

**AN INTEGRATED PRODUCT – PROCESS DEVELOPMENT  
(IPPD) BASED APPROACH FOR ROTORCRAFT DRIVE SYSTEM  
SIZING, SYNTHESIS AND DESIGN OPTIMIZATION**

A Thesis

Presented to

The Academic Faculty

by

Sylvester Vikram Ashok

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy in Aerospace Engineering

Daniel Guggenheim School of Aerospace Engineering

Georgia Institute of Technology

August 2013

Copyright © 2013 by Sylvester Vikram Ashok

**AN INTEGRATED PRODUCT – PROCESS DEVELOPMENT  
(IPPD) BASED APPROACH FOR ROTORCRAFT DRIVE SYSTEM  
SIZING, SYNTHESIS AND DESIGN OPTIMIZATION**

Approved by:

Dr. Daniel P. Schrage, Advisor  
Professor,  
School of Aerospace Engineering  
*Georgia Institute of Technology*

Dr. Vitali V. Volovoi  
Assistant Professor,  
School of Aerospace Engineering  
*Georgia Institute of Technology*

Dr. Dimitri N. Mavris  
Professor,  
School of Aerospace Engineering  
*Georgia Institute of Technology*

Dr. Stephen C. Skinner  
Director, Systems Engineering and  
Engineering Operations  
*Bell Helicopter, Textron*

Dr. Brian J. German  
Assistant Professor,  
School of Aerospace Engineering  
*Georgia Institute of Technology*

Date Approved: 29 May 2013

*'Ad Dei Gloriam'*

## ACKNOWLEDGEMENTS

There are many people I must thank for making my work and this thesis possible. It is but impossible for someone to go through the PhD process, in my opinion, without all the help and support rendered by many individuals, both directly and indirectly.

I would like to thank Dr. Schrage for taking me under his mentoring, for the confidence, support and care he has shown towards my academic pursuits. Dr. Schrage has been an incredible mentor and father figure for the last five years; he gave me the independence to study this topic but constantly mentored me to understand the ‘big picture’. I have had the opportunity, under his guidance, to work on multiple design competitions that gave me a strong foundation to understand the complexity of a ‘clean sheet’ design.

I thank my thesis proposal reading committee members Dr. Mavris and Dr. Brian German for their helpful advice and guidance in problem formulation, hypothesis testing and research objectives. I thank Dr. Volovoi and Dr. Steve Skinner for the involvement, suggestions and for accepting to be on my thesis committee.

Georgia Tech has given me a great platform to grow and learn; something that has been incredibly important to me. Simply gaining knowledge would have been pointless without developing a different way of thinking.

Over the past six years at Georgia Tech, I’ve had the great fortune of working with and learning from a diverse group of peers. I thank the students who worked with me on class and research projects, design competitions and conference papers. I thank Apinut (Nate) Sirirojvisuth and Mike Roberts for all the helpful discussions; Marc Mugnier, Alexander Robledo, Brian Wade and other design competition teammates who

helped me learn as we worked together towards that high-speed coaxial compound helicopter through the years.

I thank all my friends for making the last six years an incredibly enjoyable experience: Shaila, Sunil, Urmila, and Vaibhav for the many conversations and times together; Pablo, Rajiv, Ritu, Natasha, Alek, Charles and the many fellow helicopter enthusiasts with AHS-GT for all the fun times. I thank all my brothers and sisters from Grace Midtown who have been an invaluable addition to my life.

I thank my parents and my sister for giving me the support through my academic pursuits and teaching me many of life's important lessons yet.

Above all, I thank God for giving me this beautiful life and the great opportunities that have come with it.

# TABLE OF CONTENTS

ACKNOWLEDGEMENTS .....	iv
LIST OF TABLES .....	ix
LIST OF FIGURES .....	x
LIST OF EQUATIONS .....	xvi
LIST OF SYMBOLS AND ABBREVIATIONS .....	xviii
SUMMARY .....	xxv
1 INTRODUCTION .....	28
1.1 Problem Definition .....	28
1.2 Motivation .....	33
1.3 Research Objectives .....	45
2 LITERATURE REVIEW .....	46
2.1 Systems Engineering .....	46
2.2 Computer Aided Design .....	53
2.3 Drive System Design .....	57
2.4 Genetic Algorithm .....	68
2.5 Flexibility .....	71
2.6 Finite Element Analysis .....	76
2.7 Summary of Literature Research .....	84

3	RESEARCH QUESTIONS, CONJECTURES AND HYPOTHESES .....	86
4	METHODOLOGY .....	95
4.1	Fully-Relational Design .....	95
4.1.1	Requirements for Fully-Relational Design Implementation...	104
4.2	Gear Train Sizing .....	104
4.2.1	Bending Stress.....	106
4.2.2	Surface Contact Stress.....	107
4.2.3	Scuffing Hazard .....	108
4.2.4	Gear Rating .....	110
4.3	Optimization.....	114
4.3.1	Genetic Algorithm.....	117
4.4	Flexibility .....	123
4.5	Topology Optimization .....	125
5	IMPLEMENTATION AND RESULTS.....	127
5.1	Design Framework .....	127
5.1.1	Geometry integration .....	130
5.2	Genetic Algorithm and Optimization.....	132
5.3	Spacing Analysis .....	154
5.4	Flexibility of Design.....	163

5.5	Finite Element Analysis .....	176
5.5.1	Nonlinear Contact Analysis .....	176
5.5.2	Topology Optimization .....	197
5.6	Implementation of Methodology on Rotorcraft Drive System .....	206
5.6.1	Lifecycle Cost Analysis .....	217
6	CONCLUSION.....	218
6.1	Review of Research Questions, Conjecctures and Hypotheses .....	219
6.2	Review of Research Objectives.....	224
6.3	Contributions.....	226
6.4	Future Work .....	227
	APPENDIX.....	228
	REFERENCES .....	293

## LIST OF TABLES

Table 1. Gear materials .....	114
Table 2. Optimization variables .....	116
Table 3. Three-stage full factorial optimization result.....	135
Table 4. Constraint dependency information.....	138
Table 5. Summary of penalty techniques.....	145
Table 6. Parent selection – simulation results.....	146
Table 7. Spacing analysis designs.....	159
Table 8. Three stage and four stage designs .....	165
Table 9. DOE input table .....	167
Table 10. Face-mapped mesh vertex types [143] .....	180
Table 11. Summary of contact formulations [144].....	186
Table 12. ANSYS contact analysis setting .....	190
Table 13. FEA cases for ANSYS.....	192
Table 14. Topology optimization - Case 1.....	202
Table 15. Topology optimization result - Case 1.....	203
Table 16. Topology optimization - Case 2.....	204
Table 17. Weight distribution in different regions – Case 2.....	205
Table 18. Topology optimization results – Case 2 .....	205

## LIST OF FIGURES

Figure 1. Schematic for optimum parameter selection using the $R_F$ method [18].....	31
Figure 2. Obtaining installed power through vehicle design synthesis [19].....	32
Figure 3. Rotorcraft conceptual design process.....	32
Figure 4. Design process extended to drive system.....	33
Figure 5. Serial approach vs. CE approach [19].....	34
Figure 6. Design freedom, knowledge and cost relationship [22].....	36
Figure 7. Critical design phase.....	36
Figure 8. Design changes in serial approach.....	37
Figure 9. Moving from serial approach to IPPD approach.....	38
Figure 10. Requirements for IPPD approach.....	39
Figure 11. Design freedom and knowledge in traditional design.....	40
Figure 12. Design process reorganized to gain information earlier and to retain design freedom longer.....	40
Figure 13. Generic IPPD framework for rotorcraft preliminary design [1].....	41
Figure 14. Drive system parent requirements and constraints.....	42
Figure 15. Comparison of design time between manual and automated analysis.....	43
Figure 16. Gaps in analysis disciplines in rotorcraft design.....	44
Figure 17. Gaps in analysis disciplines and different fidelity creating gaps in integration.....	44
Figure 18. Georgia Tech generic IPPD methodology [19].....	47
Figure 19. Systems engineering process model [16].....	47
Figure 20. Disciplines involved in MDO environment [30].....	48
Figure 21. Information flow for design integration of a bearingless soft-in-plane rotor blade [24].....	51
Figure 22. Software integration in ModelCenter [23].....	52

Figure 23. Rotor flexure configuration optimization [23] .....	52
Figure 24. Parametric CAD model of a turbine blade [38].....	54
Figure 25. Complex FEA model generation automatically using CAD interface [16] ....	55
Figure 26. Parametric redesign of a single-piston engine [40] .....	56
Figure 27. Variable speed planetary gear drive system for a twin-engine coaxial compound configuration [53] .....	58
Figure 28. Design process and response surface methodology [47].....	58
Figure 29. Weights estimation of drive system [47].....	59
Figure 30. Multistage planetary weight minimization technique [47].....	59
Figure 31. Square cube law block [12] .....	60
Figure 32. Weight – torque relation [47] .....	61
Figure 33. Sun gear bending stress against speed for multiple cases [57].....	65
Figure 34. 2-Stage gear train optimization [58].....	66
Figure 35. Objective function and constraints for GA [58] .....	67
Figure 36. Simple model relating a system’s life span and its flexibility [89] .....	72
Figure 37. Flexible design in terms of system objectives and environment [16] .....	73
Figure 38. Finite element model of helical tooth [107] .....	79
Figure 39. Plane stress model of high contact ratio gears [108].....	80
Figure 40. Plane strain analysis of spur gear [109].....	81
Figure 41. Parametric gear modeling and FEA setup [109] .....	81
Figure 42. Contact model boundary conditions [110] .....	82
Figure 43. Bending stress at gear tooth base – tension and compression [104] .....	83
Figure 44. Stress distribution from dynamic contact analyses [111].....	84
Figure 45. Hypothesis 2 – case 1 .....	93
Figure 46. Hypothesis 2 – case 2 .....	93
Figure 47. Consequence of Hypothesis 2 .....	94
Figure 48. Relational design example [23].....	95

Figure 49. Fuselage – former relational design example.....	97
Figure 50. Relational design process for fuselage - former.....	98
Figure 51. Former member cross-section.....	99
Figure 52. Fully-relational design schematic for fuselage - former.....	100
Figure 53. Fully-relational design process for fuselage - former.....	100
Figure 54. Full-relational design logic.....	101
Figure 55. Single main rotor helicopter configuration.....	102
Figure 56. Initial helicopter fuselage and engine housing configuration.....	102
Figure 57. Parametrically offset product.....	103
Figure 58. Modified fuselage and engine housing to comply with shift in rotor axis and engine location.....	103
Figure 59. Modes of gear failure [114].....	105
Figure 60. Illustration of bending stress.....	107
Figure 61. Stresses in region of tooth contact [64].....	108
Figure 62. Initial scuffing [114].....	109
Figure 63. Moderate scuffing [114].....	109
Figure 64. Destructive scuffing [114].....	110
Figure 65. Bending and compressive stress vs. power [47].....	111
Figure 66. AGMA bending geometry factor J.....	112
Figure 67. Flash temperature along line of action.....	112
Figure 68. Two point crossover example [62].....	119
Figure 69. Genetic Algorithm Structure [123].....	120
Figure 70. Quantification of flexibility.....	124
Figure 71. Topology optimization setup in HyperMesh.....	126
Figure 72. Drive system sizing and analysis framework.....	127
Figure 73. Fully-relational design implementation for three-stage gear system.....	129
Figure 74. Spur gear geometry [65].....	130

Figure 75. 2-D gear profile generated in CATIA .....	131
Figure 76. Helical gear pair generated in CATIA.....	131
Figure 77. Gear stage sizing routine .....	133
Figure 78. Static linear penalty – stage 2 ( $r_p = 50$ ) .....	138
Figure 79. Static linear penalty – stage 2 ( $r_p = 500$ ).....	139
Figure 80. Static linear penalty – stage 3 ( $r_p = 50$ ).....	139
Figure 81. Static nonlinear penalty – stage 2 ( $r_p = 50$ ).....	140
Figure 82. Static nonlinear penalty – stage 3 ( $r_p = 50$ ).....	140
Figure 83. Dynamic linear penalty - stage 2 ( $r_p = 10$ , gen = 50) .....	142
Figure 84. Dynamic linear penalty - stage 3, ( $r_p = 10$ , gen = 50).....	142
Figure 85. Dynamic nonlinear penalty - stage 2, ( $r_p = 10$ , gen = 50).....	143
Figure 86. Dynamic nonlinear penalty - stage 3, ( $r_p = 100$ , gen = 100).....	143
Figure 87. Average fitness vs. dynamic penalty factor B.....	144
Figure 88. Average fitness vs. dynamic penalty coefficient B for dynamic linear.....	144
Figure 89. Adaptive GA results for static penalty .....	148
Figure 90. Adaptive GA results for dynamic penalty .....	148
Figure 91. Adaptive GA results for static nonlinear penalty .....	149
Figure 92. Adaptive GA results for four-stage gear train (static nonlinear).....	149
Figure 93. Parallel GA .....	150
Figure 94. Migration in GA .....	151
Figure 95. Average fitness vs. number of sub-populations .....	152
Figure 96. Genetic algorithm for gear train design.....	153
Figure 97. Three stage reduction drive inside a cylindrical housing .....	154
Figure 98. Three stage reduction drive inside a cylindrical housing (front view).....	155
Figure 99. Spacing analysis algorithm.....	156
Figure 100. Three-stage gear spacing analysis problem.....	157

