes the linearly dependent pivot wrenches as a set, a duplicate motion can still occur. Consider an example combination shown in Table 7.3. If C6 is linearly dependent to C5, then the motion generated by combination #2 will be identical to combination #1.

Combination #1	C1	C2	C3	C4	C5
Combination #2	C1	C2	C3	C4	C6

Table 7.3 A sample combination that causes duplicate motion

7.3.5 Input wrench specification

In order to properly scale the input wrench magnitude, the maximum moment arm distance needs to be calculated for each motion. The script *calc_d.m* calculates this by measuring the distance from each constraint center coordinate \vec{P} to the point $\vec{\rho}$ on the screw axis $\vec{\omega}$ and projecting this distance to the plane normal to the screw axis

$$d = \max(|\vec{\omega} \times (\vec{P} - \vec{\rho})|). \quad (7.6)$$

If this distance is more than the maximum distance between two constraints, then the maximum distance between two constraints is used instead. This only occurs when the screw axis is located outside of the part body or the envelope of constraint locations. As previously explained in Section 5.3, depending on the magnitude of the pitch h relative to

$1/d$, the motion is classified as either rotation-dominant or translation-dominant. The magnitude of the input wrench is then scaled according to Equations 5.10 and 5.17. It should be noted that for pure translation ($h = \infty$), the screw axis direction vector is stored in the $\vec{\mu}$ variable instead of $\vec{\omega}$.

7.3.6 Reaction wrench set up

The reaction wrench is set up in the scripts *rate_cp.m*, *rate_cpin.m*, *rate_clin.m*, *rate_cpln1.m*, and *rate_cpln2.m*. This section is the detailed implementation of Section 5.6. The reaction wrench \vec{C}_R in the equilibrium Equation 5.2 is composed differently depending on the type of constraint. It was mentioned that the reaction force in an HOC will be concentrated at the optimal location and oriented in the optimal orientation so that the resistance can achieve equilibrium with the least reaction force intensity. The rest of this section describes the process of selecting the optimal location and orientation within each HOC for defining the reaction wrench. All of the reaction wrench optimal location and orientation selection processes described in this section have been confirmed to be accurate by comparing this method's results with the results from running a simple optimization routine involving a single reaction HOC and an arbitrary screw axis.

Refer to Section 7.2.1 for the nomenclature related to the constraint variables. The equilibrium equation is constructed with ρ as the origin. This is true for all of the equations for the reaction wrench below. \vec{R} is the vector between the constraint center coordinate and the point on the screw axis and is equal to

$$\vec{R} = \vec{P} - \vec{\rho}. \quad (7.7)$$

With the exception of the pin constraint, for pure translation cases, the location of the reaction wrench does not affect the reaction force calculation. Therefore, its location is simply defined at the center point of the constraints.

For a point constraint, the reaction wrench is the normal direction vector with zero pitch as

$$\vec{C}_R = \begin{bmatrix} \vec{R} \times \hat{N} \\ \hat{N} \end{bmatrix}. \quad (7.8)$$

For a pin constraint, the location of the wrench is defined, but the orientation of the wrench can vary anywhere on the planar pencil. The orientation is to be determined according to the direction of the line-of-action of the respective motion. The pin constraint will resist the motion in the optimal direction, which is opposite to the line-of-action. The reaction wrench of the pin constraint must be in the span of the plane of its planar pencil. In order to calculate the optimal direction, the line-of-action is projected onto the planar pencil (Figure 7.3). The line-of-action \vec{v} is the instantaneous velocity of the point at the center of the pin constraint. Therefore, the location of the resisting wrench is located at the pin center point and its orientation \hat{N}_R is calculated based on the line-of-action

$$\vec{v} = h\vec{\omega} + \vec{\omega} \times \vec{R}. \quad (7.9)$$

For pure translation,

$$\vec{v} = \vec{\mu}. \quad (7.10)$$

The reaction wrench direction and the reaction wrench are

$$\vec{N}_R = \vec{N} \times (\vec{v} \times \vec{N}), \quad (7.11)$$

and

$$\vec{C}_R = \begin{bmatrix} \vec{R} \times \hat{\vec{N}}_R \\ \hat{\vec{N}}_R \end{bmatrix}. \quad (7.12)$$

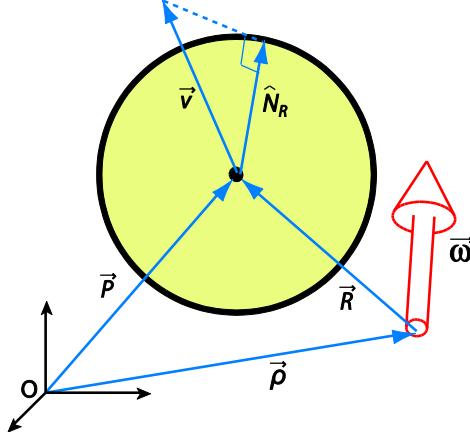


Figure 7.3 Pin constraint reaction wrench calculation

For line constraints, the orientation of the wrench is defined; it must be parallel to the normal direction of the line constraint specification. The location can be anywhere along the line constraint. The line constraint is a finite set of coplanar and parallel wrenches (two-system). For a straight line, the optimal location for the reaction wrench will be located at one of the end points of the line R_1 and R_2 (Figure 7.4). The reaction wrench \vec{C}_R is calculated for two cases, namely at each end point. The lesser of the two is to be selected as the optimal location for the reaction wrench. An exception would be when the line-of-action is directed perpendicular to the line constraint, in which case any location along the straight line will yield equal magnitude for the reaction wrench.

$$\vec{C}_{R1} = \begin{bmatrix} \vec{R}_1 \times \hat{\vec{N}} \\ \hat{\vec{N}} \end{bmatrix}, \vec{C}_{R2} = \begin{bmatrix} \vec{R}_2 \times \hat{\vec{N}} \\ \hat{\vec{N}} \end{bmatrix}. \quad (7.13)$$

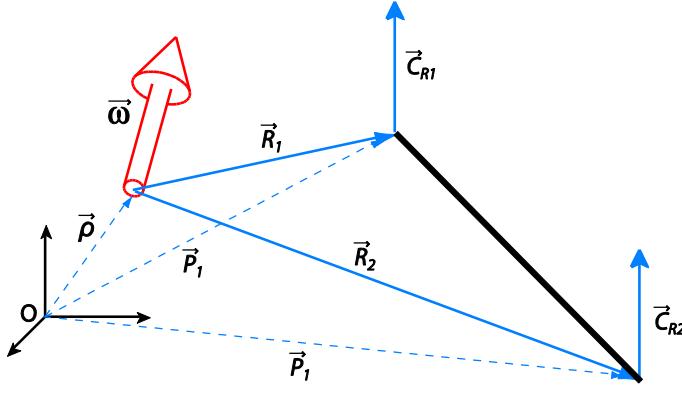


Figure 7.4 Line constraint reaction wrench calculation

There are two kinds of plane constraint: rectangular and circular. The approach to solve the rectangular plane constraint reaction wrench is similar to the line constraint. The orientation of the reaction wrench is parallel to the normal direction of the planar constraint. The location can be anywhere in the plane of the constraint within the boundary of the finite plane so that it belongs to the set of parallel wrenches spanned by the three-system. For a rectangular plane, the optimal location of the reaction wrench will be at one of the four corners for reasons similar to the line constraint. The reaction wrench is calculated for the four corner points. The least of all is to be selected as the optimal location for the reaction wrench. The same exception applies when the line-of-action is normal to the plane, in which case any location on the plane will yield an equal magnitude for the reaction wrench

$$\hat{C}_{R1} = \left[\begin{matrix} \vec{R}_1 \times \hat{N} \\ \hat{N} \end{matrix} \right], \hat{C}_{R2} = \left[\begin{matrix} \vec{R}_2 \times \hat{N} \\ \hat{N} \end{matrix} \right], \hat{C}_{R3} = \left[\begin{matrix} \vec{R}_3 \times \hat{N} \\ \hat{N} \end{matrix} \right], \hat{C}_{R4} = \left[\begin{matrix} \vec{R}_4 \times \hat{N} \\ \hat{N} \end{matrix} \right]. \quad (7.14)$$

Figure 7.5 illustrates the calculation for the rectangular plane constraint reaction wrench.

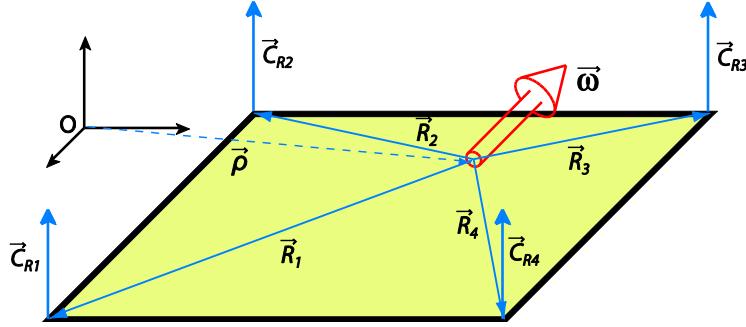


Figure 7.5 Rectangular plane constraint reaction wrench calculation

The approach to solve the circular plane constraint reaction wrench is somewhat similar to the pin constraint. The difference is that in plane constraint, the orientation is determined while the location varies across the plane. The orientation of the reaction wrench must be parallel to the normal direction of the circular plane. The location of the reaction wrench can be anywhere on the plane within the boundary of the finite plane so that it belongs to the set of parallel wrenches that form the three-system. For the circular plane, the optimal location of the reaction wrench will be anywhere along the circular boundary of the plane. This can be determined more specifically by considering the line-of-action. The optimal location will be driven toward the edge of the circular boundary that is located farthest from the motion screw axis. This location can be determined by projecting the moment arm \vec{R} onto the plane

$$\vec{R}_p = \vec{N} \times (\vec{R} \times \vec{N}). \quad (7.15)$$

The projected moment arm \vec{R}_p intersects the circular plane boundary at two points. Either of these two points can be the location of the optimal reaction wrench

depending on the forward or backward direction. The modified moment arms of the two points are calculated by

$$\vec{R}_1 = \vec{\rho} - (\vec{P} + r\hat{R}_p), \quad (7.16)$$

$$\vec{R}_2 = \vec{\rho} - (\vec{P} - r\hat{R}_p). \quad (7.17)$$

When the screw axis passes through the constraint center point, the cross product $\vec{R} \times \vec{N}$ is zero, so a modification is needed. The two optimal points of reaction can be determined by projecting the screw axis onto the plane,

$$\vec{\omega}_p = \vec{N} \times (\vec{\omega} \times \vec{N}). \quad (7.18)$$

The modified moment arms of the two points are

$$\vec{R}_1 = \vec{\rho} - (\vec{P} + r\hat{\omega}_p), \quad (7.19)$$

$$\vec{R}_2 = \vec{\rho} - (\vec{P} - r\hat{\omega}_p). \quad (7.20)$$

The possible reaction wrenches of the constraint are

$$\vec{C}_{R1} = \begin{bmatrix} \vec{R}_1 \times \vec{N} \\ \hat{N} \end{bmatrix}, \vec{C}_{R2} = \begin{bmatrix} \vec{R}_2 \times \vec{N} \\ \hat{N} \end{bmatrix}. \quad (7.21)$$

The reaction wrench is calculated for these two possible locations. The lesser of the two is selected as the optimal location for the reaction wrench. An exception would be when the line-of-action is parallel to the normal direction, in which case any location on the plane yields an equal magnitude for the reaction wrench. Figure 7.6(a) illustrates the calculation for the circular plane constraint reaction wrench in general and Figure 7.6(b) illustrates the case when the screw axis passes through the constraint center point.

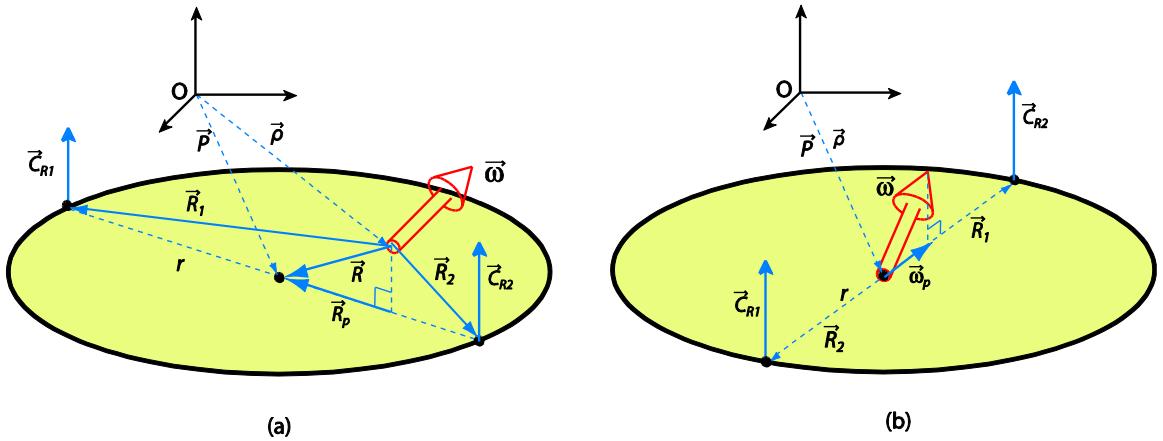


Figure 7.6 Circular plane constraint reaction wrench calculation

7.3.7 Solution to the equilibrium equation

After the pivot wrenches $\vec{C}_{P1,P5}$ are merged with the reaction wrench \vec{C}_R , the equilibrium equation

$$[\vec{C}_{P1,P5}, \vec{C}_R][\lambda_i] = -\vec{L}, \quad (7.22)$$

can be solved for the reaction force intensities λ_i . In cases when the merged constraint wrench is found to be rank-deficient (less than 6), the resistance value is set to infinity. As previously explained in Section 5.4, only the reaction force intensity λ_6 for the reaction wrench is used as the resistance value. Positive resistance values correspond to forward screw motion resistance and negative resistance values correspond to backward screw motion resistance.

7.4 Post processing

Post-processing and report generation are done by the scripts *rating.m* and *report.m*. The generated report filename is *report.html* and is contained in the same main directory of the scripts.

7.4.1 Composing the rating matrix

The rating matrix is composed in the following organization. The rows are composed of forward and backward motion. $cp_{n,m}$ is the resistance value of cp_n to motion m . The rating matrix is of size $m \times n$ and is

$$M = \begin{bmatrix} M_{fwd} \\ M_{bwd} \end{bmatrix}, \quad (7.23)$$

$$M = \begin{bmatrix} cp_{1,1} & \cdots & cp_{n,m} & cpin_{1,1} & \cdots & cpin_{n,m} & clin_{1,1} & \cdots & clin_{n,m} & cpln_{1,1} & \cdots & cpln_{n,m} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ cp_{1,m} & \cdots & cp_{n,m} & cpin_{1,m} & \cdots & cpin_{n,m} & clin_{1,m} & \cdots & clin_{n,m} & cpln_{1,m} & \cdots & cpln_{n,m} \end{bmatrix}. \quad (7.24)$$

Chapter 6 describes the equations to extract overall rating metrics such as WTR, MTR, MRR, and CA% from the rating matrix. The script displays all of the associated motions with total resistance within 10% of the WTR value. The report also contains additional information such as the number of all possible combinations, the number of linearly independent sets processed, and the number of unique motions in the final rating. Symmetrical assemblies show significant reduction from all possible combinations to the total number of unique motions evaluated.

CHAPTER 8

ASSEMBLY DESIGN STUDY

8.1 Objectives of the assembly design study

It was mentioned in Chapter 1 that one of the motivations for this dissertation is the lack of domain-specific knowledge in assembly design. Most designers do not have a scientific approach in the decision making process of DOF removal. Therefore, before developing a practical design tool that can be utilized in synthesizing optimal constraint location and orientation, this dissertation engages in a study of assembly design. The aim of this dissertation is not to create software that can be treated as a "black box" for designers to optimize their design, but rather to create a methodology to analyze and explore the design space from a kinematic point of view. Therefore, the design tool in this dissertation is developed in such a way that provides in-depth data that provide deeper insights into the performance of an assembly rather than merely giving the optimum solution for implementation purposes. It will be apparent later, especially in the discussion of case study results, that assessment of an assembly's kinematic quality is a complex task that requires much insight into the gaps between real parts, the kinematic model, and the mathematics involving the resistance value calculation.

Based on the context mentioned above, this dissertation aims to accomplish the following objectives in studying assembly using the analysis and design tool:

- Study the agreement between the kinematic constraint design tool and commonly known design principles in assembly.
- Study the trade-off between assembly resistance quality and redundancy as constraints are added or removed.
- Explore the design search space and identify possible design improvements.

8.2 Objective #1: Confirm design principles in assembly using the design tool

Because this dissertation is one of the first attempts to scientifically analyze and quantify assembly performance at an attachment strategy level, it is therefore very important to bridge the gap between the commonly known design principles in assembly and the results of the methodology used in this dissertation. It was previously mentioned that the need for a design tool is due to the complexity of assembly design at the attachment strategy level. Before the methodology is used to synthesize solutions in a complex design task, it has to be demonstrated that it can confirm existing design principles that are commonly known and have been applied by designers in an ad-hoc manner. The design principles that are to be verified in this case are the leverage principle, symmetry principle, and optimum line-of-action principle.

The leverage principle states that moving constraints farther from the most likely moment load axis results in a more effective and efficient resistance to motion. This is due to the leverage gained by increasing the moment arm length. This design principle in

general recommends placing constraints near the convex hull of the part. The isolated reaction force analysis tool model calculates the reaction force by taking into account the moment arm length of both the input force couple and the reaction constraints. For the same input magnitude, assuming identical orientations, a constraint that has greater moment arm leverage theoretically has less reaction force (better rating). This agreement needs to be confirmed in simple cases in order to trust the design tool in incorporating this principle in analyzing complex parts.

The symmetry principle states that an assembly's resistance to arbitrary motion is increased when constraints are located and oriented symmetrically with respect to the part. In the kinematic constraint analysis model, the evaluated motion set is determined based on the reciprocal system from a set of pivot constraints. In many cases, the screw axis passes through these constraint location points. Indirectly, the resistance of the reaction constraints is proportional to the distance between the constraints. If this distance between constraints is non-symmetrical, the resistance rating of the assembly tends to be stronger for some motions and weaker for other motions. With symmetrical design, the assembly will be more robust to resist more arbitrary loads. An observation study on how this behavior is manifested in the analysis results of the design tool needs to be conducted.

The optimum line-of-action principle states that constraints need to be oriented in the direction that is collinear and opposite to the instantaneous line-of-action of the input load. This principle is briefly described in Section 5.1. This is only applicable to a single motion. In the design tool, this case can be observed in the WTR rating only because the

WTR rating is the total resistance to a single motion. The other case is when the loading condition is known or specified. Note that while the leverage and symmetry principles do not optimize the quality of resistance to pure translation, the optimum line-of-action principle does. The reason is that the first two principles only concern the location and not the orientation of the constraints.

8.3 Objective #2: Trade-off study between assembly resistance quality and redundancy

The process of assembly design involves not only determining the location and orientation that are deemed to be optimal to remove DOF effectively, but also the number and types of constraints to be used. In general, using more constraints in assembly leads to a stronger assembly. Since the loads are distributed into more constraints, the reaction forces become relatively smaller compared to the input load. However, the design becomes more redundant. While redundant designs provide more reliability and load carrying capability, they have important drawbacks. One of them is sensitivity to part-to-part variation. Inaccuracies due to manufacturing tolerance lead to either locked-in stresses or loss of contact between the part and the constraints. This is the reason why minimum constraint design is desirable. One of the main disadvantages of exact constraint design is the high contact stresses or reaction forces sustained by the constraint elements.

In the context of real parts, almost all assembly design is not exact constraint design due to practical reasons such as product aesthetics, function, or reliability. This is

also usually impractical in plastic parts due to flexibility and limited load carrying capability of a single constraint.

This trade-off between overall assembly strength or resistance quality and constraint redundancy is the focus of this study. The design tool must provide data for the designer to make decisions on the number of constraints used in assembly design. The study of this trade-off is done by analyzing assembly resistance quality rating change as more constraints are added or removed from the constraint configuration. It is possible that there is a point of diminishing returns where enough improvement in resistance rating is gained and further constraint addition will increase redundancy more than motion resistance. This is accounted for by the rating metric trade-off ratio (TOR). The same occurrence can be studied in the constraint reduction process.

8.4 Objective #3: Explore design search space and identify possible design improvements

There are infinitely many ways to remove DOF between two bodies using constraints. In other words, the design space is infinite. Even after imposing geometrical limitations as well as considering manufacturability of the part, the design space is still large enough to be explored for possibilities of improvement. Much of the design search space is often unexplored because non-rectilinear motion and constraints are not intuitive. Furthermore, there is no design tool or methodology to engage in this exploration in a formal way.

This is the final application of the design tool, namely to generate design suggestions or alternatives based on the search space defined by the designer. This is

especially useful in the context of plastic parts with integral attachments, where the manufacturing technology allows for more non-rectilinear shapes and complex geometries. The implementation of the design principle of direct load path can also benefit from the design tool because it is capable of analyzing complex parts. For example, there might be a more efficient parting line that is non-planar instead of a traditional planar parting line in a typical part with threaded fasteners.

8.5 Sensitivity analysis and specified load test

In addition to the main objectives explained in this chapter, there are two additional short studies done to further utilize the constraint analysis and design tool. These short studies can take advantage of the already developed algorithms and present the result in a different application.

The first short study is to analyze the sensitivity of the assembly rating to perturbations in constraint location and orientation. Perturbations of the location and orientation of constraints from their nominal values are usually affected by manufacturing defects and dimensional variation. This is more prevalent in highly redundant constraint configurations. Although dimensional variation of one part might not be significant, as the part is assembled into the mating part, interferences or slack in fit can amplify its effect on a constraint's effectiveness.

An exaggerated example of the effect of dimensional variation on the constraint configuration is shown in Figure 8.1. If the width of the part interferes with the space in

the mating part, the part will deform and negatively impacts the constraint configuration by

- Deactivating some constraints (CLIN2, CLIN4, CPLN1).
- Rotating some constraints' normal directions (CP1, CP2, CP3, CP4, CLIN1, CLIN3).

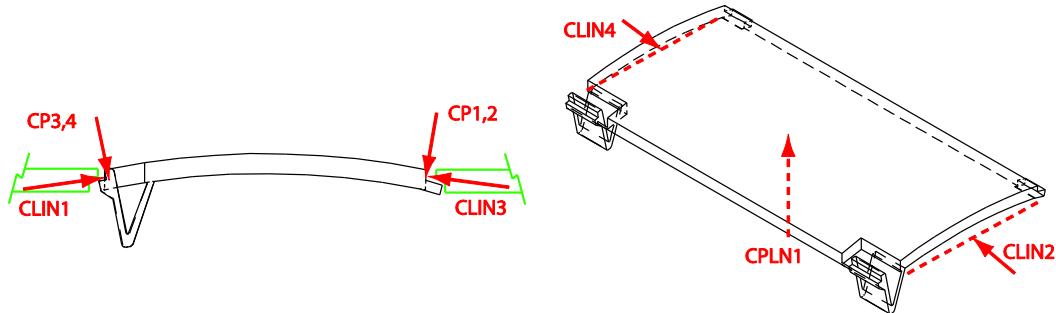


Figure 8.1 Effect of dimensional accuracy on constraint configuration (example 1)

Or in other cases such as the one shown in Figure 8.2, deformation

- Shortens or reduces the area of a planar constraint (CPLN1)
- Moves the location of the point constraints (CP1, CP2)

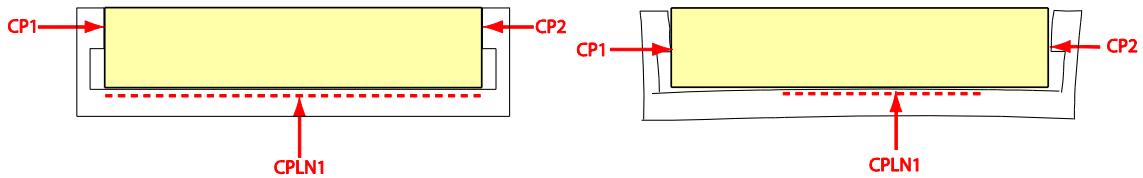


Figure 8.2 Effect of dimensional accuracy on constraint configuration (example 2)

The second additional study involves using the analysis and design tool to analyze the assembly's resistance to a specific loading condition as well as optimizing the constraint variables to provide a better constraint configuration for the specified input

load. In many cases, the most frequent or the most crucial loading condition or set of loading conditions is known. The kinematic constraint analysis and design tool is capable of analyzing and optimizing the assembly in this context. The reason this dissertation does not focus on analyzing and optimizing design for known loading conditions is that there are plenty of analytical and numerical tools to analyze and optimize designs in this manner. However, it would be a beneficial study to extend the application of this design tool to a common design context.

8.6 Case study approach

The study objectives explained in Sections 8.2 to 8.4 are to be accomplished using a case study approach. The study needs to be conducted over a variety of part geometries, ranging from simple to complex. For this purpose, a total of 5 geometries are used in this dissertation. Table 8.1 shows the mapping of case studies conducted in this dissertation to the tests to be done. The drawings of these case study geometries can be found in Appendix B.

	Baseline analysis test	Design principle study	Trade-off study	Explore design space	Sensitivity analysis & specified load test
Exact constraint (Thompson's chair)	x			x	
Simple cube	x		x		
Rectilinear assembly (battery cover)	x	x			
Axisymmetric (end cap assembly)	x		x	x	
Freeform/Non-planar assembly (printer housing assembly)	x		x	x	x

Table 8.1 List of case studies conducted

The first geometry type is an exactly constrained object. The purpose of this geometry is mainly to test the consistency of the analysis tool and to compare the results against the theoretical outcome of the analysis. Since the geometry is constrained by seven point constraints, any inconsistency or contradictory results can readily be observed. The part chosen for this geometry type is the Thompson's "chair" used by Lakshminarayana [23] as an example of an exactly constrained geometrical clamp. The case studies generated for this geometry are a baseline analysis test, one-dimensional optimization of the height variable, and two-dimensional optimization of the height and one of the trihedral point constraints.

The second geometry type is a simple cube. The 1"x1"x1" cube provides a constraint configuration that is simple enough for the human mind to visualize and intuitively analyze motion resistance. This cube is also a good candidate to generate a random number of constraints and conduct a trade-off study through constraint addition and reduction (objective #2). The case studies generated for this geometry are a baseline analysis test, scalability test, constraint addition trade-off study, and constraint reduction trade-off study.

The third geometry type is a simple rectilinear assembly involving higher order constraints. The purpose of this geometry is to test the higher order constraints model described in Chapter 4 to take the analysis tool one level higher in complexity. This geometry is used to confirm commonly known design principles in assembly constraint placement and orientation, namely the principles of leverage, symmetry, and line-of-action (objective #1). The chosen part for this geometry type is a generic battery cover

with two lugs and two snap-fits. The case studies generated for this geometry are a baseline analysis test, leverage design principle case study, symmetry design principle case study, and optimum line-of-action case study.

The fourth geometry type is an axisymmetric part. The purpose of this geometry is to explore search spaces that are unique in an axisymmetric part. The chosen part for this geometry is the end cap assembly of a medicine spray housing. The case studies generated for this geometry are a baseline analysis test, feature/constraint addition trade-off study, and snap-fit location optimization along the circular lip of the end cap.

The fifth geometry type is a freeform geometry with non-rectilinear constraints and shape. The purpose of this geometry is to explore the design search space of a freeform and relatively complex assembly. The chosen part for this geometry is a printer housing assembly from an Epson C82 Stylus printer. The case studies generated for this geometry are a baseline analysis test, threaded fastener location and orientation optimization, snap-fit location and orientation optimization, parting line optimization, line size optimization, and multi-dimensional optimization search involving a combination of the above mentioned search space.

8.7 Summary

This chapter describes the study plan for understanding the effect of constraint revision, addition, and/or removal on assembly quality. Chapter 9 discusses the assembly design study tool algorithm and implementation. Chapter 10 describes the case study geometries and constraint configurations and discusses the baseline analysis results.

Chapter 11 is dedicated to the assembly design case studies to achieve the objectives defined in this chapter. In order to understand the discussion of results in Chapters 10 and 11, it is very important to understand the assembly rating metrics (Chapter 6) and the optimization search algorithms (Chapter 9).

CHAPTER 9

DESIGN STUDY TOOL ALGORITHM

This chapter discusses the assembly design tool algorithm and methodology. The algorithm described in this chapter is used to search the design space as well as post-process the results to aid the decision making process in assembly design. The design tool module has three kinds of capabilities: constraint modification, constraint reduction, and constraint addition. Among the three, constraint modification is the one that is computationally most intensive. This is due to the many permutations in searching the specified design space. Without an efficient algorithm, the computation time can be impractical.

9.1 Constraint modification

The goal in constraint modification is to observe the change in assembly rating as the location and orientation of constraints are modified. Similar core algorithms based on the analysis tool are used. The essential functions of the analysis tool are to combine pivot constraint combinations, generate the motion, and evaluate the resistance of

reaction constraints to that motion. The design tool core functions in exploring the design space are to define the design search space, sort out recalculations involving the original motion set, generate a new motion set and the respective resistance values, and summarize the results in a response surface plot.

9.1.1 Types of design search space and optimization variable specification

The location of a constraint can be varied at discrete locations, along a straight line, along a circular curve, and across a plane. The orientation of a constraint can be varied in two different directions. The size of a constraint can also be varied. Not all of these search spaces are applicable to all constraints. For example, the size search space is not applicable to point and pin constraints for obvious reasons. The design space that is applicable for each constraint type is shown in Table 9.1.

Constraint Type	Location Search Space				Orientation search space		Size search space	
	Discrete	Line search	Curved line search	Plane	Normal direction	Line direction	Length	Area
Point	x	x	x	x	x			
Pin	x	x	x	x	x			
Line	x	x		x		x	x	
Plane	x	x		x	x			x
Search dimension	1D	1D	1D	2D	1D or 2D	1D or 2D	1D	1D
Revision Type	1	2	3	4	5, 6	7	8	9,10

Table 9.1 Design space mapping for each constraint type

Since assembly features are usually represented by multiple constraints, even after HOC modeling is utilized, the constraints need to be modified in groups. Optimization variables are typically specified as a variable group. Grouping of constraints allows multiple constraints in one group to be simultaneously relocated, reoriented, or resized.

At the beginning of each case study, these variable groups and the search space are defined in a table. The search spaces are defined in the following parameters:

- The discrete search space is a set of candidate point locations for the constraint variable. The number of constraint variables for this kind of search space is limited to one.
- The line search space (Figure 9.1a) is defined by its center point L_o , its direction vector \vec{L}_d , and the move limit in either direction δ .
- The circular curve search space (Figure 9.1b) is defined by its center point L_o , its rotation axis \vec{L}_R , and the angular move limit in either direction θ .
- The plane search space (Figure 9.1c) is defined by its center point P_o , its two principal direction vectors \vec{P}_x and \vec{P}_y , and the move limit in either direction for both axes δ_x and δ_y .
- The normal orientation search space can be defined as rotation around one or two axes, depending on the dimension of the search of interest. The search space is defined by its rotation axes \vec{r}_1 and \vec{r}_2 and the angular move limit in either direction θ_1 and θ_2 .

- The line direction orientation search space is defined by its rotation axis \vec{r} and the angular move limit θ in either direction.
- The line size search space is defined by its size limits l_l and l_u .
- The plane area size search space is defined by its size limits in its two principal directions, l_l , l_u , w_l , and w_u in the case of rectangular plane constraint and r_l and r_u in the case of circular plane constraint.

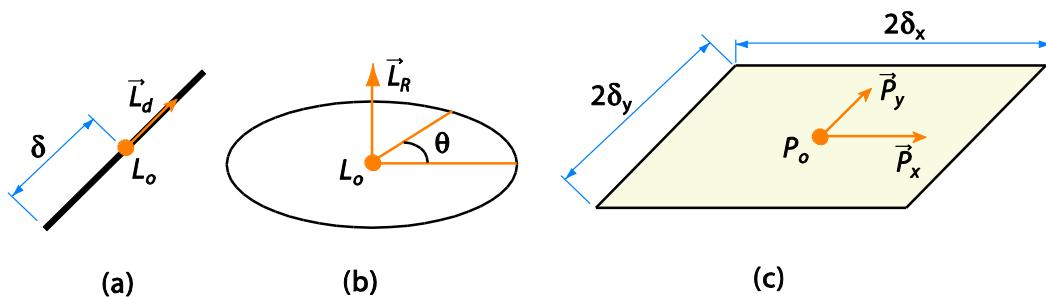


Figure 9.1 Illustration of the (a) line, (b) curved line, and (c) plane search space

The MATLAB script developed in this dissertation is capable of searching the search spaces listed above, but not all of these search spaces are used in the case studies.

9.1.2 Factorial search methodology in design space exploration

In order to understand the effect of different constraint strategies on assembly characteristics, the design space needs to be well understood. Therefore, instead of solely identifying the optimum point in the optimization search, the whole design space is explored and presented in a response surface plot.

The dimension D in the optimization process is determined by the number of constraint variable groups n and the dimension of the search space S for each constraint variable group i . The dimension is calculated by

$$D = \sum_{i=1}^n S_i. \quad (9.1)$$

A full factorial search is implemented in the MATLAB script as D number of nested for-loops.

The variable x is the increment along the search dimension. In the case of multiple dimension searches, x is an array of length D . Each search dimension limits are normalized to $-1 \leq x \leq 1$, with -1 and 1 corresponding to the minimum and maximum limits of each search dimension. In the case studies, the number of increments is typically set to 10 or less to maintain a realistic computation time while collecting enough data points to plot the optimization result.

9.1.3 Constraint modification pre-processing

It is important to understand that in the constraint modification scheme, the evaluated motion set for each constraint modification iteration is unique. The methodology of assembly constraint analysis is recursive in some ways. The motions to be evaluated are generated from the constraint configuration itself. Therefore, as the constraint configuration changes, so do the evaluated motions.

When constraints are modified, there are two types of recalculations involved in the process. One is the recalculation involved when modified constraints act as reaction constraints. The other is when they act as pivot constraints. The recalculated motion in

the first case is called recalculating the resistance for ‘base motion’ (Section 9.1.5) because the motion specifications do not change, but the modified constraint resistance values need to be recalculated. The recalculation in the second case involves generating the ‘new motion set’ (Section 9.1.6) because the motion specifications change. The motion changes because the modified constraints are members of the pivot constraint set.

The preprocessing of the constraint modification algorithm involves separating these two types of motion. Each motion is identified with the pivot constraint set to which it is reciprocal. The following procedure is done to separate the motions:

1. The constraints that are subject to be modified from all optimization variable groups are collected as a set.
2. Pivot constraint combinations that contain any of the modified constraints are identified. From these combinations, the ‘new motions’ are to be generated and recalculated.
3. The motions associated with the pivot constraint combination identified in step #2 are to be removed from the original rating matrix. However, exceptions apply to motions that are duplicates of motions associated with other pivot constraint combinations. Because they are associated to multiple pivot constraint combinations (including those that do not contain any to-be-modified constraints), they should not be removed.
4. After these exceptions are identified, the combinations identified in #2 are removed from the original rating matrix. The motions that remain are identified as the ‘base motion’ set.

5. The resistance of the modified constraints to this ‘base motion’ set will be recalculated and will replace the old resistance values.

This pre-processing is only done once before proceeding to the nested for-loop that searches the specified design space.

9.1.4 Incremental constraint modification

Each iteration in the nested for-loop begins with modifying the constraint to the next increment in the search space. For the details of this constraint, please refer to the comments in the MATLAB script for each search type attached in Appendix A.

9.1.5 Recalculation for base motion set

The resistance of the modified constraints to the base motion set needs to be recalculated. The subroutine that runs this task is similar to the main processor from the analysis tool. In Equation 7.22, the pivot constraint wrench $\vec{C}_{P1,P5}$ can be composed from the original pivot constraint combination in most cases. However, when the pivot constraint combination contains one of the modified constraints, the pivot constraint set is no longer reciprocal to the evaluated motion. These refer to the exception identified in step #3 from Section 9.1.3. In this case, the pivot constraint wrench needs to be adjusted in order to be either of rank 5 or to be replaced by an alternate pivot constraint that is also reciprocal to the motion. For the details of this adjustment, please refer to the comments in the MATLAB script attached in Appendix A. This subroutine is coded in the script *rate_motset.m*.

9.1.6 Recalculation for new motion set

The subroutine that runs this task processes the pivot constraint combinations identified in step #2 from Section 9.1.3. The main processor from the analysis tool described in Section 7.3 is used for this task (*main_loop.m*).

9.1.7 Constraint modification post-processing and optimization plots

The recalculation of the ‘base motion’ set as well as the ‘new motion’ set is done for each incremental constraint modification. A new rating matrix is created by merging the results from the recalculation procedures. The overall assembly rating is calculated from the rating matrix and stored for each incremental search of the design space.

Based on the stored results, the change in rating from the baseline rating is calculated. The change in assembly rating is reported as a response surface plot. For one-to two-dimensional optimization searches, a response surface plot is displayed to give design space information to aid design decisions. The response surface plot is useful in understanding the design space; therefore, in most case studies the optimization search is limited to two dimensions. For optimization searches of more than two dimensions, only the optimum point is identified and reported.

9.1.8 Constraint modification flowchart

The overall flowchart for the constraint modification design tool is shown in Figure 9.2.

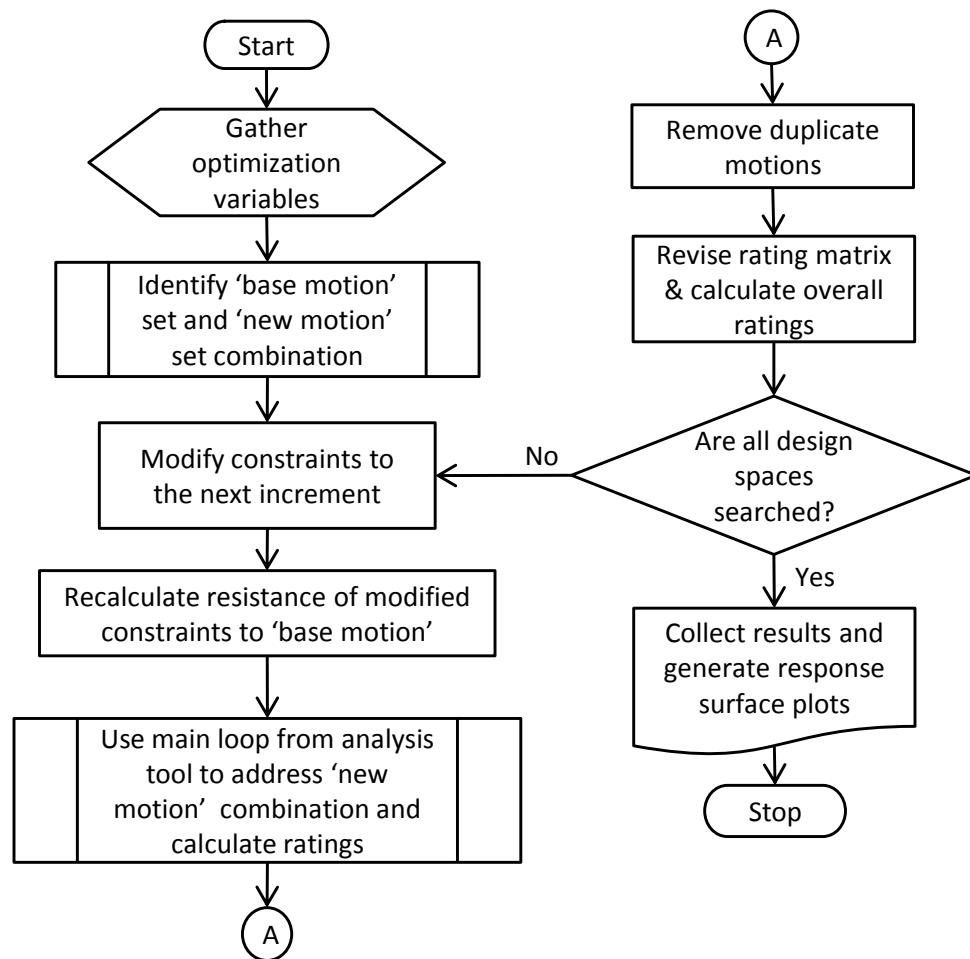


Figure 9.2 Simplified flowchart for constraint modification algorithm

9.2 Constraint reduction

The goal of the constraint reduction algorithm is to observe the change in assembly ratings as constraints are removed. Similar to the constraint modification algorithm, the goal is to observe and understand the design space. In this case, the design space is composed of the number of constraints to remove and which constraints to remove.

The number of constraints to remove is specified by the user. The MATLAB script is programmed to conduct the optimization search based on the number of constraints. It is also possible to program the script to conduct the optimization search based on the percent redundancy ratio decrease or TOR increase; however, this capability is not implemented due to the goal of exploring the design space.

If the designer wishes to remove one constraint, there are n different constraints that one can remove. Instead of using certain criteria to select the best candidate, the algorithm searches through all cases of constraint reduction and displays the assembly rating change due to each case of constraint removal. In the case where the designer wishes to remove k constraints, there are $\frac{n!}{k!(n-k)!}$ different ways to remove k constraints at a time. The assembly rating change due to each combination of constraint removal is displayed.

Because the purpose of constraint reduction is to reduce redundancy while achieving the least decrease in the assembly's capacity to resist motion, the TOR is of interest. Based on the optimization search results described in the paragraph above, the constraint reduction removal case that yields the maximum TOR increase is selected to be the best.

9.2.1 Constraint reduction algorithm

The constraint reduction algorithm is very similar to the constraint modification preprocessor (Section 9.1.3). The constraint modification preprocessor identifies the motion where the reciprocal pivot constraint contains a modified constraint and

eliminates them with the same exception identified in Section 9.1.3. The constraint reduction algorithm accomplishes a very similar task. It needs to eliminate motions where the pivot constraint contains the removed constraints. In addition to the motion elimination, the constraint reduction algorithm also removes the column in the rating matrix that is associated with the currently removed constraint. The modified rating matrix is then evaluated for the revised assembly rating, and the difference from the original assembly rating is calculated.

This process is repeated for each constraint (in the case of one-at-a-time constraint removal) or for each constraint combination (in the case of more than one at a time constraint removal). The result of the constraint reduction algorithm is displayed as a line plot where the horizontal axis refers to the constraint index. Each index value is associated with a constraint or multiple constraints being removed at a time. The multiple constraint removal combination that yields the most TOR increase is also reported.

9.3 Constraint addition

The goal of the constraint addition algorithm is to observe the change in assembly rating as constraints are added. Added constraints result in an increase in the total resistance rating as well as an increase in redundancy. This algorithm along with the constraint reduction algorithm is to be used in the trade-off case studies.

The input file specifies the type of constraint being added and the constraint itself. The MATLAB script adds these constraints one at a time. In the future, this can be

improved by adding the capability to add multiple constraints at a time to identify the combination that will yield the most gain in MTR.

9.3.1 Constraint addition algorithm

The constraint addition algorithm is fairly simple. There are two additional calculations that need to be done once a new constraint is added. First, the resistance values for the original motion need to be calculated. Second, new pivot constraint combinations that contain the newly added constraint need to be processed. The first calculation uses the same subroutine as described in Section 9.1.5, and the second calculation uses the main processor from the analysis tool. The revised assembly rating for each added constraint is reported in the result.

9.4 Specified loading condition optimization

The goal of the specified loading condition optimization is to observe the change in total resistance rating for a specific loading condition. This optimization search modifies the constraint based on the same parameter as the constraint modification optimization (Section 9.1), but evaluates only the motion that is specified as the loading condition of interest. The evaluated motion set in this case is specified by the user and not generated from a set of pivot constraint wrenches. Because of this, the vectors $\vec{C}_{P1,P5}$ in the static equilibrium equation 7.22 are not formed from a pivot wrench set. Instead, the null space of the specified motion is calculated and used as $\vec{C}_{P1,P5}$. This five-system is treated as a pseudo-pivot constraint wrench set for completing the equilibrium equation.

9.5 Sensitivity analysis study

The goal of the sensitivity analysis study is to observe the change in assembly ratings when constraints are perturbed. Due to manufacturing variation and defects, constraints might either be deactivated (not in contact) or vary from their nominal location and orientation. By perturbing some of these constraints and observing the effects on the ratings, a sensitivity analysis can be done to evaluate assembly design robustness with respect to dimensional variation. To simplify the study at this stage, only the WTR rating is observed. There are three kinds of sensitivity analysis algorithm implemented in this study, namely toggle perturbation, position perturbation, and orientation perturbation. The results are reported as the percent change in WTR for each constraint that is perturbed.

The toggle perturbation algorithm deactivates one constraint at a time and evaluates the change in WTR rating. This algorithm uses the same function from constraint reduction with constraint removal specified as one at a time (Section 9.2.1).

The position perturbation algorithm perturbs the location of one constraint at a time and evaluates the change in assembly rating. This algorithm uses the same function from the constraint modification with plane search space optimization (Section 9.1.1). The amount of position perturbation is specified by the user.

The orientation perturbation algorithm perturbs the orientation of one constraint at a time and evaluates the change in assembly rating. This algorithm uses the same

function from the constraint modification with orientation search space optimization in two angles (Section 9.1.1).

9.6 Summary

This chapter discussed the algorithm and methodology implemented in the design tool to accomplish the objectives explained in Chapter 8. Chapters 10 and 11 will discuss the case studies that use the analysis tool and the design tool

CHAPTER 10

CASE STUDY BASELINE ANALYSIS RESULTS AND MODEL

VERIFICATION

This chapter discusses the results from the case studies described in Section 8.6. Explanation of the assembly rating metrics shown in the analysis result tables can be found in Chapter 6. Explanation of the constraint variable names in the constraint configuration table can be found in Chapter 7. It should be noted that the geometries of these case studies, especially ones that are derived from real parts, are abstracted and simplified to reduce computational complexity and ease data interpretation. The simplification is done by eliminating features such as rounded corners, estimating curved that are closed to straight lines as such, and ignoring part shape that does not change the constraint configuration.

10.1 Exactly constrained geometry - Thompson's chair

The Thompson's chair is an imaginary three-legged stool. The foot of each leg is a sphere with first leg constrained by an inverted trihedral (CP1, CP2, CP3), the second by a V-groove (CP4, CP5), and the third by point-on-plane contact (CP6). The seventh

point constraint, CP7, is located on top of the stool. The constraint configuration is specified in Table 10.1 and Figure 10.1. The purpose of this case study is to observe and confirm the agreement between the analysis results and theoretical kinematic principles for an exactly constrained body in 3D space.

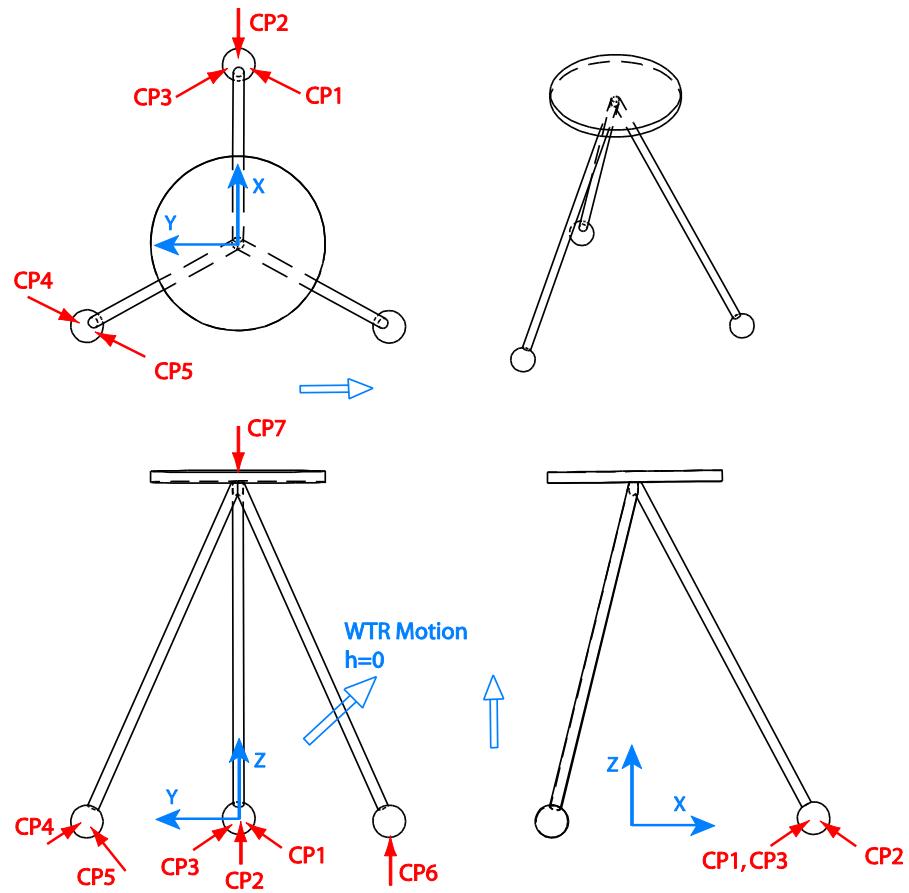


Figure 10.1 Thompson's chair constraint configuration and WTR motion

	P_x	P_y	P_z	N_x	N_y	N_z
CP1	3.570	-0.750	-0.500	0.431	0.751	0.501
CP2	4.870	0.000	-0.500	-0.867	0.000	0.498
CP3	3.570	0.750	-0.500	0.431	-0.751	0.501
CP4	-1.570	4.210	-0.500	-0.431	-0.751	0.501
CP5	-2.430	2.710	-0.500	0.431	0.751	0.501
CP6	-2.000	-3.460	-1.000	0.000	0.000	1.000
CP7	0.000	0.000	4.000	0.000	0.000	-1.000

Table 10.1 Thompson's chair constraint configuration

10.1.1 Baseline analysis results

The results are provided in Table 10.2.

Overall Rating Metric								
Weakest Total Resistance rating (WTR)			0.191	(LAR: 5.236)				
Mean Redundancy Ratio (MRR)			1.000					
Mean Total Resistance Rating (MTR)			1.001	(LAR: 0.999)				
Trade Off Ratio (TOR)			1.001					
	Screw axis direction (ω)			Screw axis coincident point (ρ)		Pitch (h)		
WTR Motion 1	0.000	-0.708	0.706	-2.000	1.724	1.731	0.001	Total Resistance Rating
	Active %	Best Resistance %		Active %		Best Resistance %		
CP1	14.3%	14.3%		CP5		14.3%		
CP2	14.3%	14.3%		CP6		14.3%		
CP3	14.3%	14.3%		CP7		14.3%		
CP4	14.3%	14.3%						

Table 10.2 Thompson's chair baseline analysis results

The WTR motion is illustrated in Figure 10.1. The WTR and MTR are relatively low compared to other case studies in this chapter. The load amplification ratio (LAR) is 5.236 (from WTR rating) and 1.001 (from MTR rating). Therefore, the reaction force of

the constraints to resist the most weakly constrained motion is about five times the input force. This is because exactly constrained assembly tends to have fewer constraints resisting the motion and hence have larger reaction forces at the support points. This is mentioned by Slocum [46] as the trade off of gaining precision of position by sacrificing load carrying capability. This trade off will be explored further in this chapter by using other case studies. The MRR is 1.000, which is expected for an exactly constrained assembly. In order to make more detailed observations, the rating matrix is provided in Table 10.3.

	CP1	CP2	CP3	CP4	CP5	CP6	CP7
Motion 1	0	0	0	0	0	0	0.329
Motion 2	0	0	0	0	0	0	0.269
Motion 3	0	0	0	0	0	0.711	0
Motion 4	0	0	0	0.829	0	0	0
Motion 5	0	0	0	0.672	0	0	0
Motion 6	0	0	0	0	0.859	0	0
Motion 7	0	0	2.871	0	0	0	0
Motion 8	0	0	1.598	0	0	0	0
Motion 9	0	0	2.078	0	0	0	0
Motion 10	0	0	0.806	0	0	0	0
Motion 11	0	2.866	0	0	0	0	0
Motion 12	0	1.595	0	0	0	0	0
Motion 13	0	0	0	0	1.381	0	0
Motion 14	0	0	0	0.401	0	0	0
Motion 15	0	0	0.747	0	0	0	0
Motion 16	0	0	0	0	0	0	0.191
Motion 17	0	0	0	0	0	0.539	0
Motion 18	1.682	0	0	0	0	0	0
Motion 19	0	0	0	1.066	0	0	0
Motion 20	0	0	0.770	0	0	0	0
Motion 21	0	0.775	0	0	0	0	0

Table 10.3 Rating matrix for Thompson's Chair (continued)

Table 10.3 Rating matrix for Thompson's Chair (continued)

Motion 22	0	0	0	0	0	0.986	0
Motion 23	0	0	0	0	0.808	0	0
Motion 24	0	0	0	0	0.712	0	0
Motion 25	0	0	0	0	0	0	0.276
Motion 26	0	0	0	0	0	0.671	0
Motion 27	0	0	0	0.859	0	0	0
Motion 28	0	0	0	0	0	0	0.640
Motion 29	0	0	0	0	0	1.068	0
Motion 30	0	0	0	0	1.391	0	0
Motion 31	0	0	0	0.539	0	0	0
Motion 32	0	0	0	0	0	0	0.634
Motion 33	0	0	0	0	0	1.058	0
Motion 34	0	2.077	0	0	0	0	0
Motion 35	0	0.604	0	0	0	0	0
Motion 36	0	0.752	0	0	0	0	0
Motion 37	0.857	0	0	0	0	0	0
Motion 38	0.806	0	0	0	0	0	0
Motion 39	0	0	0	0	1.126	0	0
Motion 40	1.593	0	0	0	0	0	0
Motion 41	0.770	0	0	0	0	0	0
Motion 42	0.770	0	0	0	0	0	0

The rating matrix shows a total of 42 evaluated motions. Because of the bi-directional nature of each screw motion, the actual number of unique pivot constraint combinations and screw motions is 21. Since each point constraint only removes a single DOF, each pivot constraint set must be composed of 5 constraint points (CP). There are 21 possible combinations in choosing 5 among the 7 constraints. In an exactly constrained geometry, each one of these pivot constraint sets needs to be a linearly independent set. This is the case for this case study because there are a total of 21 unique

combinations processed. The reciprocal motion generated by each pivot constraint set therefore is also unique.

It can be observed that for each motion (row), there are at least five constraints (for example, CP1-CP5) that are incapable of resisting the motion because they are reciprocal to it. The other remaining two constraints (CP6 and CP7) must actively resist the screw motion, or otherwise there will be unconstrained motion. It can be observed that for each motion, one of the remaining two constraints (CP6) resists the forward motion and the other (CP7) resists the backward motion. Each motion is then exactly constrained by a single active constraint, and the assembly is deterministic.

All of these observations confirm several facts summarized by Lakshminarayana [23] regarding a body constrained by seven unilateral point constraints:

- A minimum of seven unilateral point constraints is needed to achieve total restraint of a body in 3D space. If one of the constraints in this example is removed, the assembly is confirmed to lose total restraint.
- Any constraint/wrench that is linearly dependent to the pivot wrenches is unable to resist the reciprocal motion; therefore, no six of these constraints/wrenches may belong to the same linear complex; otherwise, total restraint is not achieved. When one of the constraints is modified to be linearly dependent to the set, the motion is left unconstrained, and therefore the assembly loses total restraint.
- Each motion reciprocal to five linearly independent constraints is resisted by the sixth constraint in one direction and by the seventh constraint in the other

direction. This is confirmed by the fact that only one constraint is active for each motion.

- The body constrained by seven linearly independent constraints is deterministic, namely that the actual reaction force can be found with a static equilibrium analysis because only one constraint can resist the motion.

A histogram for the total resistance for each motion is shown in Figure 10.2. The total resistance histogram is useful to observe how the most weakly constrained motions are rated relative to the rest of the evaluated motions. In this case, the motion total resistance has a relatively wide spread. Some motions are rated very strongly, while others are rated close to the average total resistance (MTR) of 1.001.

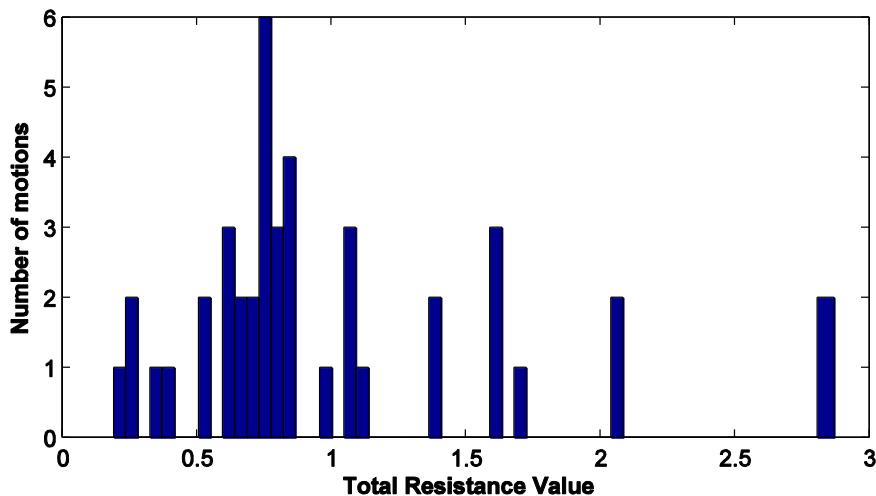


Figure 10.2 Thompson's chair total resistance histogram

10.2 Simple 3D shape geometry - generic cube

The generic cube is a 1"x1"x1" cube with point constraints on all its faces at various locations (Figure 10.3). The purpose of this geometry is to observe the behavior

of scalability of the constraint configuration as well as constraint addition and reduction trade-off study for a simple case. The constraint configurations for all 15 constraint points are specified in Table 10.4, and the different constraint combinations depending on the number of constraints for each particular case study are given in Table 10.5.

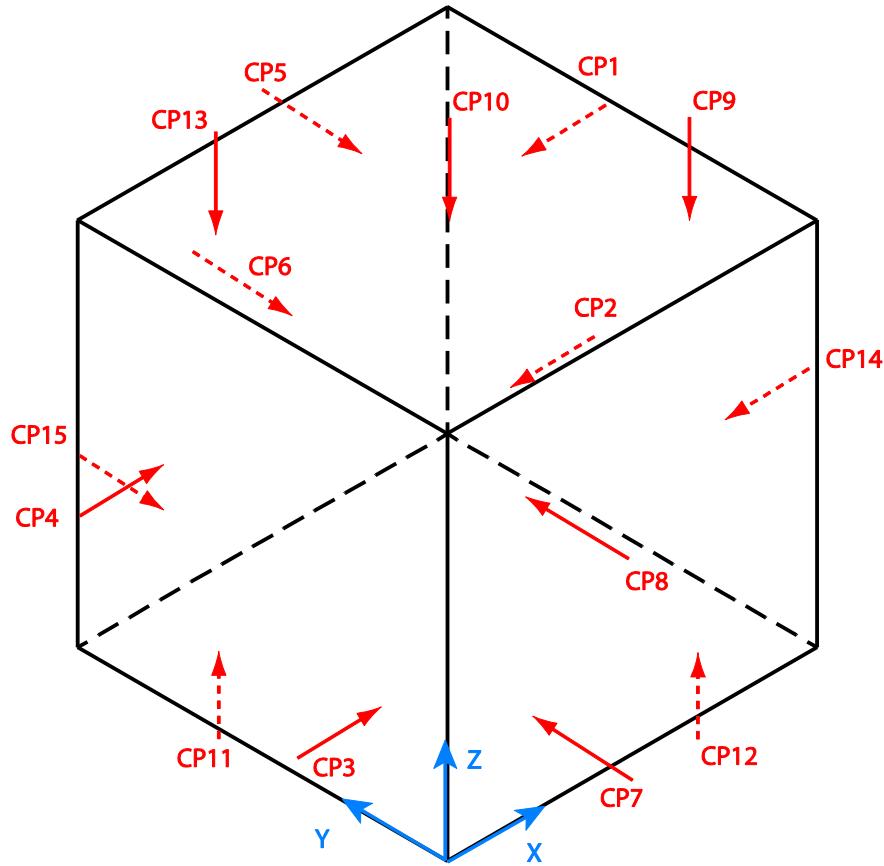


Figure 10.3 Generic cube constraint configuration

	P_x	P_y	P_z	N_x	N_y	N_z
CP1	1.000	0.750	0.750	-1.000	0.000	0.000
CP2	1.000	0.750	0.250	-1.000	0.000	0.000
CP3	0.000	0.250	0.250	1.000	0.000	0.000
CP4	0.000	0.750	0.500	1.000	0.000	0.000
CP5	0.750	1.000	0.750	0.000	-1.000	0.000
CP6	0.500	1.000	0.500	0.000	-1.000	0.000
CP7	0.250	0.000	0.250	0.000	1.000	0.000
CP8	0.250	0.000	0.750	0.000	1.000	0.000
CP9	0.750	0.250	1.000	0.000	0.000	-1.000
CP10	0.500	0.500	1.000	0.000	0.000	-1.000
CP11	0.250	0.750	0.000	0.000	0.000	1.000
CP12	0.750	0.250	0.000	0.000	0.000	1.000
CP13	0.250	0.750	1.000	0.000	0.000	-1.000
CP14	1.000	0.250	0.500	-1.000	0.000	0.000
CP15	0.250	1.000	0.250	0.000	-1.000	0.000

Table 10.4 Cube constraint configuration

Number of constraints	Constraint combination
7	CP2,CP3,CP4,CP5,CP7,CP10,CP12
8	CP1,CP2,CP3,CP4,CP5,CP7,CP10,CP12
9	CP1,CP2,CP3,CP4,CP5,CP6,CP7,CP10,CP12
10	CP1,CP2,CP3,CP4,CP5,CP7,CP8,CP10,CP11,CP12
11	CP1,CP2,CP3,CP4,CP5,CP6,CP7,CP8,CP10,CP11,CP12
12	CP1,CP2,CP3,CP4,CP5,CP6,CP7,CP8,CP9,CP10,CP11,CP12
13	CP1,CP2,CP3,CP4,CP5,CP6,CP7,CP8,CP9,CP10,CP11,CP12,CP13
14	CP1,CP2,CP3,CP4,CP5,CP6,CP7,CP8,CP9,CP10,CP11,CP12,CP13,CP14
15	CP1,CP2,CP3,CP4,CP5,CP6,CP7,CP8,CP9,CP10,CP11,CP12,CP13,CP14,CP15

Table 10.5 Cube constraint combination for different number of constraints

10.2.1 Baseline analysis results

Table 10.6 shows the baseline analysis results for the simple cube model.

Overall Rating Metric									
Weakest Total Resistance rating (WTR)				0.200	(LAR: 5.003)				
Mean Redundancy Ratio (MRR)				1.000					
Mean Total Resistance Rating (MTR)				0.486	(LAR: 2.057)				
Trade Off Ratio (TOR)				0.486					
		Screw axis direction (ω)			Screw axis coincident point (ρ)		Pitch (h)		
WTR Motion 1		0.577	-0.577	0.577	0.333	0.750	0.417	0.083	0.200
WTR Motion 2		0.667	-0.667	-0.333	0.694	0.417	0.556	-0.222	0.216
	Active %	Best Resistance %			Active %	Best Resistance %			
		CP1	14.3%	14.3%		CP5	14.3%	14.3%	
CP2	14.3%	14.3%	CP6	14.3%	14.3%	CP7	14.3%	14.3%	
CP3	14.3%	14.3%	CP4	14.3%	14.3%				

Table 10.6 Cube baseline analysis result (7 constraints)

The number of pivot constraint combinations that form a linearly dependent set is significantly higher. For the 15 constraints case, only 2.1% of the possible 4823 pivot constraint sets yield unique screw motions. This is due to the parallel and collinear nature of the relative positions and orientations of the constraints. The algorithm improves the efficiency of the analysis tool by recognizing these duplicate motions as well as the linear dependence between constraints. The overall computational improvement in this case is about one to two orders of magnitude, assuming the same amount of computation time for each iteration.

10.2.2 Scalability test case study

The purpose of this case study is to compare the analysis results between the baseline geometry and the scaled geometry. The issue of scalability was discussed in Section 3.4. The isolated reaction force couple model should resolve the scalability issues as well as the discontinuity between rotation and translation scaling that occur in other models. A scale factor of 2 is applied to the cube, and the comparison between the baseline model and the scaled model is shown in Table 10.7.

	Cube (Scale=1)	Cube (Scale=2)	% Difference
WTR	0.200	0.189	-5%
MRR	1.000	1.000	0%
MTR	0.486	0.473	-3%
TOR	0.486	0.473	-3%

Table 10.7 Cube scalability test result comparison

It can be observed that the results differ by 3-5% on average. The small difference is due to the shift in screw motion with finite pitches. Table 10.8 shows the comparison between the pitch of each motion and its associated total resistance value.

	h1	h2	h2/h1
Motion 1	Inf	Inf	-
Motion 2	0.000	0.000	-
Motion 3	0.000	0.000	-
Motion 4	0.000	0.000	-
Motion 5	0.000	0.000	-
Motion 6	Inf	Inf	-
Motion 7	0.000	0.000	-
Motion 8	0.000	0.000	-
Motion 9	-0.250	-0.500	2.000
Motion 10	0.000	0.000	-
Motion 11	0.125	0.250	2.001
Motion 12	0.250	0.500	2.000
Motion 13	0.000	0.000	-
Motion 14	0.000	0.000	-
Motion 15	0.000	0.000	-
Motion 16	0.042	0.083	1.998
Motion 17	0.167	0.333	1.999
Motion 18	-0.222	-0.445	2.001
Motion 19	0.111	0.222	2.000
Motion 20	-0.167	-0.333	1.999
Motion 21	0.083	0.167	2.001
Motion 22	Inf	Inf	-
Motion 23	0.000	0.000	-
Motion 24	0.000	0.000	-
Motion 25	0.000	0.000	-
Motion 26	0.000	0.000	-
Motion 27	Inf	Inf	-
Motion 28	0.000	0.000	-
Motion 29	0.000	0.000	-
Motion 30	-0.250	-0.500	2.000
Motion 31	0.000	0.000	-
Motion 32	0.125	0.250	2.001
Motion 33	0.250	0.500	2.000
Motion 34	0.000	0.000	-
Motion 35	0.000	0.000	-
Motion 36	0.000	0.000	-
Motion 37	0.042	0.083	1.998
Motion 38	0.167	0.333	1.999
Motion 39	-0.222	-0.445	2.001
Motion 40	0.111	0.222	2.000
Motion 41	-0.167	-0.333	1.999
Motion 42	0.083	0.167	2.001

Table 10.8 Cube scalability test – motion pitch comparison

It can be observed that for motions with zero or infinite pitch, the total resistance for the motion is identical. The reason that the resistance values are identical for zero-pitch motions (pure rotation) is as follows. When the motion is a rotation-dominant, the input torque is also scaled according to the maximum distance between the screw axis and the constraint positions. This distance is scaled by 2 for the scaled model. If the input load magnitude were not changed, the scaled model would have better resistance to the motion. This is what caused the scalability issue of the previous model by Bausch [7] and Bozzo [9]. By modeling the input load as a force couple with a properly scaled input

moment arm, the resistance value stays constant for the pure rotation motions. In the case of infinite pitch motions (pure translation), the total resistance stays constant because the resistance value for pure translation in this case depends only on the orientations of the constraints and not their positions. The orientations of the constraints do not change due to the scaling.

The difference in total resistance values then, occurs in the case where the pitch of the motion is non-zero and finite. This difference is caused by a difference in the motion pitch. The following is an explanation of the motion change. The motion change due to the scaling can be explained by understanding the change in the motion's screw geometry as the part geometry is scaled. Recall that each motion is reciprocal to a set of pivot constraints. The motion's virtual displacement at each of the pivot constraints location 'slides' perpendicularly with respect to each pivot constraint's normal. When the cube is scaled, the pivot constraints' relative distances from the origin and from each other are scaled. However, the motion must remain reciprocal to the set. Therefore, the motion reciprocal to the scaled pivot constraint locations also changes. The rotational component's virtual displacement at the pivot constraints is proportional to $\vec{\omega} \times \vec{R}$. Since the distance \vec{R} is scaled by 2, the rotational component's virtual displacement is also scaled by 2. This is not true for the translational component. Since the translational component of the virtual displacement is proportional to $h\vec{\omega}$, the scaling of the part size does not change the magnitude of the translational component. In order to maintain reciprocity, the translational component of the screw motion must be scaled by 2. This is

manifested in the pitch h being scaled by 2. Hence, the scaled motion reciprocal to the scaled pivot constraint location has a pitch multiplied by the scaling factor.

It can be observed in Table 10.8 that for finite-pitch motions, the motion pitch for the scaled model is twice that of the motion pitch for the base model. This is equal to the scale factor, which is two, except for rounding errors in the computation. Because the motion is not exactly identical, the resistance values to these kinds of motion are not the same.

Although the total resistance rating is slightly different for finite pitch motions, the difference in assembly rating is minimal (3-5%). In addition, a comparison between the total resistance histograms of the cube with scale factors of 1 and 2 shows that there is no noticeable difference in pattern of the motion resistance ratings (Figure 10.4 and Figure 10.5).

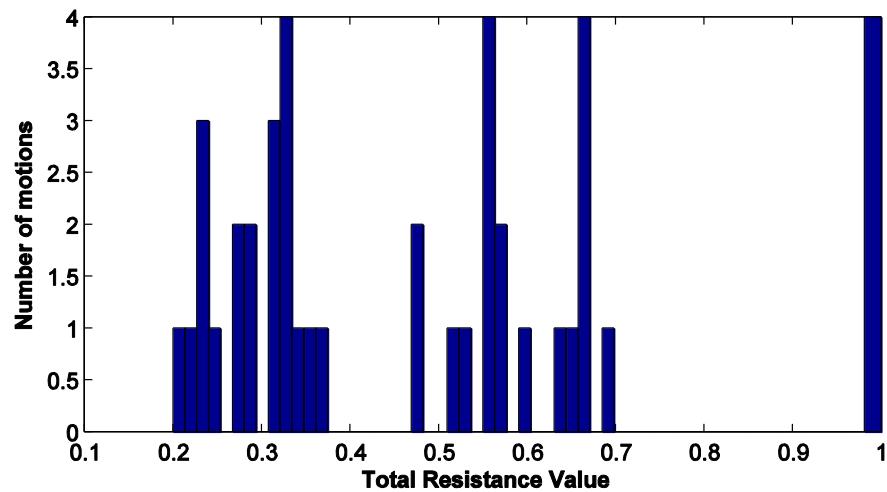


Figure 10.4 Total resistance histogram for cube (scale factor = 1)

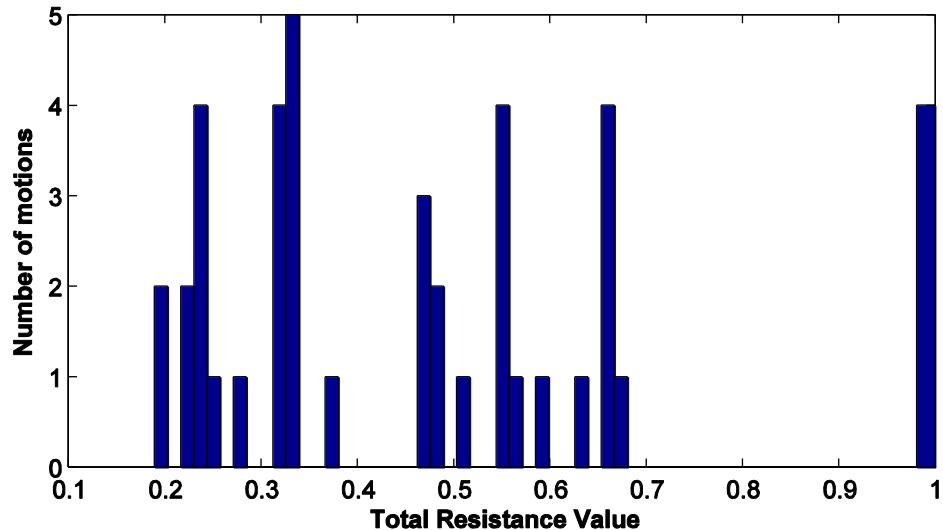


Figure 10.5 Total resistance histogramm for cube (scale factor = 2)

In summary, the scalability test case study demonstrated that the analysis tool achieved far greater consistency and scalability compared to previous models by using the isolated reaction force couple model. The motion shift for non-zero and non-infinite pitch motions can be explained by understanding the change in the screw motion geometry, which happens when pivot constraint positions are modified.

10.3 Rectilinear geometry - battery cover assembly

The third case study is a battery cover assembly. The battery cover is assembled by inserting the two lugs at an angle, lowering the part, and engaging the two snap-fits into the mating part. The constraints are provided by two cantilever snap-fits, modeled as two point constraints, and two lugs, also modeled as two point constraints. Their length-to-overall-part-length ratio is too small to be modeled as a line constraint. There are also line constraints around the perimeter of the part and a plane constraint from the bottom of

the part. The constraint configuration of the battery cover is specified in Figure 10.6 and Table 10.9.

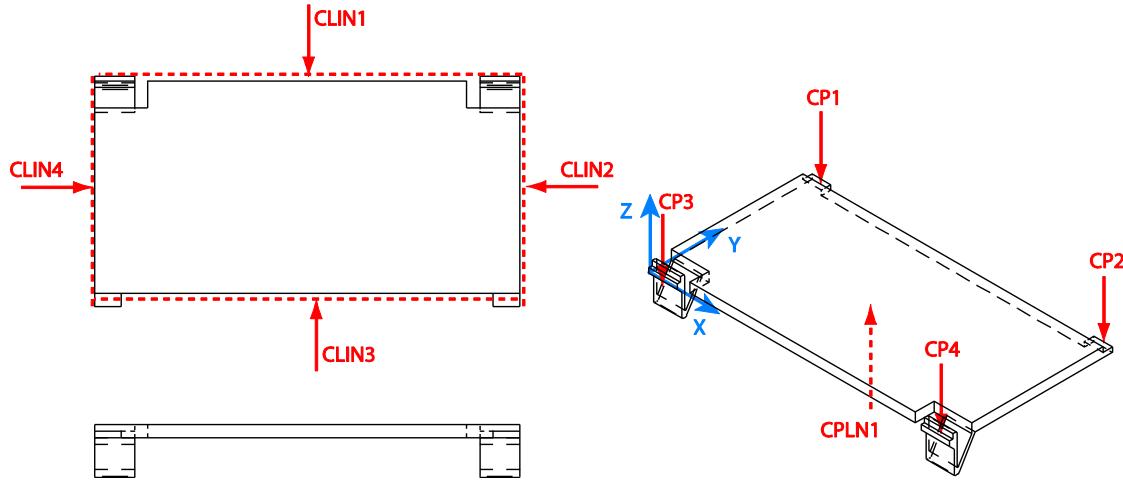


Figure 10.6 Battery cover constraint configuration

	P_x	P_y	P_z	N_x	N_y	N_z				
CP1	1	2	0	0	0	-1				
CP2	3	2	0	0	0	-1				
CP3	1	0	0	0	0	-1				
CP4	3	0	0	0	0	-1				
	P_x	P_y	P_z	w_x	w_y	w_z	N_x	N_y	N_z	L
CLIN1	0	1	0	0	1	0	1	0	0	2
CLIN2	4	1	0	0	1	0	-1	0	0	2
CLIN3	2	0	0	1	0	0	0	1	0	4
CLIN4	2	2	0	1	0	0	0	-1	0	4
	P_x	P_y	P_z	N_x	N_y	N_z	Type			
CPLN1	2	1	0	0	0	1	1			
	u_x	u_y	u_z	L_1	v_x	v_y	v_z	L_2		
CPLN_PROP1	1	0	0	4	0	1	0	2		

Table 10.9 Battery cover baseline constraint configuration

The constraint configuration in this assembly contains higher order constraints (HOC) such as line and plane constraints. The snap-fit lock is technically a line contact,

but is significantly short compared to the overall length of the part in the same direction. Therefore, it should be modeled as a point constraint. The other line contact occurs around the perimeter of the lip. Since the line contact between the lips of the battery cover and its mating part extends as long as the part length, it is obvious that this should be modeled as a line constraint. The planar constraint provided by the matic part constraints motion in the positive z direction.

10.3.1 Baseline analysis results

The baseline analysis results are shown in Table 10.10.

Overall Rating Metric							
Weakest Total Resistance rating (WTR)			2.000	(LAR: 0.500)			
Mean Redundancy Ratio (MRR)			1.500				
Mean Total Resistance Rating (MTR)			2.000	(LAR: 0.500)			
Trade Off Ratio (TOR)			1.333				
		Screw axis direction (ω)			Screw axis coincident point (ρ)		Pitch (h)
		0.000	0.000	0.000	0.000	0.000	Total Resistance Rating
WTR Motion 1		-1.000	0.000	0.000	0.000	0.000	2.000
WTR Motion 2		-1.000	0.000	0.000	0.000	2.000	2.000
WTR Motion 3		-0.707	-0.707	0.000	0.500	-0.500	2.000
WTR Motion 4		-0.707	0.707	0.000	1.500	1.500	2.000
WTR Motion 5		0.000	-1.000	0.000	3.000	0.000	2.000
WTR Motion 6		0.000	-1.000	0.000	1.000	0.000	2.000
WTR Motion 7		-1.000	0.000	0.000	0.000	0.000	2.000
WTR Motion 8		0.000	-1.000	0.000	0.000	0.000	2.000
WTR Motion 9		0.000	1.000	0.000	0.000	0.000	2.000
WTR Motion 10		1.000	0.000	0.000	0.000	0.000	2.000
WTR Motion 11		0.000	1.000	0.000	1.000	0.000	2.000
WTR Motion 12		0.000	1.000	0.000	3.000	0.000	2.000
WTR Motion 13		0.707	-0.707	0.000	1.500	1.500	2.000
WTR Motion 14		0.707	0.707	0.000	0.500	-0.500	2.000
WTR Motion 15		1.000	0.000	0.000	0.000	2.000	2.000
WTR Motion 16		1.000	0.000	0.000	0.000	0.000	2.000
		Best Resistance %				Best Resistance %	
CP1		9.4%		6.3%		CLIN1	
CP2		9.4%		3.1%		CLIN2	
CP3		9.4%		3.1%		CLIN3	
CP4		9.4%		0.0%		CLIN4	
						CPLN1	
						31.3%	
						25.0%	

Table 10.10 Battery cover baseline analysis result

The analysis results for this geometry yield a very uniform rating for most of the motions. The most weakly constrained motions are 16 different but equally rated motions. In fact, they are the complete set of evaluated motions. It can also be observed that all of the motions are zero or infinite-pitch. These refer to pure rotation or pure translation motion. A comparison between the analysis results of various geometries shows that when all constraints are located in a single plane, though their orientation is three dimensional in nature, all of the evaluated motions consist exclusively of zero or infinite-pitch motions.

The plane contact is shown to be significantly more active in resisting motion as well as being the most important reaction constraint. This is typical of higher order constraints (HOC). The possibility of an HOC to resist arbitrary motion is higher because they remove more DOF. The WTR motion set shows much symmetry. There are two types of symmetry that can be observed. The first type is symmetry in the screw axis direction $\vec{\omega}$ and $-\vec{\omega}$. For example, WTR motion 1 and 16 are symmetrical in this way. The second type is symmetry in the location of the screw axis $\vec{\rho}$ with respect to the part geometry's line of symmetry. For example, WTR motion 1 and 2 are symmetrical with respect to the part line of symmetry in the x direction. These occurrences are expected for a part with a symmetrical constraint configuration.

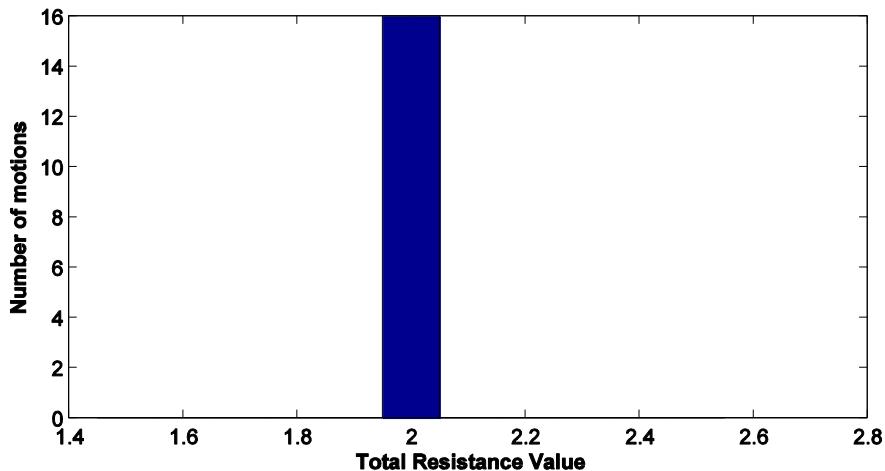


Figure 10.7 Total resistance histogram for battery cover

The total resistance histogram shows a single set of motions with a total resistance of 2.000 (Figure 10.7). The uniform rating for this case study is mainly due to the fact that almost all of the constraints are located at the edges of the part. It is common for zero and infinite pitch screw axes to pass through constraint points or be coincident with constraint directions. In these cases, the input wrench force couple moment arm is equal to the moment arm of the resisting constraints, yielding a resistance value of 1. When there are two constraints involved in this case, the total resistance rating becomes 2.

Assembly with a histogram that is very narrow tends to have very uniform strength in many different directions. In a way, it is stronger to resist more arbitrary loads compared to assembly with a wider spread in the total resistance histogram. A large deviation in the total resistance histogram implies that the assembly is stronger in some directions and weaker in others. The total resistance histogram is useful in determining the omni-directional tendency of an assembly's strength.

10.4 Axisymmetric geometry - end cap assembly

The axisymmetric assembly geometry is represented by the end cap assembly.

The end cap is part of an assembly of a medicine spray housing. The locking feature is provided by two cantilever snap-fits located symmetrically at the lips of the end cap assembly. The part is also constrained by a pin constraint and a plane constraint, both provided by the lip mating interface. The constraint configuration of the end cap assembly is specified in Figure 10.8 and Table 10.11.

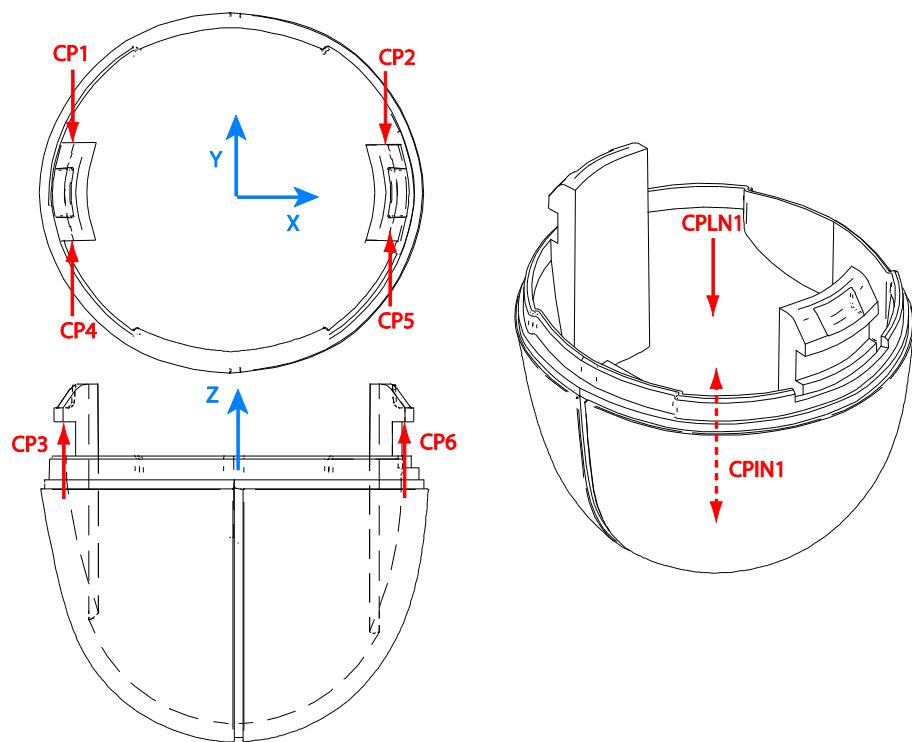


Figure 10.8 Endcap geometry and constraint configuration

	P_x	P_y	P_z	N_x	N_y	N_z
CP1	-0.588	0.161	0.188	0.000	-1.000	0.000
CP2	0.588	0.161	0.188	0.000	-1.000	0.000
CP3	-0.588	0.000	0.188	0.000	0.000	1.000
CP4	-0.588	-0.161	0.188	0.000	1.000	0.000
CP5	0.588	-0.161	0.188	0.000	1.000	0.000
CP6	0.588	0.000	0.188	0.000	0.000	1.000
	P_x	P_y	P_z	w_x	w_y	w_z
CPIN1	0.000	0.000	0.000	0.000	0.000	1.000
	P_x	P_y	P_z	N_x	N_y	N_z
CPLN1	0.000	0.000	0.000	0.000	0.000	-1.000
	r					
CPLN_PROP1	0.625					

Table 10.11 End cap constraint configuration

The assembly contacts that need to be considered for higher order constraint modeling in this case are the curved line contact in the z-axis direction, the curved line contact in the radial direction on the XY-plane, and the line contact between the snap-fit retention face and the mating part.

The curved line contact that constrains the end cap in the negative z-direction exhibits a circular plane constraint. Although both rectangular and circular plane constraints have an identical wrench system, it is important to note the circular shape of the plane because in the isolated reaction force model, the finite size of the plane of the reaction wrench is taken into account. The curved line contact that constrains the end cap in the radial direction exhibits a pin constraint because it eliminates translational DOF in the X and Y directions.

The line contact between the snap-fit retention face and the mating part does not extend to the overall part length, but its length is not insignificant. However, since the

wrench system that belongs to the line contact is also a subsystem of the plane contact, it is safe to model this as a point contact to allow more motions to be evaluated. In other words, if the snap-fit is modeled as a line constraint, the line constraint is linearly dependent on the plane constraint. If the line contact is modeled as a point contact, there is an additional combination of pivot constraints that adds more motions to consider. When too many HOC are added to a part, the number of motions evaluated can be reduced to a very small number. Hence, in order to take a more conservative approach, computational efficiency is sacrificed for the sake of evaluating more motions and gaining more data to understand the assembly. This limitation is discussed further in Chapter 12.

10.4.1 Baseline results

Table 10.12 shows the results of the analysis.

Overall Rating Metric									
Weakest Total Resistance rating (WTR)				1.000	(LAR: 1.000)				
Mean Redundancy Ratio (MRR)				1.277					
Mean Total Resistance Rating (MTR)				1.811	(LAR: 0.552)				
Trade Off Ratio (TOR)				1.418					
		Screw axis direction (ω)			Screw axis coincident point (ρ)			Pitch (h)	
WTR Motion 1		0.000	-1.000	0.000	0.588	0.000	0.000	0.000	1.030
WTR Motion 2		-1.000	0.000	0.000	0.000	0.000	0.000	Inf	1.000
WTR Motion 3		1.000	0.000	0.000	0.000	0.000	0.000	Inf	1.000
WTR Motion 4		0.000	1.000	0.000	-0.588	0.000	0.000	0.000	1.030
		Active %		Best Resistance %		Active %		Best Resistance %	
CP1		10.0%		5.0%		CPIN1		10.0%	
CP2		10.0%		5.0%					
CP3		5.0%		5.0%		CPLN1		40.0%	
CP4		10.0%		0.0%				30.0%	
CP5		10.0%		0.0%					
CP6		5.0%		5.0%					

Table 10.12 End cap baseline analysis results

The baseline analysis results show that the WTR rating for the assembly is 1.000.

Similar with the battery cover geometry, most of the constraints in this assembly are located at the edges of the parts, and therefore most of the reaction constraints have a resistance rating of 1 or greater. The total resistance histogram (Figure 10.9) shows three sets of motions that are rated similarly. The histogram is useful in observing that the motions are grouped into these sets. When the WTR motion is to be increased, the

designer needs to improve the resistance to *all* of the motions that are most weakly constrained (in this case, there are 4 motions).

Similar to the battery cover, the end cap assembly exhibits symmetry in the evaluated motion set with respect to screw axis direction (WTR motion 1 and 4). Another similarity is that the evaluated motion set consists exclusively of zero and infinite-pitch motions because all of the constraints are located on a single plane.

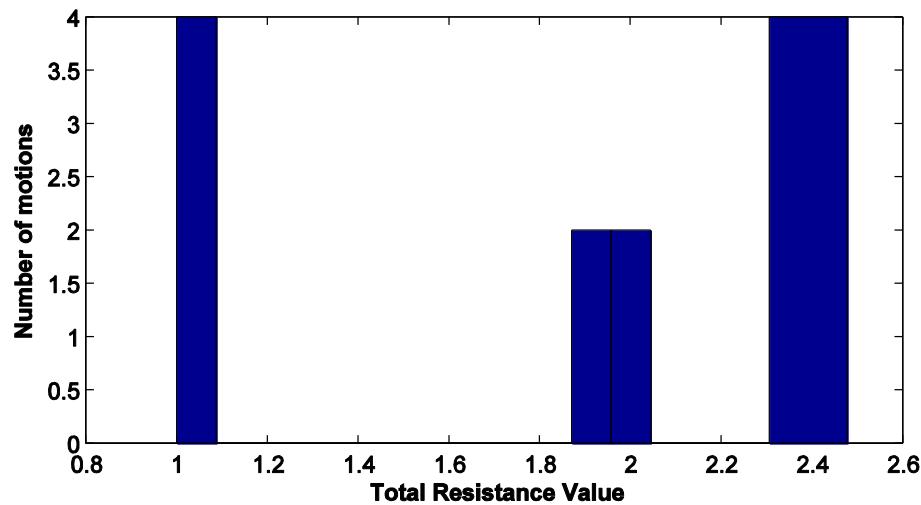


Figure 10.9 Total resistance histogram for end cap

10.4.2 HOC model vs. non-HOC model

In order to observe the computational efficiency and accuracy of the higher order constraint (HOC) modeling, a contrast between the HOC model and a non-HOC model is studied in this section. The non-HOC model discretizes the pin and plane constraints in the end cap assembly into point constraints. The pin constraint is discretized into 8 point constraints with the normal vector directed to the center of the pin. The plane constraint is

discretized into 8 parallel point constraints with the normal direction in the negative z-direction. Table 10.13 specifies the constraint configuration of this model.

	P_x	P_y	P_z	N_x	N_y	N_z
CP1	-0.588	0.161	0.000	0.000	-1.000	0.000
CP2	0.588	0.161	0.000	0.000	-1.000	0.000
CP3	-0.588	-0.161	0.000	0.000	1.000	0.000
CP4	0.588	-0.161	0.000	0.000	1.000	0.000
CP5	-0.588	0.161	0.000	0.000	0.000	1.000
CP6	0.588	0.161	0.000	0.000	0.000	1.000
CP7	-0.588	-0.161	0.000	0.000	0.000	1.000
CP8	0.588	-0.161	0.000	0.000	0.000	1.000
CP9	-0.633	0.000	0.000	0.000	0.000	-1.000
CP10	0.000	-0.633	0.000	0.000	0.000	-1.000
CP11	0.633	0.000	0.000	0.000	0.000	-1.000
CP12	0.000	0.633	0.000	0.000	0.000	-1.000
CP13	-0.448	-0.448	0.000	0.000	0.000	-1.000
CP14	0.448	-0.448	0.000	0.000	0.000	-1.000
CP15	-0.448	0.448	0.000	0.000	0.000	-1.000
CP16	0.448	0.448	0.000	0.000	0.000	-1.000
CP17	-0.633	0.000	0.000	1.000	0.000	0.000
CP18	0.000	-0.633	0.000	0.000	1.000	0.000
CP19	0.633	0.000	0.000	-1.000	0.000	0.000
CP20	0.000	0.633	0.000	0.000	-1.000	0.000
CP21	-0.448	-0.448	0.000	0.707	0.707	0.000
CP22	0.448	-0.448	0.000	-0.707	0.707	0.000
CP23	-0.448	0.448	0.000	0.707	-0.707	0.000
CP24	0.448	0.448	0.000	-0.707	-0.707	0.000

Table 10.13 End cap assembly constraint configuration (non-HOC model)

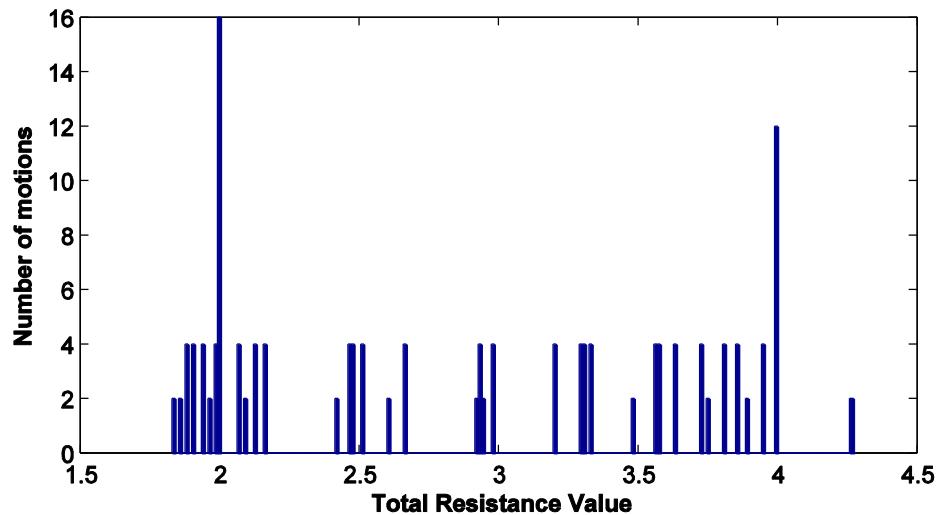


Figure 10.10 End cap total resistance histogram (non-HOC model)

	HOC model	Non-HOC model
Analysis elapsed time (sec)	0.36	12.5
Initial number of combinations	210	42504
Processed number of motion (includes duplicates due to symmetry)	20	282
Unique motion evaluated	12	148
WTR	1.000	1.829
MRR	1.277	3.028
MTR	1.811	2.855
TOR	1.418	0.943

Table 10.14 Comparison of HOC vs. non-HOC model analysis results

Table 10.14 summarizes the comparison. The computational time of the HOC model is one to two orders of magnitude less because the number of combinations the non-HOC model needs to address is much higher. The difference between the combination to begin with and the actual number of processed motions is the number of

pivot constraint combinations that are not linearly independent (the rank is less than 5).

Most often this is due to combining constraints that belong to the same HOC.

The analysis results show that the non-HOC model is rated higher in general compared to the HOC model. However, this higher rating is mainly due to the pseudo-redundancy effect as was previously explained in Section 3.4.1. The histogram (Figure 10.10) shows that the non-HOC model evaluates considerably more motions than the HOC model. A closer look reveals that many of these motions' pivot constraint sets contain only one or two wrenches that are taken from the HOC of which they are members. Therefore, these are 'fictitious' motions that do not actually need to be evaluated in the assembly. Many of them also tend to be rated higher because the resistance can come from other discretized point constraints. For example, one of the evaluated motions is specified in Table 10.15.

	Screw axis direction (ω)			Screw axis coincident point (ρ)			Pitch (h)
Motion	-0.269	-0.963	0.000	-0.587	0.164	0.000	0.000
Pivot constraints	CP1, CP2, CP5 CP9, CP23						
Active reaction constraints	CP10, CP11, CP12, CP13, CP14, CP15, CP16						

Table 10.15 Sample motion to illustrate pseudo-redundancy

CP9 is shown as one of the pivot constraints, and the motion is actively resisted by CP10-CP16, but CP9-CP16 belongs to the same constraint feature, namely the planar constraint around the lip of the end cap. Therefore, this motion is reciprocal to *and* actively resisted by the same constraint feature. This is inconsistent with the screw theory principle that a wrench reciprocal to a screw motion cannot do any work, or in this case, resist the

motion. This motion should not be in the evaluated motion set because it is contradictory to the screw theory principle and will skew the rating of the assembly. The high rating of the non-HOC model then is mostly due to these kinds of motions. This example illustrates the pseudo-redundancy effect. The HOC model eliminates the possibility of such combinations. Hence, it offers greater efficiency as well as greater accuracy.

As a bi-product of the efficiency, the number of evaluated motions is very few compared to the non-HOC model. Using the HOC model prevents evaluation of motions that violate the screw theory principle such as the one explained in the previous paragraph. However, when a line contact that is relatively short compared to the part length is modeled as an HOC, this reduces the amount of information. When the designer wants to be on the conservative side of the analysis, one should be less likely to use HOC modeling. The likelihood of a short line to be modeled as a line constraint instead of a point constraint is directly related to the HOC usage guideline. This guideline is needed to define the critical length when a line is long enough to be modeled as a line constraint. The same case applies for plane constraints.

10.5 Freeform/non-planar geometry - printer housing assembly

The final case study is focused on a real part where constraints are not necessarily oriented in a rectilinear fashion and the shape of the part leans toward freeform geometry. The organic/freeform shapes are popular with some branches of industrial design. This is where the design tool is most useful, namely to evaluate the possibility of exploring various design spaces that are not intuitive to the designer.

The printer housing assembly from an Epson Stylus C82 printer is simplified and used to represent this geometry. The printer housing is constrained by 4 threaded fasteners (the word fastener is used to prevent confusion with the term ‘screw’ in screw motion), 5 cantilever snap-fits, and line constraints along the perimeter edges of the part. The assembly constraints are specified in Figure 10.11 and Table 10.16.

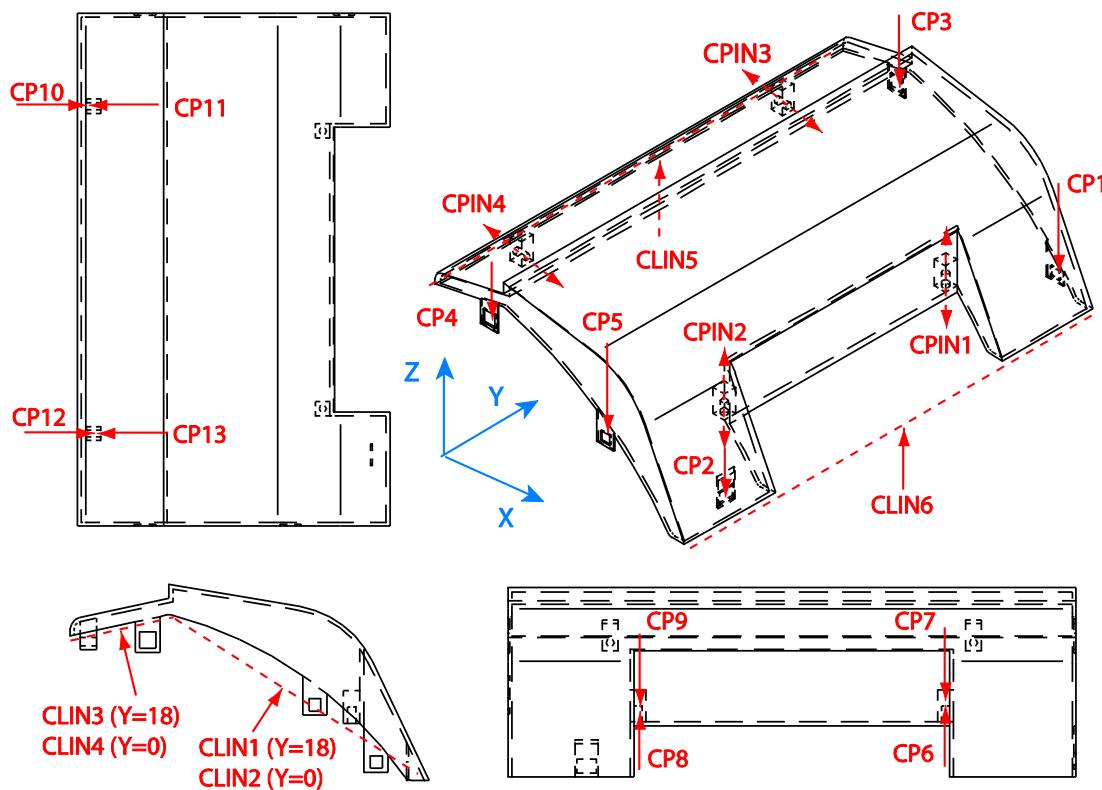


Figure 10.11 Printer housing geometry and assembly constraints

	P_x	P_y	P_z	N_x	N_y	N_z				
CP1	9.375	18.000	0.377	0.000	0.000	-1.000				
CP2	10.299	2.500	0.125	0.000	0.000	-1.000				
CP3	2.375	18.000	4.045	0.000	0.000	-1.000				
CP4	2.375	0.000	4.045	0.000	0.000	-1.000				
CP5	7.500	0.000	2.125	0.000	0.000	-1.000				
CP6	8.625	13.875	1.744	0.000	0.000	1.000				
CP7	8.625	13.875	1.744	0.000	0.000	-1.000				
CP8	8.625	4.125	1.744	0.000	0.000	1.000				
CP9	8.625	4.125	1.744	0.000	0.000	-1.000				
CP10	0.322	14.750	4.250	1.000	0.000	0.000				
CP11	0.322	14.750	4.250	-1.000	0.000	0.000				
CP12	0.322	3.250	4.250	1.000	0.000	0.000				
CP13	0.322	3.250	4.250	-1.000	0.000	0.000				
	P_x	P_y	P_z	w_x	w_y	w_z				
CPIN1	8.625	13.875	1.744	0.000	0.000	1.000				
CPIN2	8.625	4.125	1.744	0.000	0.000	1.000				
CPIN3	0.322	14.750	4.250	1.000	0.000	0.000				
CPIN4	0.322	3.250	4.250	1.000	0.000	0.000				
	P_x	P_y	P_z	w_x	w_y	w_z	N_x	N_y	N_z	L
CLIN1	7.125	18.000	3.200	-0.669	0.000	0.744	0.744	0.000	0.669	8.941
CLIN2	7.125	0.000	3.200	-0.669	0.000	0.744	0.744	0.000	0.669	8.941
CLIN3	1.551	18.000	4.754	0.978	0.000	0.211	-0.211	0.000	0.978	3.000
CLIN4	1.551	0.000	4.754	0.978	0.000	0.211	-0.211	0.000	0.978	3.000
CLIN5	0.000	9.000	4.420	0.000	1.000	0.000	0.000	0.000	1.000	11.500
CLIN6	11.000	9.000	0.000	0.000	1.000	0.000	0.000	0.000	1.000	18.000

Table 10.16 Printer housing constraint configuration

Each threaded fastener is modeled as a pin constraint with two unilateral point constraints along the fastener's axis with opposite directions. The pin constraint is due to the fastener's pin geometry, and the two-way point constraints are due to the thread in the fastener.

The perimeter edges have ribs that prevent motion in the XY plane; however, since the stiffness of the lip in this direction is must less than the unilateral constraint in the z-direction, they are neglected in the model. This is an important modeling guideline. There are features in the assembly that do not necessarily serve as constraints that remove DOF. In this case, the edge ribs/fingers serve an aesthetic function. The edge constraints force the lip of the housing and the mating part to deflect together without showing gaps. This is important for product handling because the plastic housing is flexible enough to cause these deflections, but they cannot act as effective mechanical constraints. Therefore, they are not shown in the figures for the reasons given above.

The higher order contacts that need to be considered to be modeled as HOC in this case are the line constraints along the perimeter of the housing. The lengths of line contacts modeled by CLIN1, CLIN2, CLIN5, and CLIN6 are almost equal to the part length and can readily be modeled as line constraints. Line contacts CLIN3 and CLIN4, however, are somewhat shorter than the others and slightly less than half of the overall part length in the same direction. Since CLIN3 and CLIN4 are not linearly dependent to another constraint, these lines are modeled as line constraints although they are relatively shorter.