Figure 101. Three-stage gear and housing - spacing analysis algorithm implemented in MATLAB.....	158
Figure 102. Spacing configuration for designs A and B with 13 in. radius housing .....	160
Figure 103. Spacing configuration for designs A and B with 12 in. radius housing .....	161
Figure 104. Three-stage gear train .....	163
Figure 105. Four-stage gear train.....	164
Figure 106. Distribution of input variables – case 1 .....	168
Figure 107. Three stage PMF result from simulation – case 1 .....	169
Figure 108. Four stage PMF result from simulation – case 1 .....	170
Figure 109. Three stage CDF result from simulation – case 1 .....	171
Figure 110. Four stage CDF result from simulation – case 1 .....	171
Figure 111. Distribution of input variables – case 2.....	172
Figure 112. Three stage PMF result from simulation – case 2 .....	173
Figure 113. Four stage PMF result from simulation – case 2.....	174
Figure 114. Three stage CDF result from simulation – case 2 .....	175
Figure 115. Four stage CDF result from simulation – case 2.....	175
Figure 116. FEA process flow .....	177
Figure 117. Gear geometry for FEA .....	178
Figure 118. Vertex types [142] .....	179
Figure 119. Face split into sub-mappable faces.....	180
Figure 120. Mapped mesh.....	181
Figure 121. Mesh refinement for contact stress.....	181
Figure 122. Mesh refinement for bending stress .....	182
Figure 123. Contact modeling [144].....	184
Figure 124. Normal Lagrange vs. penalty-based methods [144].....	185
Figure 125. Adjust-to-touch formulation [144] .....	188
Figure 126. Pinion – gear frictional contact.....	190

Figure 127. von-Mises stress contours for transient analysis .....	192
Figure 128. Pinion bending stress - AGMA vs. FEA .....	193
Figure 129. Contact stress – AGMA vs. FEA .....	194
Figure 130. Percentage variation between AGMA and FEA .....	194
Figure 131. Contour plot for pinion bending stress - tensile .....	195
Figure 132. Contour plot for pinion bending stress - compression .....	196
Figure 133. Contour plot for contact stress.....	196
Figure 134. HyperMesh model of gear .....	198
Figure 135. OptiStruct result without draw constraint.....	200
Figure 136. OptiStruct result few cyclic instances .....	200
Figure 137. OptiStruct result with higher level cyclic constraint .....	201
Figure 138. Topology optimization - Case 1 .....	202
Figure 139. Topology optimization result - Case 1 .....	203
Figure 140. Topology optimization - Case 2 .....	204
Figure 141. Topology optimization results - Case 2.....	205
Figure 142. Fully-relational design for drive system.....	206
Figure 143. Drive system design framework.....	208
Figure 144. Quality function deployment matrix .....	209
Figure 145. Integration of design tools in ModelCenter [32] .....	211
Figure 146. Planetary design concept (rear view) .....	212
Figure 147. Split-torque design concept (top view).....	213
Figure 148. Optimization setup [149].....	214
Figure 149. Geometry integration using CATIA [20] .....	216
Figure 150. Drive system geometry in CATIA [149].....	216

## LIST OF EQUATIONS

Equation 1. Weight - torque relation using square cube law [47] .....	60
Equation 2. Weight - torque relation as per AMCP 706 -201 [11].....	61
Equation 3. Pinion solid rotor volume [55] .....	62
Equation 4. Gear solid rotor volume [55] .....	62
Equation 5. Gear set solid rotor volume [55] .....	62
Equation 6. RTL gear box weight formula [56] .....	63
Equation 7. RTL shafting weight formula [56] .....	63
Equation 8. Boeing-Vertol main rotor drive system weight [56]. .....	64
Equation 9. Boeing-Vertol tail rotor drive system weight [56] .....	64
Equation 10. Gear weight relation [57].....	65
Equation 11. Gear weight relation [57].....	66
Equation 12. Adaptive crossover and mutation probabilities [79] .....	71
Equation 13. Weight using solid rotor volume .....	105
Equation 14. Volume of planetary gear system .....	106
Equation 15. Roulette wheel selection [121] .....	118
Equation 16. Adaptive cross-over and mutation rates .....	122
Equation 17. Static linear penalty .....	137
Equation 18. Static nonlinear penalty .....	137
Equation 19. Dynamic linear penalty.....	141
Equation 20. Dynamic nonlinear penalty.....	141
Equation 21. Coefficient values for adaptive crossover and mutation rates.....	147
Equation 22. Spacing penalty .....	162
Equation 23. Response surface equation [140].....	166
Equation 24. Pure penalty formulation .....	184
Equation 25. Augmented Lagrange formulation .....	185

Equation 26. OEC for drive system .....	213
---	-----

## LIST OF SYMBOLS AND ABBREVIATIONS

AGMA	American Gear Manufacturers Association
AMCP	Army Materiel Control Pamphlet
AOF	Aggregated Objective Function
APDL	ANSYS Parametric Design Language
C3	Command, Control and Communications
CAD	Computer Aided Design
CAE	Computer Aided Engineering
CAM	Computer Aided Manufacturing
CAPP	Computer Aided Process Planning
CBEM	Combined Blade Element and Momentum
CDF	Cumulative Distribution Function
CDM	Concurrent Design and Manufacturing
CE	Concurrent Engineering
CG	Center of Gravity
CIRADS	Concept Independent Rotorcraft Analysis and Design Software
DFA	Design for Assembly/Automation
DFC	Design for Cost
DFM	Design for Manufacturing

DFMA	Design for Manufacturing and Assembly
DOC	Direct Operating Cost
DOE	Design of Experiments
FEA	Finite Element Analysis
FFSO	Full Factorial Sub-Optimizer
FH	Flight Hours
FMS	Flexible Manufacturing Systems
FRD	Fully-Relational Design
GA	Genetic Algorithm
IMF	Installation Management File
IOC	Indirect Operating Cost
IPPD	Integrated Product and Process Development
IPT	Integrated Product Team
LCC	Lifecycle Cost
LHX	Light Helicopter Experimental
MCP	Maximum Continuous Power
MDAO	Multidisciplinary Design Analysis and Optimization
MDF	Major Dimension File
MDO	Multidisciplinary Design Optimization

MI	Manufacturability Index
MOO	Multi - Objective Optimization
OEC	Overall Evaluation Criterion
OEM	Original Equipment Manufacturer
PDF	Probability Density Function
PLE	Product Lifecycle Engineering
PLM	Product Lifecycle Management
PMF	Probability Mass Function
PRG	Product Relation Geometry
PSD	Preference Set-based Design
QE	Quality Engineering
QFD	Quality Function Deployment
RAH	Reconnaissance and Attack Helicopter
RDS	Robust Design Simulation
RDT&E	Research, Development, Testing and Engineering
RFP	Request for Proposal
RSE	Response Surface Equation
RSM	Response Surface Methodology
SBCE	Set-Based Concurrent Engineering

SCEA	Society of Cost Estimating and Analysis
SDF	Surface Definition File
SE	Systems Engineering
SFC	Specific Fuel Consumption
SI	Structural Integrity
SMR	Single Main Rotor
TBO	Time between Overhaul
TO	Topology Optimization
VTOL	Vertical Take-off and Landing
$A_{\pi}$	Flat plate drag area
$B$	Dynamic penalty coefficient
$DL$	Disk loading
$FW$	Face Width
$G$	Constraint
$H_p$	Installed power
$HP_{mr}$	Main rotor horsepower
$HP_{tr}$	Tail rotor horsepower
$I_p$	Power loading
$K$	Surface durability factor

$K_b$	Rim thickness factor
$K_m$	Load distribution factor
$K_o$	Overload factor
$K_R$	Reliability factor
$K_s$	Size factor
$K_T$	Temperature factor
$K_v$	Dynamic factor
$M$	Gear material
$N$	Number of teeth
$N_g$	Number of gears
$P^k$	Population member
$P_d$	Diametral Pitch
$Q_{mr}$	Main rotor torque
$R_F$	Ratio of fuel weight to gross weight
$R_{mr}$	Main rotor radius
$R_{tr}$	Tail rotor radius
$S_{co}$	Scoring/scuffing hazard
$S_F$	Safety factor for bending
$S_H$	Safety factor for contact

$V_{Tmr}$	Main rotor tip speed
$V_{Ttr}$	Tail rotor tip speed
$W$	Gear weight
$W_G$	Gross weight
$W_{gs}$	Gearset weight
$Y_N$	Stress cycle factor for bending
$Z_N$	Stress cycle factor for contact
$\alpha_c$	Rack-cutter profile angle
$d_p$	Pitch diameter
$m_c$	Contact ratio
$m_g$	Gear ratio / mechanical advantage
$ocd$	Outer cone distance
$p, g$	Pinion, gear (superscript)
$p_c$	Crossover probability
$p_m$	Mutation probability
$p_{mi}$	Migration probability
$r_p$	Penalty factor
$rpm_{mr}$	Main rotor RPM
$rpm_{tr}$	Tail rotor RPM

$t_{oil}$	Oil temperature
$t_{oil\ avg}$	Average oil temperature
$t_{oil\ in}$	Oil in temperature
$t_{oil\ out}$	Oil out temperature
$w_p$	Weight penalty function
$\psi$	Helical angle
$\varphi$	Pressure angle
$\eta_{tr}$	Transmission efficiency
$\Omega_{mr}$	Main rotor rotational velocity
$\Omega_{pt}$	Power turbine rotational velocity
$\Omega_{tr}$	Tail rotor rotational velocity
$\sigma$	Solidity
$\sigma_{ab}$	Allowable bending stress
$\sigma_{ac}$	Allowable contact stress
$\sigma_b$	Bending stress
$\sigma_c$	Contact stress
$\sigma_b'$	Permissible bending stress
$\sigma_c'$	Permissible contact stress

## SUMMARY

Engineering design may be viewed as a decision making process that supports design tradeoffs. The designer makes decisions based on information available and engineering judgment. The designer determines the direction in which the design must proceed, the procedures that need to be adopted, and develops a strategy to perform successive decisions. The design is only as good as the decisions made, which is in turn dependent on the information available. Information is time and process dependent. This thesis work focuses on developing a coherent bottom-up framework and methodology to improve information transfer and decision making while designing complex systems. The rotorcraft drive system is used as a test system for this methodology.

The traditional serial design approach required the information from one discipline and/or process in order to proceed with the subsequent design phase. The Systems Engineering (SE) implementation of Concurrent Engineering (CE) and Integrated Product and Process Development (IPPD) processes tries to alleviate this problem by allowing design processes to be performed in parallel and collaboratively.

The biggest challenge in implementing Concurrent Engineering is the availability of information when dealing with complex systems such as aerospace systems. The information is often incomplete, with large amounts of uncertainties around the requirements, constraints and system objectives. As complexity increases, the design process starts trending back towards a serial design approach. The gap in information can be overcome by either “softening” the requirements to be adaptable to variation in information or to delay the decision. Delayed decisions lead to expensive modifications and longer product design lifecycle. Digitization of IPPD tools for complex system enables the system to be more adaptable to changing requirements. Design can proceed

with “soft” information and decisions adapted as information becomes available even at early stages.

The advent of modern day computing has made digitization and automation possible and feasible in engineering. Automation has demonstrated superior capability in design cycle efficiency [1]. When a digitized framework is enhanced through automation, design can be made adaptable without the requirement for human interaction. This can increase productivity, and reduce design time and associated cost. An important aspect in making digitization feasible is having the availability of parameterized Computer Aided Design (CAD) geometry [2]. The CAD geometry gives the design a physical form that can interact with other disciplines and geometries. Central common CAD database allows other disciplines to access information and extract requirements; this feature is of immense importance while performing systems syntheses. Through database management using a Product Lifecycle Management (PLM) system, Integrated Product Teams (IPTs) can exchange information between disciplines and develop new designs more efficiently by collaborating more and from far [3].

This thesis focuses on the challenges associated with automation and digitization of design. Making more information available earlier goes jointly with making the design adaptable to new information. Using digitized sizing, synthesis, cost analysis and integration, the drive system design is brought in to early design. With modularity as the objective, information transfer is made streamlined through the use of a software integration suite. Using parametric CAD tools, a novel ‘Fully-Relational Design’ framework is developed where geometry and design are adaptable to related geometry and requirement changes. During conceptual and preliminary design stages, the airframe goes through many stages of modifications and refinement; these changes affect the sub-system requirements and its design optimum. A fully-relational design framework takes this into account to create interfaces between disciplines. A novel aspect of the fully-

relational design methodology is to include geometry, spacing and volume requirements in the system design process.