10.5.1 Baseline analysis results

Table 10.17 shows the analysis results.

Overall Rating Metric									
Weakest Total Resistance rating (WTR)			2.437	(LAR: 0.410)					
Mean Redundancy Ratio (MRR)			4.555						
Mean Total Resistance Rating (MTR)			17.628	(LAR: 0.057)					
Trade Off Ratio (TOR)			3.870						
		Screw axis direction (ω)			Screw axis coincident point (p)		Pitch (h)		
WTR Motion 1		0.000	0.000	-1.000	0.322	3.250	0.000	0.000	2.437
WTR Motion 2		0.000	0.000	-1.000	0.322	0.000	0.000	0.000	2.438
WTR Motion 3		0.000	0.000	1.000	0.323	14.756	0.000	0.000	2.437
WTR Motion 4		0.000	0.000	1.000	0.323	17.987	0.000	0.000	2.439
		Active %		Best Resistance %					
CP1		27.2%		2.3%					
CP2		26.0%		1.5%					
CP3		27.2%		2.9%					
CP4		27.2%		2.3%					
CP5		27.2%		0.3%					
CP6		23.7%		0.0%					
CP7		23.7%		0.0%					
CP8		23.7%		0.0%					
CP9		23.7%		0.0%					
CP10		17.9%		0.0%					
CP11		17.9%		5.5%					
CP12		17.9%		0.0%					
CP13		17.9%		5.2%					

Redundant designs usually result in higher MTR value. The MTR value of 12.734, or an equivalent load amplification ratio of 0.079, shows that each constraint only carries less than 1/10 of the input load on average. This design is capable of handling large loads due to its redundancy.

A quick observation on the individual constraint's active and best resistance percentage shows that the resistance is fairly well distributed. Although the higher order constraints such as the pin and line constraints generally are more active, the point constraints' activity is not negligible.

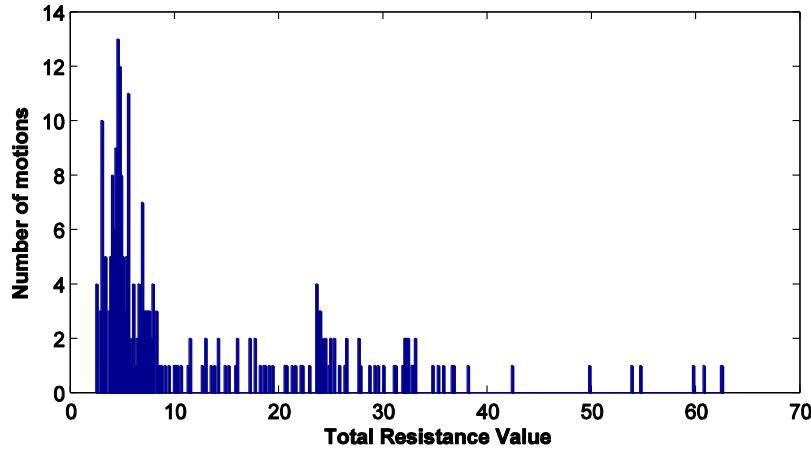


Figure 10.12 Total resistance histogram for printer housing assembly

The total resistance histogram (Figure 10.12) shows that the distribution is skewed to the left, which means that although the average total resistance (MTR) is 12.73, the majority of the motions are rated below 8.0. The symmetry in this part is less compared to the previous case studies, hence, less symmetry is observed in the evaluated motion set. Most symmetrical motions with symmetrical reaction constraints are usually

rated identically. This is not shown to be dominant in the histogram as the motion ratings are more randomly rather than uniformly distributed.

10.5.2 Snap-fits vs. threaded fasteners

If the snap-fits in the printer housing assembly are replaced by all threaded fasteners, the MRR increases by 84% (Table 10.18). Although a significant resistance quality gain is obtained (81%), the redundancy increase might be unacceptable because it leads to more internal stresses. This trade-off will be explored further in the next chapter.

	Design with 4 locating pins and 5 snap-fits	Design with 9 threaded fasteners	% Difference
WTR	1.578	4.481	184.0%
MRR	3.5283	6.494	84.0%
MTR	14.8045	26.757	80.7%
TOR	4.1959	4.121	-1.8%

Table 10.18 Comparison between snap-fit and threaded fasteners

CHAPTER 11

DESIGN OPTIMIZATION AND TRADE-OFF CASE STUDY

RESULTS

This chapter discusses the design optimization case studies described in Section 8.6. All of the optimization plots in this chapter show normalized optimization variables X1 and X2 as the horizontal axis. -1 and 1 refer to the lower and upper move limits of the optimization variable or variable group. Each variable group is defined at the beginning of each case study. Each optimization search dimension is divided into 10 increments, so there are 11 data points across each horizontal axis. The vertical axis refers to the change (in percent) in the overall assembly rating, namely the weakest total resistance (WTR), mean redundancy ratio (MRR), mean total resistance (MTR), and trade-off ratio (TOR). Explanation of these rating metrics can be found in Chapter 6. The case study geometries and constraint configurations can be found in Chapter 10. In this chapter, the word optimum is to be understood as the optimum within the design spaces, which act as the optimization constraints.

11.1 Exactly constrained geometry - Thompson's chair

11.1.1 One-dimensional optimization of the height

The purpose of this study is to optimize the height variable. The objective of the optimization is to maximize the WTR rating. The optimized variable is the position of CP7 along the z-axis. This is done by conducting a one-dimensional line search. The optimization variable and move limits are specified in Table 11.1

	Constraint variables	Line search center point	Line search direction	Move limits
Variable group X1	CP7	[0 0 4]	[0 0 1]	$-1 \leq x_1 \leq 1$

Table 11.1 Optimization variables and search space for Thompson's chair height

The optimization plot (Figure 11.1) shows that the WTR rating remains unchanged for any height. Although CP7 actively resists the most weakly constrained motion, its resistance value does not change as the height is varied. The reason is that the moment arm length between the screw axis and CP7 does not change along the line search space. The MRR also does not change for any height value because throughout the line search, the linear independence of CP7 to the rest of the constraints is not changed. Total restraint is still maintained. MTR is increased as the chair height is reduced and improved 0.8% at the minimum height of 3 inches. Therefore, a shorter chair has better

resistance to the overall evaluated motion than a higher chair. In general, the closer an object is to a uniform length in all directions the more uniform resistance it has in all directions.

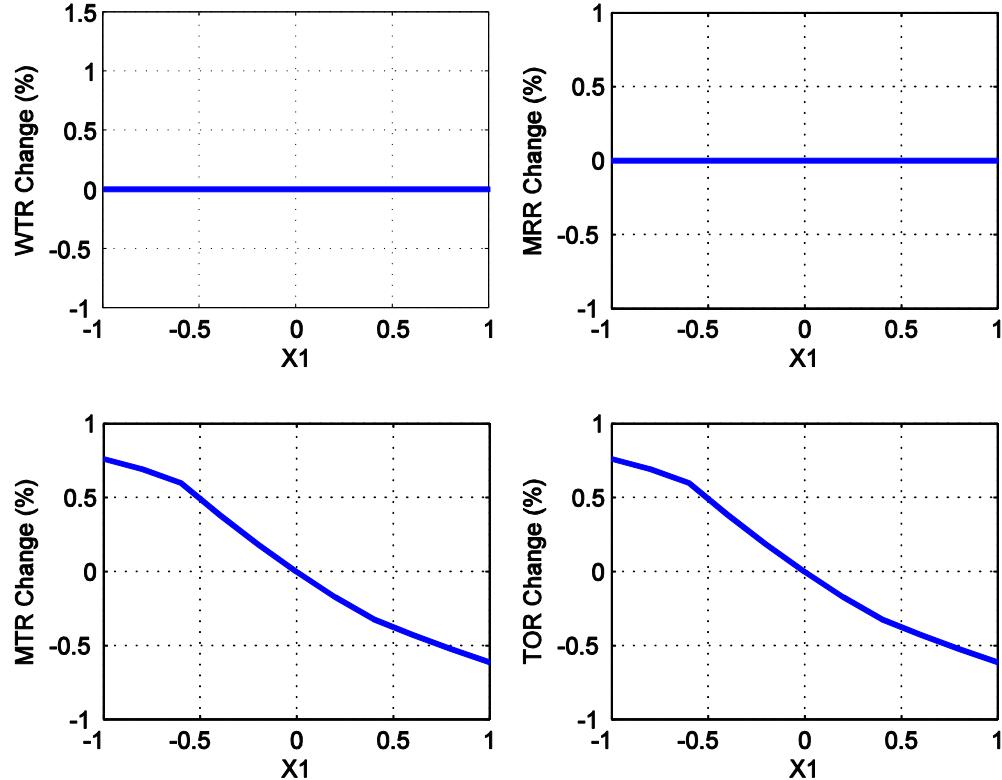


Figure 11.1 Optimization plot for Thompson's chair height variable

11.1.2 Two-dimensional optimization of the height and trihedral angle

The purpose of this study is to add one more variable to the optimization study in the previous section so as to understand the interaction between the two variables. In this study the trihedral constraint direction is added as an optimization variable. The MATLAB script does not have the capability to vary the angle of all three trihedral point

constraints as one parameter. Therefore, only the orientation of CP2 is optimized. The optimization variables and move limits are specified in Table 11.2. All positive signs in rotation angle refer to the clockwise direction.

	Constraint variables	Line search center point	Line search direction	Move limits
Variable group X1	CP7	[0 0 4]	[0 0 1]	$-1 \leq x_1 \leq 1$
	Line search direction	Orientation rotation axis		Rotation angle limits
Variable group X2	CP2	[0 1 0]		$-30^\circ \leq \theta_1 \leq 30^\circ$

Table 11.2 Optimization variables and search space for Thompson's chair height and trihedral angle

The response plot is shown in Figure 11.2. The WTR rating plot shows a discontinuity in some regions of the surface ($X2=0.2$). When a discontinuity occurs in the response surface plots (WTR), the optimum solution shifts to a different WTR motion. This results in a totally different value range. The WTR plot shows that the assembly is stronger as the angle of CP2 is changed. As the angle of CP2 is changed, the reciprocal motions where CP2 is one of the pivot constraints change. These reciprocal motions are resisted more strongly than in the previous solution. The response surface plots shows no interaction between the two variables. It can be observed that the effect of the angle of CP2 has more effect on the MTR compared to the position of CP7. MTR increases as the orientation of CP2 is closer to the z-axis. The design recommendation then is to rotate CP2 in a clockwise direction and shorten the height of the chair to increase both the WTR and the MTR ratings.

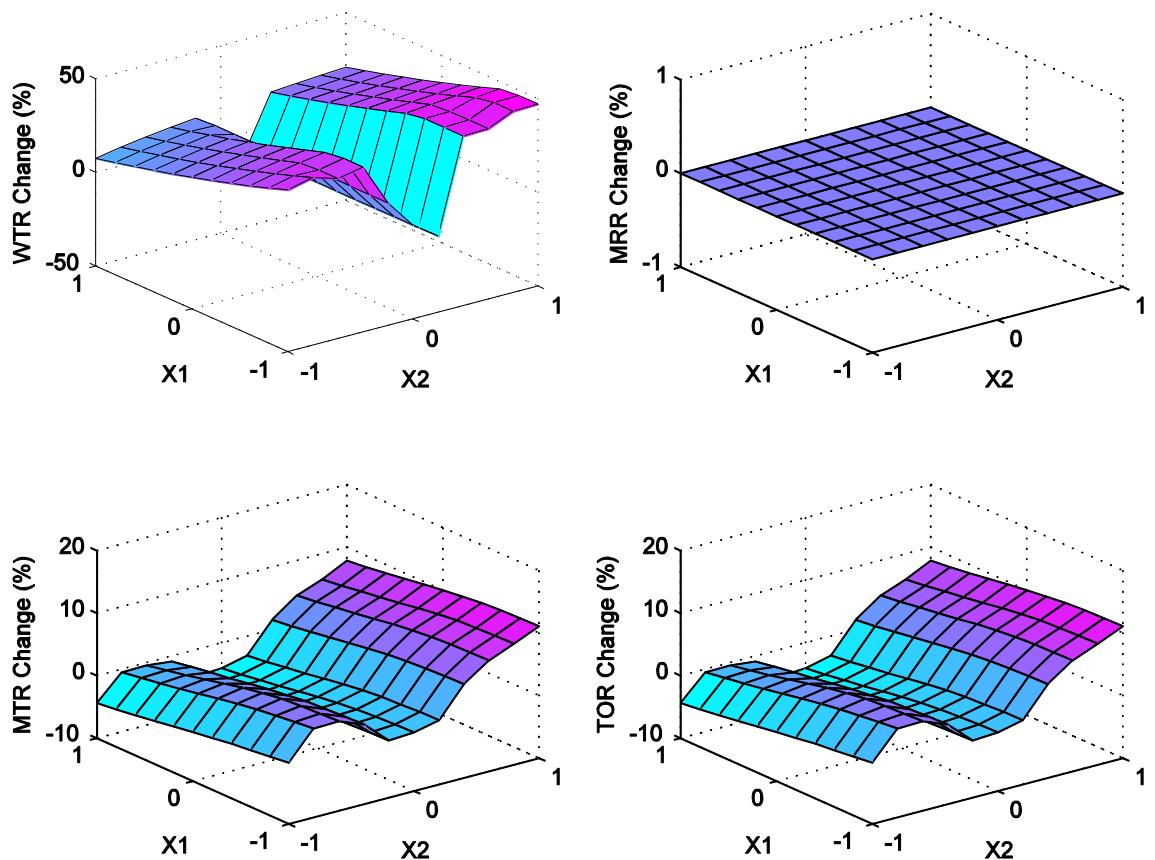


Figure 11.2 Optimization plot for Thompson's chair height and CP3 angle variable

11.2 Simple 3D shape geometry - generic cube

11.2.1 Constraint addition trade-off case study

The purpose of this case study is to gain an understanding of assembly behavior as the number of constraints is increased. Table 10.5 shows the different constraint combinations as the constraints are added.

Number of constraints	Constraint combination
7	CP2,CP3,CP4,CP5,CP7,CP10,CP12
8	CP1,CP2,CP3,CP4,CP5,CP7,CP10,CP12
9	CP1,CP2,CP3,CP4,CP5,CP6,CP7,CP10,CP12
10	CP1,CP2,CP3,CP4,CP5,CP7,CP8,CP10,CP11,CP12
11	CP1,CP2,CP3,CP4,CP5,CP6,CP7,CP8,CP10,CP11,CP12
12	CP1,CP2,CP3,CP4,CP5,CP6,CP7,CP8,CP9,CP10,CP11,CP12
13	CP1,CP2,CP3,CP4,CP5,CP6,CP7,CP8,CP9,CP10,CP11,CP12,CP13
14	CP1,CP2,CP3,CP4,CP5,CP6,CP7,CP8,CP9,CP10,CP11,CP12,CP13,CP14
15	CP1,CP2,CP3,CP4,CP5,CP6,CP7,CP8,CP9,CP10,CP11,CP12,CP13,CP14,CP15

Table 11.3 Cube constraint combinations for different number of constraints

Table 11.4 and Figure 11.3 show the overall ratings of the assembly as constraints are added by random selection. The WTR and MTR rating increases monotonically as constraints are added. However, the redundancy rating MRR also increases. The redundancy increases at approximately the same rate as the overall resistance quality, yielding an approximately constant trade-off ratio (TOR).

No. of Constraints	WTR	MRR	MTR	TOR
7	0.200	1.000	0.486	0.486
8	0.200	1.165	0.594	0.510
9	0.200	1.389	0.678	0.488
10	0.216	1.745	0.930	0.533
11	0.381	1.981	0.975	0.492
12	0.381	2.115	1.058	0.500
13	0.381	2.239	1.156	0.516
14	0.381	2.547	1.285	0.505
15	0.555	2.693	1.376	0.511

Table 11.4 Overall rating increase as constraints are added

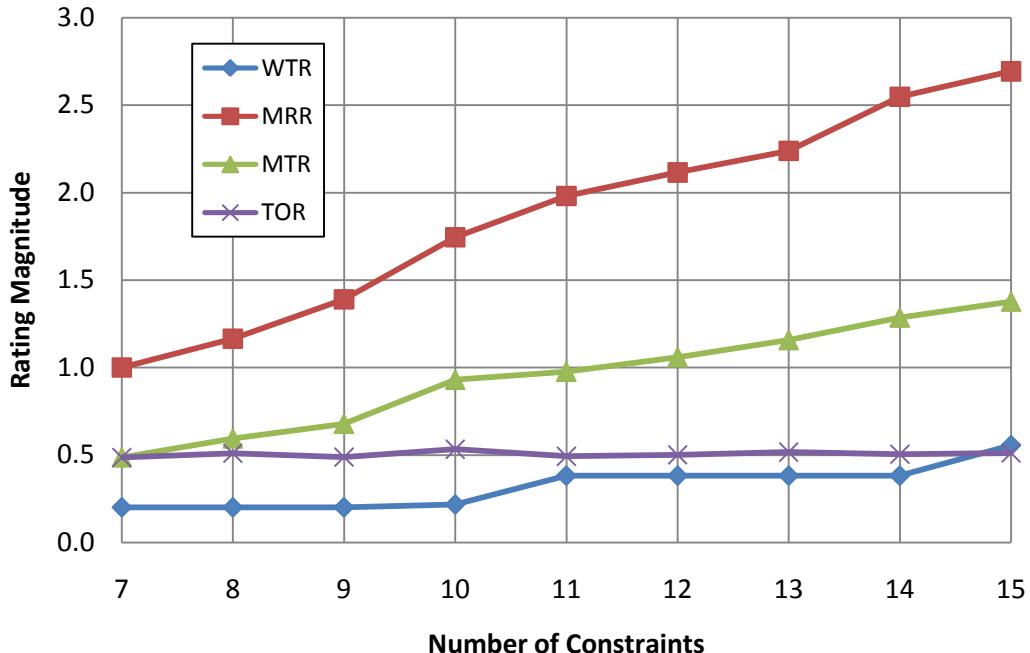


Figure 11.3 Overall rating increase as constraints are added

A closer look at the WTR rating increase shows that WTR does not always increase as constraints are added. Because WTR rating is the total resistance for the most weakly constrained motion, it is possible that newly added constraints are not actively resisting the most weakly constrained motion. This happens when the new constraint is linearly dependent to the pivot constraint set of the most weakly constrained motion or when the line-of-action causes a loss of contact from the new constraint. In this case, it does not provide additional resistance to the most weakly constrained motion, and hence no increase is observed in the WTR rating. Furthermore, by observing cases when WTR increases after the addition of constraints, better candidate constraints that are more critical to the most weakly constrained motion can be identified. When this is the concern, this procedure can be used to select better candidates among a few possible

constraints. For example, the increase of the WTR rating from 9 constraints to 10 constraints is achieved by CP11, from 10 constraints to 11 constraints by CP6, and from 14 constraints to 15 constraints by CP15.

11.2.2 Constraint reduction trade-off case study

The purpose of this case study is to gain an understanding of assembly behavior as constraints are reduced. Reduction of constraints aims to reduce redundancy in assembly DOF removal, while keeping the negative impact on assembly resistance quality minimal. This is where the rating metric TOR becomes useful. TOR is the ratio between MTR and MRR. Removing constraints can increase the TOR when the constraint to be removed is selected optimally.

Figure 11.4 shows a plot of the rating change as each constraint is removed one at a time. The horizontal axis refers to the constraint index. This refers to individual constraints regardless of their type. This index can be cross-referenced to a particular constraint. This figure is useful to identify which constraints are critical to the resistance of the assembly configuration as well as the ones that are responsible for adding redundancy. Constraints that have a negative impact on the WTR or MTR rating when removed are the ones critical to providing resistance (CP3, CP8, and CP11). Constraints that have a negative impact on MRR when removed are the ones responsible for adding redundancy (CP3, CP8). Constraints that have a negative impact on TOR when removed are ones that add more strength than redundancy. It is desirable to keep these constraints. Constraints that have a positive impact on TOR when removed are ones that add more

redundancy than strength. These constraints are the better candidates to remove (CP6, CP8, CP10, and CP14). The goal is then is to identify constraints that yield the maximum increase in TOR and remove them to optimize assembly in the context of constraint reduction. In this case, CP3 is the most critical constraint to the resistance quality. In fact, when CP3 is removed, the assembly loses total restraint. CP6 is identified as the best candidate to increase TOR when removed.

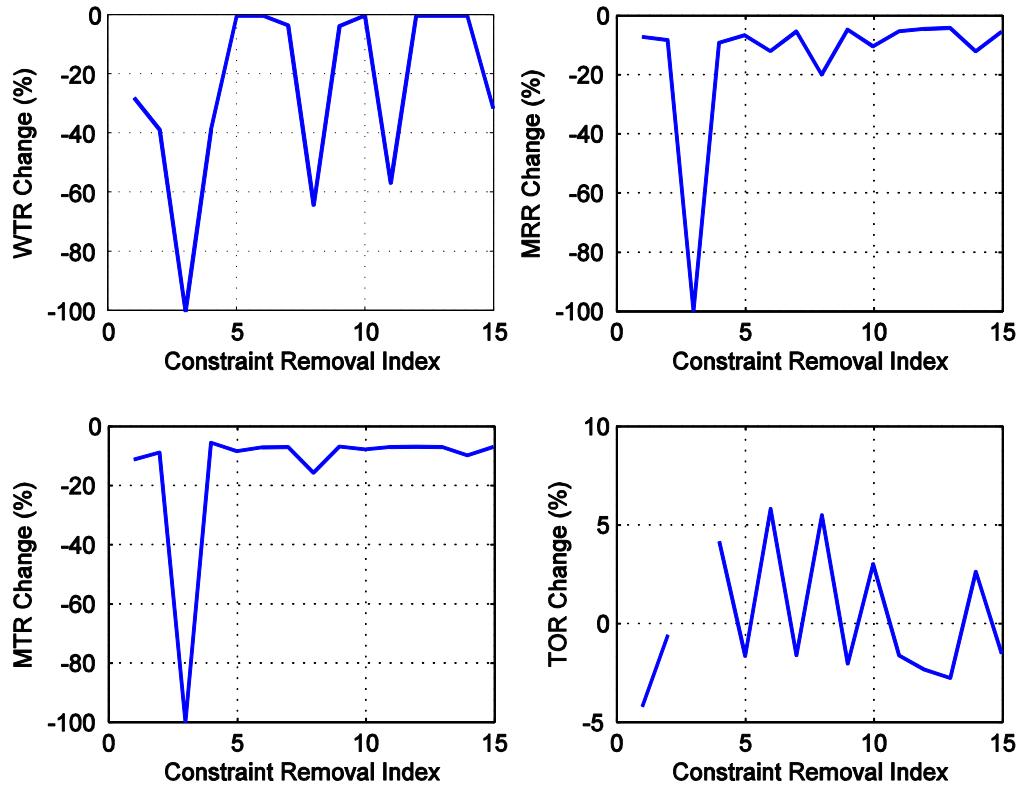


Figure 11.4 Rating change due to constraint removal (one at a time)

When the goal of the designer is to remove more than one constraint, a scheme to pick combinations of two constraints is used. Figure 11.5 shows a plot where two

constraints are removed at one time. The x-axis signifies the index of the scheme of different combinations. In this case, there are a total of 105 different combinations of 2 constraints to remove from 15 constraints.

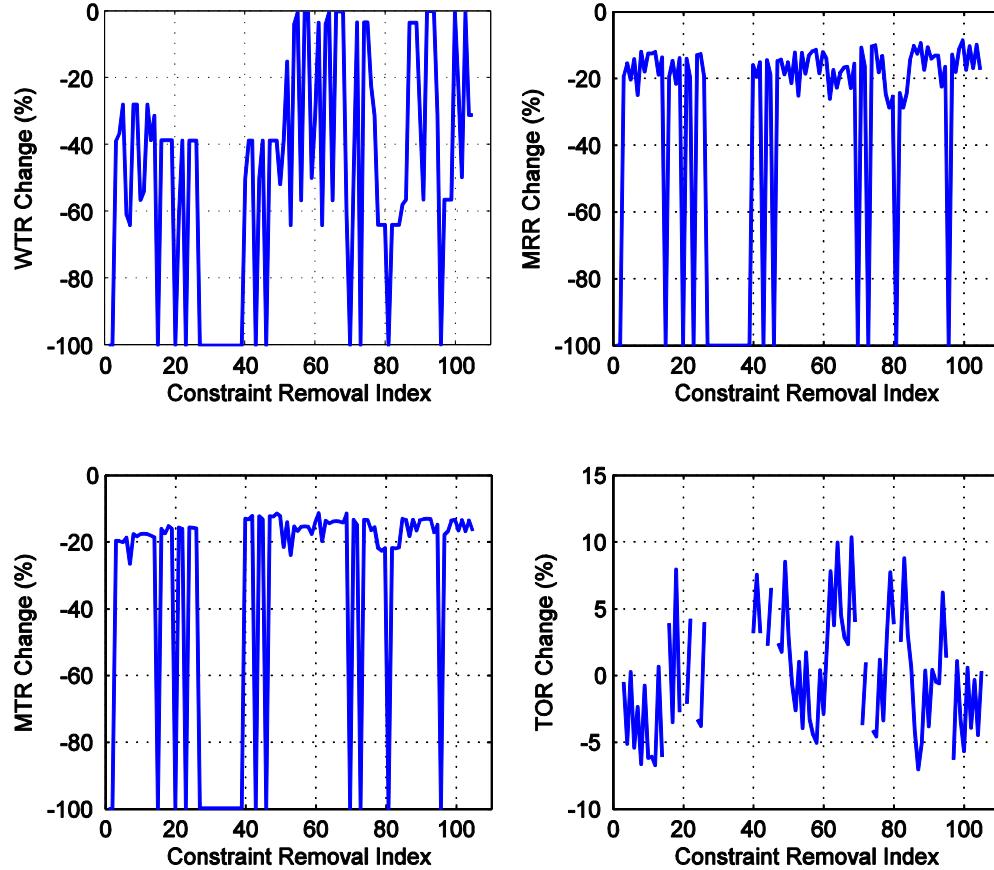


Figure 11.5 Rating change due to constraint removal (two at a time)

In the plot, there are more cases where the assembly loses a significant resistance quality. This is possible because as more constraints are removed, the possibility for an assembly to lose its ability to resist motion is significantly higher. In some cases, the assembly loses total restraint altogether. It is very important to note constraint

combinations that are critical in maintaining total restraint. In this case, removing constraint combination #68 yields the maximum possible TOR increase. This constraint combination index cross-references to constraints CP6 and CP14. In the rating matrix, there is a motion that is resisted by CP6 and CP14, but not by any other constraints. Once these constraints are eliminated, the motion becomes unconstrained.

The procedure explained above can be further extended to remove more constraints at a time up to the maximum possible number. In this case study, it is possible to remove up to 8 constraints at a time. It is impossible to remove more than 8 constraints because the number of constraints remaining will be below the required seven point constraint requirement to achieve total restraint. Table 11.5 and Figure 11.6 show the overall rating changes as more constraints are removed.

No of Constraints removed	Maximum Possible TOR increase (%)	WTR	MRR	MTR	Removed Constraints							
0	0.0	0.555	2.693	1.376								
1	5.8	0.555	2.370	1.282	6							
2	10.4	0.555	2.091	1.180	6	14						
3	15.5	0.555	1.820	1.075	6	14	10					
4	19.0	0.340	1.555	0.945	6	14	10	1				
5	17.2	0.340	1.393	0.834	6	14	10	1	2			
6	17.2	0.340	1.262	0.756	6	14	10	7	2	4		
7	15.0	0.340	1.131	0.665	6	14	10	7	2	4	12	
8	11.9	0.340	1.000	0.572	6	14	10	7	2	4	12	13

Table 11.5 Overall rating change as more constraints are removed

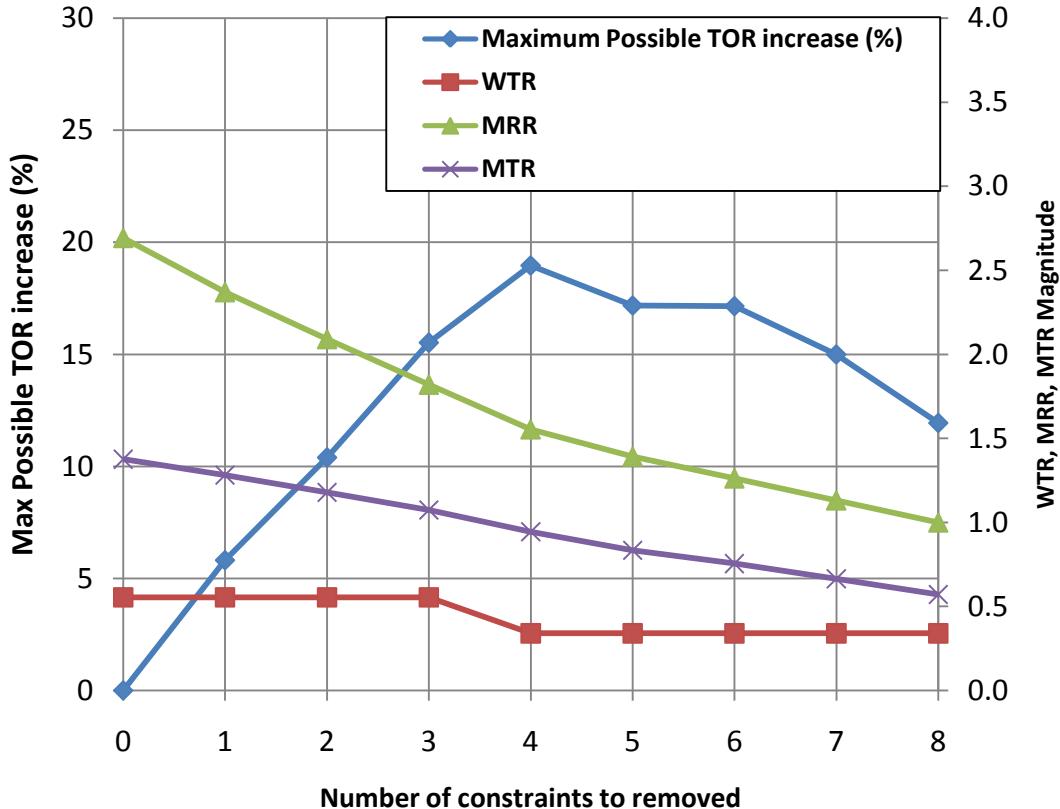


Figure 11.6 Overall rating change as constraints are removed

Note that there are two vertical axes on the plot. The left side vertical axis is for the TOR increase (%), and the right side is for the assembly ratings WTR, MRR, and MTR ratings. The first set of results is the baseline results (zero constraint removed, zero TOR increase). It can be observed that as more constraints are removed at a time, the assembly ratings WTR, MRR, and MTR monotonically decrease. MRR reduces to 1.0, which is expected for an exactly constrained assembly (7 constraints). The constraints to remove are selected optimally in a way that minimizes the reduction in MTR and maximizes the reduction in MRR or in other words, maximizing TOR increase. There might be multiple possibilities to increase TOR, but the selected ones are the best

candidates. The WTR rating stays constant until 4 constraints are removed. This means that the constraint combinations identified up to 3 constraints (CP6, CP14, and CP10) that do not actively resist the most weakly constrained motion.

An important observation in this case study is that the maximum possible gain in TOR keeps increasing until removal of 4 constraints at a time (19% increase) and decreases afterwards. This means that removing 5 constraints from the assembly does not yield a better result compared to removing 4 constraints. In other words, a pattern of diminishing return occurs in the context of optimizing the assembly by reducing redundancy.

Another important observation is the final constraint configuration after 8 constraints are removed. At this point, the assembly is composed of 7 constraints. However, the constraint configuration here is different compared to the baseline model at the beginning of this case study. The reason is that in the constraint reduction scheme, constraints are removed from 15 to 7 in an optimal manner to maximize TOR increase. As a result, it is very likely that the final configuration is different and better than the baseline configuration. Table 11.6 shows the comparison between the 7-constraint baseline configuration and the 7-constraint improved configuration.

	Baseline 7cp cube	Improved 7cp cube	% Difference
WTR	0.200	0.340	41%
MRR	1.000	1.000	0%
MTR	0.486	0.572	15%
TOR	0.486	0.572	15%

Table 11.6 Comparison between baseline and improved 7 constraint cube

The constraint reduction scheme described in this section can be used as a design tool to optimize constraints by way of adding all possible constraint candidates and reducing until the desired number of constraints is reached. In this case, the procedure was:

1. The baseline model was composed of randomly selected 7 constraints to start with, assuming that 7 is the goal number of constraints.
2. All possible candidate constraints are added to the constraint configuration (in this case, up to 15 constraints).
3. Constraints are reduced optimally until the number of constraints is back to 7 (the goal number of constraint).

11.3 Simple rectilinear geometry – planar battery cover

These studies serve the purpose of verifying the agreement between the analysis and design tool and the design principles. Therefore, the actual amount of percentage increase or decrease in the overall ratings is of less interest.

11.3.1 Design principle test case study – leverage

The purpose of this case study is to verify that the design tool demonstrates the commonly known design principle of leverage. The goal of the optimization is to maximize the WTR rating. In order to simulate this, the optimization variables are the locations of constraints CP1 and CP3. The optimization variables and search space are specified in Table 11.7 and Figure 11.7.

	Constraint variables	Line search center point	Line search direction	Move limits
Variable group X1	CP1	[1 2 0]	[1 0 0]	$-1 \leq x_1 \leq 1$
Variable group X2	CP3	[1 0 0]	[1 0 0]	$-1 \leq x_2 \leq 1$

Table 11.7 Optimization variables and search space for leverage case study

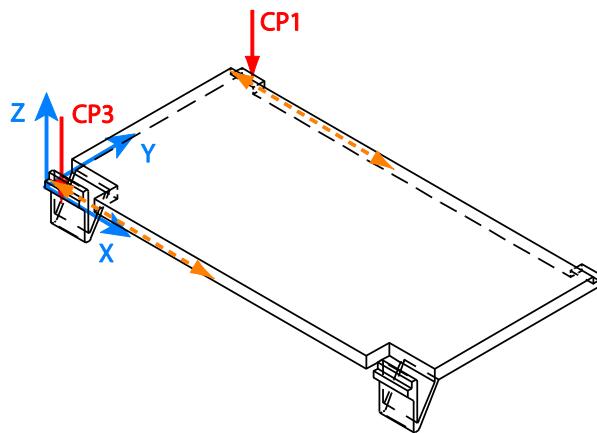


Figure 11.7 Optimization variables and search space for leverage case study

The response surface plot in Figure 11.8 shows that the constraints CP1 and CP3 reach optimal locations with respect to the WTR rating at the edge of the part. This confirms the design principle of leverage because constraints are more effective in resisting moments as they move to the edges of the part because of the increased moment arm for the reaction torque. The MTR plot also shows that CP1 and CP3 are most effective in resisting overall motion when they are located symmetrically with respect to the part. This principle of symmetry is tested in the case study described in the next section. The design therefore is already optimal by locating the snap-fit and lug at the edges of the part.

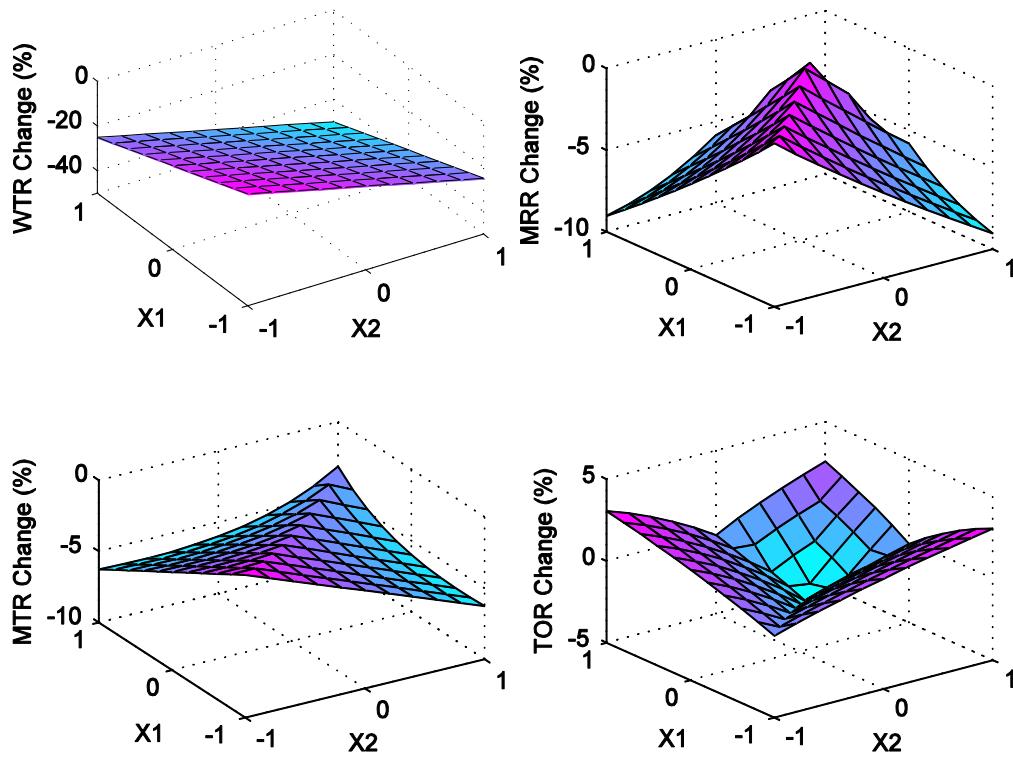


Figure 11.8 Response surface plot for leverage design principle case study

The MRR plot shows that redundancy is maximal at positions where CP1 and CP3 are parallel and symmetrical with respect to the part's line of symmetry. When CP1 and CP3 are located symmetrically, the pivot constraint combination that contains both CP1 and CP3 becomes a linearly dependent set of rank less than 5. This indirectly increases the redundancy of the constraint configuration. When CP1 and CP3 are perturbed from their symmetrical positions, the redundancy decreases. This is called the linear dependency bifurcation shift and will be discussed as a limitation of the response surface plots in Chapter 12. Because the MRR change is due to the linear dependency bifurcation point, the MRR does not accurately measure the true redundancy of the

assembly. The TOR plot increase at the edges of the move limits is mainly influenced by the reduction of redundancy. Therefore, the MRR and TOR plots are only useful when constraints are added or reduced. They are not of much interest in the context of constraint modification redesign because the number of constraints remains the same. From this point forward, comments on the MRR and TOR plot will be minimal unless constraint reduction or addition is involved.

11.3.2 Design principle test case study – symmetry

The purpose of this case study is to verify that the design tool demonstrates the commonly known design principle of symmetry. The goal of the optimization is to maximize the WTR rating. In order to simulate this, the baseline model is modified slightly. Instead of having two point constraints at the longer edge of the part (CP1, CP2, CP3, and CP4), one point constraint is placed on each edge (Figure 11.9, Table 11.8).

	P_x	P_y	P_z	N_x	N_y	N_z
CP1	2	2	0	0	0	-1
CP2	2	0	0	0	0	-1
CP3	0	1	0	0	0	-1
CP4	4	1	0	0	0	-1

Table 11.8 Battery cover symmetry case study point constraint

The locations of these constraints are the optimization variables. The constraints CP1 and CP2 are linked and move together along the top and bottom lines in the illustration, and constraints CP3 and CP4 are linked and move together along the side

lines in the illustration. The optimization variables and search space are specified in Table 11.9 and Figure 11.9.

	Constraint variables	Line search center point	Line search direction	Move limits
Variable group X1	CP1, CP2	[2 2 0]	[1 0 0]	$-1 \leq x_1 \leq 1$
Variable group X2	CP3, CP4	[0 1 0]	[0 1 0]	$-1 \leq x_2 \leq 1$

Table 11.9 Optimization variables and search space for symmetry case study

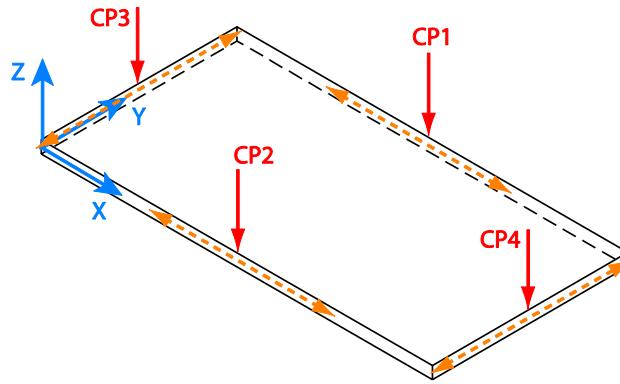


Figure 11.9 Optimization variables and search space for symmetry case study

The response surface plots (Figure 11.10) show that the constraints reach their optimal locations at the middle of the part where symmetry occurs both horizontally and vertically. This is applicable in optimizing the WTR as well as the MTR rating.

The following is an explanation for the WTR and MTR rating decrease as the constraints move away from their symmetrical locations. In this case, the screw axes of the reciprocal motions pass through the locations of some of the pivot constraints. Therefore, whenever one of the modified constraints (CP1, CP2, CP3, and CP4) is a

member of the pivot constraint combination, the reciprocal motion screw axis changes. A closer observation shows that the reason for the decrease in the WTR and MTR ratings is a consequence of the change in motion. This reveals that there are two different ways a rating can change (positively or negatively). One way is due to changing the locations or orientations of the constraints that actively resist a set of unchanged motions. The other way is due to changing the location or orientation of at least one of the constraints that belongs to the pivot constraints set, which then changes the reciprocal motion to be resisted by a set of unchanged constraints. In this case, the decrease is due to the latter.

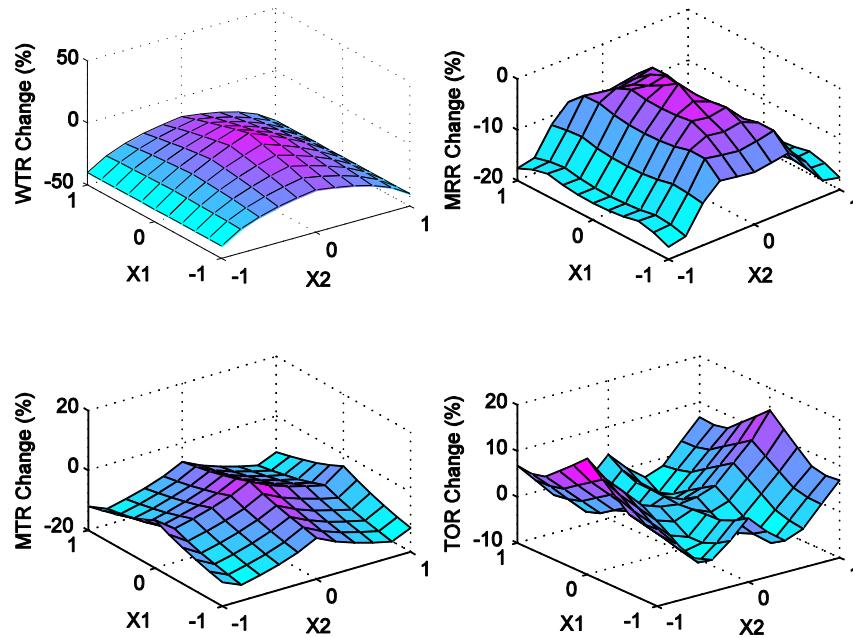


Figure 11.10 Response surface plot for symmetry design principle case study

Similar to the leverage case study in the previous section, the redundancy (MRR) is maximal where constraints are symmetrically located due to the fact that when

constraints are not symmetrical, they are less likely to be linearly dependent to the motions. This effect can be ignored in practical design applications. This is one of the limitations of this work and will be discussed in Chapter 12.

11.3.3 Study type: Design principle test – optimal line-of-action

The purpose of this case study is to confirm the design principle of line-of-action. The goal is to maximize the WTR rating. The optimization variables are the orientations of CP1, CP2, CP3, and CP4. The angular search space is the rotation about two different axes with a rotation angle limit of 45 degrees in both directions. The optimization variables and search space are specified in Table 11.10 and Figure 11.11.

	Constraint variables	Orientation rotation axis	Rotation angle limits
Variable group X1	CP1, CP2, CP3, CP4	[1 0 0]	$-45^\circ \leq \theta_1 \leq 45^\circ$
Variable group X2	CP1, CP2, CP3, CP4	[0 1 0]	$-45^\circ \leq \theta_2 \leq 45^\circ$

Table 11.10 Optimization variables and search space for line-of-action case study

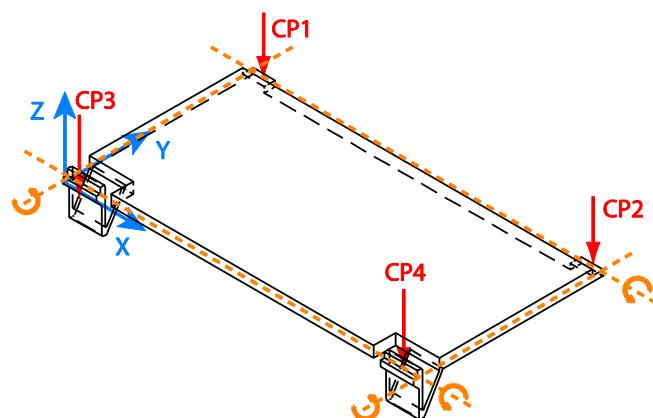


Figure 11.11 Optimization variables and search space for line-of-action case study

The response surface plot (Figure 11.12) for this case study shows that WTR reaches a maximum in the middle of the plot where the constraints are oriented perpendicular to the plane of the geometry ($X_1=0$, $X_2=0$). The WTR rating is the total resistance to this motion only. The orientation that provides optimal resistance with the least reaction force is the perpendicular orientation. This confirms the design principle of line-of-action.

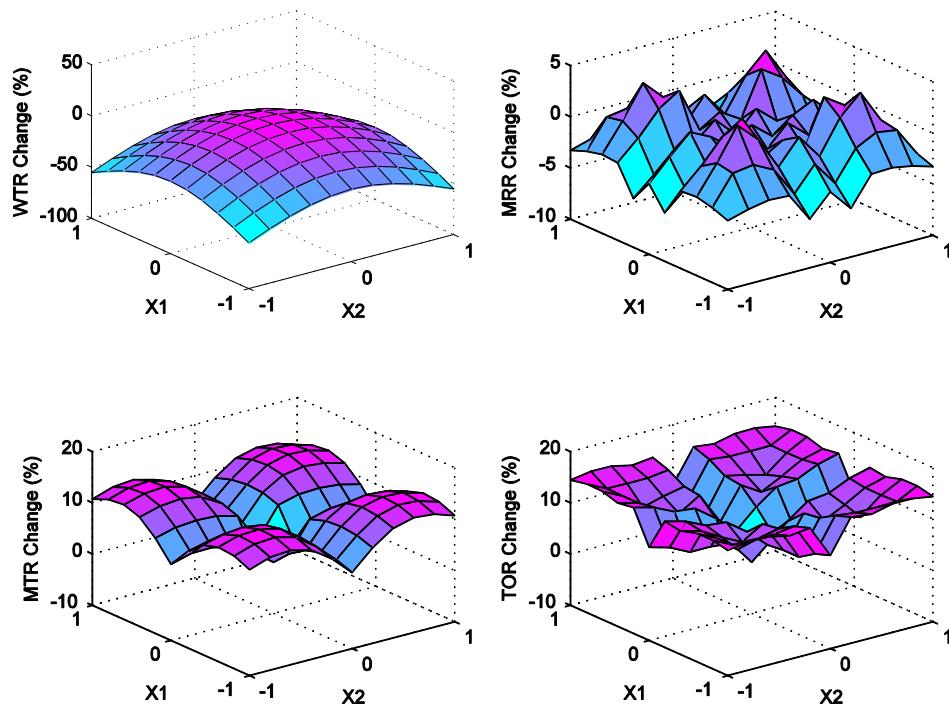


Figure 11.12 Response surface plot for line-of-action design principle case study

The MTR plot show that its maximum value occurs at a different angle. The reason for this is that the motion set changes significantly. A slight perturbation in the angular direction usually causes new motions that were not considered previously because the pivot constraint set was linearly dependent. The slight perturbation causes the constraint to be linearly independent from the set, although it is very close to being

linearly dependent. The new motions that are generated due to the new linearly independent pivot constraint set are usually rated as strongly resisted. This skews the average total resistance rating (MTR) to be higher. Due to this limitation, it is recommended that the orientation optimization be used only to optimize the most weakly constrained motion (WTR) or when critical or common design loading directions are known. The MRR plot shows many discontinuities due to the bifurcation shifts in linear dependency among constraints. This limitation will be discussed further in Chapter 12.

11.4 Axisymmetric geometry - end cap assembly

11.4.1 Design optimization case study – constraint addition

The purpose of this case study is to optimize the end cap assembly by adding constraints. The focus of this case study is to observe the extent to which adding constraints benefits the assembly and to identify the point of diminishing returns. In this case, the goal is to optimize the increase of the WTR and MTR ratings compared to the number of constraints being added. The constraint features being added are snap-fits around the lip of the end cap. The constraints are added as a set because each snap-fit is modeled as three unilateral point constraints. Figure 11.13 illustrates the radially symmetrical position of the snap-fits.

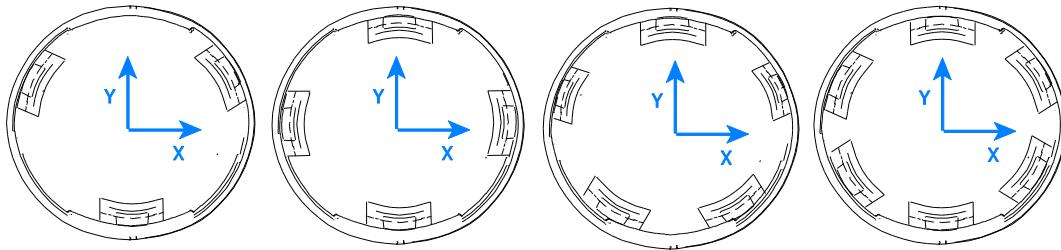


Figure 11.13 End cap snap-fit addition configuration

The plot in Figure 11.14 shows that adding more snap-fit features beyond 4 snap-fits does not yield any significant gain in the WTR rating. In addition, adding more snap-fit features beyond 5 does not yield any significant gain in any rating. As snap-fits are added, the MRR increases at the same rate as the MTR; therefore, the TOR remains approximately constant. This study is useful in determining the point at which adding more constraints, in this case more snap-fit features, yields diminishing returns in assembly rating.

The design recommendation based on these results is to increase the number of snap-fits up to 3 or 4. There is significant gain in the WTR for increasing from 2 to 3 snap-fits (68%) and from 3 to 4 snap-fits (32% additional). The trade-off ratio decreases, but the redundancy level at this point might be acceptable.