Enabling fully-relational design has certain challenges, requiring suitable optimization and analysis automation. Also it is important to ensure that the process does not get overly complicated. So the method is required to possess the capability to intelligently propagate change.

There is a need for suitable optimization techniques to approach gear train type design problems, where the design variables are discrete in nature and the values a variables can assume is a result of cascading effects of other variables. A heuristic optimization method is developed to analyze this multimodal problem. Experiments are setup to study constraint dependencies, constraint-handling penalty methods, algorithm tuning factors and innovative techniques to improve the performance of the algorithm.

Inclusion of higher fidelity analysis in early design is an important element of this research. Higher fidelity analyses such as nonlinear contact Finite Element Analysis (FEA) are useful in defining true implied stresses and developing rating modification factors. The use of Topology Optimization (TO) using Finite Element Methods (FEM) is proposed here to study excess material removal in the gear web region.

# CHAPTER 1

## INTRODUCTION

### *1.1 Problem Definition*

Vertical flight has a well-established utility in many operations, and rotorcraft systems are an indispensable asset in military applications and in many commercial sectors because of their unique capability to hover and takeoff vertically. Their almost – certainly unique ability to operate from unprepared ground, cover large areas of operation over land and sea, and transport payload make them essential to the military. Hover capability is vital in military operations for reconnaissance, security, attack, insertion, command, control and communications (C3) [4], combat search and rescue missions, and in civilian operations for providing humanitarian aid and medevac missions in emergencies. Although hover is the main rationale for its niche in aviation, improved forward flight capability has become more important and demanding, requiring designers to expand the envelope of rotorcraft performance making them more complex [5, 6].

Rotorcraft design, like any complex aerospace system design, is a multidisciplinary process, requiring the analysis and exploration of many areas such as rotor aerodynamics, rotor structures and dynamics, fuselage aerodynamic, fuselage structures, propulsion, drive system, noise and cost [7, 8]. There exists a large capability gap in aerospace system design with multidisciplinary integration. The capability gap is primarily experienced in the conceptual design stage, where many design decisions are made without many changes studied and tradeoffs being performed. Delayed decision, as recommended by the Set-Based Concurrent Engineering (SBCE) methods [9, 10], has been used extensively in the automotive design world; but this philosophy leads to

expensive design decisions in complex systems and undesirable delays in product design cycle.

In design of rotorcraft, the drive system has been studied much later and its characteristics driven by surrounding design features that are locked-in, leaving the drive system designers with less freedom in development [11]. Rotorcraft drive system design is a moderately complex task at the conceptual and preliminary stages. The task gets extremely complicated in the detailed stages. However, a significant amount of decisions are made in the conceptual and preliminary stages [12]. These decisions need to be information driven, necessitating the availability of more information in the early stages of design [13]. This requires a new look at the methodology employed in designing such systems in the early stages [14].

There is also a need to develop a sound drive system design and optimization technique. Current state-of-the-art design techniques are complex or insufficient and don't serve the needs in early design integration. There is a need for a fast and accurate design technique that takes into consideration structural and geometry requirements [7].

Modern technologies have enabled more efficient designs through the use of better materials, manufacturing processes, design tools etc. However, the overall process of putting the disciplinary designs together has not changed much. The process is still much serialized and neither time nor cost efficient. There is a need for a sound schematic to enable streamlined information flow and control of design objectives. Transfer of information between and within the different components of a disciplinary analysis is also important. There is a need to develop design methods that can model different degrees of collaboration and help resolve the conflicts between different disciplines [15]. Although the idea of automating tools and integrating multiple disciplines and facets are not new, there are some bottlenecks in efficiently implementing them. For example, high fidelity FEA has not been automated to produce satisfactory results. However, low

fidelity FEA with automatic tetrahedral meshing and basic static analyses has been automated to produce excellent results. Finding the right mix of capability and complexity is essential to successful implementation of an integrated concurrent engineering framework [16].

There is also a need to study the timely introduction of high fidelity analysis. The manner in which this analysis is introduced and studied is of particular importance to the design community [16]. Information is only valuable to the extent that it leads to better decisions [10].

Rotorcraft conceptual design and pre-design process begins with the development of a basic concept, pre-vehicle configuration geometry and sizing through fuel and power balance. Given mission and performance requirements, the necessary data for preliminary design of an aircraft is determined using a graphical technique, employed by Hiller Helicopters which is known as the  $R_F$  method [17]. This method uses parametric evaluation of helicopter configurations to determine a combination that yields minimum gross weight. The  $R_F$  method derives its name from the ratio of fuel weight to gross weight. The mission requirements dictate the  $R_F$  required for a specified endurance or radius of action. The weights obtained from the mission requirements that specify payload and crew weight drive the  $R_F$  available. Each configuration yields a gross weight where the  $R_F$  available and required is equal. This method and the process is shown in Figure 1.

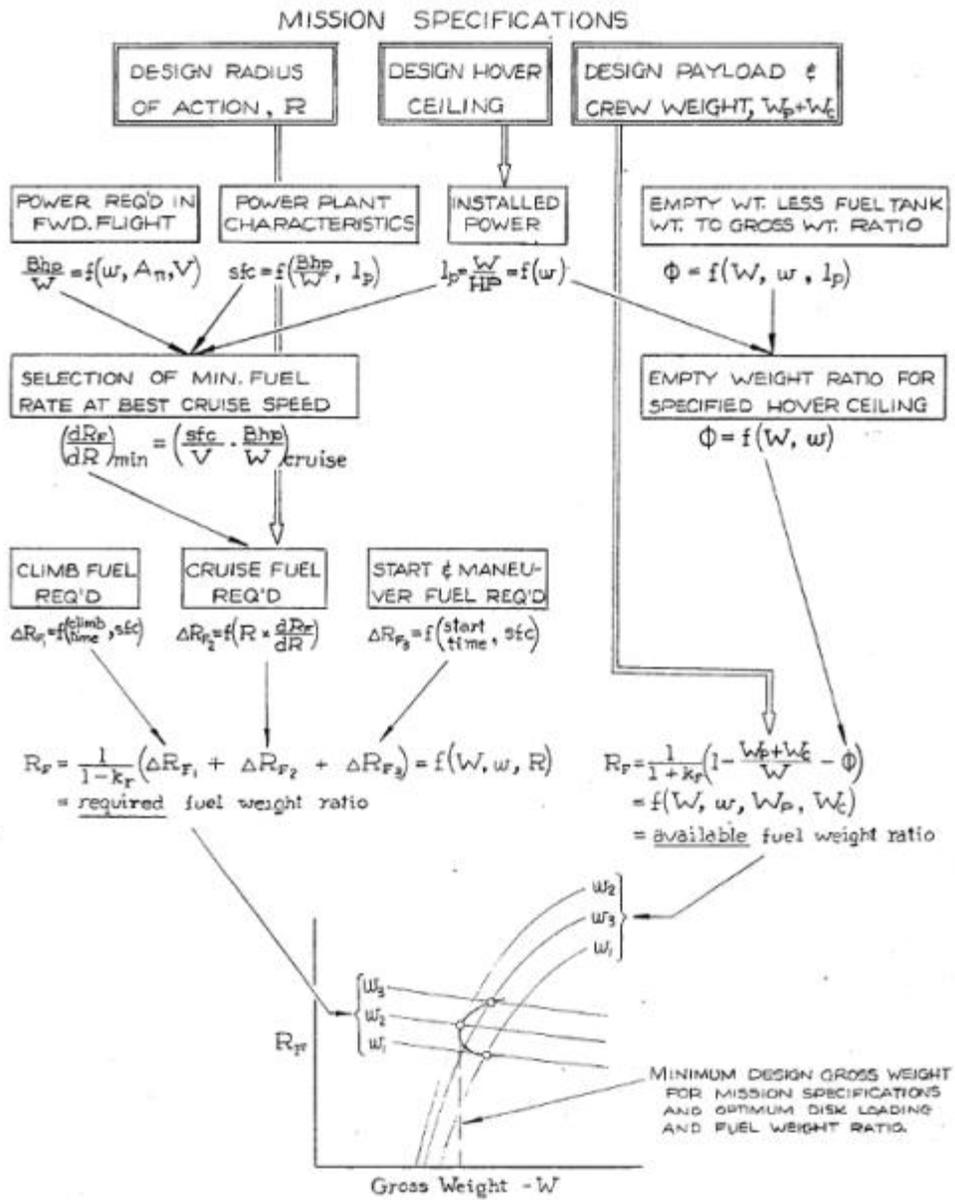


Figure 1. Schematic for optimum parameter selection using the  $R_F$  method [18]

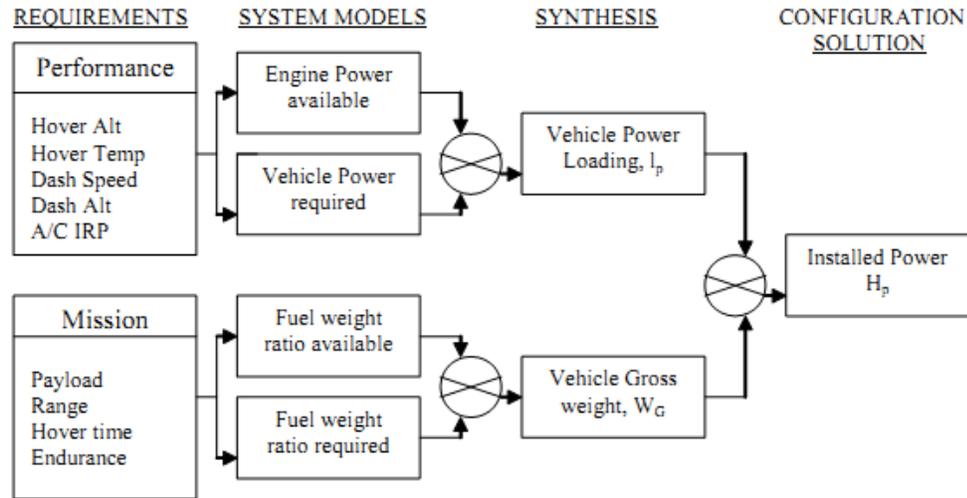


Figure 2. Obtaining installed power through vehicle design synthesis [19]

Using the power and fuel balance and performing the conceptual design the required vehicle parameters such as gross weight, power loading, and optimal disk loading, tip speed etc. are obtained (Figure 2). A preliminary vehicle geometry can be generated using the pre vehicle configuration geometry that is generated as a part of the conceptual design.

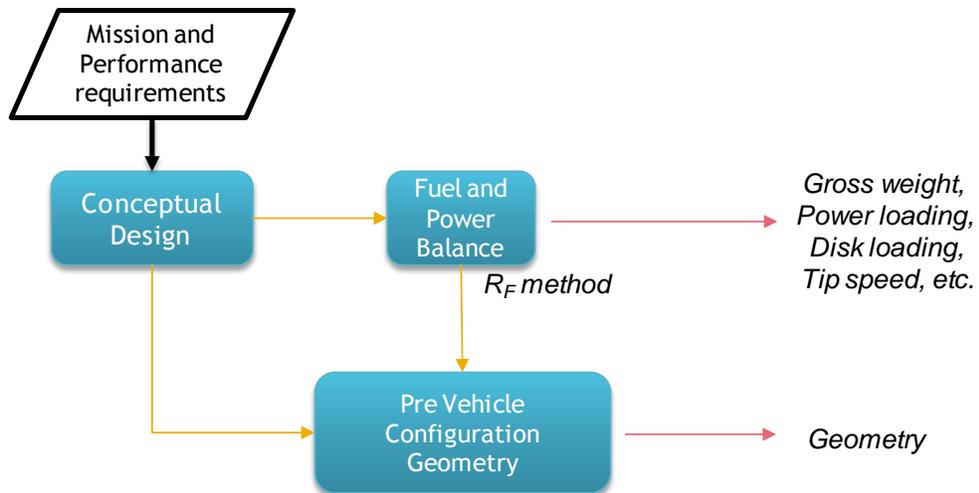


Figure 3. Rotorcraft conceptual design process

The design process extended to the drive system is shown in Figure 4. Power loading gives the installed power requirement which is the basic propulsion requirement. The propulsion design is used to generate the horsepower per engine ( $HP$ ) and the power turbine rpm ( $\Omega_{pt}$ ). In early stages of design,  $\Omega_{pt}$  may be assumed based on a known set of engine deck data and historical information. Based on conceptual design studies, rotor and tail rotor specifications are derived.

Although this study is vehicle concept independent, the implementation is performed for a single main rotor helicopter and the description of the analysis is simplified for this case. The geometry of the fuselage, airframe and engine housing are taken into account to maintain consistency. Structure arrangement, shaft locations and spacing constraints need to be obtained from the surrounding geometry. All these requirements and constraints should be dynamically used in the design process.

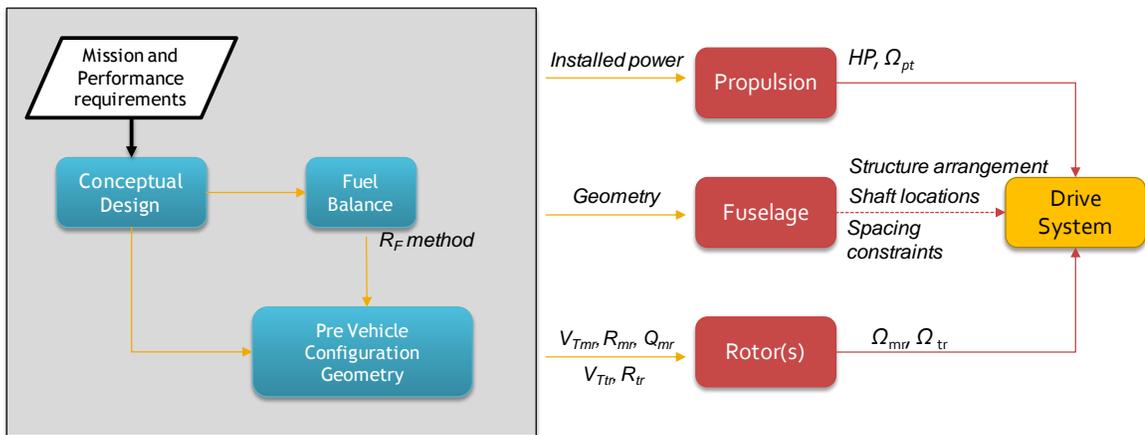


Figure 4. Design process extended to drive system

## 1.2 Motivation

Rotorcraft gear trains are sized for torque at each stage and are optimized for weight with consideration of other factors. The factors that influence the design directly arise from propulsion parameters, rotor performance requirements and airframe layout.

The propulsion and rotor parameters are easily quantified and can be used to resize the drive system. However, the airframe integration, as is with any other sub-system/component, poses a level of complexity and obscurity to the designer. During conceptual and preliminary design stages, the airframe goes through many stages of modifications and refinement; these changes affect the sub-system requirements and its design optimum [20]. The primary motivation in developing a new methodology is to create suitable interfaces between the design disciplines and enable Concurrent Engineering (CE) or an Integrated Product and Process Development (IPPD) approach. A serial approach is the traditional way of design, where the design of the system is handled one discipline at a time. Figure 5 illustrates a serial approach vs. a CE/IPPD approach.

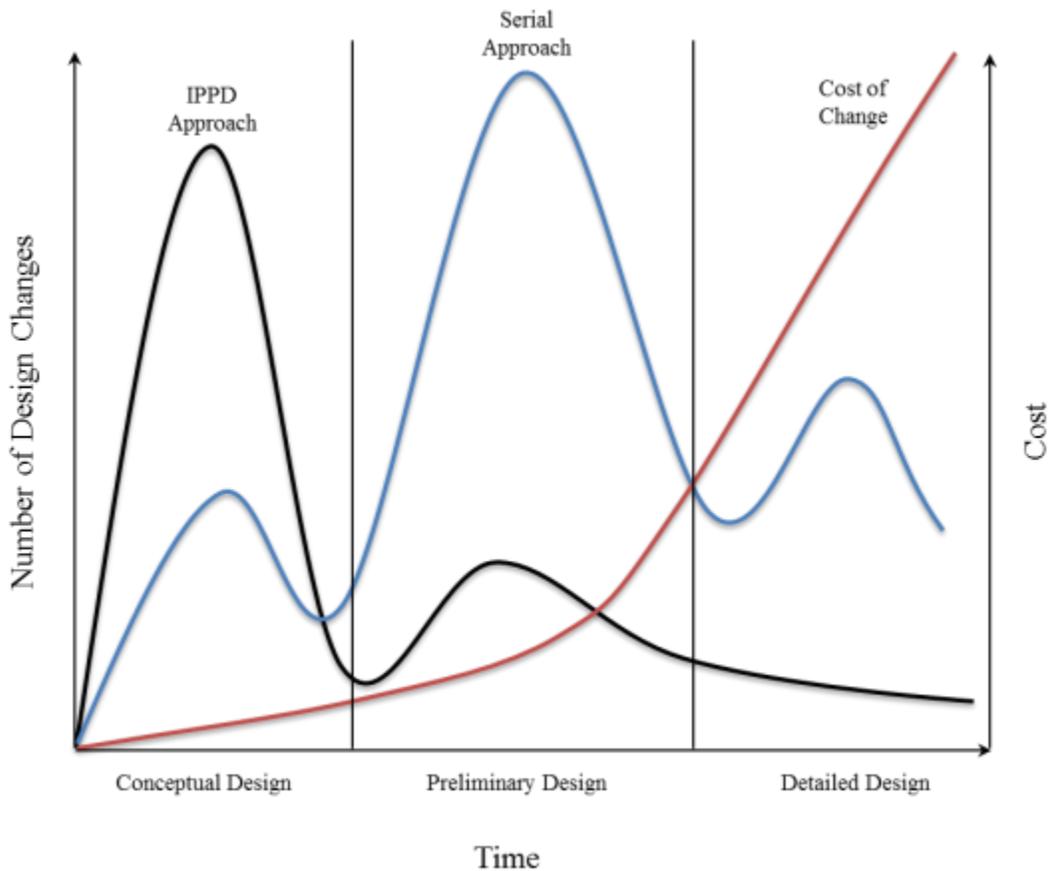


Figure 5. Serial approach vs. CE approach [19]

The cost of change, shown in Figure 5, increases exponentially as time progresses; since most design changes occur later, in the case of a serial approach, the associated cost of change is much higher as compared to that of the IPPD approach. Relation of time and design is not limited to number of changes and cost of change; Figure 6, below, compares today's design process with desired future design process in terms of knowledge about design, design freedom and cost committed. The cost in Figure 5 is indicative of the cost of one design change, thus the cost of changing the design increases exponentially as time progresses. On the other hand, the cost curves in Figure 6 represent the cumulative cost committed to the design process, not just an individual change. In current design practice, a majority of the total cost of design is committed very early in the design process, freezing most design features at the conceptual and preliminary design stages [21]. A traditional design organization would dedicate its resources for analysis and manufacturing very early in the design. This approach does not leave adequate design freedom in subsequent stages to make improvements. The consequent decline in design freedom for present and future design processes is also depicted in Figure 6. The premise of this modern systems engineering approach adopted in this thesis is to be more efficient in early design, to have as much design freedom as possible, minimize cost committed, increase the knowledge available at the early stages and most importantly incorporate design-for-change. This enables the designer to make more informed decisions before committing to costs, and reduces the time and cost of change.

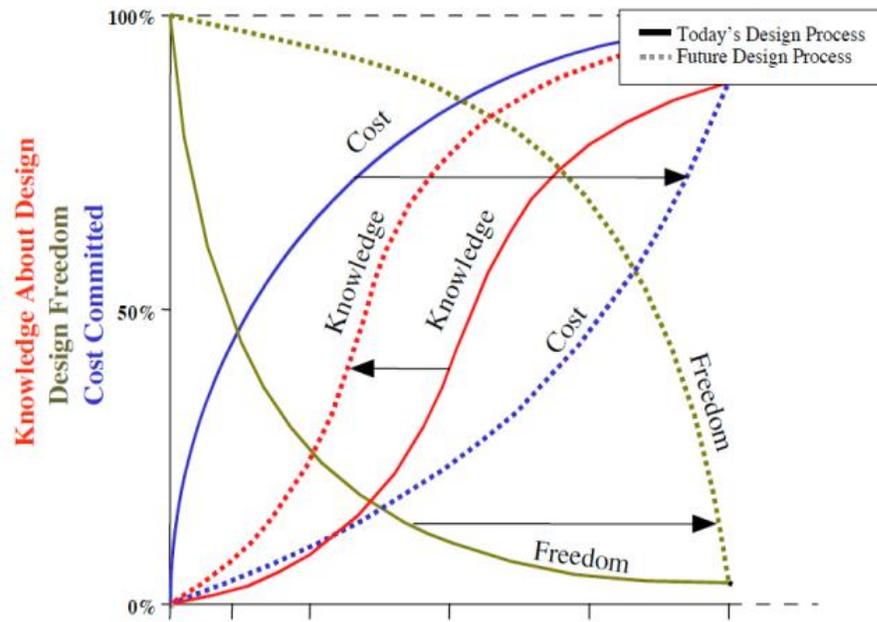


Figure 6. Design freedom, knowledge and cost relationship [22]

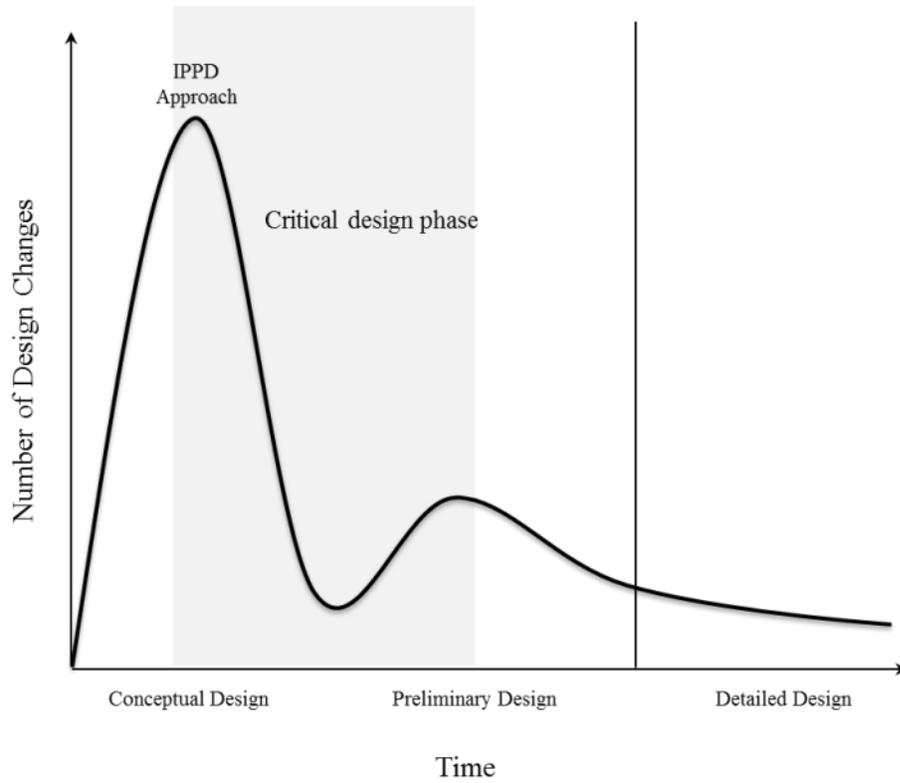


Figure 7. Critical design phase

Figure 7 shows the critical design phase in the IPPD approach. This phase covers most of the changes and tradeoffs that will be performed. In the conceptual design phase, series of design tradeoffs are performed to maximize vehicle capability. In the serial approach (Figure 8), fewer changes are made in the early design stages because of the lack of capability in obtaining information and executing change. Limited information in the conceptual design stages is the primary reason for this. Although it is not required to make all changes earlier, mid-cycle changes and changes in detailed stages are very expensive. As design changes get delayed in the design lifecycle, the performance of the system degrades, leading the design to approach a sub-optimal configuration.

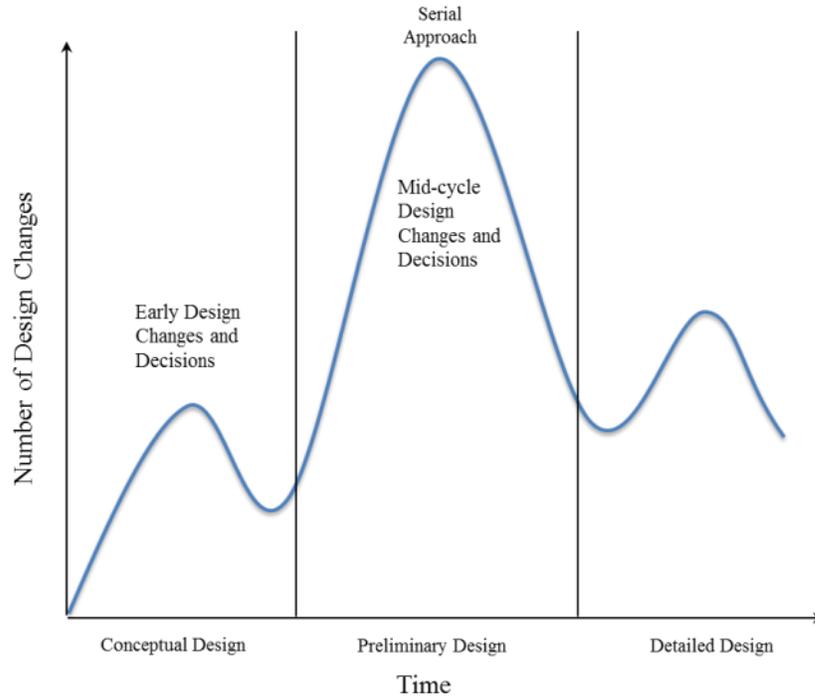


Figure 8. Design changes in serial approach

Figure 9 shows the desired paradigm shift in moving from serial to IPPD (CE) design approach; enabling more changes in the conceptual stage, reducing the number of required design changes in the preliminary stage, and eliminating the need for detailed design-stage changes. Therefore, the feasibility of adopting early changes in the IPPD

approach relies on obtaining more quality information in early phases. Tradeoff studies require multidisciplinary integration to study interactions and response of overall capability to changes. The other critical stage, where many changes occur, is in the preliminary design stage. Here, changes are being performed to refine design, mitigate conflicting interactions and further enhance overall capability through technology infusion.