No of snap-fits	WTR	MRR	MTR	TOR
2	1.000	1.277	1.811	1.418
3	1.680	1.955	2.328	1.191
4	2.000	2.367	2.603	1.099
5	1.921	2.580	3.513	1.362
6	2.000	2.935	3.544	1.208

Table 11.11 Rating change as the number of snap-fits is increased

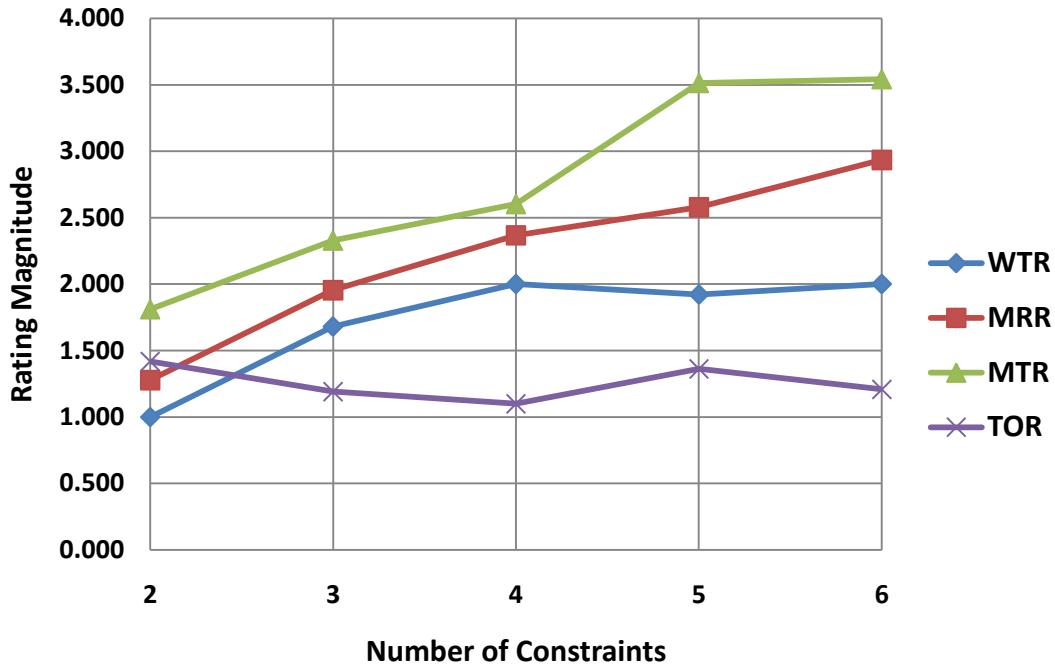


Figure 11.14 Rating change as the number of snap-fits is increased

11.4.2 Design optimization case study – snap-fit location

The purpose of this case study is to optimize the location of two of the snap-fits (CP1 through CP6) along the end cap lip curved line. The goal is to maximize the WTR rating. The optimization variables are the locations of two of the snap-fit features. The point constraints that are associated with a single feature are grouped and move together as the feature is modified. There are 3 point constraints that are associated with each snap-fit feature. Each snap-fit feature, however, is moved independently. Therefore, this is a 2-dimensional optimization search. The first group of variables (X_1) consists of CP1, CP2, and CP3 and is associated with the first snap-fit feature. The second group of

variables (X2) consists of CP4, CP5, and CP6 and is associated with the second snap-fit feature.

The search space is a circular curve. The location of each snap-fit is varied along this curve. The circular search space is defined by an axis about which the snap-fit location is rotated. Table 11.12 specifies the optimization variables, the rotation axis location and orientation, and the rotation angle limits. Figure 11.15 illustrates the search space and move limits.

	Constraint variables	Curved line search center of rotation	Curved line search rotation axis	Rotation angle limits
Variable group X1	CP1, CP2, CP3	[0 0 0]	[0 0 1]	$-45^\circ \leq \theta_1 \leq 45^\circ$
Variable group X2	CP4, CP5, CP6	[0 0 0]	[0 0 1]	$-45^\circ \leq \theta_1 \leq 45^\circ$

Table 11.12 Optimization variables and search space for snap-fit location

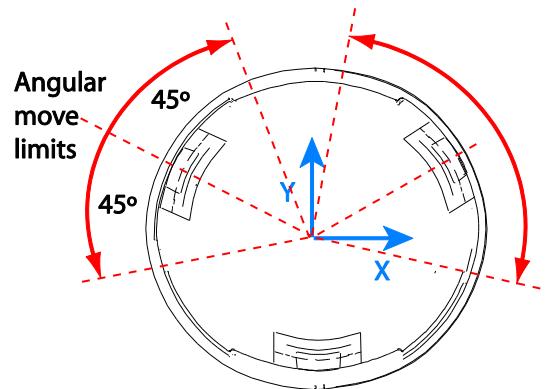


Figure 11.15 Optimization variables and search space for snap-fit location

The response surface plot is shown in Figure 11.16. It can be observed that WTR is maximal when the constraints are located in a radially symmetric pattern. The MTR

plot shows that rotating the snap-fit location up to approximately ± 30 degrees does not immediately result in significant decrease in rating. Further rotation will decrease the MTR rating. The snap-fit is most effective when located at the radial symmetric position. This indirectly confirms the symmetry design principle discussed in the previous case study.

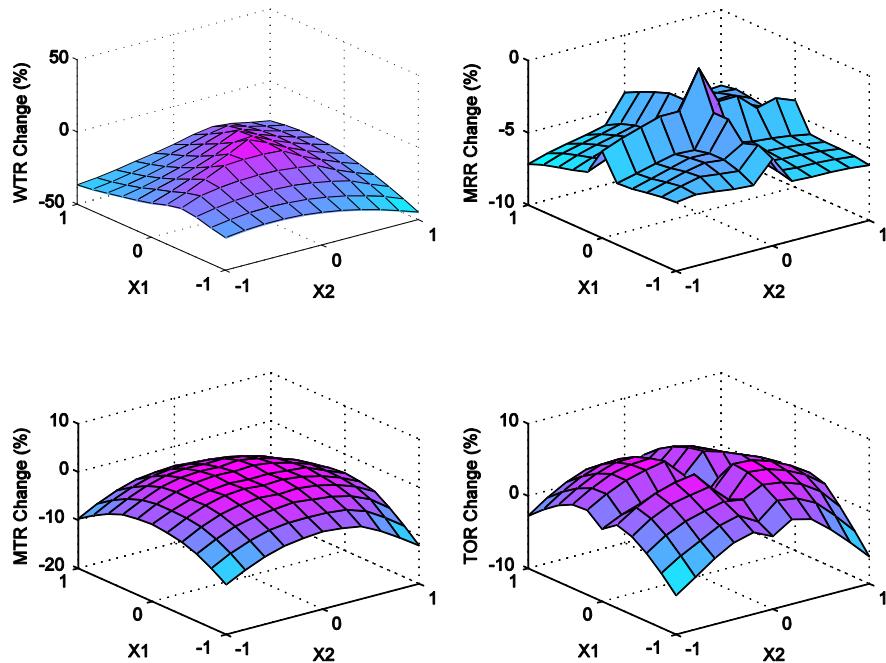


Figure 11.16 Response surface cplot for end cap snap-fit location optimization

It is also helpful to explore the areas in the WTR plot where the rating decreases significantly. The WTR rating decreases the most (up to more than 30%) as both snap-fits move closer to the bottom snap-fit and to each other. This causes all three snap-fits to be located in one half of the circular plane of the assembly. This constraint configuration and the most weakly constrained motion screw axis (zero-pitch) are illustrated in Figure 11.17. There is another weakly constrained motion screw axis that is symmetrical (in the

y-axis) to the one shown in the figure. The screw axis of the motion passes through two of the snap-fits' center locations and is resisted weakly by the third snap-fit.

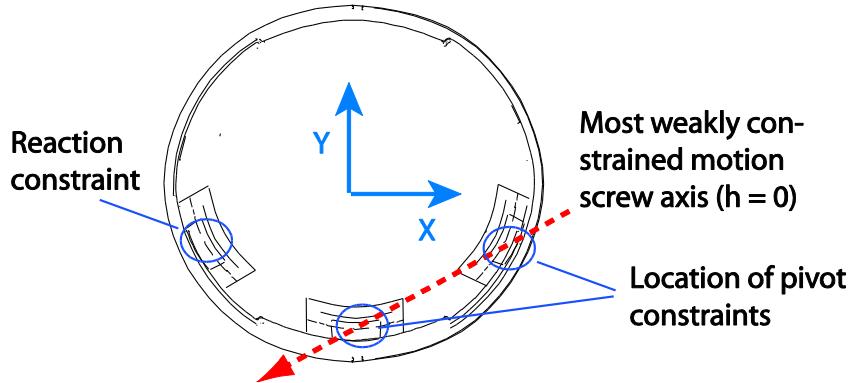


Figure 11.17 Worst constraint configuration and its associated WTR motion

11.5 Freeform/non-planar geometry - printer housing assembly

The following subsections explore the various search spaces available to the designer. In all these case studies, the manufacturability and cost to manufacture are ignored for design study purposes. Also, for visualization and design study purposes, the optimization for each design space in the following sections is limited to a two-dimensional search. Due to this limitation, each case study can only be done for a limited number of design variables. These localized design spaces are individually searched in a sequence of case studies, and at the end, the design spaces where possible improvements are identified are merged and searched simultaneously. An improved design recommendation for the printer housing based on the results is synthesized.

For all case studies under this section, the objective is to maximize the WTR rating. When there is no change in the WTR rating, maximizing the MTR rating becomes the secondary objective.

11.5.1 Design optimization – threaded fastener orientation search

The design space for the threaded fastener orientation is searched by allowing the orientation of all four fasteners to move in two angles (rotation around the x-axis and y-axis). The search space is defined as the rotation axes and the angular move limits. The optimization variables are shown in Figure 11.18 and listed in Table 11.13.

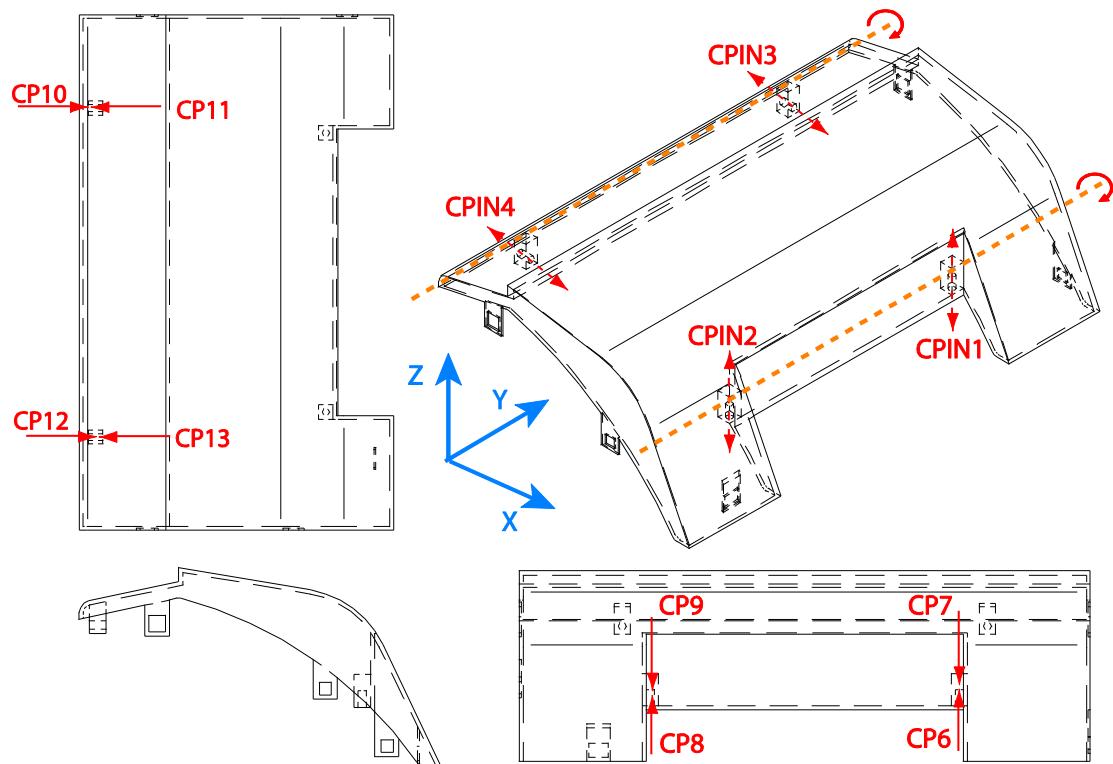


Figure 11.18 Optimization variables and search space for fastener orientation

There are two groups of variables. The first group of variables (X_1) involves two fasteners near the bottom of the housing. The constraints that belong to this group are

CPIN1, CPIN2, CP6, CP7, CP8, and CP9. The second group (X2) consists of CPIN3, CPIN4, CP10, CP11, CP12, and CP13.

	Variables	Orientation rotation axis	Rotation angle limits
Variable group X1	CPIN1, CPIN2, CP6, CP7, CP8, CP9	[0 1 0]	$-90^\circ \leq \theta_1 \leq 90^\circ$
Variable group X2	CPIN3, CPIN4, CP10, CP11, CP12, CP13	[0 1 0]	$-90^\circ \leq \theta_2 \leq 90^\circ$

Table 11.13 Optimization variables and search space for fastener orientation

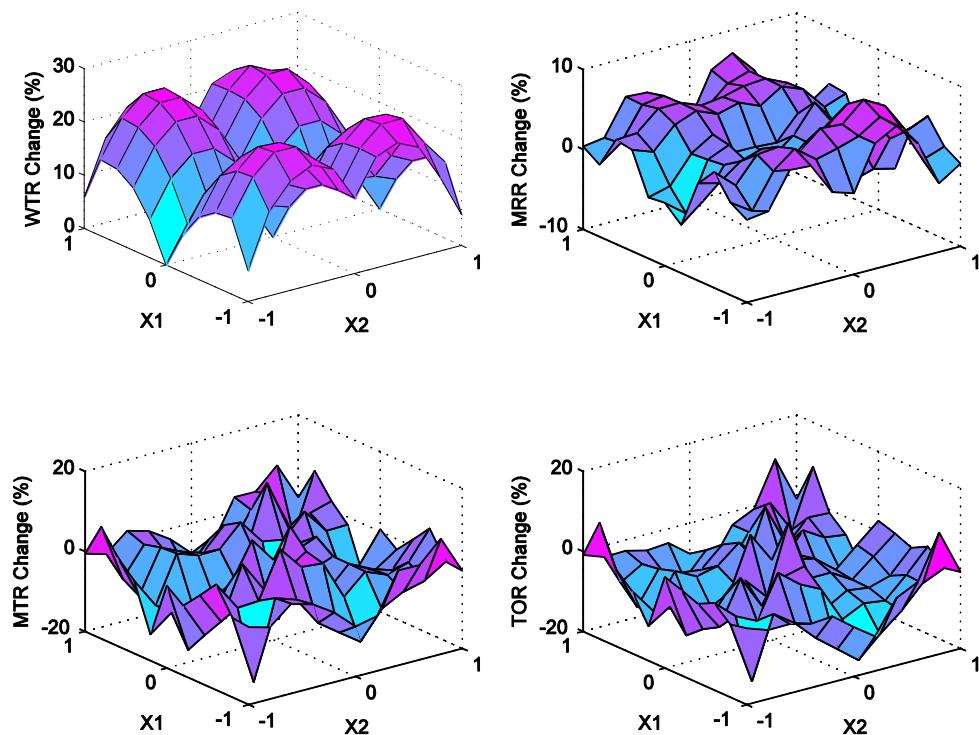


Figure 11.19 Response surface plot for fastener orientation optimization

The response surface plot is shown in Figure 11.19. In Section 11.1.2, it was explained that orientation searches yield more consistent results in the WTR rating; therefore, in this case, only the WTR plot is being evaluated. The WTR plot shows that

the WTR rating can be increased up to 24.6%. The WTR rating reaches a maximum value for the orientation combinations shown in Table 11.14 (all rotation is about the positive y-axis).

	X1 Rotation about positive y-axis (degrees)	X2 Rotation about positive y-axis (degrees)	WTR change (%)
Optimum design 1	72	-36 to -53	24.6%
Optimum design 2	72	36 to 53	24.6%
Optimum design 3	-72	-36 to -53	24.6%
Optimum design 4	-72	36 to 53	24.6%

Table 11.14 Optimum design for fastener orientation search space

This increase is due to the fact that the pivot constraint set for the WTR motion contains at least one of the modified constraints. As this constraint variable is modified, so is the motion reciprocal to the pivot constraints. The resulting motion is resisted more effectively and hence the WTR rating is improved.

As a side note, the MRR plot shows relatively random changes because of the of the linear dependency bifurcation shift. This is the one of the reasons MRR rating metric is irrelevant in orientation search spaces.

11.5.2 Design optimization – threaded fastener location

The design space for the threaded fastener location is searched by allowing the location of all four fasteners to move along two lines. The threaded fasteners at the locations of CPIN1 and CPIN3 are linked and move together. The same case applies for fasteners at the locations of CPIN2 and CPIN4. This is done to reduce the dimension of

the optimization search. The optimization variables and search space are specified in Table 11.15 and Figure 11.20.

	Variables	Line search center point	Line search direction	Move limits
Variable group X1	CPIN1, CPIN3, CP6, CP7, CP10, CP11	[8.625 9 1.744]	[0 1 0]	$-4.875 \leq x_1 \leq 4.875$
Variable group X2	CPIN2, CPIN4, CP8, CP9, CP12, CP13	[8.625 9 1.744]	[0 1 0]	$-4.875 \leq x_2 \leq 4.875$

Table 11.15 Optimization variables and search space for fastener location

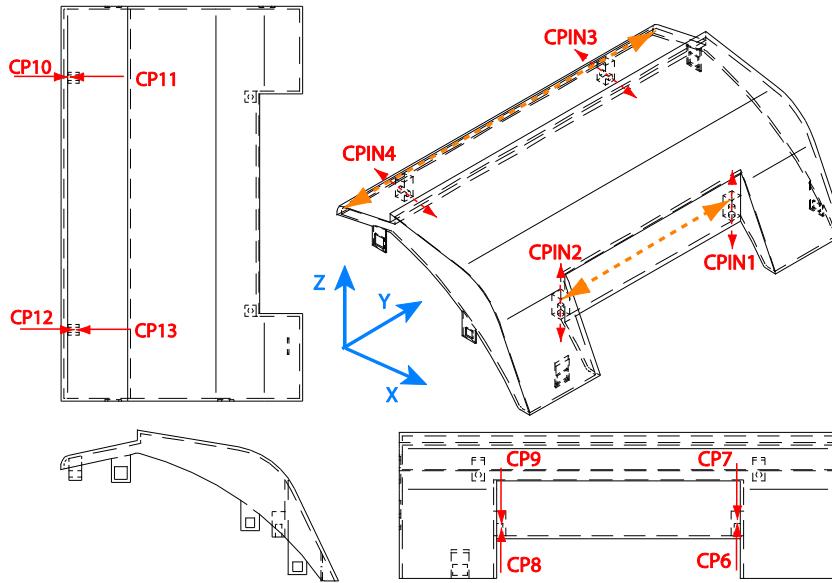


Figure 11.20 Optimization variables and search space for fastener location

The WTR plot in Figure 11.21 shows that the rating cannot be increased from the baseline model. The plot shows that the WTR rating is relatively constant for the cases where the two groups of fasteners are located symmetrically, namely they are the same

distance from the part symmetry line. The regions of the WTR plot where the rating decreases refer to locations where symmetry does not occur. The MTR plot also shows no improvement possible in this design space, but it is best to keep the fasteners at their original location. The discontinuity in the MTR plot is due to the fact that when the fastener locations coincide with one another, a certain motion set is eliminated due to linear dependency. The ‘diagonal’ in the MTR plot where the discontinuity occurs refers to the case where the four fasteners essentially become two fasteners because they overlap or almost overlap one another. The MRR plot shows that the redundancy is reduced at this overlapping location.

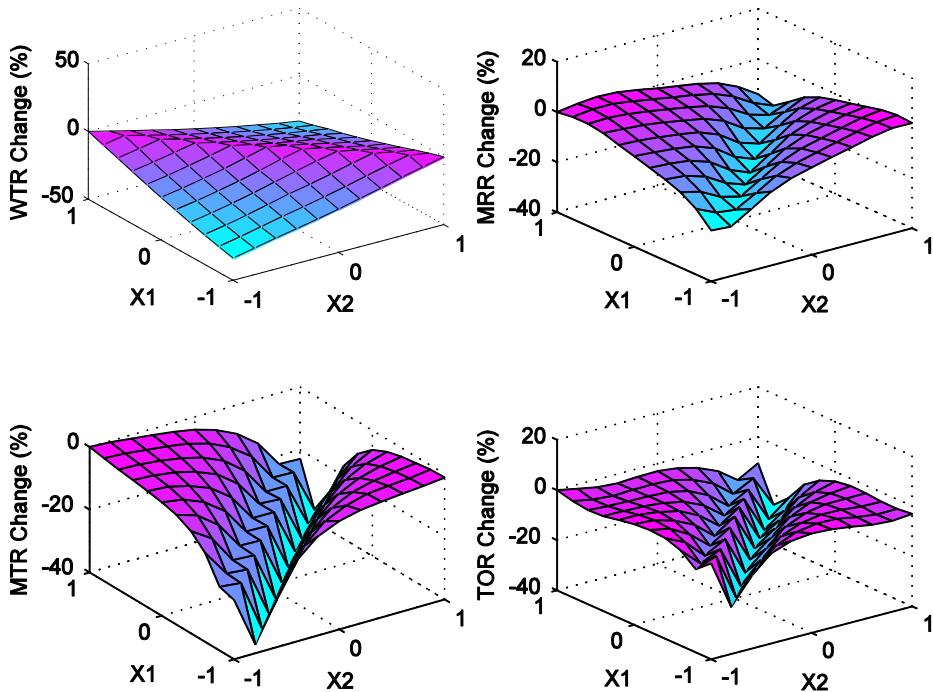


Figure 11.21 Response surface plot for fastener location optimization

Based on these results, there are two options. One is to keep the baseline design. The alternative is to reduce the number of fasteners from four to two and locate them along the line of symmetry. Table 11.16 shows that this reduction of fasteners yields a significant reduction in the redundancy ratio (MRR), which is desirable. This is done at the sacrifice of the WTR and MTR ratings. Overall, the alternate design is a better trade-off, shown by a higher TOR value. However, this is not an option when a decrease in WTR or MTR is not acceptable.

	Baseline design with 4 fasteners	Alternate design with 2 fasteners	% Difference
WTR	2.437	1.378	-43.5%
MRR	4.555	3.020	-33.7%
MTR	17.628	11.927	-32.3%
TOR	3.870	3.950	2.1%

Table 11.16 Comparison between assembly design with 4 vs. 2 fasteners

11.5.3 Design optimization – snap-fit orientation

The design space for the snap-fit orientation is searched by allowing the orientations of four snap-fits (CP1, CP3, CP4, and CP5) to rotate around the y-axis. CP1 and CP5 are linked and rotated together. The same case applies for CP3 and CP5. This is done to reduce the dimension of the optimization search. The optimization variables and search space are specified in Table 11.17 and Figure 11.22.

	Variables	Orientation rotation axis	Rotation angle limits
Variable group X1	CP1, CP5	[0 1 0]	$-60^\circ \leq \theta_1 \leq 60^\circ$
Variable group X2	CP3, CP4	[0 1 0]	$-60^\circ \leq \theta_1 \leq 60^\circ$

Table 11.17 Optimization variables and search space for snap-fit orientation

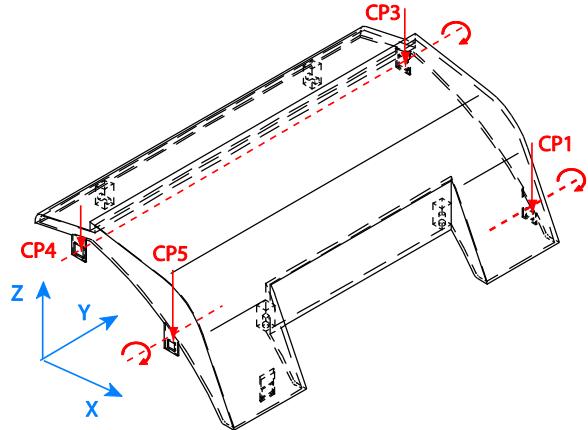


Figure 11.22 Optimization variables and search space for snap-fit orientation

The WTR plot (Figure 11.23) shows that an increase of 30.7% is achievable by re-orienting the snap-fits from the baseline model. There are two local maxima that can be identified (Table 11.18). In order to improve the rating, the two constraint variable orientations must be rotated in opposite directions. This is accomplished by rotating snap-fits CP1 and CP5 clockwise and snap-fits CP3 and CP4 counterclockwise about the y-axis. The design can also be improved by rotating snap-fits CP1 and CP5 counterclockwise and snap-fits CP3 and CP4 clockwise about the y-axis.

Although it was mentioned that manufacturing factors are not considered in this study, if this design were to be pursued, improved design #1 might be preferable because its normal direction is close to being perpendicular with the line constraints. This might reduce the mold complexity.

	X1 Rotation (degrees)	X2 Rotation (degrees)	WTR change (%)
Optimum design 1	60	-24 to -36	20.2%
Optimum design 2	-12 to -60	60	30.5%

Table 11.18 Optimum design for snap-fit orientation search space

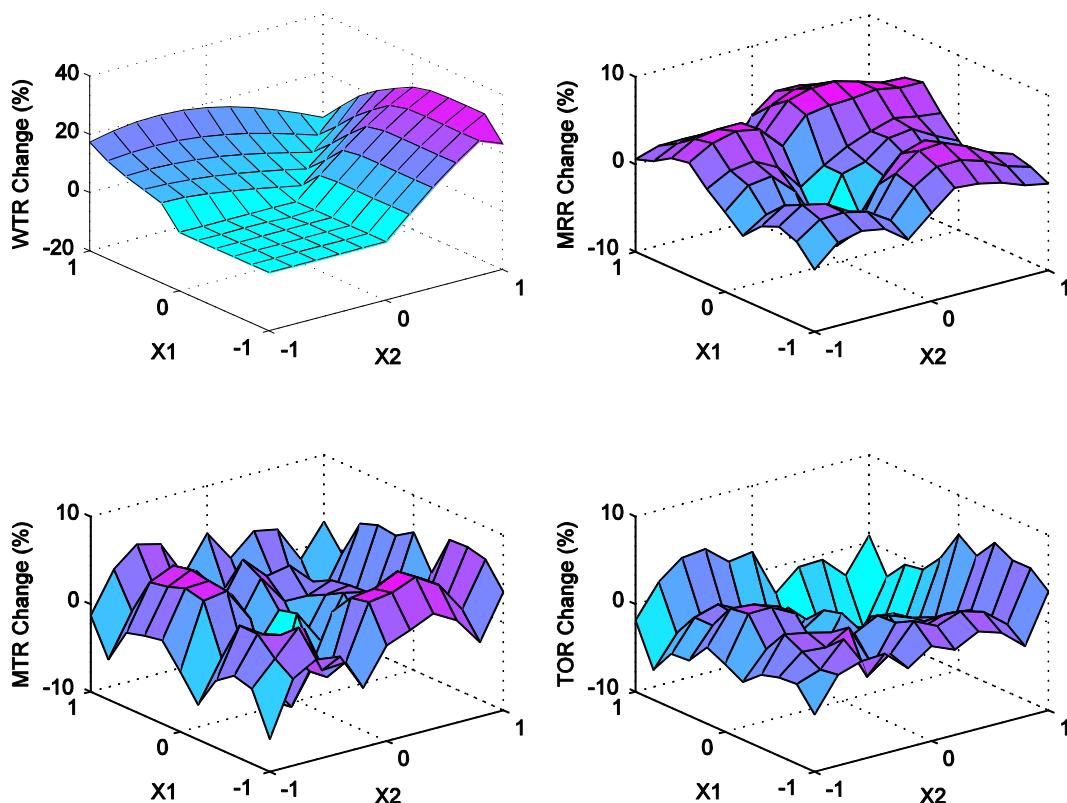


Figure 11.23 Response surface plot for snap-fit orientation optimization

11.5.4 Design optimization – snap-fit location

The design space for the snap-fit location is searched by allowing the locations of four snap-fits (CP1, CP3, CP4, and CP5) to move along the line constraints (CLIN1,

CLIN3, CLIN4, and CLIN2). The optimization variables and search space are specified in Table 11.19 and Figure 11.20.

	Variables	Line search center point	Line search direction	Move limits
Variable group X1	CP1, CP5	[7.125 18 3.200]	[-3.093 0 3.441]	$-4.5 \leq x_1 \leq 4.5$
Variable group X2	CP3, CP4	[1.551 18 4.754]	[2.000 0 0.431]	$-1.5 \leq x_2 \leq 1.5$

Table 11.19 Optimization variables and search space for snap-fit location

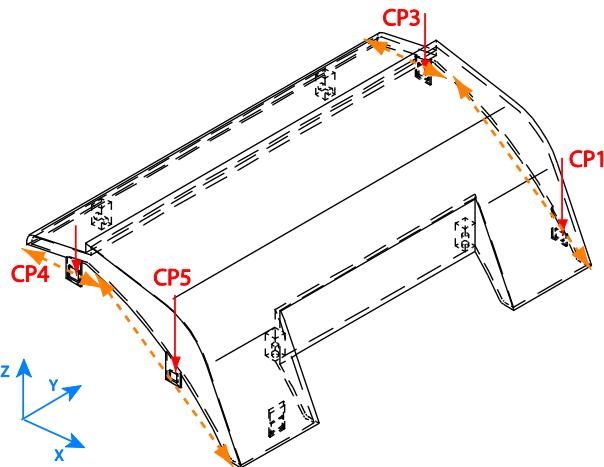


Figure 11.24 Optimization variables and search space for snap-fit location

The response surface plot (Figure 11.21) shows that the WTR rating does not change for any value of the optimization variables. A closer look at the rating matrix shows that the constraints being optimized are not active in resisting the most weakly constrained motion. MTR increases (4.5%) as the constraints CP1 and CP3 are moved higher along the line. The maximum for MTR occurs where CP1 is located between the

coordinates [5.320 18 5.208] and [5.922 18 4.539] along the line search. There is no effect on the MTR rating from moving CP4 and CP5.

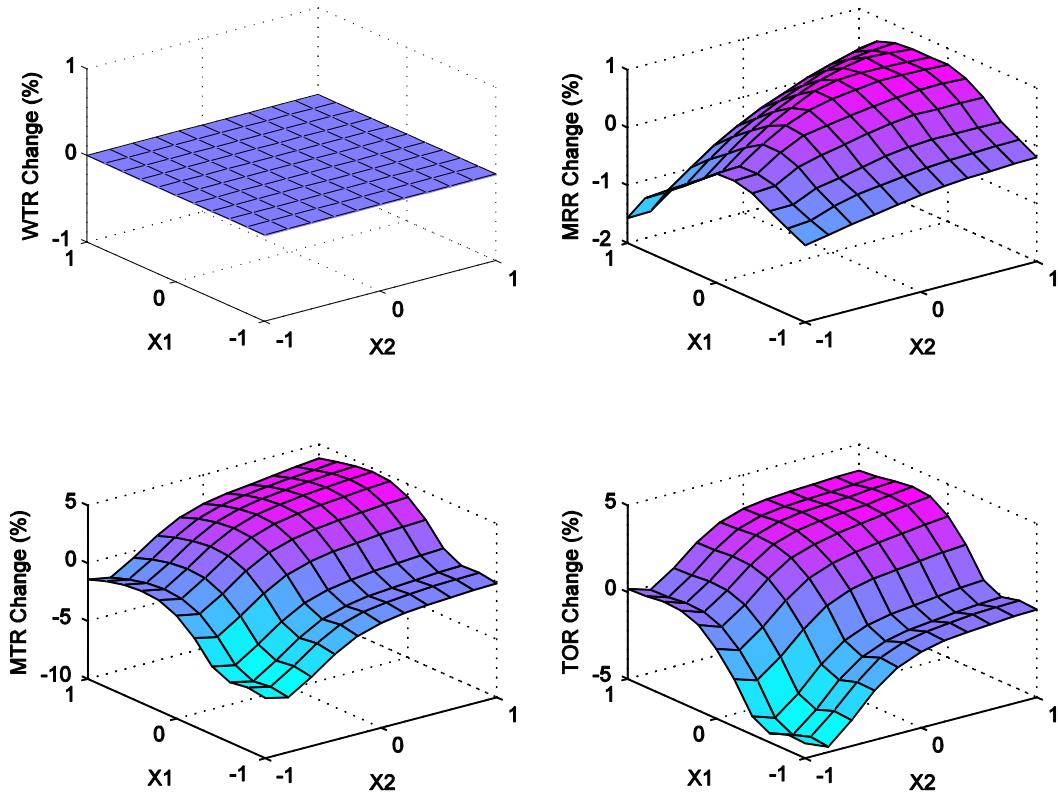


Figure 11.25 Response surface plot for snap-fit location optimization

11.5.5 Design optimization – parting line

The design space for the parting line is searched by allowing the orientations of line constraints CLIN1 and CLIN2 to be rotated about the y-axis. Rotating the orientation of the parting line, however, requires that CLIN3, CLIN4, CLIN5, and the two fasteners follow the parting line change by moving along the z-axis. Ideally, the two search spaces are linked, but this is not possible in the MATLAB script algorithm, so they are treated as

two independent design variables. The optimization variables and search space are specified in Table 11.20 and Figure 11.26.

	Variables	Orientation rotation axis		Rotation angle limits
Variable group X1	CLIN1, CLIN2	[0 1 0]		$-60^\circ \leq \theta_1 \leq 60^\circ$
		Line search center point	Line search direction	Move limits
Variable group X2	CLIN5, CLIN3, CLIN4, CPIN3, CPIN4, CP3, CP4, CP10, CP11, CP12, CP13,	[0 9 3]	[0 0 1]	$-1 \leq x_1 \leq 1$

Table 11.20 Optimization variables and search space for parting line

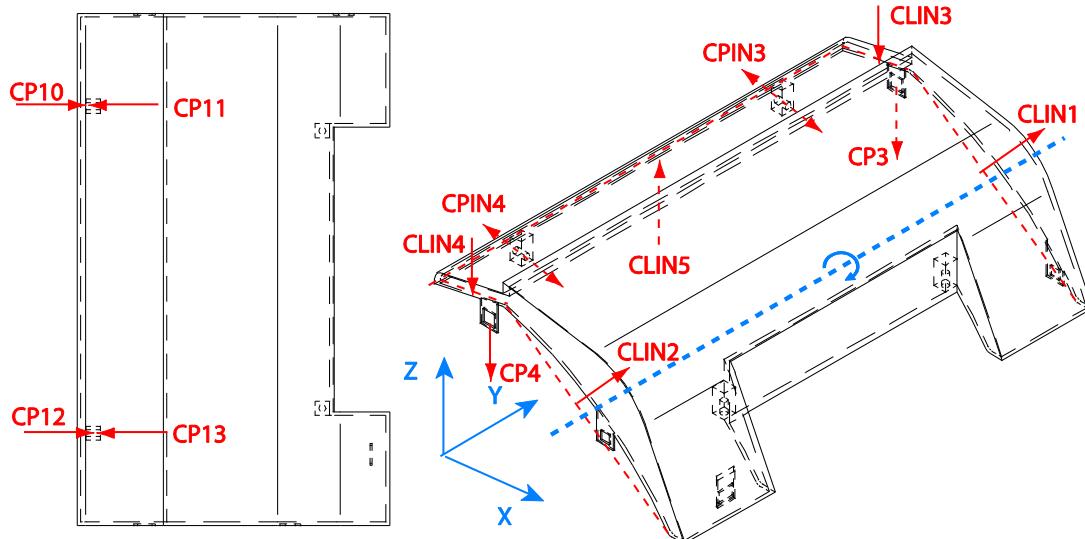


Figure 11.26 Optimization variables and search space for parting line

The response surface plot (Figure 11.27) shows that the WTR rating is only affected by the orientation of CLIN1 and CLIN2. There is no possible improvement from the baseline model by changing the variables in this case study, but the effect of rotating

the parting line orientation can be observed in the response surface plots. As CLIN1 and CLIN2 are rotated closer to the horizontal plane, the WTR rating decreases significantly. Rotating CLIN1 and CLIN2 in the opposite direction yields no gain in the WTR rating.

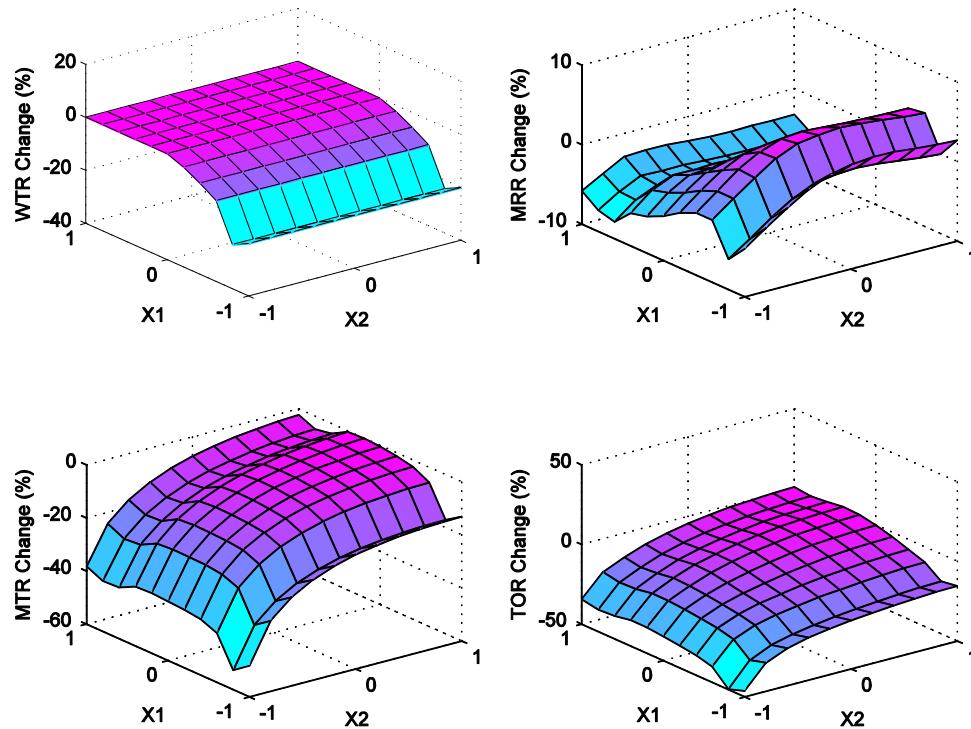


Figure 11.27 Response surface plot for parting line optimization

While moving the position of the second group of variables (CLIN3, CLIN4, CLIN5, and two fasteners on the top of the housing) in the z-direction does not affect the WTR rating, the effect on the MTR rating is significant. The MTR plot shows that not only does rotating the parting line toward the XY plane decrease the rating, but lowering the position of the second group of variables in the z-direction amplifies the rating decrease. There is a slight interaction between the two design search spaces.

These results show that the baseline design of the parting line already yields maximum WTR and MTR ratings. If the printer housing were designed with the

traditional planar parting line ($X1 = -1$), the rating would be much lower than the baseline results (Table 11.21).

	Baseline design with non-planar parting line	Alternate design with horizontal parting line	% Difference
WTR	2.437	1.869	-23.3%
MRR	4.555	4.577	0.5%
MTR	17.628	13.295	-24.6%
TOR	3.870	2.905	-24.9%

Table 11.21 Comparison of non-planar parting line vs. planar parting line design

11.5.6 Design optimization – line constraint length

There are two case studies done to search the line constraint length design space.

The first study involves the lines CLIN5 and CLIN6 as the optimization variables. The second study involves the lines CLIN1, CLIN2, CLIN3, and CLIN4.

For the first line size search study, the optimization variables and search space are specified in Table 11.22 and Figure 11.28. The second line size search study optimization variables and search space are specified in Table 11.23 and Figure 11.29. In the second study, to maintain geometric compatibility, the length of CLIN1 must be inversely proportional to the length of CLIN3. In the same way, the length of CLIN2 must be inversely proportional to the length of CLIN4. However, this relationship cannot be programmed in the MATLAB script, and therefore they are treated as separate optimization variable groups.

	Variables	Line length limits
Variable group X1	CLIN5	$8.5 \leq L_1 \leq 14.5$
Variable group X2	CLIN6	$15 \leq L_2 \leq 21$

Table 11.22 Optimization variables and search space for line size (CLIN5, CLIN6)

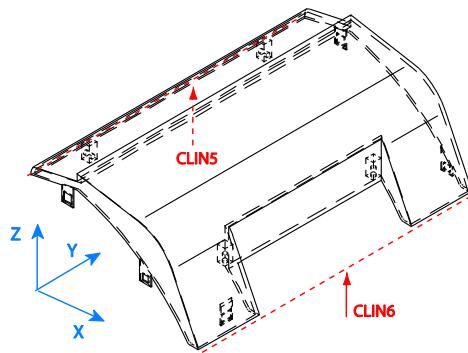


Figure 11.28 Optimization variables and search space for line size (CLIN5, CLIN6)

	Variables	Line length limits
Variable group X1	CLIN1, CLIN2	$7 \leq L_1 \leq 11$
Variable group X2	CLIN3, CLIN4	$1 \leq L_2 \leq 5$

Table 11.23 Optimization variables and search space for line size (CLIN1-CLIN4)

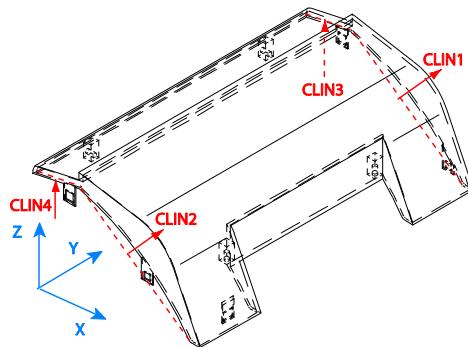


Figure 11.29 Optimization variables and search space for line size (CLIN1-CLIN4)

The response surface plots in Figure 11.30 and Figure 11.31 show that both the WTR and MTR ratings would be maximal when the lengths of the line constraints are the

longest. The reason is that a longer line has a greater moment arm length to resist moment loads. Since the line constraints in the baseline model are at maximum length, there is no further design improvement possible. There is no effect on the WTR rating because the optimization variables in this particular case are not active in resisting the WTR motion.

The design implication of this study is that a line constraint size is always optimal at its longest allowable length. The same case applies for the area of a planar constraint.

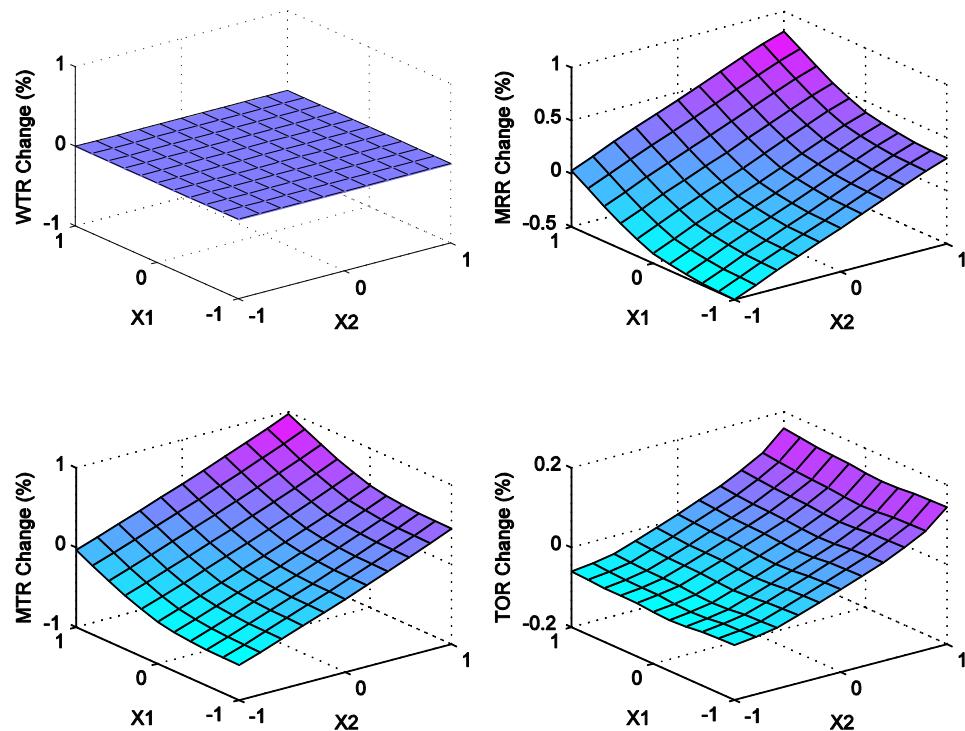


Figure 11.30 Response surface plot for line size optimization (CLIN5, CLIN6)

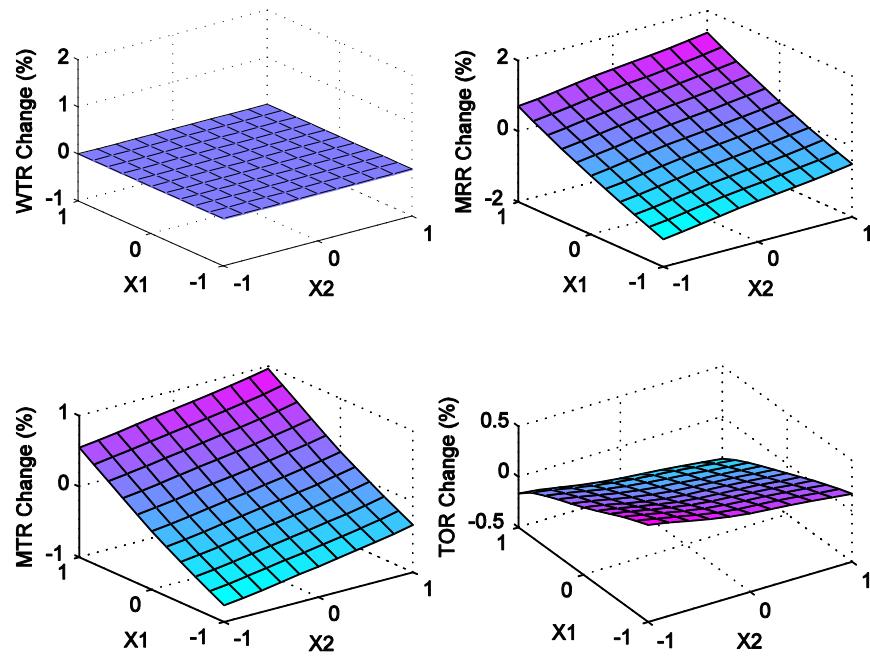


Figure 11.31 Response surface plot for line size optimization (CLIN1-CLIN4)

11.5.7 Synthesis of improved design

The design study conclusions on the printer housing are summarized in Table 11.24. Based on the design space exploration, a few design recommendations can be suggested. Table 11.25 summarizes the possible design improvements for the printer housing based on the results of the optimization studies done so far.

Optimization Study	Optimization variables	Search type	Study conclusions
Fastener orientation search	CPIN1, CPIN2, CP6, CP7, CP8, CP9	Orientation	24.6% improvement in WTR is possible
	CPIN3, CPIN4, CP10, CP11, CP12, CP13	Orientation	
Fastener location search	CPIN1, CPIN3, CP6, CP7, CP10, CP11	Line search	No improvement possible, keep baseline design. 2 fastener design increases TOR by 12.8%
	CPIN2, CPIN4, CP8, CP9, CP12, CP13	Line search	
Snap-fit orientation search	CP1, CP5	Orientation	30.7% improvement in WTR is possible
	CP3, CP4	Orientation	
Snap-fit location search	CP1, CP5	Line search	4.5% improvement in MTR is possible
	CP3, CP4	Line search	
Parting line search	CLIN1, CLIN2	Line direction orientation	Non-rectilinear parting line is preferred, which is the baseline design. Traditional horizontal parting line decreases WTR by 23.3% and MTR by 24.6%
	CLIN5, CLIN3, CLIN4, CPIN3, CPIN4, CP3, CP4, CP10, CP11, CP12, CP13,	Line search	
Line length search	CLIN5	Line size search	No improvement possible, keep line the longest possible without increasing overall part size.
	CLIN6	Line size search	
	CLIN1, CLIN2	Line size search	
	CLIN3, CLIN4	Line size search	

Table 11.24 Summary of design study conclusions

	Case Study	Design Change
Design revision A	Fastener orientation search	Rotate (CPIN1, CPIN2, CP6, CP7, CP8, CP9) 72 degrees and rotate (CPIN3, CPIN4, CP10, CP11, CP12, CP13) -36 degrees about the y-axis
Design revision B	Snap-fit orientation search	Rotate CP1, CP5 -12 to -60 degrees and rotate CP3, CP4 60 degrees about the y-axis
Design revision C	Snap-fit location search	Move CP1 and CP5 to X=5.320, Z=5.208

Table 11.25 Summary of design recommendation for each design revision

The design improvements for each design space can be implemented one at a time. In this section, an attempt to merge these design changes and observe whether the assembly rating improvements accumulate is done. When design changes involving multiple variables are done simultaneously, there is a possibility of interaction between the factors being changed. In order to identify a simultaneous search optimum involving all factors that allow improvements in assembly rating, a higher dimensional search must be done. The current MATLAB script allows up to a five-dimensional search. Due to limitations in computational capability, the number of increments in this search is limited to 6. This slightly reduces the accuracy of the results. The variables identified for improving the assembly configuration are

- Orientation of two threaded fasteners (2D search)
- Orientation of four snap-fits (CP1, CP3, CP4, and CP5) (2D search)
- Location of two snap-fits (CP1 and CP3) (1D search)

A search of the specified design space for all five optimization variables was conducted. The objective is to maximize the WTR rating. The optimum point within the optimization constraints and the corresponding design change is shown in Table 11.26.

	Normalized Units	Optimization variables	Search type	Design Change
X1	-1.0 or 1.0	CPIN1, CPIN2, CP6, CP7, CP8, CP9	Orientation	Rotate CPIN1, CPIN2, CP6, CP7, CP8, and CP9 90 or -90 degrees about the y-axis
X2	-0.6	CPIN3, CPIN4, CP10, CP11, CP12, CP13	Orientation	Rotate CPIN3, CPIN4, CP10, CP11, CP12, and CP13 -54 degrees about the y-axis
X3	0.6	CP1, CP5	Orientation	Rotate CP1 and CP5 54 degrees about the y-axis
X4	-0.6	CP3, CP4	Orientation	Rotate CP3 and CP4 -54 degrees about the y-axis
X5	-0.6	CP1, CP5	Location	Move CP1 to [7.526 18 2.754] and CP5 to [7.526 0 2.754]

Table 11.26 Simultaneous search optimum in the design search space

Based on this higher dimension optimum search, the design recommendations for only some variables are similar to those in Table 11.25. The recommended design change for variable group X1 and X2 is similar to the previous results when the design spaces were searched individually. This difference can be attributed to interactions among the design variables and/or to the reduction of the search resolution. The design recommendations for X3 and X4 are different. They are oriented in the opposite direction compared to the previous design recommendation. However, based on Figure 11.23, this new design recommendation still falls near to the local optimum in the response surface plot. The design recommendation for X5 is also different. For variable X5, the design

recommendation is to move CP1 and CP5 in the opposite direction. Recall that in the individual design space study (Section 11.5.4) the design recommendation is based on the MTR rating improvement, not the WTR rating. This might be the reason for the difference in design change recommendation.

For this particular case study, orientation modification is more effective in improving WTR and location modification is more effective in improving MTR. The difference between the individual optimization searches and the higher dimension optimization search shows that there are interactions among the design variables. Therefore, it is important to search the design space simultaneously when multiple design variables are involved.

Table 11.27 and Figure 11.32 summarize the design improvements across the possible design revisions. It is found that the optimum (design revision D) has the best WTR rating, but is not best in all ratings. Furthermore, the rating gains obtained by individual design changes do not always accumulate when all of them are implemented. In a real design context, more design changes lead to higher cost, except when done in the conceptual design stage. In this context, it is important to observe that the assembly rating improvement does not necessarily accumulate with each design change implementation.

	WTR		MRR		MTR		TOR	
	Rating	% Diff.						
Baseline design	2.437		4.555		17.628	0.0%	3.870	
Design revision A	3.036	24.6%	4.806	5.5%	17.203	-2.4%	3.579	-7.5%
Design revision B	3.180	30.5%	4.562	0.1%	18.827	6.8%	4.075	5.3%
Design revision C	2.437	0.0%	4.564	0.2%	18.421	4.5%	3.994	3.2%
Design revision D	3.507	43.9%	4.766	4.6%	18.323	3.9%	3.844	-0.7%

Table 11.27 Summary of design alternative ratings

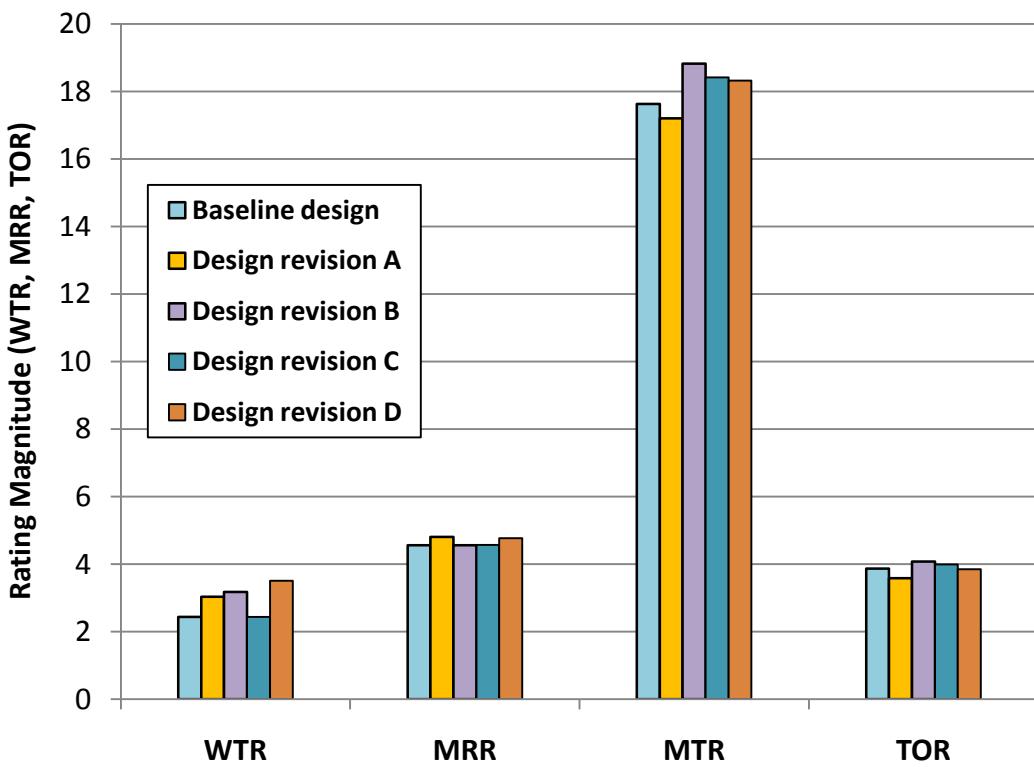


Figure 11.32 Summary of design alternative ratings

11.5.8 Design optimization - known loading direction

The purpose of this study is to optimize the constraint configuration for a specified loading condition. In this case, the specified motion is a pure rotation around the screw axis

$$\omega = [0 \ 1 \ 0], \rho = [2 \ 0 \ 4], h = 0$$

The optimization variables chosen for this study are the same as in the snap-fit location optimization case study. The optimization variable and search space are identical to Table 11.19 and Figure 11.24.

Note that for this response surface plot (Figure 11.33), the vertical axis is in absolute value, not a percent change. Since there is only one motion being evaluated, only the WTR plot is of interest. The response surface plot shows that the location of variable group X2 (CP3 and CP4) does not affect the WTR rating. The location of variable group X1 (CP1 and CP5) is better at X1=-1. This is contradictory to the results in Section 11.5.4. The recommended modification in the study where the design space is only comprised of the line search is to move the constraints CP1 and CP5 closer to CLIN3 and CLIN4. In this study, the recommended modification is to move CP1 and CP5 away from CLIN3 and CLIN4. Therefore, when a constraint configuration is optimized for a particular loading condition, the overall rating of the assembly might actually decrease from the baseline design.

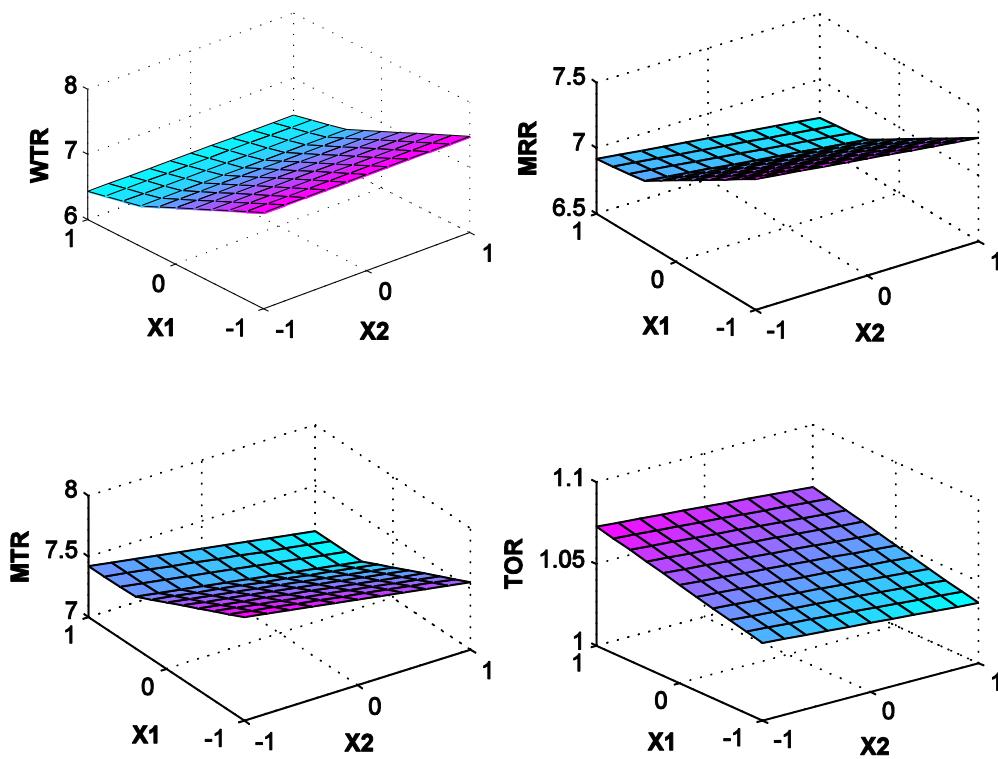


Figure 11.33 Response surface plot for snap-fit location optimization (specified loading condition)

11.5.9 Design optimization – constraint reduction

The purpose of this study is to apply the constraint reduction scheme to the printer housing case study. The goal in this study is to maximize the trade-off ratio (TOR). The constraint reduction scheme shows that the trade-off ratio can be increased significantly as more constraints are removed (Figure 11.34). The selected constraints to remove that yield the most TOR increase are shown in Table 11.28. Constraint index 15, 16, 17 occur frequently as the most appropriate constraints to remove in order to decrease redundancy with minimal impact on MTR. These constraint indexes refer to CPIN2, CPIN3, and

CPIN4. Unfortunately, the removal of the fasteners that are represented by these pin constraints also eliminates the point constraints associated with them.

No of Constraints removed	Maximum Possible TOR increase (%)	WTR	MRR	MTR	TOR	Removed Constraints Index						
0	0.0%	2.437	4.555	17.628	3.870							
1	3.9%	2.437	4.354	17.500	4.019	1						
2	6.1%	2.437	4.237	17.399	4.106	6	7					
3	17.3%	1.290	4.044	18.360	4.540	10	11	15				
4	21.5%	1.290	3.811	17.926	4.704	1	10	11	15			
5	27.4%	1.000	2.909	14.344	4.931	8	9	14	16	17		
6	38.7%	1.000	2.639	14.167	5.368	1	6	7	15	16	17	
7	45.6%	1.000	2.329	13.125	5.635	1	6	7	15	16	17	18

Table 11.28 Effect of constraint reduction to assembly rating for printer housing

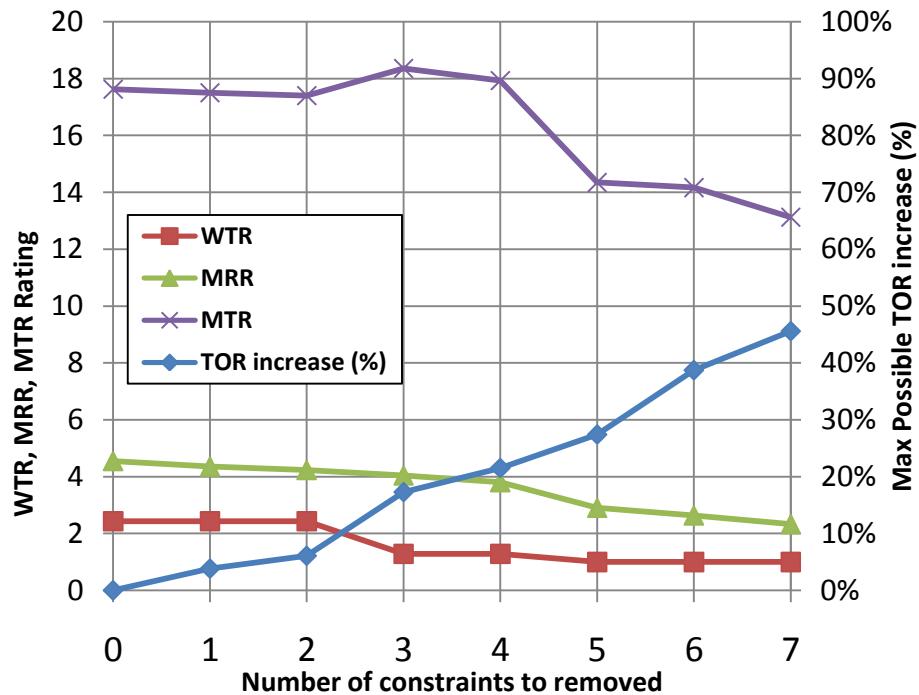


Figure 11.34 Effect of constraint reduction to assembly rating for printer housing

While removing more than 2 constraints yields even greater gain in the TOR, the WTR rating decreases significantly. Removing more than 2 constraints reduces the WTR rating from 2.437 to 1.290. This is about a 47% decrease. Note that the TOR is the ratio between MTR and MRR without regard to the WTR rating. Therefore, to minimize the impact on the WTR rating, the constraint removal should be limited to 2 constraints.

The result shows that CP6 and CP7 are the candidate constraints to be removed. The fastener that represents these pin constraints is also associated with CPIN1. In order to remove CP6 and CP7, CPIN 1 (one threaded fastener) must also be removed. An analysis of the design with the whole threaded fastener removed yields a decrease of 59% in the WTR rating. Therefore, the two-way constraints CP6 and CP7 must be removed without removing the pin constraint. This can be done by modifying or “downgrading” the assembly feature from a threaded fastener to a locating pin. This shows the advantages of integral attachment features compared to traditional bolted joints. In addition, integral attachment features provides more flexibility by allowing the removal of fewer DOF at a time during constraint reduction scheme. Table 11.29 shows the results for removing CP6 and CP7, labeled as alternate design 1.

Another alternative is to apply the same feature downgrading technique to the other threaded fasteners. Alternate design 2 is the case where the threaded fastener associated with CPIN2 is transformed into pin constraints and alternate design 3 is the case where all threaded fastener is transformed to pin constraints. Table 11.29 shows that the recommended design is alternate design 2 because there is very minimal decrease in WTR but much reduction in redundancy. This can also be observed in the TOR increase.

Alternate design 3 offers more reduction of redundancy, but at this point, the WTR rating decreases 35%.

	Baseline design	Alternate design 1 (CP6, CP7 removed)	% Diff.	Alternate design 2 (CP6-CP9 removed)	% Diff.	Alternate design 3 (CP6-CP13 removed)	% Diff.
WTR	2.437	2.4367	0.0%	2.4367	0.0%	1.578	-35.2%
MRR	4.555	4.2372	-7.0%	3.989	-12.4%	3.5283	-22.5%
MTR	17.628	17.3993	-1.3%	17.3467	-1.6%	14.8045	-16.0%
TOR	3.870	4.1063	6.1%	4.3487	12.4%	4.1959	8.4%

Table 11.29 Comparison of rating between baseline and alternate designs for constraint removal (printer housing)

11.5.10 Sensitivity analysis

The sensitivity of each constraint's location and orientation is evaluated. In this case a position perturbation of 0.5 inch (about 2-3% of overall part length) and an orientation perturbation of 10° are applied. Table 11.30 summarizes the change in rating due to perturbation in each rating. It can be observed that the WTR rating is most sensitive when CPIN1 and CPIN2 are deactivated, CLIN1 and CLIN2 position perturbed, and CLIN3 and CLIN4 orientations perturbed. The toggle perturbation has a much more significant effect compared to a position or orientation perturbation because in toggle perturbation the constraint is removed instead of being perturbed. Constraints that show zero percent change when perturbed are either not actively resisting the WTR motion or their perturbations do not change the resistance quality of the respective constraints.

As a result of the sensitivity study, the constraints that are more critical to deliver the intended performance of the assembly are identified. In the tooling design, certain

decisions such as the location of gates and cooling lines in the mold influence the areas that are subject to more shrinkage than others and directions in which warpage occurs. The identification of critical constraints might influence these tooling design decisions or vice versa.

	Toggle Perturbation WTR Change (%)	Position Perturbation worst WTR Change (%)	Orientation perturbation worst WTR Change (%)
CP1	0.0%	0.0%	0.0%
CP2	0.0%	0.0%	-0.2%
CP3	0.0%	0.0%	-0.2%
CP4	0.0%	0.0%	-0.2%
CP5	0.0%	0.0%	0.0%
CP6	0.0%	0.0%	0.0%
CP7	0.0%	0.0%	0.0%
CP8	0.0%	0.0%	0.0%
CP9	0.0%	0.0%	0.0%
CP10	0.0%	-0.1%	0.0%
CP11	-28.9%	-1.2%	-0.8%
CP12	0.0%	-0.1%	0.0%
CP13	-28.8%	-1.2%	-0.8%
CPIN1	-31.6%	-1.6%	-0.9%
CPIN2	-31.6%	-1.6%	-0.9%
CPIN3	-0.1%	0.0%	0.0%
CPIN4	0.0%	0.0%	0.0%
CLIN1	-21.8%	-1.7%	-7.4%
CLIN2	-21.8%	-1.7%	-7.3%
CLIN3	-14.8%	-0.4%	-13.2%
CLIN4	-14.8%	-0.5%	-13.2%
CLIN5	0.0%	0.0%	0.0%
CLIN6	0.0%	0.0%	0.0%

Table 11.30 Sensitivity analysis result for printer housing

CHAPTER 12

CONCLUSIONS AND FUTURE WORK

12.1 Research contributions

This dissertation has made several contributions to understanding the design of mechanical part assembly. In Chapter 1, it was mentioned that the literature and domain-specific knowledge in the locations and orientations of assembly features are much less compared to the knowledge in how to design individual features. There is a need for a practical assembly constraint analysis and design tool that provides a quantitative metric of assembly quality and is able to optimize a design at the attachment strategy level.

This dissertation has developed a model and methodology for analyzing an assembly constraint configuration. This is implemented in an analysis tool that is able to:

- Accurately and efficiently model contact interfaces in assembly such as point, pin, line, and plane contact with an equivalent wrench/screw system. The higher order constraint (HOC) model in this dissertation eliminates the need to discretize HOC contact interfaces into many point constraints.

- Evaluate constraint effectiveness based on the reaction forces to resist motion.

This isolated reaction force (IRF) model is consistent for finite and infinite pitch motion. By modeling the evaluated motion input wrench as a force couple, the evaluation method consistently computes the resistance for rotation-dominant motion as well as translation-dominant motion. A scheme to separate the motion into two different cases depending on the pitch is able to resolve the pitch discontinuity problem. The resistance value calculation and the main rating metric (total resistance) are based on the notion of load amplification ratio, which can be interpreted as the magnitude of the reaction force generated for a given input load. The load amplification ratio produced by this model has direct design applications. Therefore, this evaluation method also has better physical significance and interpretation in the context of assembly.

- Provide multiple rating metrics that measure different aspects of assembly performance. Although the main purpose of assembly is to remove DOF and resist motion, there are other characteristics of assembly that need to be considered. Among these are redundancy, resistance of the most weakly constrained motion vs. overall motion, the tradeoff between resistance capacity and redundancy, and how active a constraint is in resisting motion.

This dissertation also has developed a methodology and algorithm for studying the design space of and optimizing the constraint configuration of an assembly. This is implemented in a design tool that is able to:

- Efficiently search through the design space as constraints are modified. Every constraint modification changes not only the resistance values of the modified constraints but also the allowed motion set for each pivot constraint combination. The algorithm separates the motion that remains identical in the modified constraint configuration while also identifying new motions that need to be evaluated. The results are displayed in a response surface plot when the search is 2-dimensional or less. The optimum design configuration, based on the rating metric of interest, is also identified.
- Evaluate different possibilities for constraint reduction and select the best candidate for reducing redundancy while achieving minimum decrease in assembly total resistance. In a way, the different possibilities for constraint reduction, namely the number of constraints to remove and the selection of the best candidate for removal is the design space that is explored in this scheme.
- Observe the change in assembly ratings as constraints are added. Designers then can use this tool to identify the point at which further constraint addition yields a diminishing return.
- Identify an optimal constraint configuration for a specified loading condition and design search space. This optimum might or might not be the same as the optimal constraint configuration for overall motion evaluation.
- Identify constraints that are more sensitive to perturbation. The results in this study are applicable in the manufacturing design and allocation of tolerances

in a part. When used properly, this information can improve the robustness of a design to manufacturing variability.

This dissertation also has developed case studies that demonstrate the validity and usefulness of the methodology implemented in an analysis and design tool. The case studies in Chapters 10 and 11 accomplish the following goals:

- Demonstrate the advantages of the higher order constraint (HOC) and isolated reaction force (IRF) model in assembly analysis. The case study in Section 10.4.2 shows that the HOC model provides the accuracy and efficiency in generating possible motions. The case study in Section 10.2.2 demonstrates scalability and accuracy in rotation- and translation-dominant motion.
- Verify basic theoretical kinematic principles for an exactly constrained object in the context of assembly. An example of the Thompson's chair is used to demonstrate these principles.
- Verify commonly known design principles in assembly such as leverage, symmetry, and optimal line-of-action.
- Understand the trade-off in assembly redundancy as constraints are added and reduced. This results in more domain-specific knowledge in assembly design. A summary of this knowledge is discussed in the next section.
- Demonstrate how the design tool can be used to explore various design spaces and to identify areas where potential design improvements are possible and not possible. A synthesis of design improvements that searches through

multiple design spaces and identifies the simultaneous search optimum is also done.

12.2 Development of assembly domain-specific knowledge

The case studies in Chapters 10 and 11 also serve the purpose of understanding assembly design and generating domain-specific knowledge in this area. This section summarizes what is learned about assembly in this dissertation through the case studies.

12.2.1 Assembly performance analysis

- The case studies confirm the theoretical kinematic principles for an exactly constrained geometry. When an assembly is constrained by seven point contacts, six of which do not belong to the same linear complex, it is exactly constrained. The equivalent constraint configuration can also be composed of an HOC as long as the total degree of the constraints sums to seven. Because any combination of pivot constraints in this case is always a linearly independent set, each motion is resisted by exactly one constraint. The forward vs. backward motion is not necessarily resisted with the same effectiveness. This is one of the reasons for separating the rating for forward and backward motion.
- In assembly design, there are cases when a part shape that extends beyond the enveloping constraint configuration decreases the resistance quality of assembly. This happens when external loads are applied to the assembly not

on the constraint points but onto the part surface. This increases the moment arm for the force couple input to the assembly. This is not necessarily quantified by the analysis tool in this dissertation, but it was proposed that if the input wrench were to be scaled according to part shape and not only the constraint configuration, this behavior of assembly would be properly captured.

- There is a tendency that the closer a part shape or constraint configuration is to a ‘cubic’ shape (equal dimension along all coordinate axes), the total resistance histogram is more uniform. In other words, the total resistance across all evaluated motions does not spread into a cluster of strongly resisted motions and a cluster of weakly resisted motions.
- For the same input force, especially in pure rotation motion, a larger part will have better resistance compared to a smaller part. However, because a larger part is subject to larger moment loads as well, the assembly rating remains the same. This scalability feature is made possible by scaling the input wrench according to the size of the part or constraint configuration.
- The most weakly constrained motion tends to be symmetrical with respect to the part when the constraint configuration is symmetrical. Both directions of the motion need to be taken into consideration in the design process, especially with the unidirectional nature of IAF-dominated assembly constraints.

- In general, HOC are more active than point constraints in resisting motion.
This can be observed in the active percentage rating that tends to be higher on average than point constraints. One needs to be careful if these constraints are to be removed to reduce redundancy. The possibility of losing total restraint or being close to losing total restraint is higher when an HOC is removed.
- Some assembly features, especially in plastic parts, are not meant to function as mechanical constraints and should not be modeled as such. For example, the lip edges in the printer housing case study are meant for aesthetic and fitment purposes. They do not have enough stiffness to act as mechanical constraints.

12.2.2 Assembly resistance quality and redundancy trade-off

- When assembly constraints are added, the resistance rating of an assembly monotonically increases, while redundancy also increases. The trade-off ratio (TOR) is a measure of the resistance quality of an assembly (MTR) gain at the cost of redundancy (MRR). When TOR decreases, additional constraints create more redundancy than resistance capability. Constraint addition is useful in a design context where more resistance capability is desired and some redundancy is acceptable. In this procedure, better candidate constraints to add can be identified by observing the improvement each additional constraint yields in the WTR or MTR rating. An observation in the WTR and MTR ratings shows that in some cases there may be a point of diminishing

returns. At this point, the rating increase is negligible, and additional constraints might lead to more cost and design complexity rather than assembly resistance quality.

- When assembly constraints are reduced, redundancy and resistance capability usually decrease simultaneously. Constraint reduction in most cases can increase TOR. As long as TOR increases in constraint reduction, the amount of redundancy that is reduced is more significant compared to the sacrifice in resistance capability. Some constraints when removed yield more TOR increase than others. They are the better candidates for removal in this kind of design goal. The number of constraints being removed should be limited to the point at which removing more constraints does not yield more increase in TOR.
- In a configuration that is close to an exactly constrained design, some constraints are very critical to achieving total restraint. Such assembly constraints must be identified, especially ones that would cause the loss of total restraint when removed. Even in highly redundant design, when certain multiple constraints are removed, the assembly can lose enough WTR rating that it becomes very weak.
- A useful procedure to determine the optimum number of constraints is to add all possible constraints to the constraint configuration and start removing constraints one at a time or multiples at a time. This procedure can lead the designer to a more nearly optimal selection of constraints with the best TOR.

Hypothetically, if one starts from an infinite number of constraints that spans all possible locations and orientations, one can reduce this to the most efficient constraint configuration possible for the geometry. In reality, this kind of unconstrained optimization is impossible because the number of locations and orientations possible is limited by the part geometry, inaccessible locations or orientations, and manufacturing cost.

- Because IAF in assemblies usually remove one or two DOF for every constraint feature, an assembly with integral attachment features is a better candidate for achieving less redundant design. Section 10.5.2 shows that replacing the snap-fits with threaded fasteners increases redundancy by 42%. An assembly with threaded fasteners is less likely to achieve less redundant design because each feature removes at least 3 DOF at a time. Removing a constraint in bolted joints requires the removal of all constraints associated with the threaded fastener. In this manner, IAF has advantages over bolted joints not only in yielding less redundant designs, but also in allowing more flexibility in removing constraints fewer DOF at a time in constraint reduction process.

12.2.3 Assembly design optimization

- Design principles such as leverage, symmetry, and optimal line-of-action are confirmed to be desirable in constraint design. The closer a constraint is to the edge of a part; the better it is in resisting rotational motion in particular. A

symmetrical configuration of constraints is also desirable because it is more robust in resisting motion in many different directions (see Section 11.4.2).

The symmetry principle also applies in axisymmetric parts. While the assembly rating of a symmetrical constraint configuration is higher, the constraints tend to have higher probability of being linearly dependent in the pivot constraint combination, and therefore, such a configuration tends to increase redundancy.

- The optimal line-of-action principle usually optimizes the resistance of a constraint configuration to a single motion. This principle should be used to optimize resistance to the most weakly constrained motion. It is not applicable in optimizing resistance to overall motion. With orientation change, a resistance that was effective for one motion screw axis might not be as effective for others. Orientation optimization is more effective in improving the WTR rating, while location optimization can improve both the WTR and MTR ratings. In order to improve the WTR rating using constraint modification, the responsible constraints must be identified first.
- In implementing design changes for multiple design variables, it was found that there is interaction, especially between orientation search spaces. It is hypothesized that interaction between orientation design spaces is more likely compared to interaction between location and orientation design spaces or between location design spaces. The search spaces of line and plane constraint

size show no interaction. It was found that longer line and larger plane constraints yield better resistance ratings in both WTR and MTR.

- Because interactions between design factors are possible, the simultaneous search optimum is different from the optima of a single design space. Therefore, whenever possible, the global design space (merging all possible potential constraint variables and the search space for each) must be searched. It was found, however, that although the simultaneous search optimum design shows better increase in assembly rating compared to an optimized design in localized design spaces, the increase is not cumulative. For example, if design improvements A and B individually yield 10% increase in rating, implementing both design improvements does not necessarily yield 20% increase in rating. Since redesign cost in manufacturing is different for different redesign solutions, all design alternatives must be weighed according to the associated redesign cost. The best design is the one that yields the most assembly rating increase with minimal cost of design implementation. This can be further implemented in a rating metric that measures the trade-off ratio between assembly resistance quality and cost of design implementation (i.e. dollar per additional features)

12.3 Application guideline in using the analysis and design tool in assembly design

Figure 12.1 summarizes how the analysis and design tool can be utilized in a design process.

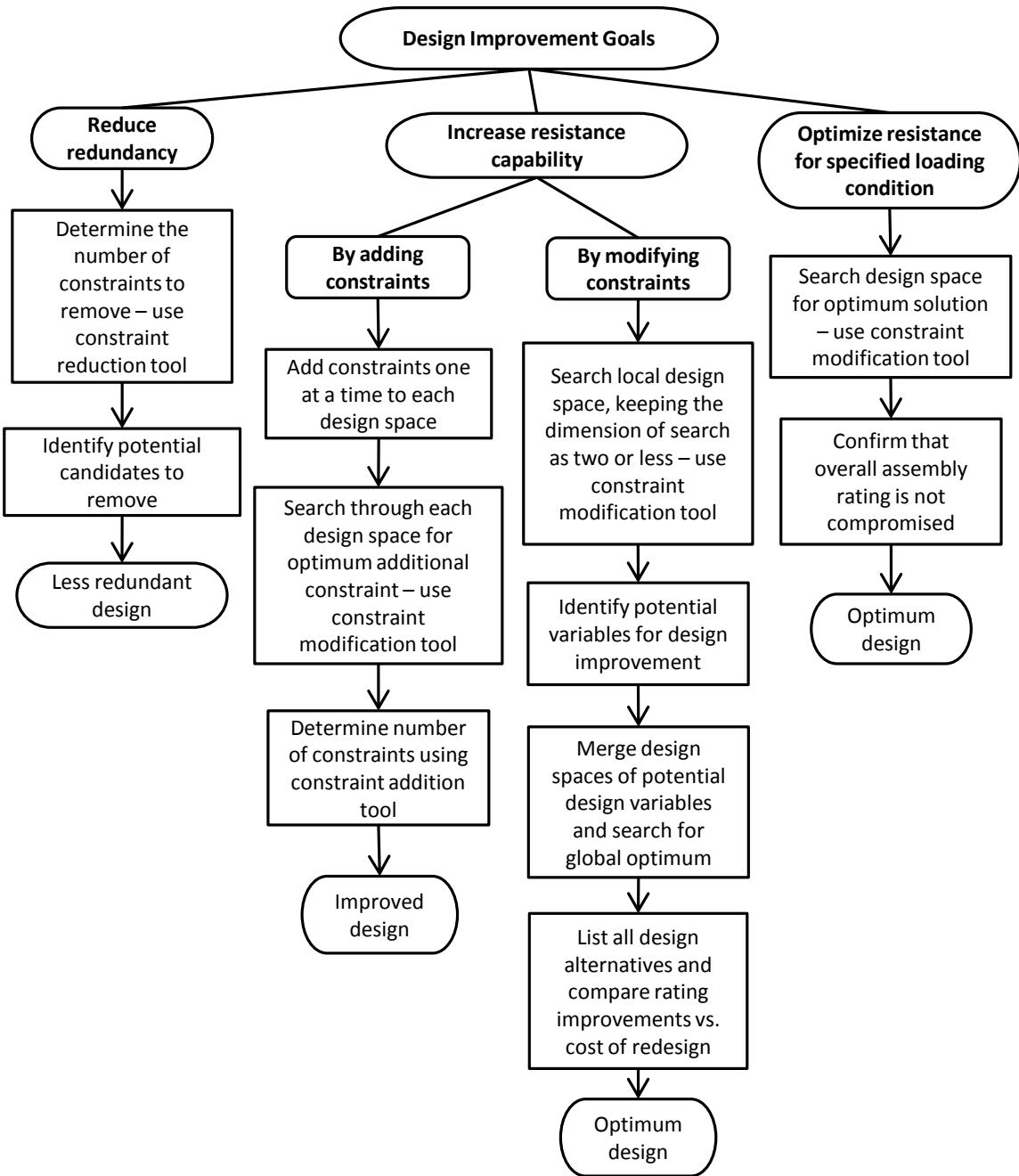


Figure 12.1 Application guidelines in utilizing analysis and design tool

A few notes and qualifications in using the analysis and design tool need to be made. Most of these limitations will be explained more thoroughly in Section 12.4.

- Line and plane contact should be modeled following the examples in Sections 10.4 and 10.5 as well as understanding the limitations set forth in later sections of this chapter.
- The MRR and TOR metrics and response surface plots should be mostly used in constraint addition and reduction, but not in constraint revision. In addition, the response surface plots for the orientation search space should be limited to the WTR plot only.
- The input wrench is modeled as a force couple, and the moment arm distance is used for scaling. This might not be intuitive. In making more detailed observations in the rating matrix, each motion must be considered in light of the input wrench scaling factor.

Future work is needed to address the limitations of the methodology and further expand the functionality of this design and analysis tool. The future work is discussed in order of importance. It is suggested that the issues be addressed in the same order.

12.4 Future work to address model and methodology limitation

The first category of the future work is to address the limitations of the models developed in this dissertation to analyze assembly design. There are three main areas in this category. The first is related to the rigid body assumption; the second is called the

bifurcation point in linear dependency, and the third is the need for guidelines in HOC modeling.

12.4.1 Incorporating stiffness into the analysis model

The model developed in this dissertation makes two important assumptions that limit the application of the conclusions drawn from it, namely the rigid body and frictionless assumptions. The rigid body assumption assumes that there is no deformation of the parts or the constraining elements, and consequently there is no shift of motion due to part deformation. The frictionless assumption assumes that there is no friction at the constraint interface.

Due to the rigid body assumption, the methodology ignores motion that might be possible due to a change in part shape. It also does not measure an assembly's load carrying capacity in terms of the stiffness of the geometry or of the materials. A constraint's relative effectiveness is determined by its location, orientation, and stiffness. The methodology assesses the first two factors, but not the stiffness. In redundant assemblies, loads tend to be transferred by the stiffest constraints. Therefore, the actual distribution of reaction forces among the constraints is not only dependent on the location and orientation, but also the relative stiffness between constraints. An assembly that is very strongly rated in this methodology might be weak in actuality due to flexibility of the part or the constraints. Due to the frictionless assumption, the methodology tends to underestimate the resistance quality of an assembly and therefore is conservative. It should be added that good practice and design codes require that friction be ignored in

the design of joints so this assumptions is often appropriate. Due to the clamping load provided by a bolted joint, these joints will be more conservative than snap-fits in a plastic part.

In Section 10.5, the printer housing perimeter lip has ribs that resist translation along the y-axis. However, further observation reveals that this constraint serves an aesthetic function. Since the wall thickness of the housing on its sides is flexible, the lip should deflect ‘together’ with the mating part lip for aesthetics and consumer perception of the product’s quality. Because its stiffness is much less compared to the constraint in the positive z-direction, the constraint in the y-direction is omitted. This is an example of the modeling gap where the analysis tool in this dissertation is incapable of taking these stiffnesses into consideration. The designer must consider part stiffness with intuition and decide whether a particular constraint effectively functions as a mechanical constraint or not.

One way to incorporate stiffness into the model already developed in this dissertation is to give proper weight to the resistance values in the rating matrix. For example, if constraint A has stiffness two times that of constraint B in its normal direction, then the resistance value of constraint A should be multiplied by 2 while constraint B is multiplied by 1. This weight coefficient must be relative to the rest of the constraints instead of absolute. Since they are weighted properly, the stiffer constraints will be rated higher because they play a more important role in resisting the constraints. This solution does not include stiffness as its absolute value. As a result, the load amplification ratio interpretation of the resistance values might be skewed.

A more fundamental solution might be to consider using the minimum energy principle to evaluate constraint reaction force. The methodology by Marin and Ferreira [29, 30] can be adapted to calculate the reaction forces simultaneously. This can be done with or without stiffness values. Stiffness can be incorporated by modeling the constraints as linear springs and solving the indeterminate problem as a linear programming problem. This solution can naturally incorporate friction into the model as well. This approach can simultaneously address the inaccuracies explained in the first paragraph of Section 12.5.1.

12.4.2 Bifurcation problem in linear dependency

Another limitation that exists in the model is the bifurcation problem in constraint linear dependency. In Sections 11.1.2 and 11.5.2, this limitation was briefly mentioned. The WTR and MTR exhibit some plots with discontinuities and local peaks. The discretization in the response surface plots is mainly caused by the bifurcation shift in linear dependency. This bifurcation point occurs at the limit as two constraints become almost collinear or parallel. When two constraints are collinear, they are linearly dependent. When more than three constraints are parallel, they are linearly dependent. A pivot constraint set that contains a linearly dependent set is not processed because it does not define a five system. However, a slight perturbation of the location (in the case of collinear constraints) or of the orientation (in the case of parallel constraints) eliminates their linear dependency. The pivot constraint set that contains this combination now is a five system and creates a new motion to be evaluated. Therefore, the resistance ratings

for this motion change the rating matrix. In general, when constraints are perturbed from their parallel or collinear condition, the number of evaluated motions suddenly increases. This is the bifurcation point that causes some discontinuities and unpredictable variation in the response surface plots. Near the bifurcation point, the inaccuracy is manifested as a motion with a zero pitch screw axis located very far from the part. This is analogous to a pure translation with an infinite pitch.

A possible solution is to perturb each constraint location and orientation proactively in the analysis phase. This will generate motion that is reciprocal to the perturbed pivot constraint set. This anticipation of additional motion to evaluate might reduce or eliminate the discontinuities due to the change in the evaluated motion set in the rating matrix. At the bifurcation point, the rank calculation might yield unpredictable solutions. A procedure to ensure that the rating is continuous at this bifurcation point must be implemented. To prevent inaccurate ratings due to a zero-pitch screw axis located very far from the origin, the algorithm should detect this and replace it with an infinite-pitch pure translation screw axis.

12.4.3 HOC modeling guidelines

In this dissertation, a line or plane contact is modeled as a higher order constraint (HOC) when the length or area span is longer than half of the overall part length in the same direction. In addition, the decision making process also involves observing the linear dependency between the constraint in question and the rest of the constraint configuration. Dilemmas in the decision making usually occur when the length of the line

is close to half of the overall part length in the same direction. The modeling process for the line constraints in Sections 10.4 and 10.5 are examples of this. At this point, the process of determining whether a constraint should be modeled as an HOC in this dissertation is done without a concrete and substantiated approach.

In order to solve many engineering problems, physical parts must be modeled mathematically. In the process, some inaccuracies due to assumptions always exist. At the same time, without assumptions the engineering problem often becomes too complex for practical use or requires overwhelming resources to solve. Therefore, assumptions must be carefully evaluated whether the inaccuracies that follow are acceptable. As fewer assumptions to simplify the problem are made, the problem grows increasingly complex. This fact applies for the HOC modeling challenge.

There is a decision continuum on the use of HOC to model higher order constraints. On one hand, it has been demonstrated that without HOC modeling, the constraint analysis becomes overwhelming and inaccurate due to the pseudo-redundancy effect. On the other hand, modeling very short lines and small planes with an HOC precludes the motion that should have been evaluated. Consider the extreme case where every single line or plane contact is modeled as an HOC in the constraint configuration. It is possible that the number of evaluated motions will be zero because none of the pivot constraint combination yields a five system. This is not wrong. The assembly indeed has achieved total restraint, i.e. there is no admissible motion. However, this analysis does not yield useful information for the designer. Recall that one of the motivations for this dissertation is the lack of quantitative metric (instead of binary test) to determine whether

total restraint is achieved or not. If the constraint configuration is modeled in a way that reduces the number of motions evaluated, the analysis tool is only useful as a binary test of total restraint, which defeats the purpose of the model.

Therefore, not using HOC in cases where lines and planes are relatively short provides a more conservative analysis at the risk of some pseudo-redundancy effects that skew the rating. At the same time, too many HOC can restrict the number of allowed and evaluated motions for the analysis calculation. In a way, if too many relatively short line or small plane contacts are modeled as HOC, the behavior of the assembly is not well understood. A ratio between the relative length or area of a line or plane and the part size can be used as the guideline. This becomes the critical length above which a line should be modeled as a line constraint. There might be other factors such as linear dependence with other constraints in the configuration, or the stiffness of the contact interface that can influence this decision. For example, more compliant contact interfaces are more likely to maintain line or plane contact areas under load. This guideline is important because most assembly contact interfaces are line or plane contacts and rarely true point contacts. There is a need for a formal study in creating and substantiating this guideline.

12.5 Future work to expand the functionality of the analysis and design tool

The second category of future work is to expand the functionality of the current design and analysis tool. This category is more practical in nature, and the future work related to the implementation of the model tends to be extensions based on the models developed in this dissertation. This is geared more toward improving the usefulness of the

design tool. The first set of improvements is for the purpose of improving the accuracy, efficiency, and capacity of the optimization search. The second set of improvements is for the purpose of adding features to the analysis and design tool.

12.5.1 Improvements in the accuracy, efficiency, and capacity of the optimization search

In the recalculation of the ‘base motion’ set mentioned in Sections 9.1.5, the pivot wrenches $\vec{C}_{P1,P5}$ sometimes involve modified constraints. This sometimes changes the rank of the pivot wrench matrix and causes some inaccuracies. Modifications to the pivot wrenches can cause the recalculated solution to differ from the original solution with non-modified pivot wrenches. This discrepancy might be resolved with using pseudo-pivot wrenches for the variables $\vec{C}_{P1,P5}$ in all resistance value calculations, including the baseline analysis subroutine. This was used in the specified loading condition optimization procedure. The procedure to calculate pseudo-pivot wrenches can be found in Section 9.4 and the associated MATLAB routine *main_spec_mot_rev.m*.

In Section 5.3, the input wrench magnitude is scaled using the maximum perpendicular distance between the screw axis and the locations of the constraints. Although a part assembly transfers load through the constraints, arbitrary loads can be applied on any of the body surfaces with or without constraints. The input wrench magnitude was scaled using the moment arm distance between the screw axis and the constraint location because this is the magnitude of the moment load applied as a force couple. By calculating this scaling factor for the input wrench force couple, the input

wrench magnitude also does not change during the optimization search. This will provide better and more predictable ratings for design optimization. In order to implement this, the input file must contain not only the set of constraint coordinate points but also the vertices of the part.

The MATLAB script computation time is highly dependent on the number of for-loops that need to be executed. MATLAB has a distributed computing toolbox that can take advantage of the full processing power of multi-core and multi-processor systems. To utilize this capability, the for-loop must be converted into a parallel for-loop (*parfor* function). The commands executed inside a parallel for-loop, however, must be independent of each other. In other words, the variables that are used and modified inside each iteration must be programmed in such a way that they have a definitive value regardless of the iteration index. The current MATLAB script requires significant coding changes in order to implement this. The parallel for-loop will increase the computational efficiency up to the number of processor cores. This is an important improvement in order to make the design tool capable of searching for a simultaneous search optimum. In many design contexts, the design space is far more than two dimensional in nature. For example, in the printer housing case study (Section 11.5.7), the higher dimensional optimum search involves five design variables. The computation time for 6 increments for each design variable takes more than 1 day on an 8 core computer system. Variable grouping of these design variables accelerates computation but actually reduces the precision of locating the simultaneous search optimum. Re-coding this algorithm in C

language can also improve computing performance and possibly take advantage of a multi-core graphics processing unit (GPU), which is optimized for vector computation.

12.5.2 Additional features to expand functionality of the design and analysis tool

The possible improvements in this area are:

- Expand the HOC model to include threaded fasteners and dowel pins. This was briefly discussed in Section 4.2.5. The threaded fastener can be modeled as a pin constraint and two opposite point constraints along the axis of the threaded fastener. A dowel pin can be modeled as a cylindrical joint. In addition to resisting translation in the plane normal to the pin axis, it can resist moments transverse to the pin axis.
- In Section 11.5.9, the constraint reduction algorithm recommends removal of a constraint apart from its feature. In this case, it suggests removing the pin constraint of a threaded fastener but not the point constraints. The constraint reduction algorithm can be improved by eliminating the possibility of removal of constraints apart from the assembly feature to which they belong.
- In the constraint reduction scheme and the higher dimension optimum search (higher than 2D optimization search), solutions that are close to being optimal can be identified to provide more than one alternative for design improvements. This is not necessary in 1D or 2D optimization searches because the design space is displayed as a response surface plot.

- In the optimization search, constraint variables can be grouped so that they are modified together. This ability to link constraints can be expanded to include parameterization of several design variables. An example of this is the capability to vary the trihedral constraint angle (in the Thompson's chair case study), which involves three point constraints.

12.6 Future work in understanding assembly design and attachment strategy

The third category of future work is to use the design and analysis tool for further understanding of assembly design and attachment strategy. The case studies in this dissertation serve both as a demonstration of the analysis and design tool capabilities as well as to develop an understanding of the dynamics of assembly constraint configurations. Further case studies need to be developed to:

- Understand the interaction between design factors. It was observed in the case studies that interaction tends to be higher between orientation search spaces compared to other search spaces. A formal case study that is aimed at studying these interactions will yield significant knowledge about assembly design. It was also observed that the assembly rating increase due to multiple design changes across different search spaces is not always cumulative.

Understanding whether interaction occurs across design spaces allows the designer to implement one or two design changes without having to explore the global design space. It might also be possible to quantify the statistical significance of the interaction using statistical inference.

- Generate new design principles in assembly design. If the model in this dissertation is improved by incorporating stiffness, there is a new area of case study to explore.
- Understand the design trade-off between optimizing assembly design for a specified loading condition and optimizing assembly design for arbitrary loads.
- Extend the application of the sensitivity analysis tool. If the model's bifurcation point limitation (Section 12.4) is resolved, the sensitivity analysis tool can be more useful in observing the changes not only in the WTR rating, but also the MTR rating.
- Analyze removable assembly case studies. Some assemblies are designed to be removable. The case study should investigate if the removal direction is the most weakly constrained motion. If it is not, then the most weakly constrained motion that is not the removal direction needs to be addressed.

LIST OF REFERENCES

1. Adams, J. D., Gerbino, S., Whitney, Daniel E., 1999, "Application of Screw Theory to Motion Analysis of Assemblies of Rigid Parts", *Proceedings of the IEEE International Symposium on Assembly and Task Planning*, Porto, Portugal, pp. 75-80.
2. Adams, J. D. and Whitney, D. E., "Application of Screw Theory to Constraint Analysis of Assemblies of Rigid Parts", *Proceedings of the IEEE International Symposium on Assembly and Task Planning*, Porto, Portugal, pp. 69-74.
3. Aicardi, M. G., Cannata, G., Casalino, 1992, "Contact Forces Decomposition for The Grasping of Rigid Objects" *Proceedings of the 1992 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Raleigh, NC, **3**, pp.1635-1641.
4. Asada, H., and By, A. B., 1985, "Kinematic Analysis of Workpart Fixturing for Flexible Assembly with Automatically Reconfigurable Fixtures," *IEEE Journal of Robotics and Automation*, RA-1, **2**, pp. 86–94.
5. Asada, H. and Kitagawa, M., 1988, "Form Closure Grasping by a Reconfigurable Universal Gripper." *ASME 1988 Manufacturing International*, ASME Press, **3**, pp. 85-90.
6. Ball, R. S., 1990, *A Treatise on the Theory of Screws*, Cambridge University Press, Cambridge, UK.
7. Bausch, J. J. and Youcef-Toumi, K., 1990, "Kinematic Methods for Automated Reconfiguration Planning." *IEEE Conference on Robotics and Automation*.
8. Blanding, D., 1999, *Exact Constraint: Machine Design Using Kinematic Principles*, ASME Press, New York.
9. Bozzo, T., 1997, "A Constraint Rating Scheme for Snap-fit Assemblies", M.S. Thesis, The Ohio State University, Columbus, OH.

10. Cai, W., Hu, S. J., Yuan, J. X., 1997, "A Variational Method of Robust Fixture Configuration Design for 3-D Workpieces," *ASME J. Manuf. Sci. Eng.*, **119**, pp. 593–602.
11. Chen, Y.-C. and Walker, I. D., 1994, "Visualization of Form-Closure and Force Distribution for Grasping Solid Objects", *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, **1**, pp. 154-159.
12. Chou, Y.-C., Chandru, V., Barash, M. M., 1989, "A Mathematical Approach to Automatic Configuration of Machining Fixtures: Analysis and Synthesis", *Journal of Engineering for Industry-Transactions of the ASME*, **111**(4), pp. 299-306.
13. Dai, J. S. and Kerr, D. R., 1996, "Analysis of Force Distribution in Grasps Using Augmentation." *Journal of Mechanical Engineering Science*, **210**(3), pp. 15-22.
14. DeMeter, E. C., 1993, "Selection of Fixture Configuration for the Maximization of Mechanical Leverage", *Proceedings of the 1993 ASME Winter Annual Meeting, New Orleans, LA*, 64, pp. 491-506.
15. DeMeter, E. C., 1994, "Restraint Analysis of Fixtures Which Rely on Surface Contact", *Journal of Engineering for Industry-Transactions of the ASME*, **116**(2), pp. 207-215.
16. Genc, S. R., Messler, R. W., Jr., Gabriele, G. A., 1997, "Methodology for Locking Feature Selection in Integral Snap-fit Assembly", *ASME Design and Engineering Technical Conference, Sacramento, CA*, ASME press, pp.68-74.
17. Hirai, S. and Asada, H., 1993, "Kinematics and Statics of Manipulation Using the Theory of Polyhedral Convex Cones" *International Journal of Robotics Research*, **12**(5), pp. 434-447.
18. Hunt, K. H., 1978, *Kinematic Geometry of Mechanisms*, Oxford University Press.
19. Kerr, D. R. and Sanger, D. J., 1983, "The Analysis of Kinematic Restraint" *Proceedings of the Sixth World Congress on Theory of Machines and Mechanisms*, pp.299-303.
20. Kerr, D. R. and Sanger, D. J., 1985, "The Synthesis of Point Contact Restraint of a Rigid Body", *Journal of Mechanisms, Transmissions, and Automation in Design*, **107**, pp. 521-525.
21. Kerr, J. and Roth, B., 1986, "Analysis of Multifingered Hands", *International Journal of Robotics Research*, **4**(4), pp. 3-17.

22. Kriegel, J. M., 1994, "Exact Constraint Design", ASME International M E Congress and Exhibition (Winter Annual Meeting), paper 94-WA/DE-18.
23. Lakshminarayana, K., 1978, "Mechanics of Form Closure", *ASME Paper 78-DET-32, New York*.
24. Lee J. D., Hu S. J., Ward A. C., 1999, "Workspace Synthesis for Flexible Fixturing of Stampings", *Journal of Manufacturing Science and Engineering*, **121**(3), pp. 478-484.
25. Lee, J. D. and Haynes, L. S., "Finite-Element Analysis of Flexible Fixturing System", *Journal of Engineering for Industry-Transactions of the ASME*, **109**(2), pp. 134-139
26. Luscher, A., 1996, "Part Nesting as a Plastic Snap-fit Attachment Strategy" *ANTEC 1996 Society of Plastic Engineers Annual Technical Conference and Exhibit*, Indianapolis, IN, SPE Press.
27. Luscher, A., Bonenberger, P., Gabriele, G. A., 1995, "Attachment Strategy Guidelines for Integral Attachment Features" *ANTEC 1995 Society of Plastic Engineers Annual Technical Conference and Exhibit*, Boston, MA, SPE press.
28. Marin, R. A. and Ferreira, P. M., 2001, "Kinematic Analysis and Synthesis of Deterministic 3-2-1 Locator Schemes for Machining Fixtures", *Journal of Manufacturing Science and Engineering*, **123**(4), pp. 708-719.
29. Marin, R. A. and Ferreira, P. M., 2002, "Optimal Placement of Fixture Clamps: Minimizing the Maximum Clamping Forces", *Journal of Manufacturing Science and Engineering-Transactions of the ASME*, **124**(3), pp. 686-694.
30. Marin, R. A. and Ferreira, P. M., 2002, "Optimal Placement of Fixture Clamps: Maintaining Form Closure and Independent Regions of Form Closure", *Journal of Manufacturing Science and Engineering-Transactions of the ASME*, **124**(3), pp. 676-685.
31. Markenscoff, X. and Papadimitriou, C. H., 1989, "Optimum Grip of a Polygon." *The International Journal of Robotics Research*, **8**(2), pp. 17-29.
32. Mantripragada, R. and Whitney, D. E., 1998, "The Datum Flow Chain: A systematic approach to assembly design and modeling", *Research in Engineering Design*, **10**, pp. 150-165.
33. Menassa, R. J. and DeVries, W. R., 1988, "Optimization Methods Applied to Selecting Support Positions in Fixture Design", *Proceedings of the USA-Japan Symposium on Flexible Automation - Crossing Bridges: Advances in Flexible Automation and Robotics*. Minneapolis, MN.

34. Nguyen, V. (1988), "Constructing Force-Closure Grasps", International Journal of Robotic Research, **7**(3), pp. 3-16.
35. Nguyen, V. (1989), "Constructing Stable Grasps", International Journal of Robotic Research, **8**(1), pp. 26-37.
36. Ohwovorile, M. S. and Roth, B., 1981, "An Extension of Screw Theory", *ASME Journal of Mechanical Design*, **103**, pp. 725-735.
37. Panagiotopoulos, P. D., Al-Fahed, A. M., 1994, "Robot Hand Grasping and Related Problems: Optimal Control and Identification", *The International Journal of Robotics Research*, **13**(2), pp. 127-136.
38. Pearce, E., Parkinson, A., Chase, K., 2004, "Tolerance Analysis and Design of Nesting Forces for Exactly Constrained Mechanical Assemblies", *Research in Engineering Design*, **15**, pp. 182-191.
39. Phillips, J., 1990, "Freedom in machinery v.1 and v.2", Cambridge University Press.
40. Reuleaux, F., 1876, "The Kinematics of Machinery", London, MacMillan and Co.
41. Rimon, E. and Burdick, J., 1996, "On Force and Form Closure for Multiple Finger Grasps", *Proceedings of the 1996 IEEE International conference on robotics and automation, Minneapolis, MN*, pp. 1795-1800.
42. Salisbury, J. K. and Roth, B, 1983, "Kinematic and Force Analysis of Articulated Mechanical Hands", *Journal of Mechanisms, Transmissions, and Automation in Design*, **105**(1), pp. 35-41.
43. Schimmels, J. M., Peshkin, M. A., 1992, "Admittance Matrix Design for Force-Guided Assembly", *IEEE Transactions on Robotics and Automation*, **8**(2), pp. 213-227.
44. Schmiedeler, J., 2007, *ME 851 Class Notes*, The Ohio State University.
45. Shapiro, A., Rimon, E., Burdick, J. W., 2001, "Passive Force Closure and its Computation in Compliant-Rigid Grasps" *IEEE International Conference on Intelligent Robots and Systems*, **3**, pp. 1769-1775.
46. Slocum, A. H., 1992, *Precision Machine Design*, Prentice-Hall, New Jersey.
47. Trinkle, J. C., 1992, "A Quantitative Test for Form Closure Grasps." *IEEE/RSJ International Conference of Intelligent Robots and Systems*.
48. Trinkle, J. C., 1992, "On the Stability and Instantaneous Velocity of Grasped Frictionless Objects", *IEEE Journal of Robotics and Automation*, **8**(5), pp. 560-572.