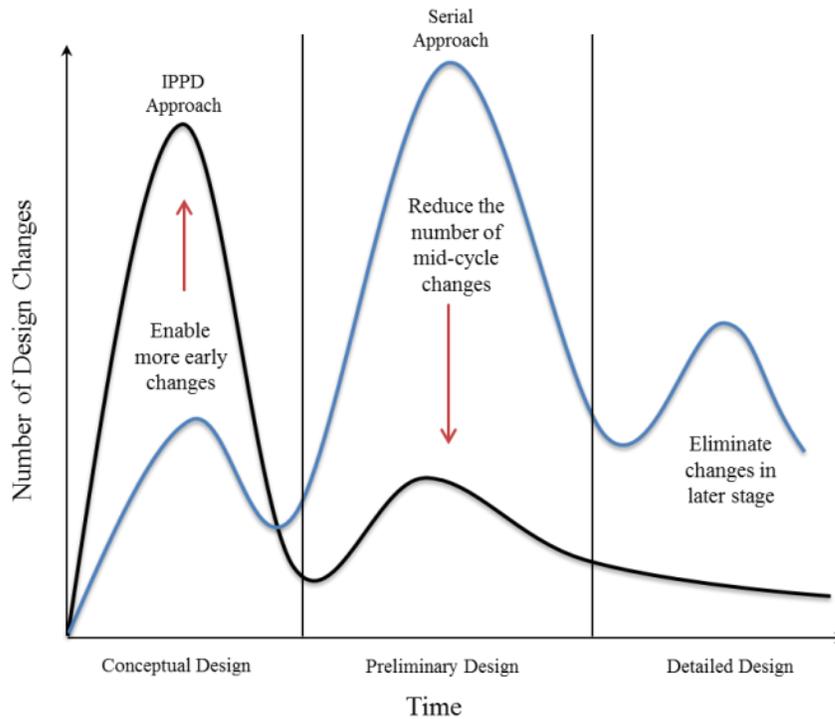


Figure 9. Moving from serial approach to IPPD approach

The requirements for effectively implementing IPPD approach are (shown in Figure 10):

1. Tools that allow for more changes to be studied and implemented.
2. Methodology to select a design that can be adapt to downstream changes.

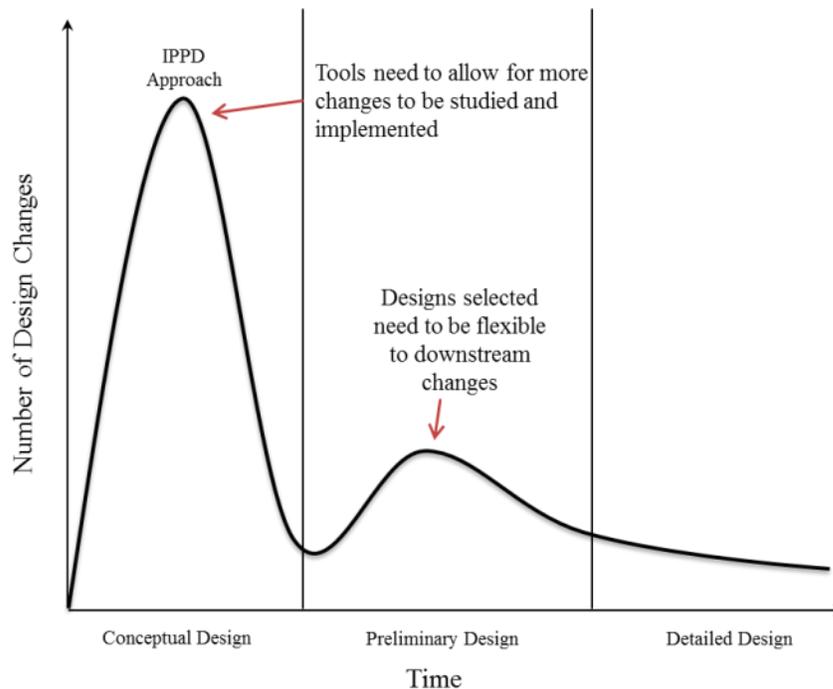


Figure 10. Requirements for IPPD approach

Rotorcraft design comprises of multiple disciplines and it is usually not possible to obtain knowledge in all relevant disciplines, in equal amounts, in a traditional design setup, as shown in Figure 11. The amount of information available in the early phases of design is scattered and may be more limited in some disciplines than others. This uneven distribution of disciplines does not allow the use of design freedom to improve quality and integrate disciplines for optimization. The IPPD approach focuses on improving this situation as shown in Figure 12. The detailed design time is reduced by up to one third based on the use of more upfront design knowledge, and a more evenly distributed effort of disciplines is provided in the conceptual and preliminary design phases [23].

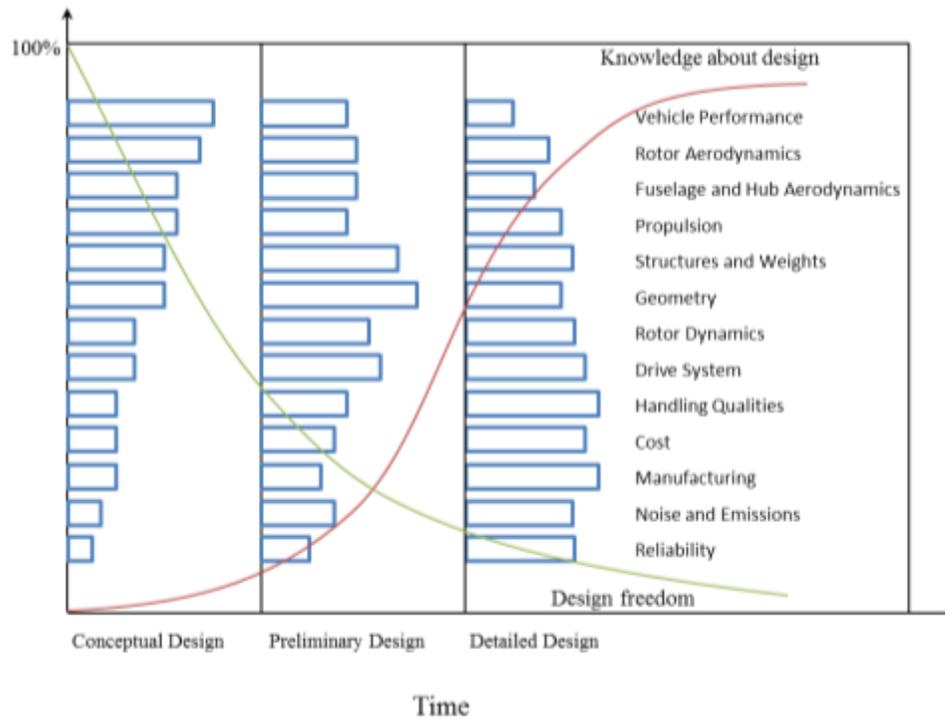


Figure 11. Design freedom and knowledge in traditional design

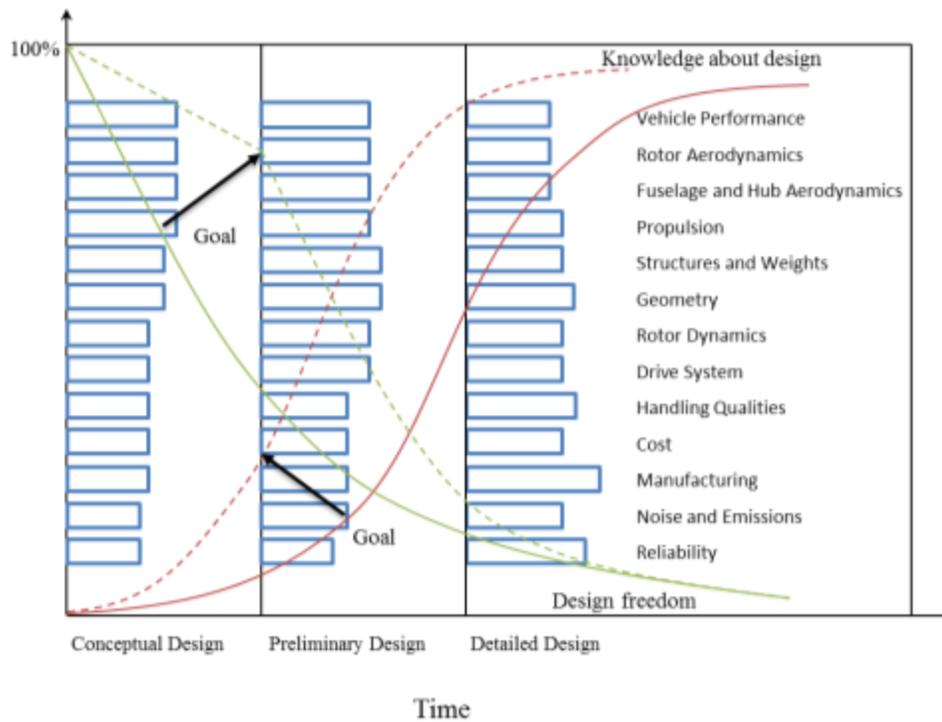


Figure 12. Design process reorganized to gain information earlier and to retain design freedom longer

IPPD methods help designers starting with the conceptual design stage, where there is great design freedom and almost infinite number of concepts to explore. In the preliminary design stage, however, the design space gets much narrower than that of the conceptual design stage. Moreover, the evaluation of each concept requires more complex analyses; therefore it is necessary to organize the analysis in a systematic manner. An IPPD framework for preliminary analysis was developed by Schrage [19] and modified by Chae et al. [1, 24] for rotorcraft design, where design and analysis tools are systematically arranged and merged for the rotorcraft preliminary design stage. This framework, shown in Figure 13, is utilized in developing a drive system preliminary design methodology.

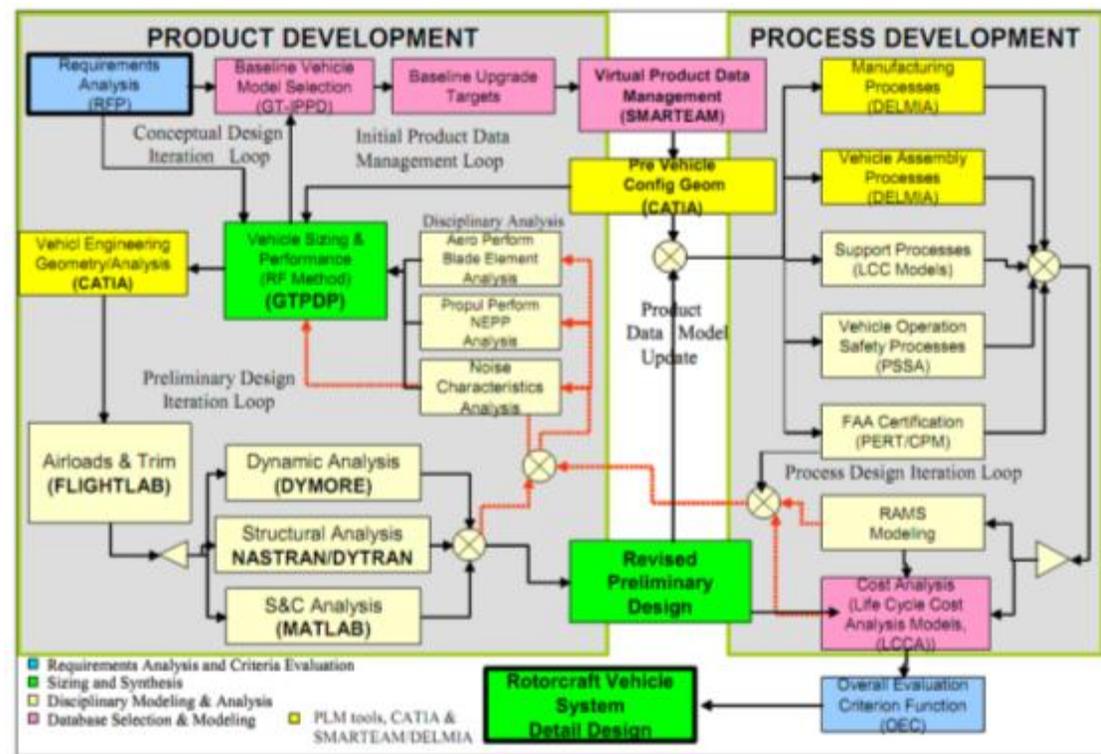


Figure 13. Generic IPPD framework for rotorcraft preliminary design [1]

To enable early changes in conceptual design phase, a transformation to the ‘design process’ is proposed. This design process is called ‘fully-relational design’ and is discussed in detail in Section 4.1 (page 95). The process enhances ‘change’ capability and allows for streamlined transfer of information to maximize available knowledge.