49. Waldron, K., 1993, *ME 851 Class Notes*, The Ohio State University.
50. Waldron, K., 1996, "The Constraint Analysis of Mechanisms", *Journal of Mechanisms*, **1**, pp. 101-114.
51. Waldron, K. J., 1972, "A Method of Studying Joint Geometry", *Mechanism and Machine Theory*, **7**.
52. Wang, M. Y. and Pelinescu, D. M., 2003, "Contact Force Prediction and Force Closure Analysis of a Fixture Rigid Workpiece with Friction", *Journal of Manufacturing Science and Engineering-Transactions of the ASME*, **125**(2), pp. 325-332
53. Whitehead T. N., 1954, *The Design and Use of Instruments and Accurate Mechanism*, Dover Press, New York.
54. Whitney, D., 2000, *Mechanical Assemblies*, Oxford University Press.
55. Whitney, D., Mantripragada, R., Adams, J. D., Rhee, S. J., 1999, "Designing Assemblies", *Research in Engineering Design*, **11**, pp. 229–253.
56. Wolter, J. D. and Trinkle, J. C., 1994, "Automatic Selection of Fixture Points for Frictionless Assemblies", *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 528-534.
57. Xiong, C.-H., Wang, M. Y., Tang, Y., Xiong, Y.-L., 2005, "Compliant Grasping with Passive Forces" *Journal of Robotic Systems*, **22**(5), pp. 271-285.
58. Xiuwen, G., Fuh, J. Y. H., Nee, A. Y. C., 1996, "Modeling of frictional elastic fixture-workpiece system for improving location accuracy", *IIE Transactions*, **28**(10), pp. 821-827.
59. Yoshikawa, T. and Nagai, K., 1991, "Manipulating and Grasping Forces in Manipulation by Multifingered Robot Hands", *IEEE Transaction on Robotics and Automation*, **7**(1), pp. 67-77.
60. Yoshikawa, T., 1996, "Passive and Active Closures by Constraining Mechanisms", *Proceedings - IEEE International Conference on Robotics and Automation*, **2**, pp. 1477-1484.

APPENDICES

APPENDIX A

MATLAB SCRIPTS

```

% Kinematic Constraint Analysis and Synthesis for Mechanical Part
Assembly
% filename: main.m
% Purpose: Analyze and optimize mechanical assembly constraint
configuration
%
% Called functions: inputfile_check, input_preproc, cp_to_wrench,
combo_preproc,
%                   main_loop, rating, result_open, histogr,
result_close,
%                   report, optim_main_rev, optim_main_red,
optim_postproc,
%                   sens_analysis_pos, sens_analysis_orient,
main_specmot_orig
%
%
% Copyright 2008 Leonard Rusli
% The Ohio State University

clc
clear variables
clear global
% Initialize variables
global cp cpin clin cpln cpln_prop no_cp no_cpin no_clin no_cpln
total_cp
global grp_members grp_rev_type grp_srch_spc
global combo_combo_proc_org wrench_proc input_wr_proc d_proc
global timestamp
global wr_all combo_dup_idx
global pts max_d
% Request input file from user
[cp, cpin, clin, cpln, cpln_prop, inputfile...
    grp_members, grp_rev_type, grp_srch_spc, ...
    add_cp_type, add_cp]=input_menu();
% Time stamp for stop watch
tic % Start stopwatch to measure elapsed time
timestamp=clock;
clc
disp('Initializing input file...')
% Check for input file for consistency & format
inputfile_check
if inputfile_ok==false
    disp(inputfile_error)
    return
end
% Input file preprocessor to normalize constraint normal vector
and count number of constraints for each type
input_preproc
% Transform constraints into wrenches
[wr_all pts max_d]=cp_to_wrench(cp,cpin,clin,cpln,cpln_prop);
% Create combination scheme for composing pivot constraint sets
[combo]=combo_preproc(total_cp,'original');
% Main for-loop subroutine to process pivot constraint sets
clc
[mot_half R combo_proc_org input_wr_proc_org ...
    d_proc_org]=main_loop(combo,wr_all,1,'original');

% Stop further analysis if there is no evaluated motion
if size(mot_half,1)==0
    clc;fprintf('There is no motion to be evaluated. \n');
    return
end
fprintf('Calculating final ratings... \n')

% Motion post-processing
mot_half_rev=[mot_half(:,1:6) mot_half(:,7:10)]; % Inverse om
and mu to get reverse motion
mot_half_org=mot_half; % Store evaluated motions from
original/baseline analysis
combo_dup_idx_org=combo_dup_idx; % Keep track duplicate motion
references
mot_all_org=[mot_half;mot_half_rev]; % Merge forward and reverse
motion
no_mot_half=size(combo_proc_org,1); % Count the number of half-
duplex/forward motion only
no_mot=size(mot_all_org,1); % Count the number of total motion in
original/baseline set

% Rating post-processing
Ri=1./R;
Ri=(round(Ri.*1e4)).*1e-4; % Round rating matrix values to 4
decimal places

% Remove duplicate motions
[mot_all_org_uniq uniq_idx]=unique(mot_all_org, 'rows'); %
Identify and remove extra duplicate motion
Ri_uniq=Ri(uniq_idx,:); % Remove rating matrix rows affiliated
with duplicate motion

% Calculate overall assembly ratings
[Rating_all_org WTR_idx_org free_mot free_mot_idx best_cp
rowsum]=rating(Ri_uniq, mot_all_org_uniq);

% Generate analysis report
[result]=result_open(inputfile); % Open result file for writing
report % Write report in result file

% Plot total resistance histogram
histogr(Rating_all_org, rowsum)

result_close % Close results file

toc % Stop stopwatch and display elapsed time

% Show design optimization menu
fprintf('\n0. Skip optimization')
fprintf('\n1. Constraint revision')
fprintf('\n2. Constraint reduction')
fprintf('\n3. Constraint addition')
fprintf('\n4. Sensitivity analysis position')
fprintf('\n5. Sensitivity analysis orientation')
fprintf('\n6. Known loading condition study\n\n')
optim_type = input('Run which type of optimization? ');

tic; % Start stopwatch to measure optimization elapsed time

if optim_type == 1
    no_step=input('How many steps for each variable? '); % Prompt
user input for optimization search resolution
    optim_main_rev % Start constraint modification optimization
routine
        optim_postproc(no_step,no_dim, WTR_optim_chg,
MRR_optim_chg,... % MRR_optim_chg, TMR_optim_chg, inputfile)
elseif optim_type == 2
    no_red=input('How many constraints to remove at a time? ');
    optim_main_red % Start constraint reduction optimization
routine
elseif optim_type == 3
    disp('THE optim_main_add.m PRELIMINARY CODE CONTAINS ERRORS.
IN THE DISSERTATION CONSTRAINT ADDITION IS STILL DONE MANUALLY.')
    optim_main_add % Start constraint addition optimization
routine
elseif optim_type == 4
    sens_analysis_pos % Start position sensitivity analysis
routine
elseif optim_type == 5
    sens_analysis_orient % Start orientation sensitivity analysis
routine
elseif optim_type == 6
    % Prompt user input for specified loading condition screw
axis
    mot_input=input('Specify loading screw axis (0 for mot_set): ');
    if mot_input==0
        specmot=mot_set; % Use variable mot_set if multiple
loading screw axis is defined
    end
    [Ri_spec_mot mot_proc]=main_specmot_orig(specmot) % Start
specified loading condition routine
else
    disp('Optimization skipped');
end

fprintf('Calculation done.\n'); toc; % Display optimization
elapsed time
save(inputfile) % Store workspace as 'inputfile'.mat


---


% Checks input file for consistency
% filename: inputfile_check
%
inputfile_ok=true;
if isempty(cp)==1 & isempty(cpin)==1 & isempty(clin)==1 &
isempty(cpln)==1
    inputfile_ok=false;
    inputfile_error='All CP variable is empty';
end
if isempty(cp)==0 & size(cp,2)~=6
    inputfile_ok=false;
    inputfile_error='CP size is incorrect';
end
if isempty(cpin)==0 & size(cpin,2)~=6
    inputfile_ok=false;
    inputfile_error='CPIN size is incorrect';
end
if isempty(clin)==0 & size(clin,2)~=10
    inputfile_ok=false;
    inputfile_error='CLIN size is incorrect';
end
if isempty(cpln)==0 & size(cpln,2)~=7 & size(cpln_prop,2)~=8
    inputfile_ok=false;
    inputfile_error='CPLN or CPLN_PROP size is incorrect';
end
if size(cpln,1)~size(cpln_prop,1)
    inputfile_ok=false;

```

```

    inputfile_error='CPLN and CPLN_PROP does not have the same
    number of constraints';
end



---


function [cp, cpin, clin, cpln, cpln_prop, inputfile...
    grp_members, grp_rev_type, grp_srch_spc, ...
    add_cp_type,add_cp]=input_menu()
% Menu to select input file
% filename: input_menu.m
% Purpose:
% - Request input file from user
% - Store the input for report filename
%
% Input variables: -
% Output variables: cp, cpin, clin, cpln, cpln_prop, inputfile,
%                   grp_members, grp_rev_type, grp_srch_spc,
%                   add_cp_type,add_cp
% Called functions: -
%
cp=[]; cpin=[]; clin=[]; cpln=[]; cpln_prop=[];
cp_rev_allow=[]; pt_srch=[]; lin_srch=[]; pln_srch=[];
dir_srch=[];
lin_size_srch=[]; pln_size_srch=[];
cp_rev_grp=[]; cp_grp_prop=[];
grp_members=[];
grp_rev_type=[];
grp_srch_spc=[];
add_cp_type=[];
add_cp([]);

% Create labels for each input file
f1='case1a_chair_height';
f2='case1b_chair_height_angle';
f3='case2a_cube_scalability';
f4='case2b_cube_tradeoff';
f5='case3a_cover_leverage';
f6='case3b_cover_symmetry';
f7='case3c_cover_orient';
f8='case4a_endcap_tradeoff';
f9='case4b_endcap_circlnsrch';
f10='case5a_printer_4screws_orient';
f11='case5b_printer_4screws_line';
f12='case5c_printer_snap_orient';
f13='case5d_printer_snap_line';
f14='case5e_printer_partingline';
f15='case5f1_printer_line_size';
f16='case5f2_printer_sideline_size';
f17='case5g_printer_5d';
f18='case5rev_a_printer_2screws';
f19='case5_printer_allscrews';
f20='case5rev_d_printer_remove2_bot_screw';
f21='case5rev_b_printer_flat_partingline';

% Display input file choices & prompt user
fprintf('\n1: %s\n2: %s\n3: %s\n4: %s\n5: %s\n6: %s\n7: %s\n8:
%s\n9: %s\n10: %s\n...
    ,f1,f2,f3,f4,f5,f6,f7,f8,f9,f10);
fprintf('11: %s\n12: %s\n13: %s\n14: %s\n15: %s\n16:
%s\n17: %s\n18: %s\n19: %s\n20: %s\n21:
%s\n',f1,f2,f3,f4,f5,f6,f7,f8,f9,f10);
cp_set = input('Which input file to run? ');

if cp_set == 1
    case1a_chair_height
    inputfile=f1;
elseif cp_set == 2
    case1b_chair_height_angle
    inputfile=f2;
elseif cp_set == 3
    case2a_cube_scalability
    inputfile=f3;
elseif cp_set == 4
    case2b_cube_tradeoff
    inputfile=f4;
elseif cp_set == 5
    case3a_cover_leverage
    inputfile=f5;
elseif cp_set == 6
    case3b_cover_symmetry
    inputfile=f6;
elseif cp_set == 7
    case3c_cover_orient
    inputfile=f7;
elseif cp_set == 8
    case4a_endcap_tradeoff
    inputfile=f8;
elseif cp_set == 9
    case4b_endcap_circlnsrch
    inputfile=f9;
elseif cp_set ==10
    case5a_printer_4screws_orient
    inputfile=f10;
elseif cp_set ==11
    case5b_printer_4screws_line
    inputfile=f11;
elseif cp_set ==12
    case5c_printer_snap_orient
    inputfile=f12;
elseif cp_set ==13
    case5d_printer_snap_line
    inputfile=f13;
elseif cp_set ==14
    case5e_printer_partingline
    inputfile=f14;
elseif cp_set ==15
    case5f1_printer_line_size
    inputfile=f15;
elseif cp_set ==16
    case5f2_printer_sideline_size
    inputfile=f16;
elseif cp_set ==17
    case5g_printer_5d
    inputfile=f17;
elseif cp_set ==18
    case5rev_a_printer_2screws
    inputfile=f18;
elseif cp_set == 19
    case5_printer_allscrews
    inputfile=f19;
elseif cp_set == 20
    case5rev_d_printer_remove2_bot_screw
    inputfile=f20;
elseif cp_set == 21
    case5rev_b_printer_flat_partingline
    inputfile=f21;
end



---


% Input file preprocessor
% filename: input_preproc.m
% Function:
% - Normalizes the constraint normal vectors
% - Count the number of constraints for each constraint type
%
% Input variables: cp, cpin, clin, cpln
% Output variables: no_cp, no_cpin, no_clin, no_cpln, total_cp
% Called functions: -
%
% Normalize direction cosines for each constraint type
for i=1:size(cp,1)
    cp(i,4:6)=cp(i,4:6)./norm(cp(i,4:6));
end
for i=1:size(cpin,1)
    cpin(i,4:6)=cpin(i,4:6)./norm(cpin(i,4:6));
end
for i=1:size(clin,1)
    clin(i,4:6)=clin(i,4:6)./norm(clin(i,4:6));
    clin(i,7:9)=clin(i,7:9)./norm(clin(i,7:9));
end
for i=1:size(cpln,1)
    cpln(i,4:6)=cpln(i,4:6)./norm(cpln(i,4:6));
end
no_cp=size(cp,1); % no_cp is the number of contact points
no_cpin=size(cpin,1); % no_cpin is the number of contact pins
no_clin=size(clin,1); % no_clin is the number of contact line
no_cpln=size(cpln,1); % no_cpln is the number of contact plane
total_cp = no_cp+no_cpin+no_clin+no_cpln; % total_cp is the total
number of constraints


---


function [wr_all pts
max_d]=cp_to_wrench(cp,cpin,clin,cpln,cpln_prop)
% Transforms constraints into wrenches
% filename: cp_to_wrench.m
% Purpose:
% - Transforms point, pin, line, and plane constraints into
wrench systems
% - Create discretized constraint location points for moment arm
calculation
% - Calculates the maximum distance between discretized location
points
%
% Input variables: cp, cpin, clin, cpln, cpln_prop
% Output variables: wr_all, pts, max_d
% Called functions: -
%
wr_pt=[];wr_pin=[];wr_lin=[];wr_pln=[];wr_all=[]; % Initialize
default values for variables
j=1; % Initialize index for wr_all

% Transform all point contacts
for i = 1:size(cp,1)
    wr=[cp(i,4:6),cross(cp(i,1:3),cp(i,4:6))];
    wr_all{j}=wr;
    j=j+1; % Advance index
end

```

```

% Transform all pin contacts
% Pin syntax cpin=[centerpoint(x,y,z), pinaxis(x,y,z)]
for i = 1:size(cpin,1)
    axes=null(cpin(i,4:6)); % Solve for basis vectors of plane
    normal_to_pin_axis
    om_axis1=axes(:,1)'; % Wrench principle axis 1 zero pitch
    om_axis2=axes(:,2)'; % Wrench principle axis 2 zero pitch
    mu_axis1=cross(cpin(i,1:3),om_axis1); % mu principle axis 1
    zero_pitch
    mu_axis2=cross(cpin(i,1:3),om_axis2); % mu principle axis 2
zero_pitch
wr_all{j}=[om_axis1 mu_axis1;om_axis2 mu_axis2]; % Merge
wrench_principle_axes
j=j+1; % Advance index
end

% Transform line contact
% Line syntax clin=[midpoint(x,y,z), line dir(x,y,z), constraint
dir(x,y,z), length of line]
for i = 1:size(clin,1)
    om_axis1=clin(i,7:9); %zero pitch
    om_axis2=[0 0 0]; %inf pitch
    mu_axis1=cross(clin(i,1:3),om_axis1); %zero pitch
    mu_axis2=cross(clin(i,4:6),clin(i,7:9)); %inf pitch
    wr_all{j}=[om_axis1 mu_axis1;om_axis2 mu_axis2];
    j=j+1;
end

% Plane syntax cpln=[midpoint(x,y,z), normal(x,y,z), type (1 is
rect 2 is circ)
% cpln_prop= [xaxisdir(x,y,z),x-length,yaxisdir(x,y,z),y-length]
% or for circular cpln_prop=[radius];
% or for circular cpln_prop=[radius];

% Transform plane contact
for i = 1:size(cpln,1)
    axes=null(cpln(i,4:6));
    om_axis1=cpln(i,4:6); %zero pitch
    om_axis2=[0 0 0]; %inf pitch
    om_axis3=[0 0 0]; %inf pitch
    mu_axis1=cross(cpln(i,1:3),om_axis1); %zero pitch
    mu_axis2=axes(:,1)';
    mu_axis3=axes(:,2)';
    wr_all{j}=[om_axis1 mu_axis1;om_axis2 mu_axis2;om_axis3];
    mu_axis3];
    j=j+1;
end
clear i j
pts=[];
if isempty(cp)==0
    pts=[pts;cp(:,1:3)];
end
if isempty(cpin)==0
    pts=[pts;cpin(:,1:3)];
end
if isempty(clin)==0
    for j=1:size(clin,1)
        pts=[pts;clin(j,1:3)+clin(j,10)/2.*clin(j,4:6)];
        pts=[pts;clin(j,1:3)-clin(j,10)/2.*clin(j,4:6)];
    end
end
if isempty(cpln)==0
    for j=1:size(cpln,1)
        if cpln(j,7)==1
            pts=[pts;cpln(j,1:3)+cpln_prop(j,4)/2.*cpln_prop(j,1:3)+cpln_prop
(j,8)/2.*cpln_prop(j,5:7)];
            pts=[pts;cpln(j,1:3)+cpln_prop(j,4)/2.*cpln_prop(j,1:3)-
cpln_prop(j,8)/2.*cpln_prop(j,5:7)];
            pts=[pts;cpln(j,1:3)-cpln_prop(j,4)/2.*cpln_prop(j,1:3)+cpln_prop(j,8)/2.*cpln_prop(j,
5:7)];
            pts=[pts;cpln(j,1:3)-cpln_prop(j,4)/2.*cpln_prop(j,1:3)-
cpln_prop(j,8)/2.*cpln_prop(j,5:7)];
        else
            axes=null(cpln(j,4:6));
            pts=[pts;cpln(j,1:3)+cpln_prop(j,1).*axes(:,1)'];
            pts=[pts;cpln(j,1:3)-cpln_prop(j,1).*axes(:,1)'];
            pts=[pts;cpln(j,1:3)+cpln_prop(j,1).*axes(:,2)'];
            pts=[pts;cpln(j,1:3)-cpln_prop(j,1).*axes(:,2)'];
        end
        pts=[pts;cpln(j,1:3)+cosd(45)*cpln_prop(j,1).*axes(:,1)'+cosd(45)
*cpln_prop(j,1).*axes(:,2)'];
        pts=[pts;cpln(j,1:3)+cosd(45)*cpln_prop(j,1).*axes(:,1)-
cosd(45)*cpln_prop(j,1).*axes(:,2)'];
        pts=[pts;cpln(j,1:3)-cosd(45)*cpln_prop(j,1).*axes(:,1)-
cosd(45)*cpln_prop(j,1).*axes(:,2)];
    end
end

```

```

end
c=nchoosek(1:size(pts,1),2);
for a=1:size(c,1)
    distance(a)=((pts(c(a,1),1)-pts(c(a,2),1))^2 + ...
    (pts(c(a,1),2)-pts(c(a,2),2))^2 + ...
    (pts(c(a,1),3)-pts(c(a,2),3))^2)^.5;
end
max_d=max(distance);

function [combo]=combo_prepoc(total_cp,set)
% Create an exhaustive combination that form the five-system
wrench
% filename: combo_prepoc.m
% Input variables: cp, cpin, clin, cpln, cp_rev, cpin_rev,
clin_rev, cpln_rev
% Output variables: combo
% Called functions: -
%
global cp cpin clin cpln cp_rev cpin_rev
global cp_rev cpin_rev clin_rev cpln_rev cpln_prop_rev
combo2=[];combo3=[];combo4=[];
% This allows the function to be used for both the baseline
analysis and the optimization routine
if strcmp(set,'original')==1
    set_cp=cp; set_cpin=cpin; set_clin=clin; set_cpln=cpln;
    set_cpln_prop=cpln_prop;
elseif strcmp(set,'revised')==1
    set_cp=cp_rev; set_cpin=cpin_rev; set_clin=clin_rev;
    set_cpln=cpln_rev; set_cpln_prop=cpln_prop_rev;
end

% If there is a plane (3 DOF) constraint start with 2 constraint
combinations because 2 planes can compose pivot constraint set
if isempty(set_cpin)==0
    combo2=nchoosek(1:total_cp,2); % Combination involving 2
constraints
    combo2=[combo2,zeros(size(combo2,1),3)]; % Substitute blanks
with zeros
    combo3=nchoosek(1:total_cp,3); % Combination involving 3
constraints
    combo3=[combo3,zeros(size(combo3,1),2)];
    combo4=nchoosek(1:total_cp,4); % Combination involving 4
constraints
    combo4=[combo4,zeros(size(combo4,1),1)];
    combo5=nchoosek(1:total_cp,5); % Combination involving 5
constraints

% If there is a pin or line (2 DOF) constraint, start with 3
constraint combinations because 3 pins/lines can compose pivot
constraint set
elseif isempty(set_cpin)==0 || isempty(set_clin)==0
    combo3=nchoosek(1:total_cp,3); % Combination involving 3
constraints
    combo3=[combo3,zeros(size(combo3,1),2)];
    combo4=nchoosek(1:total_cp,4); % Combination involving 4
constraints
    combo4=[combo4,zeros(size(combo4,1),1)];
    combo5=nchoosek(1:total_cp,5); % Combination involving 5
constraints

% If there is no HOC, 5 point constraints are always needed to
compose pivot constraint set
else
    combo5=nchoosek(1:total_cp,5);
end
combo=[combo2;combo3;combo4;combo5]; % Merge all combination



---


function [mot_half R combo_proc input_wr_proc
d_proc]=main_loop(combo,wr_all,dispbar,set)
% Main processor for computing reciprocal motions from pivot
constraint sets and rate resistance quality
% filename: main_loop.m
% Function:
% - Create pivot wrench set
% - Check linear independence in pivot wrench matrix
% - Calculate reciprocal motion
% - Test for duplicate motion
% - Create input wrench and reaction wrench
% - Rate resistance quality of each constraints to the motion
% - Merge resistance values
%
% Input variables: cp, cpin, clin, cpln, cp_rev, cpin_rev,
clin_rev, cpln_rev
% Output variables: mot_half R combo_proc input_wr_proc d_proc
% Called functions: form_combo_wrench, rec_mot, input_wr_compose,
react_wr_5_compose
%           rate_cp, rate_cpin, rate_clin, rate_cpln1,
rate_cpln2
%
global cp cpin clin cpln cpln_prop no_cp no_cpin no_clin no_cpln
global cp_rev cpin_rev clin_rev cpln_rev cpln_prop_rev

```

```

global wrench_proc combo_dup_idx
global pts max_d pts_rev max_d_rev

% This allows the function to be used for both the baseline
analysis and optimization routine
if strcmp(set,'original')==1
    set_cp=cp; set_cpin=cpin; set_clin=clin; set_cpln=cpln;
set_cpln_prop=cpln_prop;
set_pts=pts; set_max_d=max_d;
else
    set_cp=cp_rev; set_cpin=cpin_rev; set_clin=clin_rev;
set_cpln=cpln_rev; set_cpln_prop=cpln_prop_rev;
set_pts=pts_rev; set_max_d=max_d_rev;
end

% Initialize variables
mot_hold=[];m=1;
Rcp_pos=[];Rcp_neg=[];Rcpin=[];Rclin_pos=[];Rclin_neg=[];
Rcpln_pos=[];Rcpln_neg=[];
combo_dup_idx=zeros(size(combo,1),1);

if dispbar==1
    prog_bar=waitbar(0,'Base Calculation progress'); % Display
progress bar
end

for i = 1:size(combo,1)
    % Update progress bar
    if dispbar==1
        if size(combo,1)<=1000
            waitbar(i/size(combo,1),prog_bar);
        else
            if mod(i,50)==0
                waitbar(i/size(combo,1),prog_bar);
            end
        end
    end

    % Compose pivot wrench matrix
    [pivot_wr]=form_combo_wrench(i,combo, wr_all);
    rank_check = rank(pivot_wr); % Calculate the number of
linearly independent wrench in the set
    if rank_check==5 % Proceed if the wrench is a 5-system

        % Solve the reciprocal motion
        [mot]=rec_mot(pivot_wr);

        % Check for duplicate motion
        [isnbr_idx]=ismember(mot,mot_hold, 'rows');
        if isnbr==0
            mot_hold=[mot_hold;mot];
        else
            combo_dup_idx(i)=idx; % Record the motion index to
which the motion is a duplicate
            continue
        end

        % Compose input wrench
        [input_wr d]=input_wr_compose(mot, set_pts, set_max_d);

        % Set up matrix of constraining wrench from pivot
        % constraints (rho as origin)

        [react_wr_5]=react_wr_5_compose(combo(i,:),mot(7:9)',set);

        % Calculate reaction value for each CP (reaction wrenches
are calculated with these functions)
        for j = 1:no_cp
            [Rcp_pos(m,j) Rcp_neg(m,j)]=rate_cp(mot,react_wr_5,
input_wr, set_cp(j,:));
        end

        % Calculate resistance value for each CPIN
        for j = 1:no_cpin
            [Rcpin(m,j)]=rate_cpin(mot,react_wr_5,input_wr,set_cpin(j,:));
        end

        % Calculate resistance value for each CLIN
        for j = 1:no_clin
            [Rclin_pos(m,j)]=rate_clin(mot,react_wr_5,input_wr,set_clin(j,:));
        end

        % Calculate resistance value for each CPLN
        for j = 1:no_cpln
            if set_cpln(j,7)==1
                [Rcpln_pos(m,j)]=rate_cpln(mot,react_wr_5,
input_wr, set_cpln(j,:),set_cpln_prop(j,:));
            elseif set_cpln(j,7)==2
                [Rcpln_pos(m,j)]=rate_cpln2(mot,react_wr_5,
input_wr, set_cpln(j,:),set_cpln_prop(j,:));
            end
        end
    end

    % Collect the calculated moment arms
    [Rcpln_pos(m,j)]=rate_cpln2(mot,react_wr_5,
input_wr, set_cpln(j,:),set_cpln_prop(j,:));
    end

    % Collect the calculated moment arms
    [Rcpln_pos(m,j)]=rate_cpln2(mot,react_wr_5,
input_wr, set_cpln(j,:),set_cpln_prop(j,:));
    end

    % Collect processed combinations
    wrench_proc(m,:)=pivot_wr; % Collect processed pivot
wrenches
    input_wr_proc(m)=input_wr; % Collect input wrenches
    m=m+1; % Advance rating matrix row index
else
    continue
end

mot_half=mot_hold; % Grab the evaluated motion set

Rcp=[Rcp_pos;Rcp_neg]; % Merge resistance value for fwd and rev
motion
Rcpin=[Rcpin;Rcpin]; % Resistance values for fwd and rev motion
is the same for cpin
Rclin=[Rclin_pos;Rclin_neg]; % Merge resistance value for fwd and
rev motion
Rcpln=[Rcpln_pos;Rcpln_neg]; % Merge resistance value for fwd and
rev motion
R=[Rcp Rcpin Rclin Rcpln]; % Merge all ratings across different
constraints
if dispbar==1
    close(prog_bar); %close the progress bar
end

```

```

function [rat WTR_idx free_mot free_mot_idx best_cp
rowsum]=rating(Ri, mot_all)
% Calculate the assembly rating metrics
% filename: rating.m
% Input variables: Ri, mot_all
% Output variables: rat, WTR_idx, free_mot, free_mot_idx,
% best_cp, rowsum
% Called functions: -

[max_of_row best_cp]=max(Ri,[],2); % Identify maximum of each row
rowsum=sum(Ri,2); % Sum the rows (row refers to motions)

%Find unconstrained motion
free_mot_idx=find(rowsum==0); %find any motion that is rated zero
and classify as free motion
free_mot=mot_all(free_mot_idx,:);

%Redundancy ratio
if min(rowsum)>=0 % If no unconstrained motion exists
    MRR=mean(rowsum./max_of_row);
    WTR=min(rowsum);
    WTR_idx=find(rowsum>=WTR & rowsum<=WTR*1.1);
    MTR=mean(rowsum);
else
    WTR=0;WTR_idx=0;
    MRR=0;
    MTR=0;
end
% Report all rating
rat=[WTR MRR MTR/MRR];

```

```

function [result]=result_open(inputfile)
% Open html result file for writing
% filename: result_open.m
% Input variables: inputfile
% Output variables: result
% Called functions: -

% Create filename according to inputfile
filename=['Result - ' inputfile '.html'];
result = fopen(filename, 'wt');

% Result header for html format
fprintf(result,'<HEAD><nTITLE>\nResult file for
%</TITLE><n>',inputfile);
fprintf(result,'<STYLE TYPE="text/css">\n!!--\nT1\n  {\n';
fprintf(result,'font-family:sans-serif; \nfont-size:10pt;\n}\n-->
</STYLE>\n');
fprintf(result,'</HEAD>\n');
fprintf(result,'<FONT SIZE=3 FACE="helvetica">\n');

% Close html result file
% filename: result_close.m
%
timestop=clock;

totaltime=etime(timestop,timestart)/60; % total time in minutes

```

```

totaltime_min=fix(totaltime);
totaltime_sec=mod(totaltime,1)*60;
fprintf(result,'

\nTotal analysis time: %5.0f minutes %2.0f
seconds \n<p>',totaltime_min,totaltime_sec);

% fprintf(result,'< /T1 >\n');
fprintf(result,'< /font >\n');
fclose(result);


---


% Generate analysis report
% filename: report.m
%
global totaltime_min totaltime_sec timestamp

fprintf(result,'<b>Input File: %s <p>\n</b>',inputfile);

%Print Main Ratings
fprintf(result,'Weakest Total Resistance rating (WTR): %5.4f
(LAR: %6.3f)<br>n',Rating_all_org(1), 1/Rating_all_org(1));
fprintf(result,'Mean Redundancy Ratio (MRR): %5.4f
<br>n',Rating_all_org(2));
fprintf(result,'Mean Total Resistance Rating (MTR): %5.4f (LAR:
%6.3f)<br>n',Rating_all_org(3),1/Rating_all_org(3));
fprintf(result,'Trade Off Ratio (TOR): %5.4f
<p>\n</p>',Rating_all_org(4));

% Print unconstrained motion if exists
if isempty(free_mot)==0
    fprintf(result,'<b>Unconstrained Motion: </b><br>n');
    free_mot_idx_dummy=zeros(size(free_mot,1),1);
    dummy=zeros(size(free_mot,1),1);
    table_mot(result, free_mot, dummy);
    return
end

% Print most weakly constrained motion
else
    fprintf(result,'<b>There is no unconstrained motion.
</b><p>\n</p>');

    fprintf(result,'<p>\n<b>Weakest Constrained Motion (according
to WTR): <br></b></p>');
    WTR_mot=mot_all_org.uniq(WTR_idx_org,:);
    TR=rowsum(WTR_idx_org);
    WTR_cp=best_cp(WTR_idx_org);
    table_mot(result, WTR_mot, TR); % Print motion in tabular
format
end

% Identify best resistance constraints for each motion
for i=1:size(Ri_uniq,2)
    non_zero_cnt_in_col(i)=nnz(Ri_uniq(:,i)); %%ok<AGROW>
    b=find(best_Cp==i);
    cp_best_count(i)=size(b,1); %%ok<AGROW> %Counter for # of
times a cp provide best resistance
end

% Individual CP rating
cp_col=1:total_cp;
cp_invd_rat = sum(Ri_uniq,1)./non_zero_cnt_in_col;

% Calculate CA% and CBR%
cp_active_pct=non_zero_cnt_in_col/no_mot*100;
cp_best_pct=cp_best_count./no_mot.*100;

% Merge constarints ratings in a matrix
cp_table=[cp_col', cp_invd_rat', cp_active_pct', cp_best_pct'];

fprintf(result,'<p><TABLE BORDER=2>\n');
fprintf(result,'<b> <font size=3
FACE="helvetica"><tr><th>CP#</th> <th>Individual Rating</th>
<th>Active %</th> <th>Best Resistance %%</th> <tr></b>\n'');

% Print constraints' individual ratings
for i=1:size(cp_table,1)
    fprintf(result,'<tr><td>%d</td> <td>%5.4f</td>
<td>%4.1f%%</td> <td>%4.1f%%</td> </tr>\n',cp_table(i,:));
end

% Print computation information
fprintf(result,'</font></TABLE><p>\n');
total_combo=size(combo,1);
fprintf(result,'Total Possible Combination: %8.0f
<br>n',total_combo);
fprintf(result,'Total Linearly Independent Combination Processed:
%8.0f <br>n',size(combo_proc_org,1));
fprintf(result,'Total Unique screw motion found:
%8.0f<p>\n</p>',size(mot_all_org.uniq,1)/2);



---


function histogr(Rating_all_org, rowsum)
% Plot the total resistance histogram
% filename: histogr.m
% Input variables: Rating_all_org, rowsum
% Output variables: -
% Called functions: -


```

```

%
if Rating_all_org(1)~0
figure('Position',[100,500,600,300]);
hist(rowsum,length(rowsum)*2^.5);
xlabel('Total Resistance Value','fontWeight','b');ylabel('Number
of motions','fontWeight','b');
end

%
% Optimization main program for constraint modification
% filename: optim_main_rev.m
% Called functions: optim_rev_mot
%
global combo_proc_orig wrench_proc
global grp_members grp_rev_type grp_srch_spc cp_rev_all
global combo_dup_idx_org remain_idx_combo

% Initialize variables for rating collection
Rating_all_rev_optim=[];WTR_optim_all=[];MRR_optim_all=[];MTR_opt
im_all=[];
tot_i=0; %counters for each for loop

clc
fprintf('Calculating optimization...n')

% Check for empty optimization variables in input file
if size(grp_members,1)==0
    disp('No variable constraints are specified. Optimization
terminated');
    return
end

% Load input file for constraints to be modified
[dummy3 dummy4 cp_rev_all]=find(grp_members');

%Find combo that contains cp_rev_all
del_idx=[];
for n=1:length(cp_rev_all)
    [row_idx, dum1, dum2]=find(combo==cp_rev_all(n));
    del_idx=[del_idx;row_idx];
end
del_idx=unique(del_idx); %This is the index of combo that
contains cp_rev_all

combo_red_idx=setdiff([1:size(combo,1)],del_idx); %This is the
combo index that needs to be kept
dup_idx= combo_dup_idx_org(combo_red_idx); %dup_idx is based on
mot_half, or combo_proc, not combo
dup_idx=unique(dup_idx);
%dup_idx identifies the motion in mot_all_org that is a duplicate
of the combo index that needs to be kept and therefore must not
be removed
dup_idx(1)=[]; %remove the zero that occurs on the first element,
non 5-rank combo

%Find combo_nondup that contains cp_rev_all and mark these to
remove
del_idx_all=[];
for n=1:length(cp_rev_all)
    [row_idx, dum1,
dum2]=find(combo_proc_org(:,2:6)==cp_rev_all(n));
    del_idx_all=[del_idx_all;row_idx];
end
del_idx_all=unique(del_idx_all);
%del_idx_all is the motion index that contain removed cp, but
then
%the ones that must be kept (dup_idx) must be subtracted from the
set
del_idx_nondup=setdiff(del_idx_all,dup_idx);

remain_idx=setdiff([1:no_mot_half],del_idx_nondup);
remain_idx_full=[remain_idx ;remain_idx+no_mot_half];

combo_proc_optimbase=combo_proc_org(remain_idx,2:6);
% input_wr_proc_optimbase=input_wr_proc_org(remain_idx);
% d_proc_optimbase=d_proc_org(remain_idx);
mot_half_optimbase=mot_half_org(remain_idx,:);
mot_all_optimbase=mot_all_org(remain_idx_full,:);
Ri_optimbase=Ri(remain_idx_full,:);

combo_new= combo(del_idx,:); %Identify original combo that will
need re-creation of motion

% create mapping for x depending on the dimension for each
variable
% the rows of x_map refer to the indices of x to grab
% x is a multi-dimension index for the optim parameter. It's
dimension is
% the same size as the number of cp being optimized
% Refer to the use of x_map in optim_rev.m
row=1;
x_map=zeros(size(grp_rev_type,1),2);
for i=1:size(grp_rev_type,1)

```

```

if grp_rev_type(i)==4 || grp_rev_type(i) ==6 ||
grp_rev_type(i) ==9
    x_map(i,:)=[row row+1];
    row=row+2;
else
    %dim=1
    x_map(i,:)=row 0];
    row=row+1;
end
%calc number of dimension in optimization search
no_dim=row-1;

tot_it=(no_step+1)^no_dim; %total iteration is # of variable *
no_step

% Start optimization factorial search here (allow up to 5D
search)

if no_dim==1

    ai=1:no_step+1; % ai is the index for search increments
    a=((ai-1)/no_step*2)-1; % a is the normalized search units
from -1 to 1
    prog_bar=waitbar(0,'Optimization iteration'); % display
progress bar

    for ai=1:no_step+1

        waitbar(ai/tot_it,prog_bar); % Updates progress bar
        x=[a(ai)]; % Assign the current variable value parameter
    %
        % Call the optim_rev to conduct analysis for current
increment
        [Rating_all_rev Ri_new_uniq
mot_all_new_uniq]=optim_rev(x,x_map,...

Ri_optimbase,mot_all_optimbase,mot_half_optimbase,
combo_proc_optimbase,combo_new);
        % Collect rating response values
        WTR_optim_all(ai)=Rating_all_rev(1);
        MRR_optim_all(ai)=Rating_all_rev(2);
        MTR_optim_all(ai)=Rating_all_rev(3);
    end
    close(prog_bar);
elseif no_dim==2
    ai=1:no_step+1;
    a=((ai-1)/no_step*2)-1;
    bi=1:no_step+1;
    b=((bi-1)/no_step*2)-1;
    prog_bar=waitbar(0,'Optimization iteration'); % display
progress bar
    for ai=1:no_step+1
        for bi=1:no_step+1
            x=[a(ai);b(bi)];
            waitbar(ai*bi/tot_it,prog_bar);
            [Rating_all_rev Ri_new_uniq
mot_all_new_uniq]=optim_rev(x,x_map,...

Ri_optimbase,mot_all_optimbase,mot_half_optimbase,
combo_proc_optimbase,combo_new);
            WTR_optim_all(ai,bi)=Rating_all_rev(1);
            MRR_optim_all(ai,bi)=Rating_all_rev(2);
            MTR_optim_all(ai,bi)=Rating_all_rev(3);
        end
    close(prog_bar);
elseif no_dim==3
    ai=1:no_step+1;
    a=((ai-1)/no_step*2)-1;
    bi=1:no_step+1;
    b=((bi-1)/no_step*2)-1;
    ci=1:no_step+1;
    c=((ci-1)/no_step*2)-1;
    for ai=1:no_step+1
        for bi=1:no_step+1
            for ci=1:no_step+1
                clc
                fprintf('Calculating optimization
iteration...\n');
                fprintf('Variable 1 iteration
(%i/%i)\n',ai,no_step+1);
                fprintf('Variable 2 iteration
(%i/%i)\n',bi,no_step+1);
                fprintf('Variable 3 iteration
(%i/%i)\n',ci,no_step+1);
                x=[a(ai);b(bi);c(ci)];
                [Rating_all_rev Ri_new_uniq
mot_all_new_uniq]=optim_rev(x,x_map,...

Ri_optimbase,mot_all_optimbase,mot_half_optimbase,
combo_proc_optimbase,combo_new);
                WTR_optim_all(ai,bi,ci)=Rating_all_rev(1);
                MRR_optim_all(ai,bi,ci)=Rating_all_rev(2);
                MTR_optim_all(ai,bi,ci)=Rating_all_rev(3);
            end
        end
    end
elseif no_dim>5
    ai=1:no_step+1;
    a=((ai-1)/no_step*2)-1;
    bi=1:no_step+1;
    b=((bi-1)/no_step*2)-1;
    ci=1:no_step+1;
    c=((ci-1)/no_step*2)-1;
    di=1:no_step+1;
    d=((di-1)/no_step*2)-1;
    for ai=1:no_step+1
        for bi=1:no_step+1
            for ci=1:no_step+1
                for di=1:no_step+1
                    clc
                    fprintf('Calculating optimization
iteration...\n');
                    fprintf('Variable 1 iteration
(%i/%i)\n',ai,no_step+1);
                    fprintf('Variable 2 iteration
(%i/%i)\n',bi,no_step+1);
                    fprintf('Variable 3 iteration
(%i/%i)\n',ci,no_step+1);
                    fprintf('Variable 4 iteration
(%i/%i)\n',di,no_step+1);
                    x=[a(ai);b(bi);c(ci);d(di)];
                    [Rating_all_rev Ri_new_uniq
mot_all_new_uniq]=optim_rev(x,x_map,...

Ri_optimbase,mot_all_optimbase,mot_half_optimbase,
combo_proc_optimbase,combo_new);
                    WTR_optim_all(ai,bi,ci,di)=Rating_all_rev(1);
                    MRR_optim_all(ai,bi,ci,di)=Rating_all_rev(2);
                    MTR_optim_all(ai,bi,ci,di)=Rating_all_rev(3);
                end
            end
        end
    end
end

```

```

    disp('Total dimension to optimize exceeds limit');
    return
end

TOR_optim_all=MTR_optim_all./MRR_optim_all;
WTR_optim_chg=(WTR_optim_all-
Rating_all_org(1)./Rating_all_org(1).*100;
MRR_optim_chg=(MRR_optim_all-
Rating_all_org(2)./Rating_all_org(2).*100;
MTR_optim_chg=(MTR_optim_all-
Rating_all_org(3)./Rating_all_org(3).*100;
TOR_optim_chg=(TOR_optim_all-
Rating_all_org(4)./Rating_all_org(4).*100;



---


function [Rating_all_rev Ri_new_uniq
mot_all_new_uniq]=optim_rev(x,x_map, ...
    Ri_optimbase,mot_all_optimbase,mot_half_optimbase,
    combo_proc_optimbase,combo_new)
% Constraint modification routine called in nested for-loop from
optim_main_rev
% filename: optim_rev.m
% Called functions: move_pt_srch, move_lin_srch,
move_circlin_srch, move_pn_srch
%           orient1d_srch, orient2d_srch,
line_orient1d_srch, resize_lin_srch
%           resize_rectpln_srch, resize_circpln_srch,
cp_rev_to_wrench
%           rate_motset, main_loop, rating
%
% global cp cpin clin cpln cpln_prop
global cp_rev cpin_rev clin_rev cpln_rev cpln_prop_rev
global grp_members grp_rev_type grp_srch_spc cp_rev_all
global wr_all
global pts_rev max_d_rev
global combo_dup_idx_org
cp_rev=cp; cpin_rev=cpin; clin_rev=clin;
cpln_rev=cpln; cpln_prop_rev=cpln_prop;

% Relocate and reorient modified constraints according to rev
type
% 1 - move location, point search
% 2 - move location, straight line search
% 3 - move location, curved line search
% 4 - move location, plane search
% 5 - reorient normal, 1D angular search
% 6 - reorient normal, 2D angular search
% 7 - reorient line, 1D angular search
% 8 - resize line length
% 9,10 - resize plane length and width for rectangular plane and
circular plane

% This for loop is based on the number of group. Linked
constraints are done within the subroutines
for i=1:size(grp_members,1)
    % i is the group #
    rev_type=grp_rev_type(i);
    cp_rev_in_group=nonzeros(grp_members(i,:));
    if rev_type==1
        clc
        disp('THE POINT REV_TYPE IS PRELIMINARY AND MOST LIKELY
CONTAIN ERRORS')
        return
    % Right now the dimensions of pt_srch is still only for
one constraint
    % It also does not allow two constraints cause they can
occupy the same search space x_grp=x(x_map(i,1));
    % move_pt_srch(x_grp,cp_rev_in_group,grp_srch_spc);
    elseif rev_type==2
        x_grp=x(x_map(i,1));
        move_lin_srch(x_grp,cp_rev_in_group,grp_srch_spc(i,:));
    elseif rev_type==3
        x_grp=x(x_map(i,1));
    elseif rev_type==4
        x_grp=x(x_map(i,1:2));
        move_pn_srch(x_grp,cp_rev_in_group,grp_srch_spc(i,:));
    elseif rev_type==5
        x_grp=x(x_map(i,1));
        orient1d_srch(x_grp,cp_rev_in_group, grp_srch_spc(i,:));
    elseif rev_type==6
        x_grp=x(x_map(i,1:2));
        orient2d_srch(x_grp,cp_rev_in_group, grp_srch_spc(i,:));
    elseif rev_type==7
        x_grp=x(x_map(i,1));
        line_orient1d_srch(x_grp,cp_rev_in_group,
        grp_srch_spc(i,:));
    elseif rev_type==8
        x_grp=x(x_map(i,1));
        resize_rectpln_srch(x_grp,cp_rev_in_group,
        grp_srch_spc(i,:));
    elseif rev_type==9
        x_grp=x(x_map(i,1));
        resize_circpln_srch(x_grp,cp_rev_in_group,
        grp_srch_spc(i,:));
    elseif rev_type==10
        x_grp=x(x_map(i,1));
        resize_circpln_srch(x_grp,cp_rev_in_group,
        grp_srch_spc(i,:));
    end
end



---


% Transform revised cp into revised wrenches
[wrl_all_new_wrench]=cp_rev_to_wrench(wr_all,cp_rev_all, ...
    cp_rev, cpin_rev, clin_rev, cpln_rev, cpln_prop_rev);

%mot_all_optimbase is duplex mot format

% Re-calculate cp_rev column in rating with old motion
[R_recalc]=rate_motset(combo_proc_optimbase, ...
    mot_half_optimbase,cp_rev_all,'revised');
Ri_recalc=1./R_recalc;

% Substitute revised CP columns in the rating matrix
Ri_optimbase_recalc=Ri_optimbase;
Ri_optimbase_recalc(:,cp_rev_all)=Ri_recalc;

% Re-create combinations that contain cp_rev
% use mainloop function
dispbar=0; % disable progress bar
set='revised';
[mot_all_add R_add dmy1 dmy2
dmy3]=main_loop(combo_new,wr_all_new,dispbar,set);
Ri_add=1./R_add;
mot_all_add_rev=[mot_all_add(:,1:6) mot_all_add(:,7:10)];

% Merge reduced old rating with additional rating
Ri_new=[Ri_optimbase_recalc;Ri_add];
mot_all_new=[mot_all_optimbase;mot_all_add;mot_all_add_rev];
[mot_all_new_uniq uniq_idx]=unique(mot_all_new,'rows','first');
Ri_new_uniq=Ri_new(uniq_idx,:);

% Calculate new rating
[Rating_all_rev WTR_idx_rev free_mot_rev free_mot_idx_rev
best_cp_rev]=rating(Ri_new_uniq, mot_all_new_uniq);



---


% Main processor for constraint reduction
% filename: optim_main_red.m
%
clc;fprintf('Calculating optimization...\n');

% Initialize variables
WTR_optim_all=[];MRR_optim_all=[];MTR_optim_all=[];
del_idx=[];

% Create combination scheme, especially when removing more than
one constraints at a time
cp_del_comb=nchoosek(1:total_cp,no_red);

prog_bar=waitbar(0,'Optimization iteration progress'); % display
progress bar

for a=1:size(cp_del_comb,1)
    % Update progress bar
    if size(cp_del_comb,1)<=1000
        waitbar(a/size(cp_del_comb,1),prog_bar);
    else
        if mod(a,50)==0
            waitbar(a/size(cp_del_comb,1),prog_bar);
        end
    end
    cp_del_idx=cp_del_comb(a,:); % This is the cp that needs to be
removed

    % Find combo that contains cp_del_idx
    del_idx=[];
    for n=1:length(cp_del_idx)
        [row_idx, dum1, dum2]=find(combo==cp_del_idx(n));
        del_idx=[del_idx;row_idx];
    end
    del_idx=unique(del_idx); % This is the index of combo that
contains cp_del_idx

    combo_red_idx=setdiff([1:size(combo,1)],del_idx); % This is
the combo index that needs to be kept
    dup_idx= combo_dup_idx(combo_red_idx);
    dup_idx=unique(dup_idx);