Modifications require associated component redesign. Reflecting these changes in terms of a new optimized design is a challenge. Using a suitable framework with the right interfaces helps address interactions and maintain consistency between parent – child parts and also associative parts.

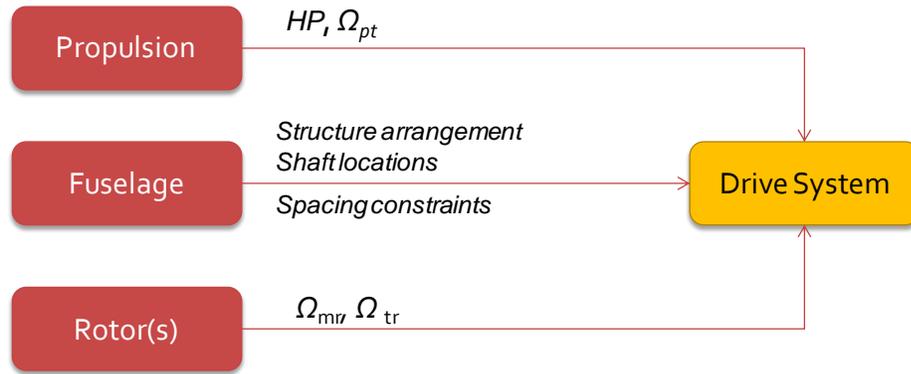


Figure 14. Drive system parent requirements and constraints

There is a requirement for automation of the update cycle while maintaining multi-disciplinary interfaces. Automation has demonstrated significant advantages to traditional design. Primary advantages being considerable reduction in design cycle time and human error, and capability to run process intensive optimization and computation for Design of Experiments (DOE). Figure 15 shows the difference in design process between manual and automated analysis and also a notional difference in time and its decomposition.

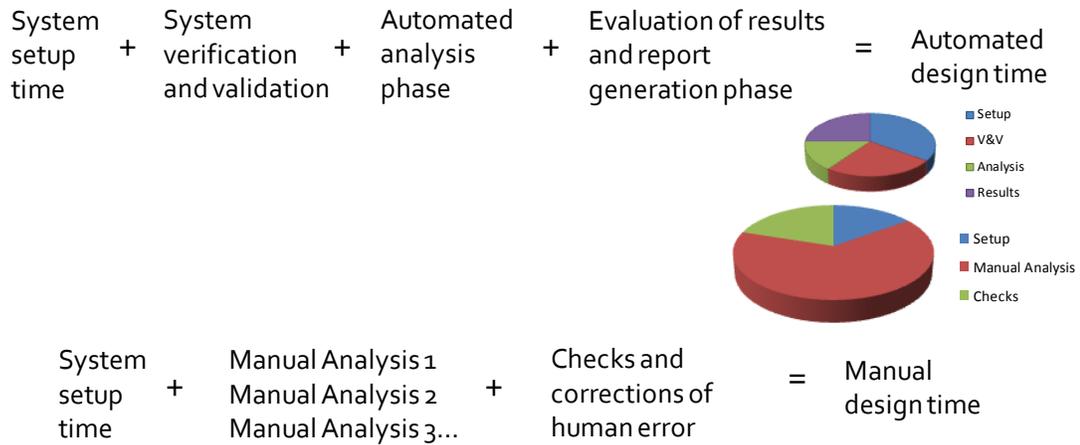


Figure 15. Comparison of design time between manual and automated analysis

A “bottleneck” in complete automation and multidisciplinary analysis, in complex aerospace systems is the lack of tools to perform full or fraction of disciplinary analysis at different levels of fidelity. A notional figure indicating gaps for fixed-wing aircraft design was developed by Nickol in 2004 [25]. He presented a table of key analysis disciplines and associated fidelities for fixed-wing aircraft, from which the following figure (Figure 16) is adopted to indicate the gaps in rotorcraft design. The real problem arises in integration, when one discipline requires information from another discipline with adequate fidelity. Figure 17, below, shows some cases of interdependency, especially with respect to geometry. Also to be noted is the lack of a formal tool to design drive systems. Transmission efficiency is an important aspect of vehicle sizing and drive system contribute a lot to overall vehicle empty weight. For example, the entire drive system for the Sikorsky UH60–A contributes to over 14% of the empty weight [26] (drive system: 1663.4 lbs., empty weight: 11620.6 lbs.).

		Analysis Discipline											
Fidelity		Rotor Aerodynamics	Fuselage and Hub Aerodynamics	Rotor Dynamics	Propulsion	Structures and Weights	Drive System	Noise	Flight Control System	Geometry	Cost	Manufacturing	Reliability
	Low	CBEM (e.g. CIRADS)	Empirical Methods (e.g. CIRADS)	GAP	Empirical Methods (e.g. CIRADS)	Empirical Methods (e.g. CIRADS)	Empirical Methods	Empirical Methods	GAP	Basic Sketch (e.g. VSP)	Empirical (e.g. Bell PC)	GAP	GAP
	Medium	Advanced Inflow	CFD (e.g. FLUENT)	Basic Rotor Dynamics (e.g. RCAS)	Basic Cycle Analysis (e.g. GasTurb)	Basic Structures	Gear Train Sizing	Basic Noise (e.g. WOP-WOP)	Trim (e.g. Flightlab)	Basic Parametric Geometry (e.g. CATIA)	GAP	GAP	Safety and Reliability (e.g. SPN)
	High	CFD (e.g. CHARM)	CFD (e.g. OVERFLOW)	Advanced Rotor Dynamics (e.g. RCAS)	Advanced Cycle Analysis (e.g. NPSS)	FEA (e.g. ANSYS)	Advanced Gear Design (FEA), Housing Design etc.	Advanced Noise	Stability Augmentation System (e.g. Flightlab)	Parametric Geometry (e.g. CATIA)	GAP	Virtual Manufacturing and Assembly (e.g. DELMIA)	GAP

Figure 16. Gaps in analysis disciplines in rotorcraft design

		Analysis Discipline											
Fidelity		Rotor Aerodynamics	Fuselage and Hub Aerodynamics	Rotor Dynamics	Propulsion	Structures and Weights	Drive System	Noise	Flight Control System	Geometry	Cost	Manufacturing	Reliability
	Low	CBEM (e.g. CIRADS)	Empirical Methods (e.g. CIRADS)	GAP	Empirical Methods (e.g. CIRADS)	Empirical Methods (e.g. CIRADS)	Empirical Methods	Empirical Methods	GAP	Basic Sketch (e.g. VSP)	Empirical (e.g. Bell PC)	GAP	GAP
	Medium	Advanced Inflow	CFD (e.g. FLUENT)	Basic Rotor Dynamics (e.g. RCAS)	Basic Cycle Analysis (e.g. GasTurb)	Basic Structures	Gear Train Sizing	Basic Noise (e.g. WOP-WOP)	Trim (e.g. Flightlab)	Basic Parametric Geometry (e.g. CATIA)	GAP	GAP	Safety and Reliability (e.g. SPN)
	High	CFD (e.g. CHARM)	CFD (e.g. OVERFLOW)	Advanced Rotor Dynamics (e.g. RCAS)	Advanced Cycle Analysis (e.g. NPSS)	FEA (e.g. ANSYS)	Advanced Gear Design (FEA), Housing Design etc.	Advanced Noise	Stability Augmentation System (e.g. Flightlab)	Parametric Geometry (e.g. CATIA)	GAP	Virtual Manufacturing and Assembly (e.g. DELMIA)	GAP

Figure 17. Gaps in analysis disciplines and different fidelity creating gaps in integration

### **1.3 Research Objectives**

The primary objective of this thesis is to implement Concurrent Engineering and develop a design process that makes early design phases more efficient and to streamline design integration. Figure 6 (page 36) outlines these goals for the future design process in relation to the present. More specifically, the design process needs to accomplish the following:

1. Increase design freedom
2. Minimize cost committed
3. Improve knowledge available
4. Enable design-for-change

The overall research objectives of this thesis are:

1. Develop a framework flexible to interfaces, fast and accurate, with integration and automation capability.
2. Improve understanding of optimization techniques for gear train design.
3. Closing the gap in high fidelity design in early stages.
4. Understanding ‘when’ and ‘where’ high fidelity analysis information is required for design decisions.
5. Develop a method to enforce geometry and space constraints through ‘fully-relational’ design.
6. Develop a method to select a ‘flexible’ configuration in conceptual design stages.

## CHAPTER 2

### LITERATURE REVIEW

#### *2.1 Systems Engineering*

Concurrent Engineering (CE) and Systems Engineering (SE) are broadly studied topics in the field of design. SE has many formal definitions, one of which is, “an interdisciplinary, collaborative approach that derives, evolves, and verifies a life-cycle balanced system solution,” as defined by INCOSE [27]. Price et al. [16] define aerospace systems engineering as a holistic approach to a product that comprises several components, namely, customer specification, conceptual design, risk analysis, functional analysis and architecture, physical architecture, design analysis and synthesis, trade studies and optimization, manufacturing, testing, validation and verification, delivery, life cycle cost, and management. Further, they claim, it also involves interaction between traditional disciplines such as Aerodynamics, Structures and Flight Mechanics with people and process-oriented disciplines such as Management, Manufacturing, and Technology Transfer.

SE has been seen as a cultural change taking place in industry and government [19]. SE has also become a methodology for organizing and managing aerospace systems production [16]. The quality revolution of the 1970s identified the need for new systems approach, concurrent engineering and IPPD based product – process simulation. The primary design/synthesis iteration, illustrated in Figure 18, is between the SE method, ‘System Synthesis through Multidisciplinary Design Optimization (MDO)’, ‘Generate Feasible Alternatives’, and the QE method - Robust Design Assessment and Optimization, to ‘Evaluate Alternatives’ and finally to update the System Synthesis [19, 22]. Price et al. [16] also define SE as a process management tool in which functional and

physical architectures are linked to enable closer coordination and management of complex aerospace systems, as shown in Figure 19.

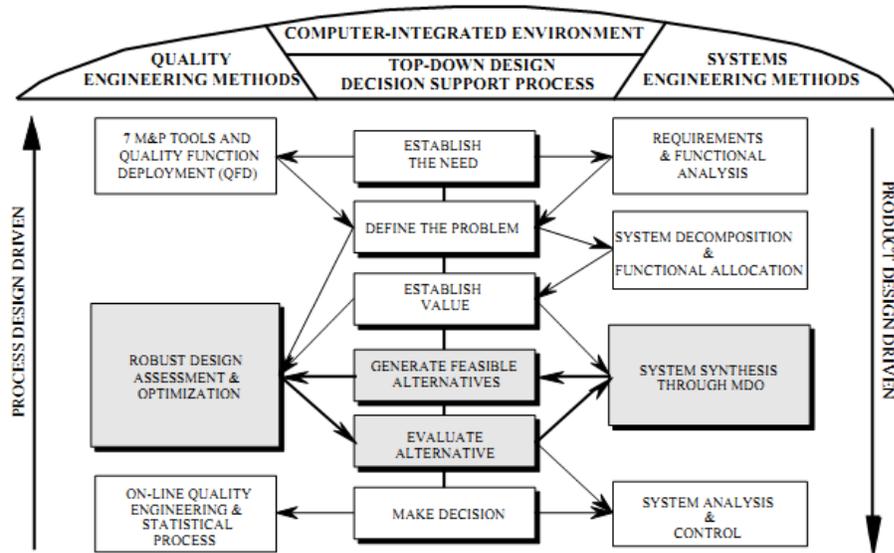


Figure 18. Georgia Tech generic IPPD methodology [19]

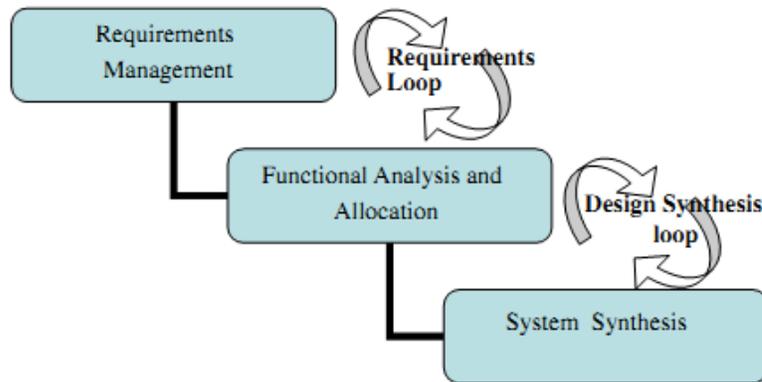


Figure 19. Systems engineering process model [16]

Multi-disciplinary Analysis and Optimization (MDO) is an important element of rotorcraft design and SE [28, 29]. Orr and Narducci [7] claim that an MDO system that infuses high-fidelity analyses quickly and consistently across the design space, will lead to improved designs because first-time decisions can confidently consider impacts to all

relevant engineering disciplines. Their research addressed ‘schedule’ as an engineering metric and suggest that adherence to schedule and budget should be improved by the automation and integration of MDAO.

Khalid, in his thesis [30], developed and implemented a preliminary design methodology using multidisciplinary design optimization for rotorcraft. He studied two MDO techniques - namely, All At Once (AAO) and Collaborative Optimization (CO), and implemented them for a light turbine training helicopter. In his study, he uses a systems engineering - modeling and simulation framework to study and integrate various disciplines, including the drive system, shown in Figure 20.

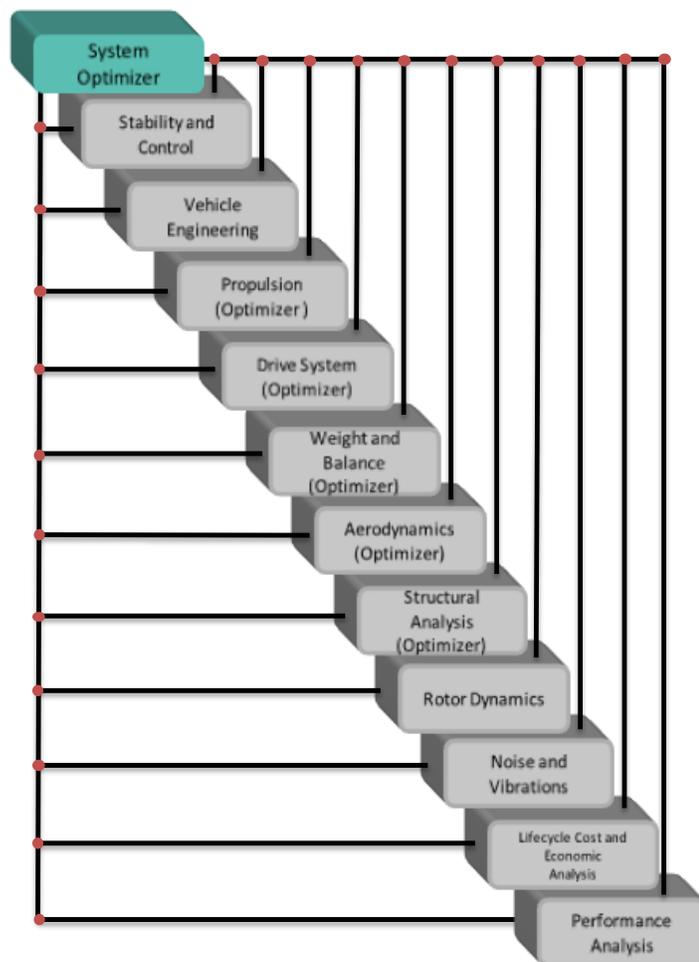


Figure 20. Disciplines involved in MDO environment [30]

The inclusion of high fidelity analysis tools raises an issue of maintaining appropriate levels of approximation across various engineering disciplines. Disparity in level of fidelity in different areas and its implications are not clearly known i.e. the appropriateness of accepting a low level of fidelity in one area while working on high-fidelity in another. For example, Orr and Narducci [7] suggest, it may be sufficient to use C81 airfoil tables for rotor loads and dynamics and engine performance maps for propulsion, while using exact airfoil geometry in CFD for aerodynamic performance.

SE and MDAO research have identified the following five points that need to be addressed in efforts towards development of a comprehensive MDAO integration framework [7, 16, 31, 32]:

1. Appropriateness of accepting low fidelity information in one discipline while working on high fidelity in another
2. Accommodating different disciplines and integrating them at the ideal level of fidelity
3. Cascading effect of data unavailability
4. Interfaces and interactions
5. Controlling emergent behavior

Past research in this area has shown that it has been very difficult to implement systems engineering principles completely through formal models and tools. Price et al. [16] also claim that research into SE provides a deeper understanding of the core principles and interactions, and helps one to appreciate the required technical architecture for fully exploiting it as a process, rather than a series of events.

Carty [33] studied MDO problems and stated that the challenge in implementing MDO and multi-disciplinary integration first arise in identifying the disciplines, that need to be included in the analyses. The disciplines being very disparate in nature make them

hard to combine and integrate. Conceptual design is influenced by many disciplines; accommodating them and integrating them at the ideal level of fidelity is the challenge. Price et al. [16] state that “Interfaces are specified by the designer but interactions may emerge as a consequence of this”.

Price et al. [16] highlight four key challenge areas that require attention:

1. Integration of design and analysis methods into a SE framework
2. Identification and measurement of interfaces and emergent behavior
3. Digital manufacturing and economics
4. Collaborative design and virtual enterprise

Schrage et al. [1, 24] studied design integration for rotorcraft and developed an integrated framework to perform the design of a bearingless soft-in-plane rotor blade. Using relational design technique, geometry of the hub and blade was integrated. This report discusses the formal set-up for information flow within analysis tools depicted in Figure 21. A process based part-level manufacturing cost/time analysis through Response Surface Equations (RSE) combined with Life-Cycle Cost (LCC) using Bell PC-based model was used in their study. Some tools from this research that are relevant to this thesis include CATIA for CAD, ANSYS for FEA pre-processing, and CIRADS for vehicle conceptual sizing and Bell PC for LCC.

Gunduz, in his thesis work [23], automated the rotor dynamic analysis using ModelCenter, shown in Figure 22. His research in the implementation of integration of complex analysis and geometry integration creates some of the foundations of design automation that this thesis is based on. His thesis explicitly details the configuration optimization procedure for a bearingless rotor and the techniques used for information flow between tools (Figure 23). His approach included the integration of non-native tools to ModelCenter such as ANSYS-VABS and DYMORE. His research focused on

structural and dynamic aspects of rotor design, rather than others. The rotor design is analyzed and optimized with respect to structural stability and dynamic response to external excitations [24].

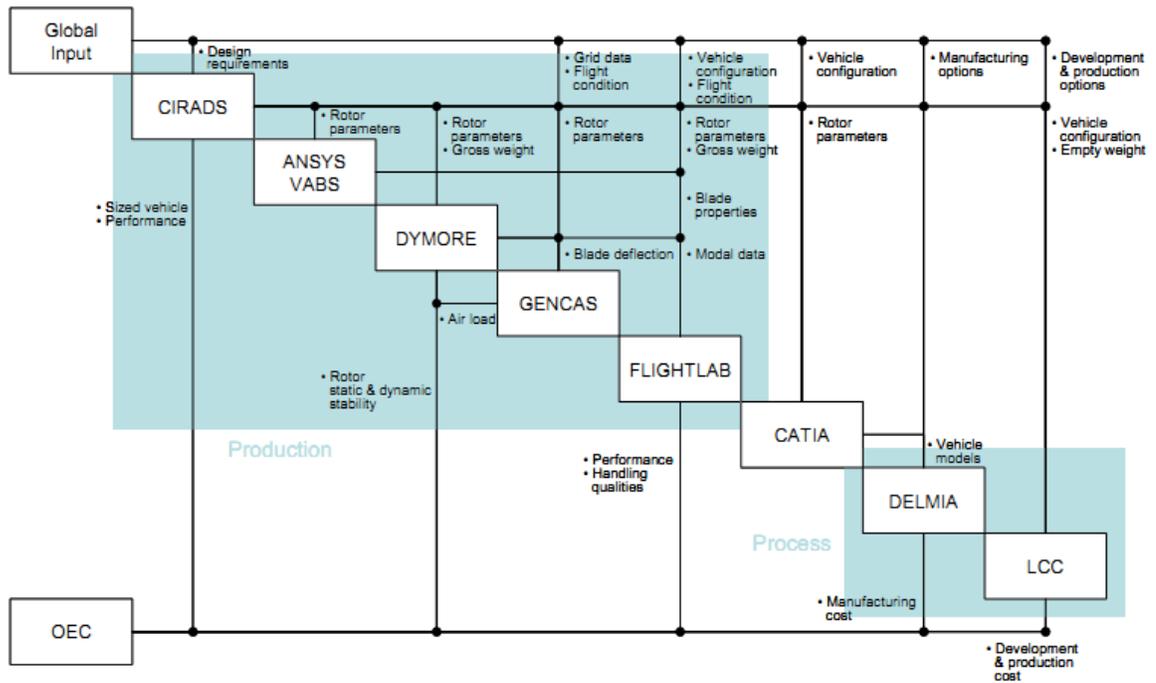


Figure 21. Information flow for design integration of a bearingless soft-in-plane rotor blade [24]

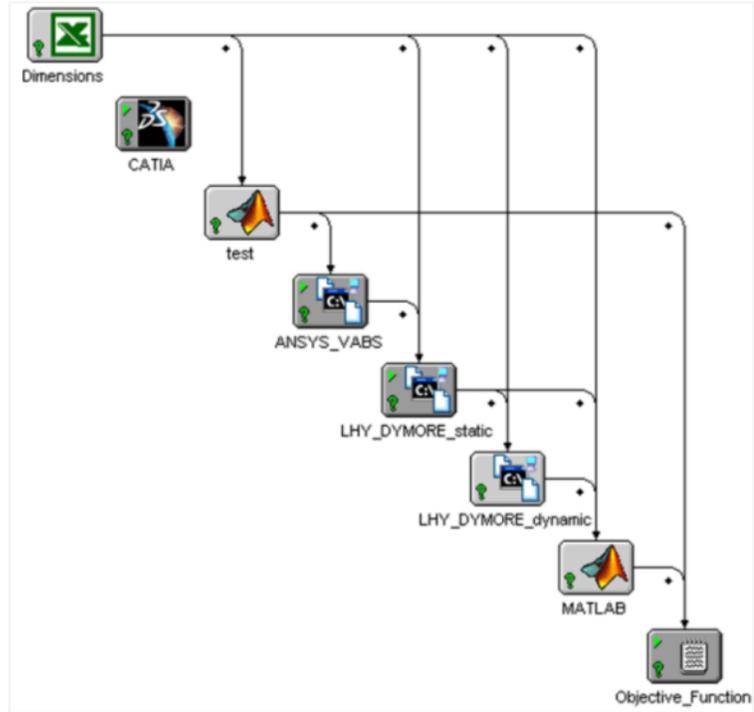


Figure 22. Software integration in ModelCenter [23]



Figure 23. Rotor flexure configuration optimization [23]

## 2.2 *Computer Aided Design*

The important advance in CAD technology that occurred in the late 1980s was the introduction of parametric CAD modeling [34]. After significant research and development since the 1980s, feature-based and parametric modeling techniques start getting adopted into mainstream CAD programs [35, 36]. These programs give the designer the capability to design parts using geometric features (points, lines, circles, etc.), with parameterized dimensions and assemble them in a digital environment. With the parameterized product model, the designer can make a design change simply by changing design variable values and asking the CAD software to automatically regenerate the parts that are affected by the change, consequently, regenerating the entire assembly.

Orr and Narducci [7] developed a framework to perform multi-disciplinary analysis, design, and optimization using high fidelity tools that covered the areas of rotor aerodynamics, rotor structures and dynamics, fuselage aerodynamics, fuselage structures, and propulsion / drive system; they found that using a central geometry database enhanced the capability of maintaining consistency among disciplines.

Parametric CAD becomes particularly important in the PLM deployment and manufacturing stage [3]. When a number of feasible design alternatives are available, the designer has to make a decision by performing tradeoff studies. Chang et al. [37] claim that a CDM approach holds the potential for shortening the overall product development cycle, improving product quality, and reducing product cost.