```

```

%dup_idx identifies the motion in mot_all_org that is a
duplicate of the combo index that needs to be kept and therefore
must not be removed
dup_idx(1)=[];
%remove the zero that occurs on the first
element, non 5-rank combo

%Find combo_nondup that contains cp_del_idx and mark these to
remove
del_idx_all=[];
for n=1:length(cp_del_idx)
    [row_idx, dum1,
    dum2]=find(combo_proc_org(:,2:6)==cp_del_idx(n));
    del_idx_all=[del_idx_all;row_idx];
end
%del_idx_all is the motion index that contain removed cp,
del_idx_all=unique(del_idx_all);
% but then the ones that must be kept (dup_idx) must be
subtracted from the set
del_idx_nondup=setdiff(del_idx_all,dup_idx);

remain_idx=setdiff([1:no_mot_half],del_idx_nondup);
remain_idx_full=[remain_idx ;remain_idx+no_mot_half];

combo_proc_red=combo_proc_org(remain_idx,2:6);
Ri_red=Ri(remain_idx_full,:);
Ri_red(:,cp_del_idx)=[];%delete column for deleted cp
mot_all_red=mot_all_org(remain_idx_full,:);

[mot_all_red_uniq uniq_idx]=unique(mot_all_red,'rows');
Ri_red_uniq=Ri(red(uniq_idx,:));

% Calculate rating after constraint reduction
[Rating_all_rev WTR_idx_rev free_mot_rev free_mot_idx_rev
best_cp_rev...]
]=rating(Ri_red_uniq, mot_all_red_uniq);
WTR_optim_all(a)=Rating_all_rev(1);
MRR_optim_all(a)=Rating_all_rev(2);
MTR_optim_all(a)=Rating_all_rev(3);
end

TOR_optim_all=MTR_optim_all./MRR_optim_all;
close(prog_bar); %close the progress bar

% t is constraint removal index
t=1:size(cp_del_comb,1);

% Calculate rating change
WTR_optim_chg=(WTR_optim_all-
Rating_all_org(1))./Rating_all_org(1).*100;
MRR_optim_chg=(MRR_optim_all-
Rating_all_org(2))./Rating_all_org(2).*100;
MTR_optim_chg=(MTR_optim_all-
Rating_all_org(3))./Rating_all_org(3).*100;
TOR_optim_chg=(TOR_optim_all-
Rating_all_org(4))./Rating_all_org(4).*100;

% Plot constraint reduction
f2=figure('Position',[100,300,640,480]);
title('Rating Change (%) Due to Constraint Reduction');
subplot(2,2,1);plot(t,WTR_optim_chg,'LineWidth',1.25);ylabel('WTR
Change (%)'); xlabel('Constraint Removal Index');grid on;
subplot(2,2,2);plot(t,MRR_optim_chg,'LineWidth',1.25);ylabel('MRR
Change (%)'); xlabel('Constraint Removal Index');grid on;
subplot(2,2,3);plot(t,MTR_optim_chg,'LineWidth',1.25);ylabel('MTR
Change (%)'); xlabel('Constraint Removal Index');grid on;
subplot(2,2,4);plot(t,TOR_optim_chg,'LineWidth',1.25);ylabel('TOR
Change (%)'); xlabel('Constraint Removal Index');grid on;

% Identify constraint removal combination that yield maximum TOR
increase
[max_tor_increase b]=max(TOR_optim_chg)
cp_del_comb(b,:);
ratings_at_TOR_increase=[WTR_optim_all(b) MRR_optim_all(b)
MTR_optim_all(b)];

% Automatically save the figures
saveas(f2,inputfile,'fig')
saveas(f2,inputfile,'eps')



---


function optim_postproc(no_step,no_dim, WTR_optim_chg,
MRR_optim_chg,...)
    MTR_optim_chg, TOR_optim_chg,inputfile)
% Plot the response surface plots
% filename: optim_postproc.m
% Input variables: no_step,no_dim, WTR_optim_chg, MRR_optim_chg,
MTR_optim_chg, TOR_optim_chg,inputfile
% Output variables: -
% Called functions: -
%
% Copyright 2008 Leonard Rusli
% The Ohio State University

% Prepare x and y axis array
x1_inc=-1:2/no_step:1;
x2_inc=-1:2/no_step:1;

if no_dim==1
    t=x1_inc;
    f2=figure('Position',[100,300,640,480]);

    subplot(2,2,1);plot(t,WTR_optim_chg,'LineWidth',2);xlabel('X1');y
label('WTR Change (%)');grid on;
    subplot(2,2,2);plot(t,MRR_optim_chg,'LineWidth',2);xlabel('X1');y
label('MRR Change (%)');grid on;
    subplot(2,2,3);plot(t,MTR_optim_chg,'LineWidth',2);xlabel('X1');y
label('MTR Change (%)');grid on;
    subplot(2,2,4);plot(t,TOR_optim_chg,'LineWidth',2);xlabel('X1');y
label('TOR Change (%)');grid on;
    % Automatically save the figures as fig and eps format
    saveas(f2,inputfile,'fig')
    saveas(f2,inputfile,'eps')

elseif no_dim==2
    [u,v]=meshgrid(x1_inc,x2_inc);
    f2=figure('Position',[100,300,640,480]);
    subplot(2,2,1);surf(u,v,WTR_optim_chg);colormap cool;
    xlabel('WTR Change (%)');axis xy;
    xlabel('X2');
    ylabel('X1');
    subplot(2,2,2);surf(u,v,MRR_optim_chg);
    xlabel('MRR Change (%)');axis xy;
    xlabel('X2');
    ylabel('X1');
    subplot(2,2,3);surf(u,v,MTR_optim_chg);
    xlabel('MTR Change (%)');axis xy;
    xlabel('X2');
    ylabel('X1');
    subplot(2,2,4);surf(u,v,TOR_optim_chg);
    xlabel('TOR Change (%)');axis xy;
    xlabel('X2');
    ylabel('X1');

    saveas(f2,inputfile,'fig')
    saveas(f2,inputfile,'eps')

elseif no_dim==3
    % Identify optimums based on different metric
    WTR_max=max(WTR_optim_all(:));
    WTR_max_idx=find(WTR_optim_all(:)==WTR_max);
    [WTR_max_x1 WTR_max_x2
WTR_max_x3]=ind2sub(size(WTR_optim_all),WTR_max_idx);

    MRR_max=max(MRR_optim_all(:));
    MRR_max_idx=find(MRR_optim_all(:)==MRR_max);
    [MRR_max_x1 MRR_max_x2
MRR_max_x3]=ind2sub(size(MRR_optim_all),MRR_max_idx);

    MTR_max=max(MTR_optim_all(:));
    MTR_max_idx=find(MTR_optim_all(:)==MTR_max);
    [MTR_max_x1 MTR_max_x2
MTR_max_x3]=ind2sub(size(MTR_optim_all),MTR_max_idx);

    TOR_max=max(TOR_optim_all(:));
    TOR_max_idx=find(TOR_optim_all(:)==TOR_max);
    [TOR_max_x1 TOR_max_x2
TOR_max_x3]=ind2sub(size(TOR_optim_all),TOR_max_idx);

    % Identify ratings at optimum point based on WTR optimum
    WTR_global_abs=WTR_optim_all(WTR_max_x1(1),
WTR_max_x2(1),WTR_max_x3(1));
    WTR_global_chg=WTR_optim_chg(WTR_max_x1(1),
WTR_max_x2(1),WTR_max_x3(1));
    MRR_global_abs=MRR_optim_all(WTR_max_x1(1),
WTR_max_x2(1),WTR_max_x3(1));
    MRR_global_chg=MRR_optim_chg(WTR_max_x1(1),
WTR_max_x2(1),WTR_max_x3(1));
    MTR_global_abs=MTR_optim_all(WTR_max_x1(1),
WTR_max_x2(1),WTR_max_x3(1));
    MTR_global_chg=MTR_optim_chg(WTR_max_x1(1),
WTR_max_x2(1),WTR_max_x3(1));
    TOR_global_abs=TOR_optim_all(WTR_max_x1(1),
WTR_max_x2(1),WTR_max_x3(1));
    TOR_global_chg=TOR_optim_chg(WTR_max_x1(1),
WTR_max_x2(1),WTR_max_x3(1));

elseif no_dim==4
    WTR_max=max(WTR_optim_all(:));
    WTR_max_idx=find(WTR_optim_all(:)==WTR_max);
    [WTR_max_x1 WTR_max_x2 WTR_max_x3
WTR_max_x4]=ind2sub(size(WTR_optim_all),WTR_max_idx);

    MRR_max=max(MRR_optim_all(:));
    MRR_max_idx=find(MRR_optim_all(:)==MRR_max);

```

```

[MRR_max_x1 MRR_max_x2 MRR_max_x3
MRR_max_x4]=ind2sub(size(MRR_optim_all),MRR_max_idx);

MTR_max=max(MTR_optim_all(:));
MTR_max_idx=find(MTR_optim_all(:)==MTR_max);
[MTR_max_x1 MTR_max_x2 MTR_max_x3
MTR_max_x4]=ind2sub(size(MTR_optim_all),MTR_max_idx);

TOR_max=max(TOR_optim_all(:));
TOR_max_idx=find(TOR_optim_all(:)==TOR_max);
[TOR_max_x1 TOR_max_x2 TOR_max_x3
TOR_max_x4]=ind2sub(size(TOR_optim_all),TOR_max_idx);

WTR_global_abs=WTR_optim_all(WTR_max_x1(1),
WTR_max_x2(1),WTR_max_x3(1),WTR_max_x4(1));
WTR_global_chg=WTR_optim_chg(WTR_max_x1(1),
WTR_max_x2(1),WTR_max_x3(1),WTR_max_x4(1));
MRR_global_abs=MRR_optim_all(WTR_max_x1(1),
WTR_max_x2(1),WTR_max_x3(1),WTR_max_x4(1));
MRR_global_chg=MRR_optim_chg(WTR_max_x1(1),
WTR_max_x2(1),WTR_max_x3(1),WTR_max_x4(1));
WTR_global_abs=MTR_optim_all(WTR_max_x1(1),
WTR_max_x2(1),WTR_max_x3(1),WTR_max_x4(1));
WTR_global_chg=MTR_optim_chg(WTR_max_x1(1),
WTR_max_x2(1),WTR_max_x3(1),WTR_max_x4(1));
TOR_global_abs=TOR_optim_all(WTR_max_x1(1),
WTR_max_x2(1),WTR_max_x3(1),WTR_max_x4(1));
TOR_global_chg=TOR_optim_chg(WTR_max_x1(1),
WTR_max_x2(1),WTR_max_x3(1),WTR_max_x4(1));

elseif no_dim==5
    WTR_max=max(WTR_optim_all(:));
    WTR_max_idx=find(WTR_optim_all(:)==WTR_max);
    [WTR_max_x1 WTR_max_x2 WTR_max_x3 WTR_max_x4
WTR_max_x5]=ind2sub(size(WTR_optim_all),WTR_max_idx);

    MRR_max=max(MRR_optim_all(:));
    MRR_max_idx=find(MRR_optim_all(:)==MRR_max);
    [MRR_max_x1 MRR_max_x2 MRR_max_x3 MRR_max_x4
MRR_max_x5]=ind2sub(size(MRR_optim_all),MRR_max_idx);

    MTR_max=max(MTR_optim_all(:));
    MTR_max_idx=find(MTR_optim_all(:)==MTR_max);
    [MTR_max_x1 MTR_max_x2 MTR_max_x3 MTR_max_x4
MTR_max_x5]=ind2sub(size(MTR_optim_all),MTR_max_idx);

    TOR_max=max(TOR_optim_all(:));
    TOR_max_idx=find(TOR_optim_all(:)==TOR_max);
    [TOR_max_x1 TOR_max_x2 TOR_max_x3 TOR_max_x4
TOR_max_x5]=ind2sub(size(TOR_optim_all),TOR_max_idx);

    WTR_global_abs=WTR_optim_all(WTR_max_x1(1),
WTR_max_x2(1),WTR_max_x3(1),WTR_max_x4(1),WTR_max_x5(1));
    WTR_global_chg=WTR_optim_chg(WTR_max_x1(1),
WTR_max_x2(1),WTR_max_x3(1),WTR_max_x4(1),WTR_max_x5(1));
    MRR_global_abs=MRR_optim_all(WTR_max_x1(1),
WTR_max_x2(1),WTR_max_x3(1),WTR_max_x4(1),WTR_max_x5(1));
    MRR_global_chg=MRR_optim_chg(WTR_max_x1(1),
WTR_max_x2(1),WTR_max_x3(1),WTR_max_x4(1),WTR_max_x5(1));
    MTR_global_abs=MTR_optim_all(WTR_max_x1(1),
WTR_max_x2(1),WTR_max_x3(1),WTR_max_x4(1),WTR_max_x5(1));
    MTR_global_chg=MTR_optim_chg(WTR_max_x1(1),
WTR_max_x2(1),WTR_max_x3(1),WTR_max_x4(1),WTR_max_x5(1));
    TOR_global_abs=TOR_optim_all(WTR_max_x1(1),
WTR_max_x2(1),WTR_max_x3(1),WTR_max_x4(1),WTR_max_x5(1));
    TOR_global_chg=TOR_optim_chg(WTR_max_x1(1),
WTR_max_x2(1),WTR_max_x3(1),WTR_max_x4(1),WTR_max_x5(1));

end



---


function []=table_mot(result, mot_list, TR)
% Create a table for screw axis motions
% filename: table_mot.m
% Input variables: result, mot_list, TR
% Output variables: -
% Called functions: -

fprintf(result,'<TABLE BORDER=2>\n');
fprintf(result,'<b> <tr><th>Om(x)</th> <th>Om(y)</th>
<th>Om(z)</th> <th>Mu(x)</th> <th>Mu(y)</th> <th>Mu(z)</th>
<th>Rho(x)</th> <th>Rho(y)</th> <th>Rho(z)</th> <th>Pitch</th>
<th>Total Resistance</th> <br></b>\n');

for i=1:size(mot_list,1)
    fprintf(result,'<tr><td>%7.4f</td> <td>%7.4f</td>
<td>%7.4f</td> <td>%7.4f</td> <td>%7.4f</td> <td>%7.4f</td>
<td>%7.4f</td> <td>%7.4f</td> <td>%7.4f</td> <td>%5.4f</td>
',...
        mot_list(i,:));
    fprintf(result,'<td>%4f</td></tr>\n',TR(i));
end
fprintf(result,'</TABLE>\n');



---


% Sensitivity analysis by perturbing constraint location
% filename: sens_analysis_pos.m
%
% This procedure uses optim_main_rev plane search space to
explore position perturbation

pert_dist=input('How much position perturbation? (same units as
input file) ');

for idx=1:total_cp

% Set up plane search space
grp_members=idx;
grp_rev_type=4;
if idx<=no_cp
    k=idx;
    cp_crp=cpr(k,1:3);cp_normal=cpr(k,4:6);
elseif idx>no_cp & idx<=no_cp+no_cpin
    k=idx-no_cp;
    cp_crp=cpr(k,1:3);cp_normal=cpr(k,4:6);
elseif idx>no_cp+no_cpin & idx<=no_cp+no_cpin+no_clin
    k=idx-(no_cp+no_cpin);
    cp_crp=cpr(k,1:3);cp_normal=cpr(k,7:9);
elseif idx>no_cp+no_cpin+no_clin & idx<no_cp+no_cpin+no_cpin
    k=idx-(no_cp+no_cpin+no_cpin);
    cp_crp=cpr(k,1:3);cp_normal=cpr(k,4:6);
end
xy=null(cp_normal);
grp_srcn_spc=[cp_crp xy(:,1)' pert_dist xy(:,2)' pert_dist];
no_step=2;
optim_main_rev
% Collect rating change
SAP_WTR(idx,:,:)=WTR_optim_chg;
SAP_MRR(idx,:,:)=MRR_optim_chg;
SAP_MTR(idx,:,:)=MTR_optim_chg;
SAP_TOR(idx,:,:)=TOR_optim_chg;

end

% Optional plots
k=input('Plot which constraint''s perturbation? (enter 0 to exit)
');

while k~=0 & k<=total_cp
    WTR=zeros(3,3);MRR=zeros(3,3);MTR=zeros(3,3);TOR=zeros(3,3);
    WTR(:,:,1)=SAP_WTR(k,:,:);MRR(:,:,1)=SAP_MRR(k,:,:);MTR(:,:,1)=SAP_MTR(k,:,:);TOR(:,:,1)=SAP_TOR(k,:,:);
    optim_postproc(no_step,no_dim,
    WTR,MRR,MTR,TOR,Rating_all_org,'sens_plot');
    k=input('Plot which constraint''s perturbation? (enter 0 to
exit) ');
end

for m=1:total_cp
    WTR=zeros(3,3);
    WTR(:,:,1)=SAP_WTR(m,:,:);
    SAP_WTR_min(m)=min(min(WTR));
end



---


% Sensitivity analysis by perturbing constraint orientation
% filename:sens_analysis_orient.m
%
% This procedure uses optim_main_rev orient2d search space to
explore orientation perturbation

pert_angle=input('How much angle perturbation? (same units as
input file) ');

for idx=1:total_cp

% Set up orient2d search space
grp_members=idx;
grp_rev_type=6;
if idx<=no_cp
    k=idx;
    cp_normal=cpr(k,4:6);
elseif idx>no_cp & idx<=no_cp+no_cpin
    k=idx-no_cp;
    cp_normal=cpr(k,4:6);
elseif idx>no_cp+no_cpin & idx<=no_cp+no_cpin+no_clin
    k=idx-(no_cp+no_cpin);
    cp_normal=cpr(k,7:9);
elseif idx>no_cp+no_cpin+no_clin & idx<no_cp+no_cpin+no_cpin
    k=idx-(no_cp+no_cpin+no_cpin);
    cp_normal=cpr(k,4:6);
end
xy=null(cp_normal);
grp_srcn_spc=[xy(:,1)' xy(:,2)' pert_angle pert_angle];
no_step=2;
optim_main_rev
% Collect rating change
SAO_WTR(idx,:,:)=WTR_optim_chg;
SAO_MRR(idx,:,:)=MRR_optim_chg;
SAO_MTR(idx,:,:)=MTR_optim_chg;
SAO_TOR(idx,:,:)=TOR_optim_chg;

```

```

end
% Optional plots
k=input('Plot which constraint''s perturbation? (enter 0 to exit)');
while k~=0 & k<=total_cp
    WTR=zeros(3,3);MRR=zeros(3,3);MTR=zeros(3,3);TOR=zeros(3,3);
    WTR(:,:,1)=SAO_WTR(k,:,1);MRR(:,:,1)=SAO_MRR(k,:,1);MTR(:,:,1)=SAO_MTR(
    k,:,1);TOR(:,:,1)=SAO_TOR(k,:,1);
    optim_postproc(no_step,no_dim,WTR,MRR,MTR,TOR,'sens_plot');
    k=input('Plot which constraint''s perturbation? (enter 0 to exit)');
end
for m=1:total_cp
    WTR=zeros(3,3);
    WTR(:,:,1)=SAO_WTR(m,:,1);
    SAO_WTR_min(m)=min(min(WTR));
end


---


% Post-process sensitivity analysis results
% filename: sens_analysis_postproc.m
%
% Using average
for k=1:total_cp
    WTR(:,:,1)=SAO_WTR(k,:,1);MRR(:,:,1)=SAO_MRR(k,:,1);MTR(:,:,1)=SAO_MTR(
    k,:,1);TOR(:,:,1)=SAO_TOR(k,:,1);
    WTR_avg(k)=mean(mean(WTR));
    MRR_avg(k)=mean(mean(MRR));
    MTR_avg(k)=mean(mean(MTR));
    TOR_avg(k)=mean(mean(TOR));
    WTR(:,:,2)=SAP_WTR(k,:,2);MRR(:,:,2)=SAP_MRR(k,:,2);MTR(:,:,2)=SAP_MTR(
    k,:,2);TOR(:,:,2)=SAP_TOR(k,:,2);
    WTRP_avg(k)=mean(mean(WTR));
    MRP_avg(k)=mean(mean(MRR));
    MTRP_avg(k)=mean(mean(MTR));
    TORP_avg(k)=mean(mean(TOR));
end
[WTR_min WTR_min_idx]=min(WTR_avg)
[WTRP_min WTRP_min_idx]=min(WTRP_avg)

% Using worst change
for k=1:total_cp
    WTR(:,:,1)=SAO_WTR(k,:,1);MRR(:,:,1)=SAO_MRR(k,:,1);MTR(:,:,1)=SAO_MTR(
    k,:,1);TOR(:,:,1)=SAO_TOR(k,:,1);
    WTR_avg(k)=min(min(WTR));
    MRR_avg(k)=min(min(MRR));
    MTR_avg(k)=min(min(MTR));
    TOR_avg(k)=min(min(TOR));
    WTR(:,:,2)=SAP_WTR(k,:,2);MRR(:,:,2)=SAP_MRR(k,:,2);MTR(:,:,2)=SAP_MTR(
    k,:,2);TOR(:,:,2)=SAP_TOR(k,:,2);
    WTRP_avg(k)=min(min(WTR));
    MRP_avg(k)=min(min(MRR));
    MTRP_avg(k)=min(min(MTR));
    TORP_avg(k)=min(min(TOR));
end
x1_inc=-1:2/no_step:1;
x2_inc=-1:2/no_step:1;
[u,v]=meshgrid(x1_inc,x2_inc);
WTRP(:,:,1)=SAP_WTR(WTRP_min_idx,:,:);
WTR(:,:,1)=SAO_WTR(WTR_min_idx,:,:);

figure;
surf(u,v,WTR);colormap cool;
zlabel('WTR Change (%)','fontweight','b');axis xy;
xlabel('X2','fontweight','b');
ylabel('X1','fontweight','b');

figure;
surf(u,v,WTRP);colormap cool;
zlabel('WTR Change (%)','fontweight','b');axis xy;
xlabel('X2','fontweight','b');
ylabel('X1','fontweight','b');



---


% Known loading condition analysis for original set of
constraints
% filename: main_specmot_orig.m
% Called functions: input_wr_compose, rate_cp, rate_cpin,
rate_clin, rate_cpln1, rate_cpln2, rating
%
global cp cpin clin cpln cpln_prop no_cp no_cpin no_clin no_cpln
global pts max_d d_proc wrench_proc input_wr_proc

% The procedure here is similar to main_loop, but only for a
specified motion.
% See comments in main_loop.m

```

```

Rcp_pos=[];Rcp_neg=[];Rcpin=[];
Rclin_pos=[];Rclin_neg=[];
RcpIn_pos=[];RcpIn_neg=[];
no_specmot=size(specmot,1);
for m=1:no_specmot
    %reverse find imaginary pivot constraints
    omu=specmot(m,1:3)./norm(specmot(m,1:3));rho=specmot(m,4:6);h=spec
    mot(m,7);
    if h~=-inf
        mu=h*omu+cross(rho,omu);
    else
        mu=omu;
        omu=[0 0 0];
    end
    mot=[omu mu rho h];
    rec_mot=[mu omu];
    pivot_wr=null(rec_mot)';
    [input_wr d]=input_wr_compose(mot,pts,max_d);
    %Rate all constraints
    %Can use pivot_wr as react_wr_5 because everything is with
respect to
    %origin
    for j = 1:no_cp
        [Rcp_pos(m,j) Rcp_neg(m,j)]=rate_cp(mot,pivot_wr,
        input_wr, cp(j,:));
    end
    for j = 1:no_cpin
        [RcpIn_pos(m,j)]=rate_cpin(mot,pivot_wr,input_wr,cpin(j,:));
    end
    for j = 1:no_clin
        [Rclin_pos(m,j)]=rate_clin(mot,pivot_wr,input_wr,clin(j,:));
    end
    for j = 1:no_cpln
        if cpin(j,7)==1
            [RcpIn_pos(m,j)]=rate_cpln1(mot,pivot_wr,
            input_wr, cpin(j,:),cpln_prop(j,:));
        elseif cpin(j,7)==2
            [RcpIn_pos(m,j)]=rate_cpln2(mot,pivot_wr,
            input_wr, cpin(j,:),cpln_prop(j,:));
        end
        d_proc(m,1)=d;
        wrench_proc{m}=pivot_wr;
        input_wr_proc{m}=input_wr;
    end
    Rcp=[Rcp_pos;Rcp_neg]; % Merge resistance value for fwd and rev
motion for cp
    RcpIn=[RcpIn_pos;RcpIn_neg]; % Resistance values for fwd and rev motion
is the same for cpin
    Rclin=[Rclin_pos;Rclin_neg];
    RcpIn=[RcpIn_pos;RcpIn_neg];
    R=[Rcp RcpIn Rclin RcpIn];
    Ri_specmot=1./;
    Ri_specmot=(round(Ri_specmot.*1e4)).*1e-4
    rowsum_specmot=sum(Ri_specmot,2)
    specmot_rev=[-specmot(:,1:3) specmot(4:7)];
    mot_proc=[specmot;specmot_rev]
    [Rating_all_org WTR_idx_org free_mot free_mot_idx best_cp
rowsum]=rating(Ri_specmot, mot_proc);


---


% Known loading condition optimization
% filename: main_specmot_optim.m
% Called functions: rate_specmot
%
global cp cpin clin cpln cpln_prop no_cp no_cpin no_clin no_cpln
global cp_rev cpin_rev clin_rev cpln_rev cpln_prop_rev
global pts max_d d_proc wrench_proc input_wr_proc
global grp_members grp_rev_type grp_srch_spc cp_rev_all

% The procedure here is similar to optim_main_rev, but we are
only evaluating a specific motion

no_step=10; % default step size
WTR_optim_specmot=[]; MRR_optim_specmot=[]; MTR_optim_specmot=[];
prog_bar=waitbar(0, 'Optimization iteration progress'); % display
progress bar

% Mapping parameter x
row=1;

```

```

x_map=zeros(size(grp_rev_type,1),2);
for i=1:size(grp_rev_type,1)
    if grp_rev_type(i)==4 || grp_rev_type(i) ==6 ||
    grp_rev_type(i) ==9
        %dim=2
        x_map(i,:)=[row row+1];
        row=row+2;
    else
        %dim=1
        x_map(i,:)=[row 0];
        row=row+1;
    end
end

% Count number of dimension
no_dim=row-1;
tot_i=0; %counters for each for loop
tot_it=(no_step+1)^no_dim; %total iteration is # of variable *
no_step

[dummy3 dummy4 cp_rev_all]=find(grp_members');

ai=1;
for a=1:2/no_step:1
    bi=1;
    for b=1:2/no_step:1
        if no_dim<2
            break
        elseif no_dim==2
            x=[a;b];
            waitbar(tot_i/tot_it,prog_bar);

            % Calculate constraints quality for each increment in
            the search space
            [Rating_all_rev Ri_specmot
            mot_specmot_all]=rate_specmot(x,x_map,specmot);

            % Collect ratings
            WTR_optim_specmot(ai,bi)=Rating_all_rev(1);
            MRR_optim_specmot(ai,bi)=Rating_all_rev(2);
            MTR_optim_specmot(ai,bi)=Rating_all_rev(3);
            TOR_optim_specmot(ai,bi)=Rating_all_rev(4);

            Ri_specmot_col(:,:,ai,bi)=Ri_specmot;
            Ri_specmot_rowsum(:,:,ai,bi)=sum(Ri_specmot,2);
            mot_specmot_col(:,:,ai,bi)=mot_specmot_all;
            tot_i=tot_i+1;
        end
        bi=bi+1;
    end
    if no_dim<1
        break
    elseif no_dim==1
        x=a;
        waitbar(tot_i/tot_it,prog_bar);

        [Rating_all_rev Ri_specmot
        mot_specmot_all]=rate_specmot(x,x_map,specmot);

        WTR_optim_specmot(ai)=Rating_all_rev(1);
        MRR_optim_specmot(ai)=Rating_all_rev(2);
        MTR_optim_specmot(ai)=Rating_all_rev(3);
        TOR_optim_specmot(ai)=Rating_all_rev(4);

        Ri_specmot_col(:,:,ai)=Ri_specmot;
        Ri_specmot_rowsum(:,:,ai)=sum(Ri_specmot,2);
        mot_specmot_col(:,:,ai)=mot_specmot_all;
        tot_i=tot_i+1;
    end
    ai=ai+1;
end
close(prog_bar); %close the progress bar
TOR_optim_specmot=MTR_optim_specmot./MRR_optim_specmot;

% The plot procedure below is similar to optim_postproc. See
% comments in optim_postproc.m
x1_inc=1:2/no_step:1;
x2_inc=1:2/no_step:1;

if no_dim==1
    t=x1_inc;

    f2=figure('Position',[100,300,640,480]);
    subplot(2,2,1);plot(t,WTR_optim_specmot,'LineWidth',2); xlabel('X1
    ','fontweight','b'); ylabel('WTR','fontweight','b'); grid on;

    subplot(2,2,2);plot(t,MRR_optim_specmot,'LineWidth',2); xlabel('X1
    ','fontweight','b'); ylabel('MRR','fontweight','b'); grid on;

    subplot(2,2,3);plot(t,MTR_optim_specmot,'LineWidth',2); xlabel('X1
    ','fontweight','b'); ylabel('MTR','fontweight','b'); grid on;

    subplot(2,2,4);plot(t,TOR_optim_specmot,'LineWidth',2); xlabel('X1
    ','fontweight','b'); ylabel('TOR','fontweight','b'); grid on;

    saveas(f2,inputfile,'fig')
    saveas(f2,inputfile,'eps')

elseif no_dim==2
    [u,v]=meshgrid(x1_inc,x2_inc);
    f2=figure('Position',[100,300,640,480]);
    subplot(2,2,1);surf(u,v,WTR_optim_specmot); colormap cool;
    xlabel('WTR','fontweight','b'); axis xy;
    xlabel('X2','fontweight','b');
    ylabel('X1','fontweight','b');
    subplot(2,2,2);surf(u,v,MRR_optim_specmot);
    xlabel('MTR','fontweight','b'); axis xy;
    xlabel('X2','fontweight','b');
    ylabel('X1','fontweight','b');
    subplot(2,2,3);surf(u,v,MTR_optim_specmot);
    xlabel('MTR','fontweight','b'); axis xy;
    xlabel('X2','fontweight','b');
    ylabel('X1','fontweight','b');
    subplot(2,2,4);surf(u,v,TOR_optim_specmot);
    xlabel('TOR','fontweight','b'); axis xy;
    xlabel('X2','fontweight','b');
    ylabel('X1','fontweight','b');

    saveas(f2,inputfile,'fig')
    saveas(f2,inputfile,'eps')
end

```

```

function [wrench]=form_combo_wrench(i,combo, wr_all)
% Creates wrench matrix for a given pivot constraint combination
%
% Input variables: i,combo, wr_all
% Output variables: wrench
% Called functions: -
%
% Form the combination set from constraint wrenches
% Compose wrench based on the number of constraints in the
combination
if nnz(combo(:,i))==2
    wrench = [wr_all{combo(i,1)};wr_all{combo(i,2)}];
elseif nnz(combo(:,i))==3
    wrench =
    [wr_all{combo(i,1)};wr_all{combo(i,2)};wr_all{combo(i,3)}];
elseif nnz(combo(:,i))==4
    wrench =
    [wr_all{combo(i,1)};wr_all{combo(i,2)};wr_all{combo(i,3)};
     wr_all{combo(i,4)}];
else
    wrench =
    [wr_all{combo(i,1)};wr_all{combo(i,2)};wr_all{combo(i,3)};
     wr_all{combo(i,4)};wr_all{combo(i,5)}];
end

```

```

function [input_wr d]=input_wr_compose(mot,set_pts,set_max_d)
% Compose input wrench for a given motion
% filename: input_wr_compose.m
% Input variables: mot, set_pts, set_max_d
% Output variables: input_wr, d
% Called functions: calc_d
%
% Assign variables from screw axis parameters
omu=mot(1:3)';mu=mot(4:6)';rho=mot(7:9)';h=mot(10);

%Determine max moment arm d for motion
if h==inf
    [d]=calc_d(omu,rho,set_pts,set_max_d);
else
    d=inf; % For pure translation, d is not used
end

hs=h; % Screw pitch
hw=1/h; % Wrench pitch

if abs(hw)>=d
    % Rotation-dominant motion, torque input
    fi=hs*d*omu;
    ti=d*omu;
elseif h==inf
    % Pure translation , force input
    fi=mu; % mu is the screw axis for pure translation
    ti=[0;0;0];
else
    % Translation-dominant motion, force input
    fi=omu;
    ti=hw*omu;
end

input_wr=-[fi;ti]; % Compose input wrench, negative sign for
static equilibrium

```

```

function [mot]=rec_mot(wrench)
% Calculates reciprocal motion for a given wrench
% filename: rec_mot.m
% Input variables: wrench
% Output variables: mot
% Called functions: -
%
x=null(wrench); % The reciprocal motion is the null space of
pivot wrench matrix
x=round(x.*1e4)./1e4; % Round to 4 decimal places
mu =(x(1:3)'; % Translation of the origin due to screw axis
om =x(4:6)'; % Screw axis

% Calculate rho, position vector of screw axis
if norm(om)==0 % Check for pure translation
    h = inf; % Pitch is infinity for pure translation
    rho = [0 0 0]; % Set rho=0 for pure translation
    muu=mu/norm(mu); % Normalize mu
    mot=[om muu rho h]; % Compose into standard screw motion
else
    h = dot(mu,om)./dot(om,om); % Calculate pitch
    rho = cross(om,mu)./dot(om,om); % Calculate rho
    omu = om/norm(om); % Normalize screw axis
    mot=[omu mu rho h]; % Compose into standard screw motion
end
mot=round(mot.*1e4)./1e4; % Round everything to 4 decimal places


---


function [react_wr_5]=react_wr_5_compose(comb,rho,set)
% Compose constraining wrench from pivot wrenches for equilibrium
equation
% filename: react_wr_5_compose.m
% Input variables: comb,rho,set
% Output variables: react_wr_5
% Called functions: -
%
global cp cpin clin cpln_prop no_cp no_cpin no_clin no_cpln
global cp_rev cpin_rev clin_rev cpln_rev cpln_prop_rev

% This allows the function to be used for both the baseline
analysis and optimization routine
if strcmp(set,'original')==1
    set_cp=cp; set_cpin=cpin; set_clin=clin; set_cpln=cpln;
set_cpln_prop=cpln_prop;
elseif strcmp(set, 'revised') == 1 || strcmp(set, 'additional') == 1
    set_cp=cp_rev; set_cpin=cpin_rev; set_clin=clin_rev;
    set_cpln=cpln_rev; set_cpln_prop=cpln_prop_rev;
end
c=1; % Row index

% The following procedure is similar to the one in cp_to_wrench
function
for a=1:nnz(comb) % Do this for each constraint in the pivot
combination
    idx=comb(a);

    if idx<=no_cp % Point constraint
        b=idx;
        cp_pos=set_cp(b,1:3)'-rho;

        react_wr_5(:,c,:)=[set_cp(b,4:6),cross(cp_pos,set_cp(b,4:6))];
        c=c+1;

        elseif idx>no_cp && idx<=no_cp+no_cpin % Pin constraint
            b=idx-no_cp;
            cpln_pos=set_cpin(b,1:3)'-rho;
            axes=null(set_cpin(b,4:6));
            om_axis1=axes(:,1)';
            om_axis2=axes(:,2)';
            mu_axis1=cross(cpln_pos,om_axis1);
            mu_axis2=cross(cpln_pos,om_axis2);
            react_wr_5(:,c+1,:)=[om_axis1 mu_axis1;om_axis2
mu_axis2];
            c=c+2;

            elseif idx>no_cp+no_cpin && idx<=no_cp+no_cpin+no_clin % Line
constraint
                b=idx-(no_cpin);
                clin_pos=set_clin(b,1:3)'-rho';
                om_axis1=set_clin(b,7:9); %zero pitch
                om_axis2=[0 0 0]; %inf pitch
                mu_axis1=cross(clin_pos,om_axis1); %zero pitch
                mu_axis2=cross(set_clin(b,4:6),set_clin(b,7:9)); %inf
pitch
                react_wr_5(:,c+1,:)=[om_axis1 mu_axis1;om_axis2
mu_axis2];
                c=c+2;
    end
end

```

```

react_wr_5(:,c+1,:)=[om_axis1 mu_axis1;om_axis2
mu_axis2];
c=c+2;

elseif idx>no_cp+no_cpin+no_clin &&
idx<=no_cp+no_cpin+no_clin+no_cpln % Plane constraint
    b=idx-(no_cp+no_cpin+no_clin);
    cpln_pos=set_cpln(b,1:3)'-rho';
    axes=null(set_cpln(b,4:6));
    om_axis1=set_cpln(b,4:6); %zero pitch
    om_axis2=[0 0 0]; %inf pitch
    om_axis3=[0 0 0]; %inf pitch
    mu_axis1=cross(cpln_pos,om_axis1); %zero pitch
    mu_axis2=axes(:,1)';
    mu_axis3=axes(:,2)';
    react_wr_5(:,c+2,:)=[om_axis1 mu_axis1;om_axis2
mu_axis3;om_axis3];
    c=c+3;
end

```

```

function [Rcp_pos Rcp_neg]=rate_cp(mot,react_wr_5, input_wr,
cp_row)
% Rate the resistance quality of point constraints
% filename: rate_cp.m
% Input variables: mot,react_wr_5, input_wr, cp_row
% Output variables: Rcp_pos, Rcp_neg
% Called functions: -
%
rho=mot(7:9);
cp_pos=cp_row(1:3)-rho; % Position vector of cp from rho
wr_pt_set=[cp_row(4:6),cross(cp_pos,cp_row(4:6))]; % Reaction
wrench

% Merge the constraining wrenches from pivot constraints and
reaction
constraints
react_wr=[react_wr_5; wr_pt_set];

if rank(react_wr)==6 % If rank is not 6 then it is linearly
dependent have infinite reaction
    static_sol=react_wr\input_wr; % Solve static equilibrium
    if static_sol(end)>=0 % Capture the Positive value
        Rcp_pos=static_sol(end);
        Rcp_neg=inf;
    else % Capture the Negative value
        Rcp_pos=inf;
        Rcp_neg=-static_sol(end);
    end
else
    Rcp_pos=inf;
    Rcp_neg=inf;
end

```

```

function [Rcpin]=rate_cpin(mot,react_wr_5, input_wr,cpin_row)
% Rate the resistance quality of pin constraints
% filename: rate_cpin.m
% Input variables: mot,react_wr_5, input_wr, cpin_row
% Output variables: Rcpin
% Called functions: -
%
omu=mot(1:3);muu=mot(4:6);rho=mot(7:9);h=mot(10);
cpin_ctr=cpin_row(1:3);
cpin_normal=cpin_row(4:6);

%In the write up describe the different cases that this can fall
into
% parallel, coincident, intersecting, non-intersecting

if h~=-inf
    % Assess the relationship between input wrench and reaction
wrench
    mom_arm=cpin_ctr-rho; % Calculate moment arm length
    if norm(mom_arm)~=0 % Non-intersecting
        line_action=h*muu+cross(omu,mom_arm);
    else % Intersecting
        line_action=[0 0 0]; %if axis is coincident and/or
intersecting, then pin cannot react
    end
else
    line_action=muu; % Pure translation case
end

% Project the line of action on to the plane of CPIN
const_dir = cross(cpin_normal,cross(line_action,cpin_normal));
const_dir = round(const_dir.*1e5)./1e5;

if norm(const_dir)~=0 % If const_dir==0, it a mark that reaction
is infinite
    const_dir=const_dir./norm(const_dir);

```

```

wr_const_dir=[const_dir,cross((cpin_ctr-rho),const_dir)]; %
Reaction wrench

% Merge the constraining wrenches from pivot constraints and
reaction constraints
react_wr=[react_wr_5; wr_const_dir];

if rank(react_wr')==6 % If rank is not 6 then it is linearly
dependent have infinite reaction
    static_sol=react_wr\input_wr; % Solve static equilibrium
    Rcpin=abs(static_sol(end)); % Capture lambda_6
else
    Rcpin=inf;
end

function [Rclin_pos
Rclin_neg]=rate_clin(mot,react_wr_5,input_wr,clin_row)
% Rate the resistance quality of line constraints
% filename: rate_clin.m
% Input variables: mot,react_wr_5,input_wr,clin_row
% Output variables: Rclin_pos, Rclin_neg
% Called functions: -
%
rho=mot(7:9);
clin_ctr=clin_row(1:3); % Line midpoint coordinate
clin_dir=clin_row(4:6)./(norm(clin_row(4:6))); % Line direction
clin_normal=clin_row(7:9); % Line normal constraint direction
clin_halflen=clin_row(10)/2; % Line half length

% Line constraint end points
clin_end1=clin_ctr-(clin_halflen.*clin_dir);
clin_end2=clin_ctr+(clin_halflen.*clin_dir);

% Reaction wrenches
wr_clin_end1=[clin_normal,cross((clin_end1-rho),clin_normal)];
wr_clin_end2=[clin_normal,cross((clin_end2-rho),clin_normal)];

% Merge the constraining wrenches from pivot constraints and
reaction constraints
react_wr1=[react_wr_5; wr_clin_end1];
react_wr2=[react_wr_5; wr_clin_end2];

if rank(react_wr1')==6 % If rank is not 6 then it is linearly
dependent have infinite reaction
    static_sol=react_wr1\input_wr; % Solve static equilibrium
    M(1)=static_sol(end); % Capture lambda_6
else
    M(1)=inf;
end

if rank(react_wr2')==6
    static_sol=react_wr2\input_wr;
    M(2)=static_sol(end);
else
    M(2)=inf;
end

Mpos=[inf inf];Mneg=[inf inf];

% Select the optimum reaction point (least reaction force) for
each direction motion
for b=1:2
    if abs(M(b)) < 0.0001, M(b)=0; end
    if M(b)>0, Mpos(b)=M(b);end
    if M(b)<0, Mneg(b)=-M(b);end
end

% Reciprocal sum when both points react together
% This happens when the line of action is in the same direction
for both locations
Rclin_pos=1/(1/Mpos(1)+1/Mpos(2)+1/Mpos(3)+1/Mpos(4));
Rclin_neg=1/(1/Mneg(1)+1/Mneg(2)+1/Mneg(3)+1/Mneg(4));

```

```

function [Rcpn_pos
Rcpn_neg]=rate_cpln2(mot,react_wr_5,input_wr,cpln_row,cpln_prop_
row)
% Rate the resistance quality of rectangular plane constraints
% filename: rate_cpln2.m
% Input variables: mot,react_wr_5,input_wr,cpln_row,cpln_prop_row
% Output variables: Rcpn_pos, Rcpn_neg
% Called functions: -
%
omu=mot(1:3);rho=mot(7:9);h=mot(10);%muu is only unit length in
trans

cpln_ctr=cpln_row(1:3); % plane midpoint coordinate
cpln_normal=cpln_row(4:6); % plane normal constraint direction
cpln_rad=cpln_prop_row(1); %circular plane radius

if h~=inf
    % Assess the relationship between input wrench and reaction
wrench
    mom_arm = cpln_ctr-rho;
    if norm(mom_arm)~=0 % Non-intersecting
        mom_arm_proj =
cross(cpln_normal,cross(mom_arm,cpln_normal));
        % If the cross mom_arm ~=0 but parallel to normal this is
fine
        % because then it means the plane cannot react
    else % Intersecting
        mom_arm_proj=cross(omu,cpln_normal);
    end

    if norm(mom_arm_proj)~=0
        mom_arm_proj=mom_arm_proj./norm(mom_arm_proj);
    end

    % Optimum reaction location
    cpln_edge_pos1=cpln_ctr+mom_arm_proj.*cpln_rad;
    cpln_edge_pos2=cpln_ctr-mom_arm_proj.*cpln_rad;
else
    % For pure translation, the location of wrench does not
matter
    cpln_edge_pos1=cpln_ctr;
    cpln_edge_pos2=cpln_ctr;
end

% Reaction wrenches
wr_line_proj1=[cpln_normal,cross((cpln_edge_pos1-
rho),cpln_normal)];
```

```

wr_line_proj2=[cpln_normal,cross((cpln_edge_pos2-
rho),cpln_normal)];

% Merge the constraining wrenches from pivot constraints and
reaction constraints
react_wr1=[react_wr_5; wr_line_proj1];
react_wr2=[react_wr_5; wr_line_proj2];

if rank(react_wr1')=6 % If rank is not 6 then it is linearly
dependent have infinite reaction
    static_sol=react_wr1\input_wr; % Solve static equilibrium
    M1=static_sol(end); % Capture lambda_6
else
    M1=inf;
end

if rank(react_wr2')=6
    static_sol=react_wr2\input_wr;
    M2=static_sol(end);
else
    M2=inf;
end

M=[M1 M2]; Mpos=[inf inf]; Mneg=[inf inf];

% Select the optimum reaction point (least reaction force) for
each direction motion
for b=1:2
    if abs(M(b)) < 0.0001, M(b)=0; end
    if M(b)>0, Mpos(b)=M(b);end
    if M(b)<0, Mneg(b)=-M(b);end
end

% Reciprocal sum when both points react together
% This happens when the line of action is in the same direction
for both locations
Rcpn_pos=1/(2*(1/Mpos(1)+1/Mpos(2)));
Rcpn_neg=1/(2*(1/Mneg(1)+1/Mneg(2)));




---


function [d]=calc_d(omu,rho,pts,max_d)
% Calculate maximum moment arm d between screw axis and
constraint locations
% filename: calc_d.m
% Input variables: omu,rho,pts,max_d
% Output variables: d
% Called functions: -
%
for a=1:size(pts,1)
    mom_arm = pts(a,:)-rho;
    dist(a)=norm(cross(omu,mom_arm)); % project moment arm to
perpendicular distance
end

d=max(dist); % pick the maximum

if d>max_d
    d=max_d; %limit d to max_d, maximum distance between
constraints
end




---


function [cp_rev cpin_rev clin_rev cpln_rev cpln_prop_rev...]
]=move_pt_src(x,cp_rev_idx, pt_src)
% Move constraints in the line search space
% filename: move_pt_src.m
%
% WARNING: THIS IS PRELIMINARY CODE AND CONTAINS ERRORS

global no_cp no_cpin no_clin no_cpln
global cp_rev cpin_rev clin_rev cpln_rev cpln_prop_rev

for i=1:length(cp_rev_in_group)

    if cp_rev_idx(i)<no_cp

        %Move cp_rev location to midpoint of line and then move
it along
        %the line with multiplier x

        cp_rev(cp_rev_idx(i),1:3)=pt_src(x,1:3);

    elseif cp_rev_idx(i)>no_cp && cp_rev_idx(i)<=no_cp+no_cpin

        cpin_rev(cp_rev_idx(i),1:3)=pt_src(x,1:3);

    elseif cp_rev_idx(i)>no_cp+no_cpin &&
cp_rev_idx(i)<=no_cp+no_cpin+no_clin

        clin_rev(cp_rev_idx(i),1:3)=pt_src(x,1:3);

    end

    elseif cp_rev_idx(i)>no_cp+no_cpin+no_clin &&
cp_rev_idx(i)<no_cp+no_cpin+no_cpln

        cpln_rev(cp_rev_idx(i),1:3)=pt_src(x,1:3);

    end

    elseif cp_rev_idx(i)>no_cp+no_cpin+no_cpln &&
cp_rev_idx(i)<no_cp+no_cpin+no_clin+no_cpln

        cpln_rev(cp_rev_idx(i),1:3)=pt_src(x,1:3);

    end

    else
        cp_rev_idx(i)=no_cp+no_cpin+no_clin+no_cpln;
    end

    if isempty(cp_rev)
        pts=[pts;cp_rev(:,1:3)];
    end

    if isempty(cpin_rev)==0
        pts=[pts;cpin_rev(:,1:3)];
    end

    if isempty(clin_rev)==0
        for j=1:size(clin_rev,1)
            pts=[pts;clin_rev(j,1:3)+clin_rev(j,7:9)/2.*clin_rev(j,4:6)];
        end
    end
end

```

```

pts=[pts;clin_rev(j,1:3)-
clin_rev(j,10)/2.*clin_rev(j,4:6)];
end

if isempty(cpln_rev)==0
    for j=1:size(cpln_rev,1)
        if cpln_rev(j,7)==1
            pts=[pts;cpln_rev(j,1:3)+cpln_prop_rev(j,4)/2.*cpln_prop_rev(j,1:3)+cpln_prop_rev(j,8)/2.*cpln_prop_rev(j,5:7)];
            pts=[pts;cpln_rev(j,1:3)+cpln_prop_rev(j,4)/2.*cpln_prop_rev(j,1:3)-cpln_prop_rev(j,5:7)];
            pts=[pts;cpln_rev(j,1:3)-cpln_prop_rev(j,4)/2.*cpln_prop_rev(j,1:3)+cpln_prop_rev(j,8)/2.*cpln_prop_rev(j,5:7)];
            pts=[pts;cpln_rev(j,1:3)-cpln_prop_rev(j,4)/2.*cpln_prop_rev(j,1:3)-cpln_prop_rev(j,8)/2.*cpln_prop_rev(j,5:7)];
        else
            axes=null(cpln_rev(j,4:6));
        end
    end
    pts=[pts;cpln_rev(j,1:3)+cpln_prop_rev(j,1).*axes(:,1)'];
    pts=[pts;cpln_rev(j,1:3)-cpln_prop_rev(j,1).*axes(:,1)'];
    pts=[pts;cpln_rev(j,1:3)+cpln_prop_rev(j,1).*axes(:,2)'];
    pts=[pts;cpln_rev(j,1:3)-cpln_prop_rev(j,1).*axes(:,2)'];
    pts=[pts;cpln_rev(j,1:3)+cosd(45)*cpln_prop_rev(j,1).*axes(:,1)+cosd(45)*cpln_prop_rev(j,1).*axes(:,2)];
    pts=[pts;cpln_rev(j,1:3)+cosd(45)*cpln_prop_rev(j,1).*axes(:,1)-cosd(45)*cpln_prop_rev(j,1).*axes(:,2)];
    pts=[pts;cpln_rev(j,1:3)-cosd(45)*cpln_prop_rev(j,1).*axes(:,1)+cosd(45)*cpln_prop_rev(j,1).*axes(:,2)];
    pts=[pts;cpln_rev(j,1:3)-cosd(45)*cpln_prop_rev(j,1).*axes(:,1)-cosd(45)*cpln_prop_rev(j,1).*axes(:,2)];
end
c=nchoosek(1:size(pts,1),2);
for a=1:size(c,1)
    distance(a)=((pts(c(a,1),1)-pts(c(a,2),1))^2+...
        (pts(c(a,1),2)-pts(c(a,2),2))^2+...
        (pts(c(a,1),3)-pts(c(a,2),3))^2)^5;
end
max_d=max(distance);



---


function []=line_orientid_srch(x_grp,cp_rev_in_group,
lin_dir_srch)
% Reorient line constraints about 1 axis
% filename: line_orientid_srch.m
%
global no_cp no_cpin no_clin no_cpln
global clin_rev

% dir1d_srch format:
% [local_rot_axis(3), angle,zeros]
% local_rot_axis is treated as the local x axis

% original line direction clin_rev(k,4:6) is the local z axis
% Later the same case for clin_rev(k,7:9), the constraint normal

% make sure follows right hand rule

angle=x_grp*lin_dir_srch(4);

% Specify dn, orientation of normal vector relative to local
frame for rotation around local x
dn=[0; -sind(angle); cosd(angle)];
% Note that non-perturbed (phi=0) --> dn=[0 0 1]
dn=dn/norm(dn);

local_x=lin_dir_srch(1:3);

for i=1:length(cp_rev_in_group)
    idx=cp_rev_in_group(i);
    if idx==no_cp
        disp('Line orientation search is not applicable to point
constraints');
        return;
    elseif idx>no_cp && idx<=no_cp+no_cpin
        disp('Line orientation search is not applicable to pin
constraints');
    end
    pts=[pts;clin_rev(k,1:3)-
clin_rev(k,10)/2.*clin_rev(k,4:6)];
    end

if isempty(cpln_rev)==0
    for j=1:size(cpln_rev,1)
        if cpln_rev(j,7)==1
            pts=[pts;cpln_rev(j,1:3)+cpln_prop_rev(j,4)/2.*cpln_prop_rev(j,1:3)+cpln_prop_rev(j,8)/2.*cpln_prop_rev(j,5:7)];
            pts=[pts;cpln_rev(j,1:3)+cpln_prop_rev(j,4)/2.*cpln_prop_rev(j,1:3)-cpln_prop_rev(j,5:7)];
            pts=[pts;cpln_rev(j,1:3)-cpln_prop_rev(j,4)/2.*cpln_prop_rev(j,1:3)+cpln_prop_rev(j,8)/2.*cpln_prop_rev(j,5:7)];
            pts=[pts;cpln_rev(j,1:3)-cpln_prop_rev(j,4)/2.*cpln_prop_rev(j,1:3)-cpln_prop_rev(j,8)/2.*cpln_prop_rev(j,5:7)];
        else
            axes=null(cpln_rev(j,4:6));
        end
    end
    pts=[pts;cpln_rev(j,1:3)+cpln_prop_rev(j,1).*axes(:,1)'];
    pts=[pts;cpln_rev(j,1:3)-cpln_prop_rev(j,1).*axes(:,1)'];
    pts=[pts;cpln_rev(j,1:3)+cpln_prop_rev(j,1).*axes(:,2)'];
    pts=[pts;cpln_rev(j,1:3)-cpln_prop_rev(j,1).*axes(:,2)'];
    pts=[pts;cpln_rev(j,1:3)+cosd(45)*cpln_prop_rev(j,1).*axes(:,1)+cosd(45)*cpln_prop_rev(j,1).*axes(:,2)];
    pts=[pts;cpln_rev(j,1:3)+cosd(45)*cpln_prop_rev(j,1).*axes(:,1)-cosd(45)*cpln_prop_rev(j,1).*axes(:,2)];
    pts=[pts;cpln_rev(j,1:3)-cosd(45)*cpln_prop_rev(j,1).*axes(:,1)+cosd(45)*cpln_prop_rev(j,1).*axes(:,2)];
    pts=[pts;cpln_rev(j,1:3)-cosd(45)*cpln_prop_rev(j,1).*axes(:,1)-cosd(45)*cpln_prop_rev(j,1).*axes(:,2)];
end
c=nchoosek(1:size(pts,1),2);
for a=1:size(c,1)
    distance(a)=((pts(c(a,1),1)-pts(c(a,2),1))^2+...
        (pts(c(a,1),2)-pts(c(a,2),2))^2+...
        (pts(c(a,1),3)-pts(c(a,2),3))^2)^5;
end
max_d=max(distance);



---


function []=move_curvlin_srch(x_grp,cp_rev_in_group,circlin_srch)
% Move constraints in the curved line search space
% filename: move_curvlin_srch.m
%
global no_cp no_cpin no_clin no_cpln
global cp_rev cpin_rev clin_rev cpln_rev

%cirvlin_srch format is relative, not absolute, basically it
rotates the
%position using the axis defined.
%[center_coord(3), rotation_axis_dir(3), angle]
%starting point x axis is defined by
angle=x_grp*circlin_srch(7);
circlin_srch(4:6)=circlin_srch(4:6)./norm(circlin_srch(4:6));
local_orig=circlin_srch(1:3); % Location of the rotation axis
center

% For loop to apply modification to all constraints in the group
for i=1:length(cp_rev_in_group)

idx=cp_rev_in_group(i);

% Find local_x - the vector from the rotation axis to the
constraint location
if idx<=no_cp
    k=idx;
    local_x=cp_rev(k,1:3)'-local_orig;
elseif idx>no_cp & idx<=no_cp+no_cpin
    k=idx-no_cp;
    local_x=cpin_rev(k,1:3)'-local_orig;
elseif idx>no_cp+no_cpin & idx<=no_cp+no_cpin+no_clin
    k=idx-no_cpin;
    local_x=clin_rev(k,1:3)'-local_orig;
elseif idx>no_cpin+no_clin & idx<=no_cpin+no_clin+no_cpln
    k=idx-(no_cpin+no_clin);
    local_x=cpln_rev(k,1:3)'-local_orig;
end

% Rotation matrix
rot_local=[ cosd(angle) -sind(angle) 0;
            sind(angle) cosd(angle) 0;
            0 0 1];

% dp is the vector from rotation center to the new position
dp=rot_local*local_x;
new_pos=local_orig+dp;

% Apply this to each constraints depending on their type
if idx<=no_cp
    k=idx;
    cp_rev(k,1:3)=new_pos';
elseif idx>no_cp && idx<=no_cp+no_cpin
    k=idx-no_cp;
    cpin_rev(k,1:3)=new_pos';
elseif idx>no_cp+no_cpin && idx<=no_cp+no_cpin+no_clin
    k=idx-(no_cpin+no_clin);
    clin_rev(k,1:3)=new_pos';
elseif idx>no_cpin+no_clin & idx<=no_cpin+no_clin+no_cpln
    k=idx-(no_cpin+no_clin);
    cpln_rev(k,1:3)=new_pos';
end



---


function []=move_lin_srch(x_grp,cp_rev_in_group,lin_srch)
% Move constraints in the line search space
% filename: move_lin_srch.m
%

```

```

global no_cp no_cpin no_clin no_cpln
global cp_rev cpin_rev clin_rev cpln_rev

% The line search space is defined for the first constraints
only, the other constraints follow

% First find the relative position vector from the first
constraints midpoint to the midpoint of line search space

idx=cp_rev_in_group(1);
lin_srch(4:6)=lin_srch(4:6)./norm(lin_srch(4:6));

if idx<=no_cp
    k=idx;
    ctr_move=lin_srch(1:3)-cp_rev(k,1:3);

elseif idx>no_cp && idx<=no_cp+no_cpin
    k=idx-no_cp;
    ctr_move=lin_srch(1:3)-cpin_rev(k,1:3);

elseif idx>no_cp && idx<=no_cp+no_cpin
    k=idx-no_cp;
    ctr_move=lin_srch(1:3)-cpln_rev(k,1:3);

elseif idx>no_cp+no_cpin && idx<=no_cp+no_cpin+no_clin
    k=idx-(no_cp+no_cpin);
    ctr_move=lin_srch(1:3)-clin_rev(k,1:3);

elseif idx>no_cp+no_cpin+no_clin &&
idx<=no_cp+no_cpin+no_clin+no_cpln
    k=idx-(no_cp+no_cpin+no_clin);
    ctr_move=lin_srch(1:3)-cpln_rev(k,1:3);

end

% Move the constraints with ctr_move, then move in the same
direction as the first constraints

for i=1:length(cp_rev_in_group)
    idx=cp_rev_in_group(i);
    if idx==0, return, end

    if idx<=no_cp
        k=idx;

        cp_rev(k,1:3)=cp_rev(k,1:3)+ctr_move+(x_grp(1)*pln_srch(7)).*pln_
srch(4:6)+(x_grp(2)*pln_srch(11)).*pln_srch(8:10);
        elseif idx>no_cp && idx<=no_cp+no_cpin
            k=idx-no_cp;

        cpin_rev(k,1:3)=cpin_rev(k,1:3)+ctr_move+(x_grp(1)*pln_srch(7)).*
pln_srch(4:6)+(x_grp(2)*pln_srch(11)).*pln_srch(8:10);
        elseif idx>no_cp+no_cpin && idx<=no_cp+no_cpin+no_clin
            k=idx-(no_cp+no_cpin);

        clin_rev(k,1:3)=clin_rev(k,1:3)+ctr_move+(x_grp(1)*pln_srch(7)).*
pln_srch(4:6)+(x_grp(2)*pln_srch(11)).*pln_srch(8:10);
        elseif idx>no_cp+no_cpin+no_clin &&
idx<=no_cp+no_cpin+no_clin+no_cpln
            k=idx-(no_cp+no_cpin+no_clin);

        cpln_rev(k,1:3)=cpln_rev(k,1:3)+ctr_move+(x_grp(1)*pln_srch(7)).*
pln_srch(4:6)+(x_grp(2)*pln_srch(11)).*pln_srch(8:10);
    end
end

function []=move_pln_srch(x_grp,cp_rev_in_group, pln_srch)
% Move constraints in the plane search space
% filename: move_pln_srch.m
%
global no_cp no_cpin no_clin no_cpln
global cp_rev cpin_rev clin_rev cpln_rev

pln_srch(4:6)=pln_srch(4:6)./norm(pln_srch(4:6));
pln_srch(8:10)=pln_srch(8:10)./norm(pln_srch(8:10));

%pln search format
% [ center (x,y,z), x-dir-search(x,y,z),x-dir one-way width,
% y-dir-search(x,y,z,),y-dir one-way width]

% The procedure here is very similar to line search space, but
applied in 2D
% See comments on move_lin_srch

idx=cp_rev_in_group(1);

if idx<=no_cp
    k=idx;
    ctr_move=pln_srch(1:3)-cp_rev(k,1:3);

elseif idx>no_cp && idx<=no_cp+no_cpin
    k=idx-no_cp;
    ctr_move=pln_srch(1:3)-cpin_rev(k,1:3);

elseif idx>no_cp+no_cpin && idx<=no_cp+no_cpin+no_clin
    k=idx-(no_cp+no_cpin);
    ctr_move=pln_srch(1:3)-cpln_rev(k,1:3);

end

for i=1:length(cp_rev_in_group)
    idx=cp_rev_in_group(i);
    if idx==0, return, end

    if idx<=no_cp
        %Move cp_rev location to midpoint of line and then move
it along
        %the line with multiplier x
        k=idx;

        cp_rev(k,1:3)=cp_rev(k,1:3)+ctr_move+(x_grp(1)*pln_srch(7)).*pln_
srch(4:6)+(x_grp(2)*pln_srch(11)).*pln_srch(8:10);
        elseif idx>no_cp && idx<=no_cp+no_cpin
            k=idx-no_cp;

        cpin_rev(k,1:3)=cpin_rev(k,1:3)+ctr_move+(x_grp(1)*pln_srch(7)).*
pln_srch(4:6)+(x_grp(2)*pln_srch(11)).*pln_srch(8:10);
        elseif idx>no_cp+no_cpin && idx<=no_cp+no_cpin+no_clin
            k=idx-(no_cp+no_cpin);

        clin_rev(k,1:3)=clin_rev(k,1:3)+ctr_move+(x_grp(1)*pln_srch(7)).*
pln_srch(4:6)+(x_grp(2)*pln_srch(11)).*pln_srch(8:10);
        elseif idx>no_cp+no_cpin+no_clin &&
idx<=no_cp+no_cpin+no_clin+no_cpln
            k=idx-(no_cp+no_cpin+no_clin);

        cpln_rev(k,1:3)=cpln_rev(k,1:3)+ctr_move+(x_grp(1)*pln_srch(7)).*
pln_srch(4:6)+(x_grp(2)*pln_srch(11)).*pln_srch(8:10);
    end
end

function [Rating_all_rev Ri_specmot
mot_specmot_all]=rate_specmot(x,x_map,spec_mot)
% Known loading condition main processor to calculate constraint
effectiveness
% filename: rate_specmot.m
% Called functions: move_pt_srch, move_lin_srch,
move_circlin_srch, move_pln_srch
%          orient1d_srch, orient2d_srch,
line_orient1d_srch, resize_lin_srch
%          resize_rectpln_srch, resize_circpln_srch,
cp_rev_to_wrench, rating
%          rate_cp, rate_cpin, rate_clin, rate_cpln1,
rate_cpln2, rating
%
global cp cpin clin cpln cpln_prop no_cp no_cpin no_clin no_cpln
global cp_rev cpin_rev clin_rev cpln_rev cpln_prop_rev
global d_proc wrench_proc input_wr_proc
global wr_all
global grp_members grp_rev_type grp_srch_spc cp_rev_all

% The procedure here is a combination of optim_rev and main_loop
procedures to analyze constraints

% Initialize variables
Rcp_pos=[];Rcp_neg=[];Rcpin=[];
Rclin_pos=[];Rclin_neg=[];
Rcpln_pos=[];Rcpln_neg=[];

no_spec_mot=size(spec_mot,1);

for m=1:no_spec_mot

    % Find null space for pseudo-pivot constraints

    omu=spec_mot(m,1:3)./norm(spec_mot(m,1:3));rho=spec_mot(m,4:6);h=
spec_mot(m,7);
    mu=h*omu+cross(rho,omu);
    mot=[omu mu rho h];

```

```

rec_mot=[mu omu];
pivot_wr=null(rec_mot)';
% Initialize constraint variables with original for each
motion
cp_rev=cp; cpin_rev=cpin; clin_rev=clin;
cpln_rev=cpln; cpln_prop_rev=cpln_prop;

for i=1:size(grp_members,1)
    rev_type=grp_rev_type(i);
    cp_rev_in_group=nonzeros(grp_members(i,:));
    if rev_type==1
        x_grp=x(x_map(i,1));
        move_pt_srch(x_grp,cp_rev_in_group, grp_srch_spc);
    elseif rev_type==2
        x_grp=x(x_map(i,1));
    move_lin_srch(x_grp,cp_rev_in_group,grp_srch_spc(i,:));
    elseif rev_type==3
        x_grp=x(x_map(i,1));
    move_circlin_srch(x_grp,cp_rev_in_group,grp_srch_spc(i,:));
    elseif rev_type==4
        x_grp=x(x_map(i,1:2));
    move_pln_srch(x_grp,cp_rev_in_group,grp_srch_spc(i,:));
    elseif rev_type==5
        x_grp=x(x_map(i,1));
        orientd_srch(x_grp,cp_rev_in_group,
grp_srch_spc(i,:));
    elseif rev_type==6
        x_grp=x(x_map(i,1:2));
        orientd_srch(x_grp,cp_rev_in_group,
grp_srch_spc(i,:));
    elseif rev_type==7
        x_grp=x(x_map(i,1));
        line_orientid_srch(x_grp,cp_rev_in_group,
grp_srch_spc(i,:));
    elseif rev_type==8
        x_grp=x(x_map(i,1));
        resize_lin_srch(x_grp,cp_rev_in_group,
grp_srch_spc(i,:));
    elseif rev_type==9
        x_grp=x(x_map(i,1));
        resize_rectpln_srch(x_grp(1:2),cp_rev_in_group,
grp_srch_spc(i,:));
    elseif rev_type==10
        x_grp=x(x_map(i,1));
        resize_circpln_srch(x_grp,cp_rev_in_group,
grp_srch_spc(i,:));
    end
end

% Transform constraints into wrenches
[w_all_new_pts_rev
max_d_rev]=cp_rev_to_wrench(wr_all,cp_rev_all, ...
cp_rev, cpin_rev, clin_rev, cpln_rev, cpln_prop_rev);

omu=mot(1:3)';mu=mot(4:6)';rho=mot(7:9)';h=mot(10);

% Calculate d for motion
if h~inf
    [d]=calc_d(omu,rho,pts_rev,max_d_rev);
else
    d=inf;
end

%Set up input wrench with true origin, unlike main_loop
hs=h; hw=1/h;
if abs(hw)>d
    %rotation dominant torque input
    fi=hs*d*omu;
    ti=d*omu+hs*d*cross(rho,omu);
elseif h~inf
    %pure translation , force dominant
    fi=mu; %mu already normalized in rec_mot
    ti=[0;0;0];
else
    %force dominant force input
    fi=omu;
    ti=hw*omu+cross(rho,omu);
end
input_wr=-[fi;ti];

% Rate all constraints
% We can use pivot_wr as react_wr_5 because everything is
with respect to origin
for j = 1:no_cp
    [Rcp_pos(m,j) Rcp_neg(m,j)]=rate_cp(mot,pivot_wr,
input_wr, cp_rev(j,:));
end

for j = 1:no_cpin
[Rcpin(m,j)]=rate_cpin(mot,pivot_wr,input_wr,cpin_rev(j,:));
end

for j = 1:no_clin
    [Rclin_pos(m,j)]
Rclin_neg(m,j)]=rate_clin(mot,pivot_wr,input_wr,clin_rev(j,:));
end

for j = 1:no_cpln
    if cpln_rev(j,7)==1
        [Rcpln_pos(m,j)]
Rcpln_neg(m,j)]=rate_cpln1(mot,pivot_wr,
input_wr,cpln_rev(j,:),cpln_prop_rev(j,:));
    elseif cpln_rev(j,7)==2
        [Rcpln_pos(m,j)]
Rcpln_neg(m,j)]=rate_cpln2(mot,pivot_wr,
input_wr,cpln_rev(j,:),cpln_prop_rev(j,:));
    end
end

mot_specmot(m,:)=mot;
mot_specmot_rev(m,:)=[-mot(1:6) mot(7:10)]; % Add reverse
motion
% Collect processed d, pivot wrenches, and input wrenches
d_proc(m,1)=d;
wrench_proc{m}=pivot_wr;
input_wr_proc{m}=input_wr;
end

mot_specmot_all=[mot_specmot;mot_specmot_rev]; % Collect motions
Rcp=[Rcp_pos;Rcp_neg]; % Merge resistance value for fwd and rev
motion for constraints
Rcpin=[Rcpin;Rcpin]; % Resistance values for fwd and rev motion
is the same for cpin
Rclin=[Rclin_pos;Rclin_neg];
Rcpln=[Rcpln_pos;Rcpln_neg];

R=[Rcp Rcpin Rclin Rcpln]; % Merge all ratings

Ri_specmot=1/R;
Ri_specmot=(round(Ri_specmot.*1e4)).*1e-4;

% Calculate final ratings
[Rating_all_rev WTR_idx_org free_mot free_mot_idx best_cp
rowsum]=rating(Ri_specmot, mot_specmot);

function [cp_rev cpin_rev clin_rev cpln_rev cpln_prop_rev...]
]=move_pt_srch(x, cp_rev_idx, pt_srch)
% Move constraints in the line search space
% filename: move_pt_srch.m
%
% WARNING: THIS IS PRELIMINARY CODE AND CONTAINS ERRORS

global no_cp no_cpin no_clin no_cpln
global cp_rev cpin_rev clin_rev cpln_rev cpln_prop_rev

for i=1:length(cp_rev_in_group)

    if cp_rev_idx(i)<=no_cp

        %Move cp_rev location to midpoint of line and then move
it along
        %the line with multiplier x

        cp_rev(cp_rev_idx(i),1:3)=pt_srch(x,1:3);

    elseif cp_rev_idx(i)>no_cp && cp_rev_idx(i)<=no_cp+no_cpin

        cpin_rev(cp_rev_idx(i),1:3)=pt_srch(x,1:3);

    elseif cp_rev_idx(i)>no_cp+no_cpin &&
cp_rev_idx(i)<=no_cp+no_cpin+no_clin

        clin_rev(cp_rev_idx(i),1:3)=pt_srch(x,1:3);

    elseif cp_rev_idx(i)>no_cp+no_cpin+no_clin &&
cp_rev_idx(i)<=no_cp+no_cpin+no_clin+no_cpln

        cpln_rev(cp_rev_idx(i),1:3)=pt_srch(x,1:3);

    end
end

% use x as the index for going through the search space. In this
case use
% grp_srch_spc with each row corresponding to each constraint.

function []=orientid_srch(x_grp, cp_rev_in_group, dir1d_srch)
% Reorient constraints about 1 axis
% filename: orientid_srch.m
%

```

```

global no_cp no_cpin no_clin no_cpln
global cp_rev cpin_rev clin_rev cpln_rev

% dir1d_srch format:
% [local_rot_axis(3), angle,zeros]
% local_rot_axis is treated as the local_x axis
% constraint normal is local_z axis
% make sure follows right hand rule

angle=x_grp*dir1d_srch(4);

% Specify dn, orientation of normal vector relative to local
frame for rotation around local_x
dn=[0; -sin(angle); cosd(angle)];
% Note that non-perturbed (phi=0) --> dn=[0 0 1]
dn=dn/norm(dn);

local_x=dir1d_srch(1:3)';

for i=1:length(cp_rev_in_group)
    idx=cp_rev_in_group(i);
    if idx<=no_cp
        k=idx;
        %extract cp location and orientation
        %locate local_x and y frame
        local_z=cp_rev(k,4:6)';
        local_y=cross(local_z,local_x);
        rot=[local_x local_y local_z]; % Transform from local to
global frame
        new_normal=rot*dn;
        cp_rev(k,4:6)=new_normal./norm(new_normal);
    elseif idx>no_cp && idx<=no_cpin+no_cpln
        k=idx;
        local_z=cpin_rev(k,4:6)';
        local_y=cross(local_z,local_x);
        rot=[local_x local_y local_z];
        new_normal=rot*dn;
        cpin_rev(k,4:6)=new_normal./norm(new_normal);
    elseif idx>no_cpin+no_cpln && idx<=no_cp+no_cpin+no_clin
        k=idx-(no_cpin+no_cpln);
        local_z=cclin_rev(k,7:9)';
        local_y=cross(local_z,local_x);
        rot=[local_x local_y local_z];
        new_normal=rot*dn;
        clin_rev(k,7:9)=new_normal./norm(new_normal);
    elseif idx>no_cpin+no_clin &&
idx<=no_cp+no_cpin+no_cpln
        k=idx-(no_cpin+no_cpln);
        local_z=cpln_rev(k,4:6)';
        local_y=cross(local_z,local_x);
        rot=[local_x local_y local_z];
        new_normal=rot*dn;
        cpln_rev(k,4:6)=new_normal./norm(new_normal);
    end
end

function []=orient2d_srch(x_grp,cp_rev_in_group, dir2d_srch)
% Reorient constraints about 2 axis
% filename: orient2d_srch.m
%
global no_cp no_cpin no_clin no_cpln
global cp_rev cpin_rev clin_rev cpln_rev

alpha=x_grp(1)*dir2d_srch(7);
beta=x_grp(2)*dir2d_srch(8);

% Specify dn, orientation of normal vector relative to local
frame for
% rotation around local_x and local_y
% constraint normal is local_z

% dir_srch format:
% [local_x_axis(3), local_y_axis(3), angle(x), angle(y)]
% make sure follows right hand rule

dn=[sind(beta); -sind(alpha); cosd(alpha)*cosd(beta)];
% Note that non-perturbed (alpha=beta=0) --> dn=[0 0 1]
dn=dn/norm(dn);

local_x=dir2d_srch(1:3)';
local_y=dir2d_srch(4:6)';

for i=1:length(cp_rev_in_group)
    idx=cp_rev_in_group(i);
    if idx<=no_cp
        k=idx;
        %extract cp location and orientation
        %locate local_x and y frame
        local_z=cp_rev(k,4:6)';
        rot=[local_x local_y local_z]; % Transform from local to
global frame
        new_normal=rot*dn;
        cp_rev(k,4:6)=new_normal./norm(new_normal);
    elseif idx>no_cp && idx<=no_cpin
        k=idx-no_cp;
        local_z=cpin_rev(k,4:6)';
        rot=[local_x local_y local_z];
        new_normal=rot*dn;
        cpin_rev(k,4:6)=new_normal./norm(new_normal);
    elseif idx>no_cpin && idx<=no_cpln
        k=idx-(no_cpin);
        local_z=cpln_rev(k,4:6)';
        rot=[local_x local_y local_z];
        new_normal=rot*dn;
        cpln_rev(k,4:6)=new_normal./norm(new_normal);
    elseif idx>no_cpln && idx<=no_clin
        k=idx-(no_cpln);
        local_z=cclin_rev(k,7:9)';
        rot=[local_x local_y local_z];
        new_normal=rot*dn;
        clin_rev(k,7:9)=new_normal./norm(new_normal);
    elseif idx>no_clin && idx<=no_cpln+no_cpin
        k=idx-(no_cpln+no_cpin);
        local_z=cpln_rev(k,4:6)';
        local_y=cross(local_z,local_x);
        rot=[local_x local_y local_z];
        new_normal=rot*dn;
        cpln_rev(k,4:6)=new_normal./norm(new_normal);
    elseif idx>no_cpln+no_cpin && idx<=no_cp+no_cpin+no_clin
        k=idx-(no_cpln+no_cpin);
        local_z=cpln_rev(k,4:6)';
        local_y=cross(local_z,local_x);
        rot=[local_x local_y local_z];
        new_normal=rot*dn;
        clin_rev(k,7:9)=new_normal./norm(new_normal);
    elseif idx>no_cp+no_cpin+no_clin && idx<=no_cpln+no_cpin+no_cpln
        k=idx-(no_cp+no_cpin+no_clin);
        local_z=cpln_rev(k,4:6)';
        local_y=cross(local_z,local_x);
        rot=[local_x local_y local_z];
        new_normal=rot*dn;
        cpln_rev(k,4:6)=new_normal./norm(new_normal);
    elseif idx>no_cpln+no_cpin+no_cpln && idx<=no_cp+no_cpin+no_cpln+no_clin
        k=idx-(no_cpln+no_cpin+no_cpln);
        local_z=cpln_rev(k,4:6)';
        local_y=cross(local_z,local_x);
        rot=[local_x local_y local_z];
        new_normal=rot*dn;
        clin_rev(k,7:9)=new_normal./norm(new_normal);
    elseif idx>no_cp+no_cpin+no_cpln+no_clin && idx<=no_cp+no_cpin+no_cpln+no_cpln
        k=idx-(no_cp+no_cpin+no_cpln+no_clin);
        local_z=cpln_rev(k,4:6)';
        local_y=cross(local_z,local_x);
        rot=[local_x local_y local_z];
        new_normal=rot*dn;
        clin_rev(k,7:9)=new_normal./norm(new_normal);
    end
end

[R]=rate_motset(combo_set,mot_half_reduc,cp_set,whatkind_cp)
% Rate a specified set of constraints to resist a specified set
of motion
% filename: rate_motset.m
% Input variables: combo_set,mot_half_reduc,cp_set,whatkind_cp
% Output variables: R
% Called functions: input_wr_compose, react_wr_5_compose
%                  rate_cp, rate_cpin, rate_clin, rate_cpln1,
rate_cpln2
%
global no_cp no_cpin no_clin no_cpln
global no_cp_rev no_cpin_rev no_clin_rev no_cpln_rev
global cp_rev clin_rev cpln_rev cpln_prop_rev pts_rev
max_d_rev
global combo_dup_idx_org remain_idx combo
m=1;

Rcp_pos=[];Rcp_neg=[];Rcpin=[];Rclin_pos=[];Rclin_neg=[];
Rcpin_pos=[];Rcpin_neg=[];Rpos=[];Rneg=[];

if strcmp(whatkind_cp,'additional')==1
    %modify using the appended cp indexes
    no_cp=no_cp_rev;
    no_cpin=no_cpin_rev;
    no_clin=no_clin_rev;
    no_cpln=no_cpln_rev;
end

%motset is half_mot format
%cpl_set is column format
% cnt1=1;cnt2=1;cnt3=1;cnt4=1;

for i=1:size(mot_half_reduc,1)

    % Note that d might change once the cp set is revised, but
it's too costly to recalculate d for all rated cp and mot

    mot=mot_half_reduc(:,i,:);
    [input_wr dmy1]=input_wr_compose(mot,pts_rev,max_d_rev);

    [react_wr_5]=react_wr_5_compose(combo_set(i,:),mot(7:9)',whatkind_
cp);

    for j=1:length(cp_set);

        % The procedure below is implemented to handle cases when
modified constraints
        % belonging to the pivot wrench changes the pivot wrench
matrix ranks
        cp_eval=cp_set(j);
        cp_eval_idx=find(combo_set(i,:)==cp_eval);
        pivot_wr=react_wr_5;
        if isempty(cp_eval_idx)==0
            pivot_wr(cp_eval_idx,:)=[];
        end
        % if want to increase accuracy, insert a linear
dependence
        % eliminator here, and get rid of the pivot_wr rank
check
        swap=pivot_wr;
        while rank(swap)>5
            swap(6,:)=[];
        end
        alt_idx=find(combo_dup_idx_org==remain_idx(i));
        if rank(pivot_wr)<5
            for s=1:size(alt_idx,1)
                swap=swap([alt_idx s],:);
            end
        end
        if rank(swap)==5
            [swap]=react_wr_5_compose(combo(alt_idx(s),:),mot(7:9)',whatkind_
cp);
        end
    end
end

```

```

        break
    end
end
pivot_wr=swap;

% Rate the constraint quality to resist motion
if cp_eval<no_cp
    k=cp_eval;
    [Rpos(m,j) Rneg(m,j)]=rate_cp(mot,pivot_wr,input_wr,
cp_rev(k,:));
elseif cp_eval>no_cp && cp_eval<=no_cp+no_cpin
    k=cp_eval-no_cp;

[Rpos(m,j)]=rate_cpin(mot,pivot_wr,input_wr,cpin_rev(k,:));
elseif cp_eval>no_cp+no_cpin &&
cp_eval<=no_cp+no_cpin+no_clin
    k=cp_eval-(no_cp+no_cpin);
    [Rpos(m,j) Rneg(m,j)]=rate_cpin(mot,pivot_wr,
input_wr,cpin_rev(k,:));
elseif cp_eval>no_cp+no_cpin+no_clin &&
cp_eval<=no_cp+no_cpin+no_cpin+no_cpin
    k=cp_eval-(no_cp+no_cpin+no_cpin);
    [Rpos(m,j) Rneg(m,j)]=rate_cpin1(mot,pivot_wr,
input_wr,cpin_rev(k,:));
elseif cpin_rev(k,7)==1
    [Rpos(m,j) Rneg(m,j)]=rate_cpin1(mot,pivot_wr,
input_wr,cpin_rev(k,:));
elseif cpin_rev(k,7)==2
    [Rpos(m,j) Rneg(m,j)]=rate_cpin2(mot,pivot_wr,
input_wr,cpin_rev(k,:));
end
end

% Just in case the procedure above did not handle the
rank problem
if rank(pivot_wr)<5
    Rpos(m,j)=inf; Rneg(m,j)=inf;
end
m=m+1; %Advance rating matrix row index
end
% Merge all ratings
R=[Rpos;Rneg];

```

```

function resize_rectpln_srch(x_grp, cp_rev_in_group, pln_size_srch)
% Resize rectangular plane constraints
% filename: resize_rectpln_srch.m
%
global no_cp no_cpin no_clin no_cpln
global cpln_prop_rev

% Adjustment because parameter x ranges from -1 to 1
pln_length=pln_size_srch(1)+(x_grp(1)+1)/2*(pln_size_srch(2)-pln_size_srch(1));
pln_width=pln_size_srch(3)+(x_grp(2)+1)/2*(pln_size_srch(4)-pln_size_srch(3));

% Apply this to all rectangular plane constraints
for i=1:length(cp_rev_in_group)
    idx=cp_rev_in_group(i);
    if idx<=no_cp
        disp('Plane size search is not applicable to point
constraints');
        return
    elseif idx>no_cp && idx<=no_cp+no_cpin
        disp('Plane size search is not applicable to pin
constraints');
        return
    elseif idx>no_cp+no_cpin && idx<=no_cp+no_cpin+no_clin
        disp('Plane size search is not applicable to line
constraints');
        return
    elseif idx>no_cp+no_cpin+no_clin &&
idx<=no_cp+no_cpin+no_cpin+no_cpin
        k=idx-(no_cp+no_cpin+no_cpin);
        cpln_prop_rev(k,10)=line_length;
        disp('Line size search is not applicable to plane
constraints');
        return
    end

```

```

function resize_circpln_srch(x_grp, cp_rev_in_group, pln_size_srch)
% Resize circular plane constraints
% filename: resize_circpln_srch.m
%
global no_cp no_cpin no_clin no_cpln
global cpln_prop_rev

% Adjustment because parameter x ranges from -1 to 1
pln_rad=pln_size_srch(5)+(x_grp+1)/2*(pln_size_srch(6)-pln_size_srch(5));

```

Dependency Report

Run Report on Current Directory

- Show child functions Show parent functions (current dir. only)
- Show subfunctions

Built-in functions and files in toolbox/matlab are not shown

files	Children (called functions)	Parents (calling functions, current dir. only)
calc_d		input_wr_compose rate_specmot
combo_preproc		main optim_main_dd
cp_rev_to_wrench		optim_rev rate_specmot
cp_to_wrench		main optim_main_dd
form_combo_wrench		main_loop
histogr		main optim_main_dd
input_menu	other : F:\Documents\Scripts\Input files\case1a_chair_height.m other : F:\Documents\Scripts\Input files\case1b_chair_height_angle.m other : F:\Documents\Scripts\Input files\case2a_cube_scalability.m other : F:\Documents\Scripts\Input files\case2b_cube_tradeoff.m other : F:\Documents\Scripts\Input files\case3a_cover_leverage.m other : F:\Documents\Scripts\Input files\case3b_cover_symmetry.m other : F:\Documents\Scripts\Input files\case3c_cover_orient.m other : F:\Documents\Scripts\Input files\case4a_endcap_tradeoff.m other : F:\Documents\Scripts\Input files\case4b_endcap_circlinsrch.m other : F:\Documents\Scripts\Input files\case5a_printer_4screws_orient.m other : F:\Documents\Scripts\Input files\case5b_printer_4screws_line.m other : F:\Documents\Scripts\Input files\case5c_printer_snap_orient.m other : F:\Documents\Scripts\Input files\case5d_printer_snap_line.m other : F:\Documents\Scripts\Input files\case5e_printer_partingline.m	main

m	other : F:\Documents\Scripts\Input files\case5f1_printer_line_size.m other : F:\Documents\Scripts\Input files\case5f2_printer_sideline_size.m other : F:\Documents\Scripts\Input files\case5g_printer_5d.m other : F:\Documents\Scripts\Input files\case5rev_a_printer_2screws.m other : F:\Documents\Scripts\Input files\case5_printer_allscrews.m other : F:\Documents\Scripts\Input files\case5rev_b_printer_flat_partingline.m	
input_preproc		main
input_wr_compose	current dir : calc_d	main_loop main_specmot_orig rate_motset
inputfile_check	unknown : cp unknown : cpin unknown : clin unknown : cpin unknown : cpin_prop	main
line_orient1dsrch		optim_rev rate_specmot
main	current dir : input_menu current dir : inputfile_check current dir : input_preproc current dir : cp_to_wrench current dir : combo_preproc current dir : main_loop current dir : rating current dir : result_open current dir : report current dir : histogr current dir : result_close current dir : optim_main_rev current dir : optim_postproc current dir : optim_main_red current dir : sens_analysis_pos current dir : sens_analysis_orient current dir : main_specmot_orig unknown : inputfile_ok unknown : inputfile_error unknown : no_dim unknown : WTR_optim_chg unknown : MRR_optim_chg unknown : MTR_optim_chg unknown : TOR_optim_chg unknown : mot_set	
main_loop	current dir : form_combo_wrench current dir : rec_mot current dir : input_wr_compose current dir : react_wr_5_compose current dir : rate_cp current dir : rate_cpin current dir : rate_clin current dir : rate_cpin1 current dir : rate_cpin2	main optim_main_dd optim_rev
main_specmot_optim	current dir : rate_specmot unknown : specmot unknown : inputfile	
main_specmot_o	current dir : input_wr_compose	main

<u>rig</u>	current dir : <u>rate cp</u> current dir : <u>rate cpin</u> current dir : <u>rate clin</u> current dir : <u>rate cpln1</u> current dir : <u>rate cpln2</u> current dir : <u>rating</u> unknown : <u>specmot</u>			<u>_orig</u> <u>rate_motset</u> <u>rate_specmot</u>
<u>move_curlin_srch</u>		optim_rev		<u>rate cpin</u>
<u>move_lin_srch</u>		optim_rev rate_specmot		<u>rate cpln1</u>
<u>move_pln_srch</u>		optim_rev rate_specmot		<u>rate cpln2</u>
<u>move_pt_srch</u>	unknown : <u>cp_rev_in_group</u>	rate_specmot		<u>rate motset</u>
<u>optim_main_add</u>	current dir : <u>cp_to_wrench</u> current dir : <u>rate_motset</u> current dir : <u>combo_preproc</u> current dir : <u>main_loop</u> current dir : <u>rating</u> current dir : <u>histogr</u> unknown : <u>Ri</u> unknown : <u>add_cp_type</u> unknown : <u>add_cp</u> unknown : <u>combo_proc_org</u> unknown : <u>mot_half_org</u> unknown : <u>mot_all_org</u>			<u>input_wr_compose</u> <u>react_wr_5_compose</u> <u>rate_cp</u> <u>rate_cpin</u> <u>rate_clin</u> <u>rate_cpln1</u> <u>rate_cpln2</u>
<u>optim_main_red</u>	current dir : <u>rating</u> unknown : <u>total_cp</u> unknown : <u>no_red</u> unknown : <u>combo</u> unknown : <u>combo_dup_idx</u> unknown : <u>combo_proc_org</u> unknown : <u>no_mot_half</u> unknown : <u>Ri</u> unknown : <u>mot_all_org</u> unknown : <u>Rating_all_org</u> unknown : <u>inputfile</u>	main		<u>move_pt_srch</u> <u>move_lin_srch</u> <u>move_pln_srch</u> <u>orientid_srch</u> <u>orient2d_srch</u> <u>line_orientid_srch</u> <u>resize_lin_srch</u> <u>resize_rectpln_srch</u> <u>resize_circpln_srch</u> <u>cp_rev_to_wrench</u> <u>calc_d</u> <u>rate_cp</u> <u>rate_cpin</u> <u>rate_clin</u> <u>rate_cpln1</u> <u>rate_cpln2</u> <u>rating</u> <u>move_circlin_srch</u>
<u>optim_main_rev</u>	current dir : <u>optim_rev</u> unknown : <u>combo_proc_org</u> unknown : <u>no_mot_half</u> unknown : <u>mot_half_org</u> unknown : <u>mot_all_org</u> unknown : <u>Ri</u> unknown : <u>no_step</u> unknown : <u>Rating_all_org</u>	main sens_analysi_s_orient sens_analysi_pos		<u>rating</u>
<u>optim_postproc</u>	unknown : <u>WTR_optim_all</u> unknown : <u>MRR_optim_all</u> unknown : <u>MTR_optim_all</u> unknown : <u>TOR_optim_all</u>	main sens_analysi_s_orient sens_analysi_pos		<u>react_wr_5_compose</u>
<u>optim_rev</u>	current dir : <u>move_lin_srch</u> current dir : <u>move_curlin_srch</u> current dir : <u>move_pln_srch</u> current dir : <u>orientid_srch</u> current dir : <u>orient2d_srch</u> current dir : <u>line_orientid_srch</u> current dir : <u>resize_lin_srch</u> current dir : <u>resize_rectpln_srch</u> current dir : <u>resize_circpln_srch</u> current dir : <u>cp_rev_to_wrench</u> current dir : <u>rate_motset</u> current dir : <u>main_loop</u> current dir : <u>rating</u>	optim_main_rev		<u>rec_mot</u>
<u>orientid_srch</u>		optim_rev rate_specmot		<u>report</u>
<u>orient2d_srch</u>		optim_rev rate_specmot		<u>resize_circpln_srch</u>
<u>rate_clin</u>		main_loop main_specmot _orig rate_motset rate_specmot		<u>resize_lin_srch</u>
<u>rate_cp</u>		main_loop main_specmot		

<u>resize_rectpln</u>		optim_rev rate_specmot
<u>result_close</u>	unknown : <u>timestart</u> unknown : <u>result</u>	main
<u>result_open</u>		main
<u>sens_analysis</u> <u>orient</u>	current dir : <u>optim_main_rev</u> current dir : <u>optim_postproc</u> unknown : <u>total_cp</u> unknown : <u>no_cp</u> unknown : <u>cp</u> unknown : <u>no_cpin</u> unknown : <u>cpin</u> unknown : <u>no_clin</u> unknown : <u>clin</u> unknown : <u>no_cpln</u> unknown : <u>cpln</u> unknown : <u>WTR_optim_chg</u> unknown : <u>MRR_optim_chg</u> unknown : <u>MTR_optim_chg</u> unknown : <u>TOR_optim_chg</u> unknown : <u>no_dim</u>	main
<u>sens_analysis</u> <u>pos</u>	current dir : <u>optim_main_rev</u> current dir : <u>optim_postproc</u> unknown : <u>total_cp</u> unknown : <u>no_cp</u> unknown : <u>cp</u> unknown : <u>no_cpin</u> unknown : <u>cpin</u> unknown : <u>no_clin</u> unknown : <u>clin</u> unknown : <u>no_cpln</u> unknown : <u>cpln</u> unknown : <u>WTR_optim_chg</u> unknown : <u>MRR_optim_chg</u> unknown : <u>MTR_optim_chg</u> unknown : <u>TOR_optim_chg</u> unknown : <u>no_dim</u> unknown : <u>Rating_all_org</u>	main
<u>sens_analysis</u> <u>postproc</u>	unknown : <u>total_cp</u> unknown : <u>SAO_WTR</u> unknown : <u>SAO_MRR</u> unknown : <u>SAO_MTR</u> unknown : <u>SAO_TOR</u> unknown : <u>SAP_WTR</u> unknown : <u>SAP_MRR</u> unknown : <u>SAP_MTR</u> unknown : <u>SAP_TOR</u> unknown : <u>no_step</u>	
<u>table_mot</u>		report

APPENDIX B

CASE STUDY GEOMETRIES

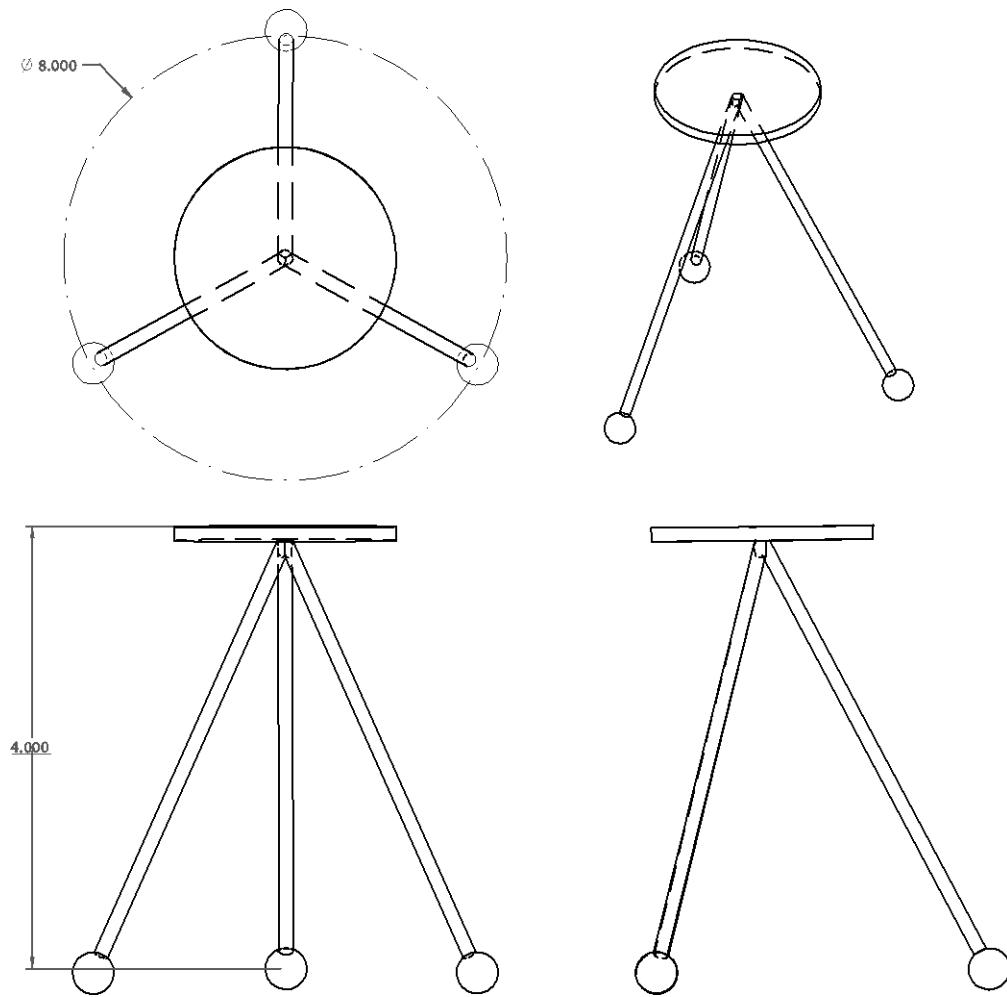


Figure B.1 Thompson's chair case study geometry

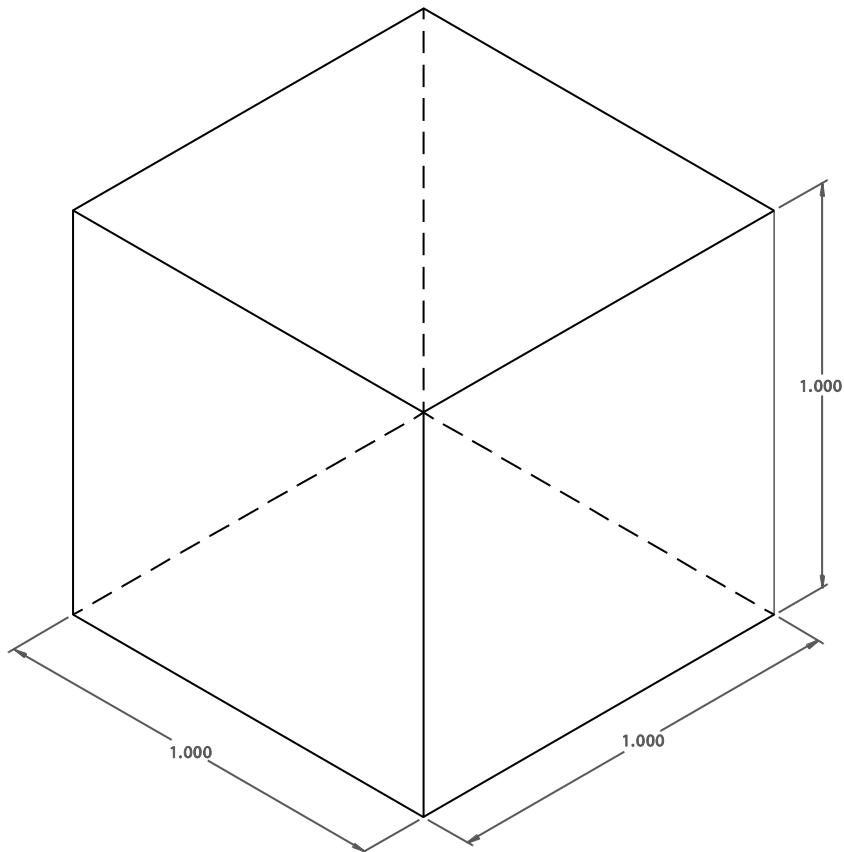


Figure B.2 Generic cube case study geometry

298

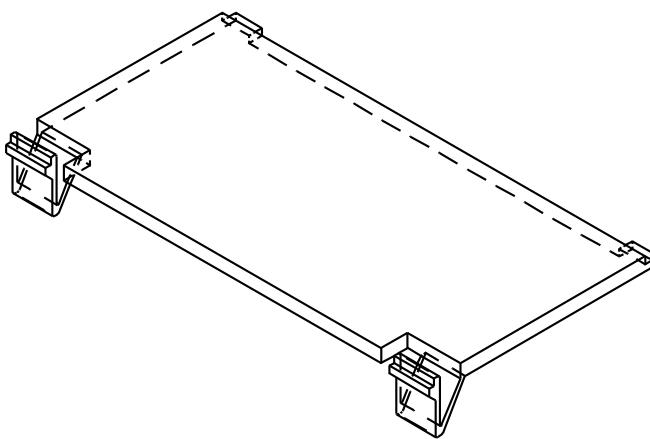
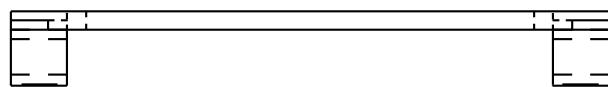
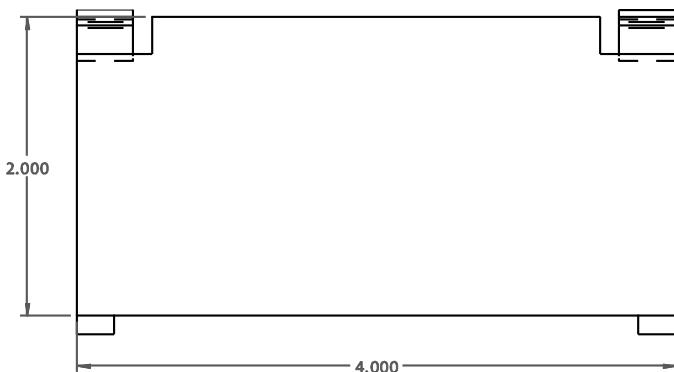


Figure B.3 Battery cover case study geometry

299

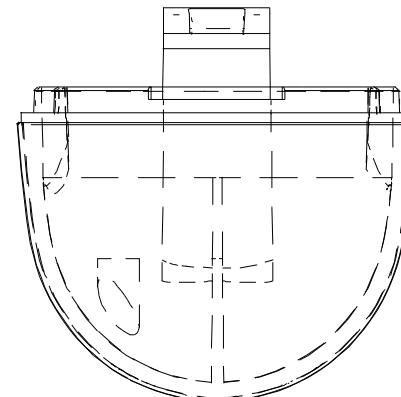
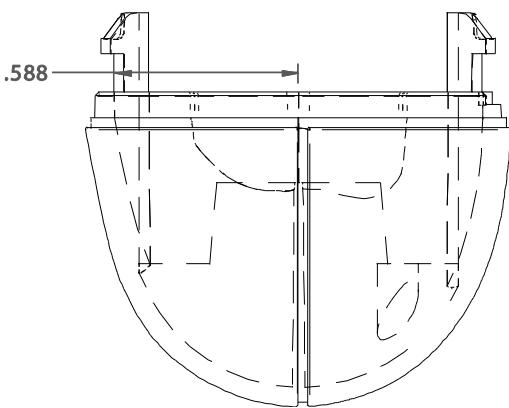
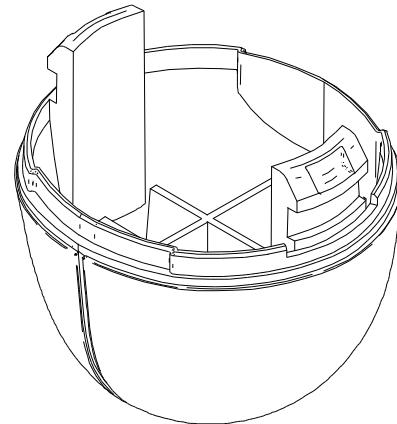
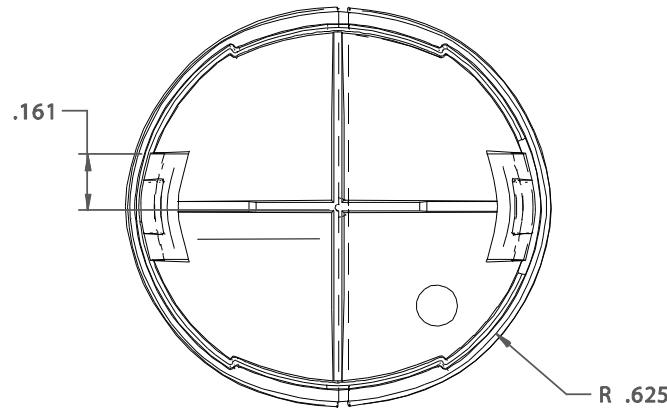


Figure B.4 End cap assembly case study geometry

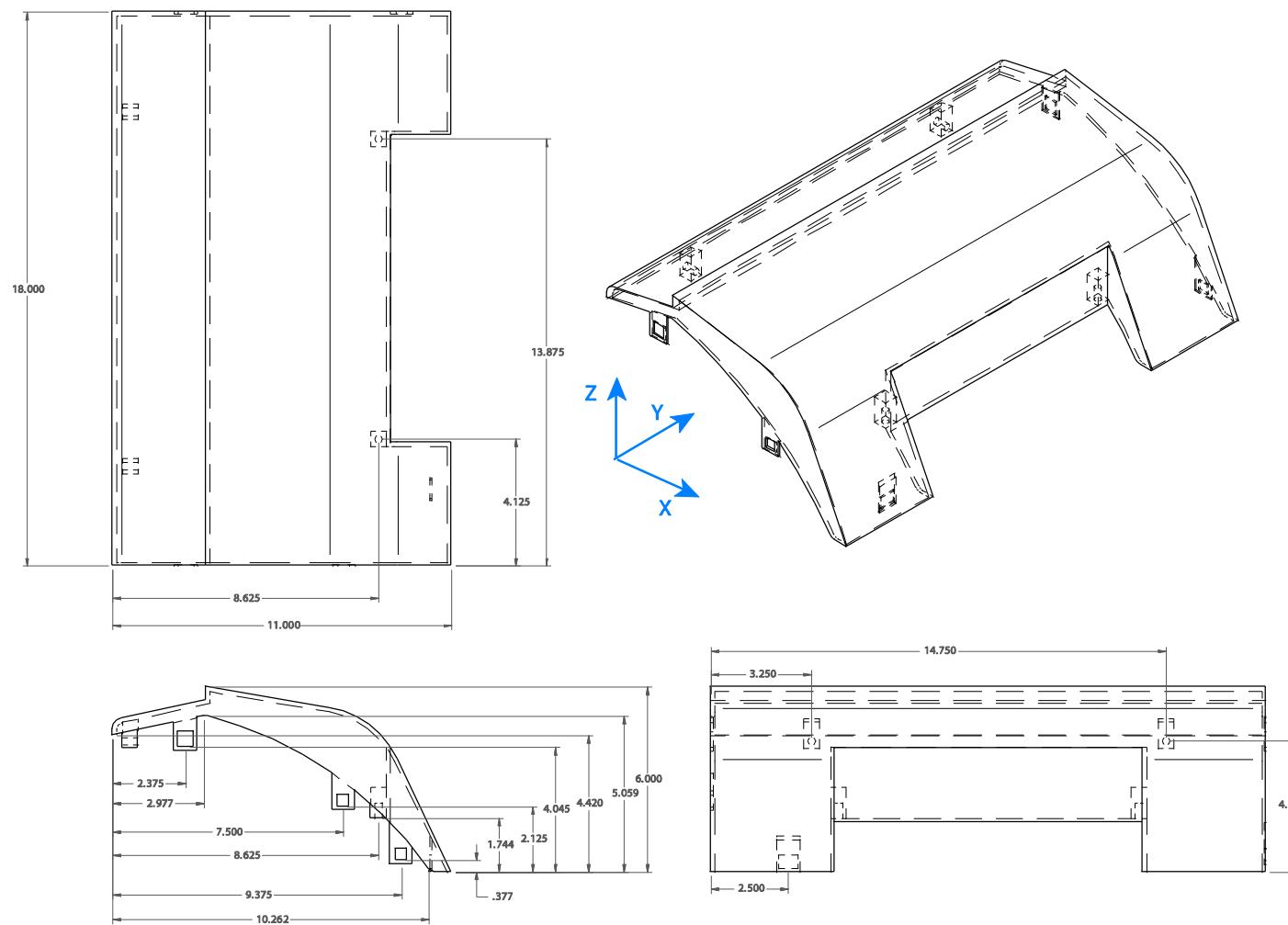


Figure B.5 Printer housing assembly case study geometry