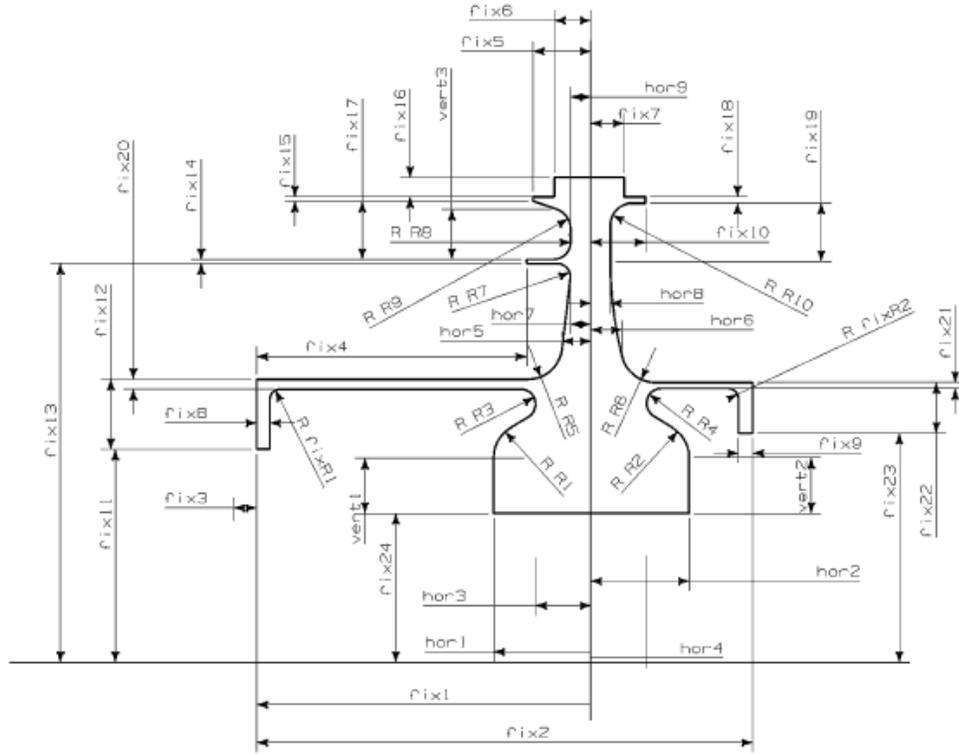


Figure 24. Parametric CAD model of a turbine blade [38]

Robinson et al. [38] present an approach to rate the quality of parameters in a CAD model for use as optimization. In their study parametric effectiveness is computed as the ratio of change in performance achieved by perturbing the parameters in the optimum way, to the change in performance that would be achieved by allowing the boundary of the model to move without the constraint on shape change enforced by CAD parameterization. They applied this to 2D and 3D FEA and CFD problems. In their study they look at the design of a parametric turbine disc (Figure 24).

While parametric optimization of geometries has progressed, the influence of associated systems has not been approached from a concurrent engineering standpoint. Shape optimization and study of variation in geometries can be enhanced and need to be influenced by associated systems, volume and spacing to get a complete understanding of optimal design [39].

The complexity of the design changes is multiplied when the product design involves multiple engineering disciplines. Very often, a simple change in one part may propagate to its neighboring parts, therefore, affects the entire product assembly. Both parts and assembly must be regenerated for a physically valid product model, at the same time, the regenerated product model must meet designers expectations [40].

Stark [41] discusses the importance of CAE and the role it plays in support of design. He claims that CAD - CAE provides the capability to provide solutions at a fraction of cost and time, and that it has enabled collaboration between disciplines and enterprises [42-44].

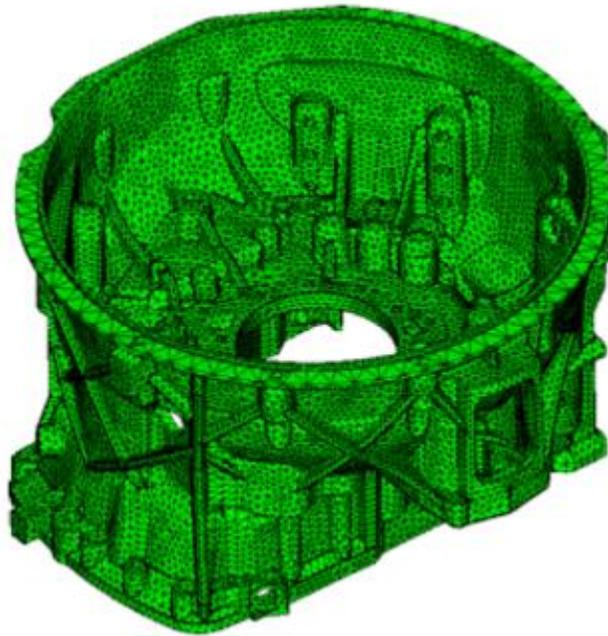
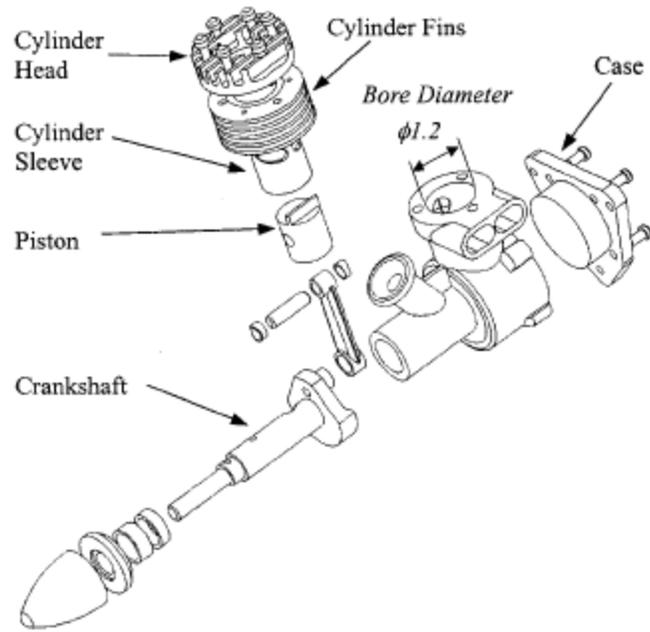
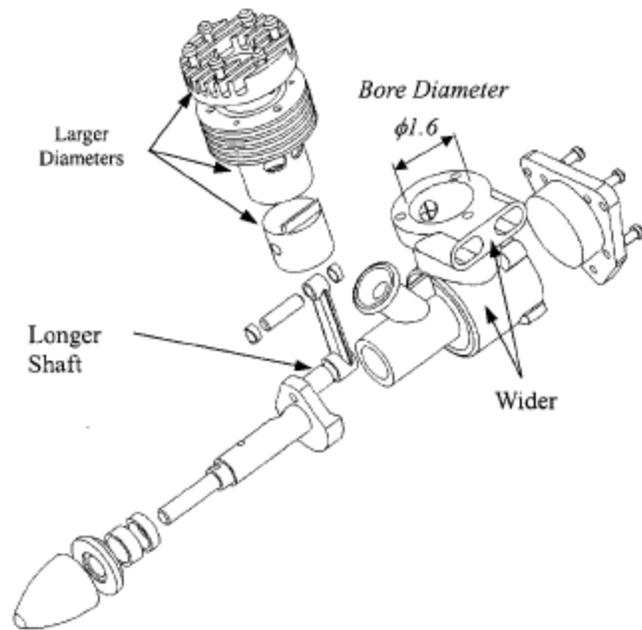


Figure 25. Complex FEA model generation automatically using CAD interface [16]



(a) Bore Diameter 1.2"



(b) Bore Diameter 1.6"

Figure 26. Parametric redesign of a single-piston engine [40]

Designers must identify an optimal solution that satisfies a number of performance requirements by repeating analysis and modification of CAD models. They need to process CAE results until an improved solution is obtained. This sort of design process is both time consuming and expensive: the design process can be improved greatly by automating the iterative process.

### **2.3 *Drive System Design***

For rotorcraft, transmission gearing has to provide the required reduction ratio while transferring power from the engines to the rotors (Figure 27). Gears are found in all types of machinery and are used to transmit power from one axis to another, with the capability of adding a mechanical advantage in terms of increase in torque and reduction in angular velocity [45, 46]. Bellocchio, in his thesis research [47], developed a drive system design methodology for a single main rotor heavy lift helicopter. His design process was parameterized to be able to run a DOE and generate a RSE (Figure 28). His research included a detailed analysis of gears using AGMA and AMCP standards and recommendations [11, 48-52].

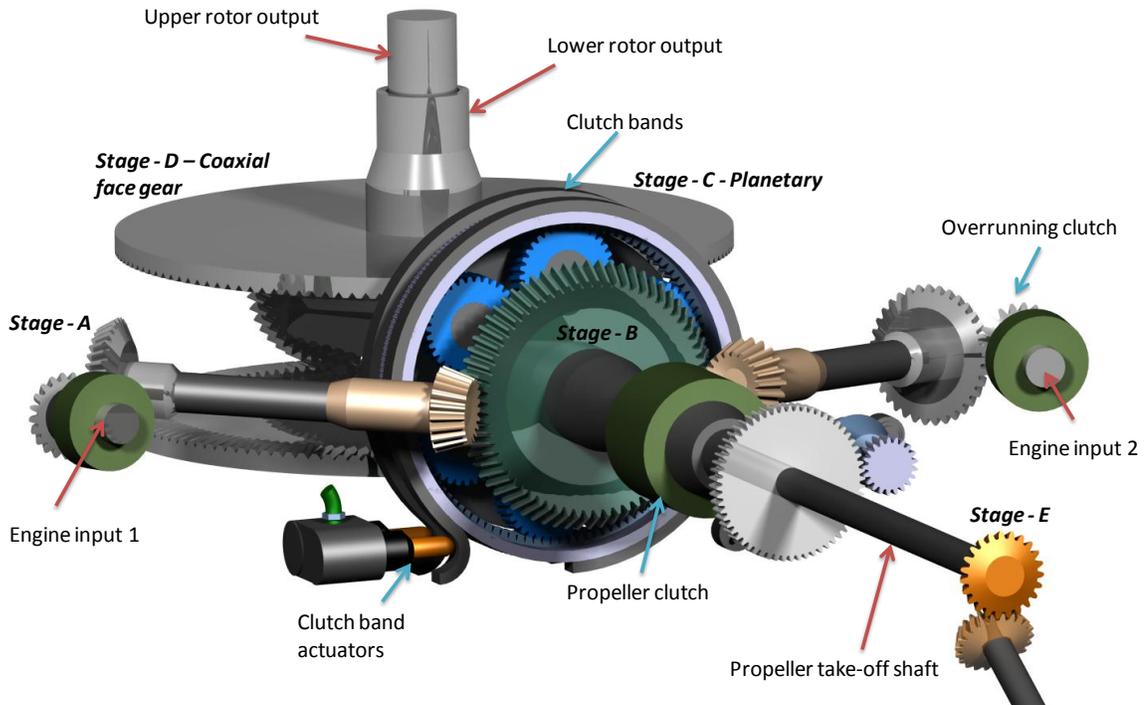


Figure 27. Variable speed planetary gear drive system for a twin-engine coaxial compound configuration [53]

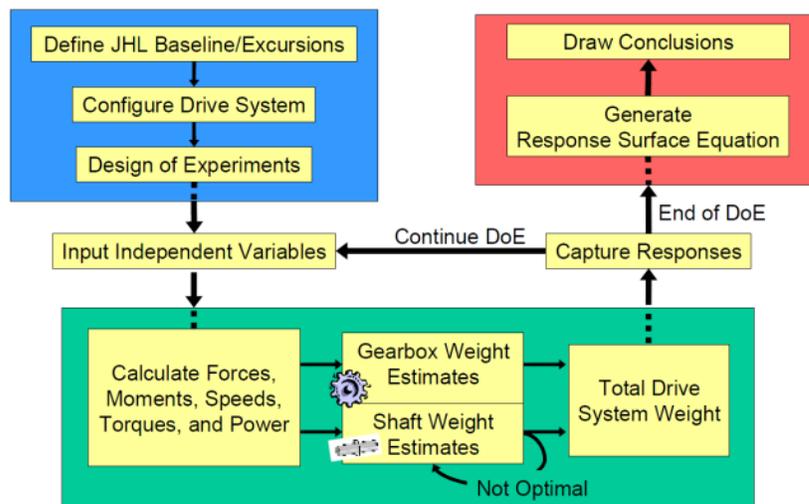


Figure 28. Design process and response surface methodology [47]

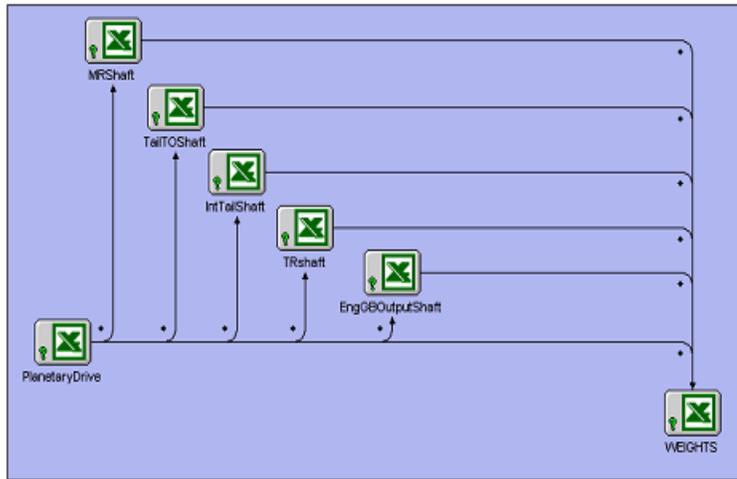


Figure 29. Weights estimation of drive system [47]

His study evaluated split-torque and planetary drive system concepts for heavy lift helicopter application. His study used a graphical method to optimize planetary stages for weight (Figure 30). His extensive study of the drive system provides a great insight into the design requirements and shortcoming of existing methodology.

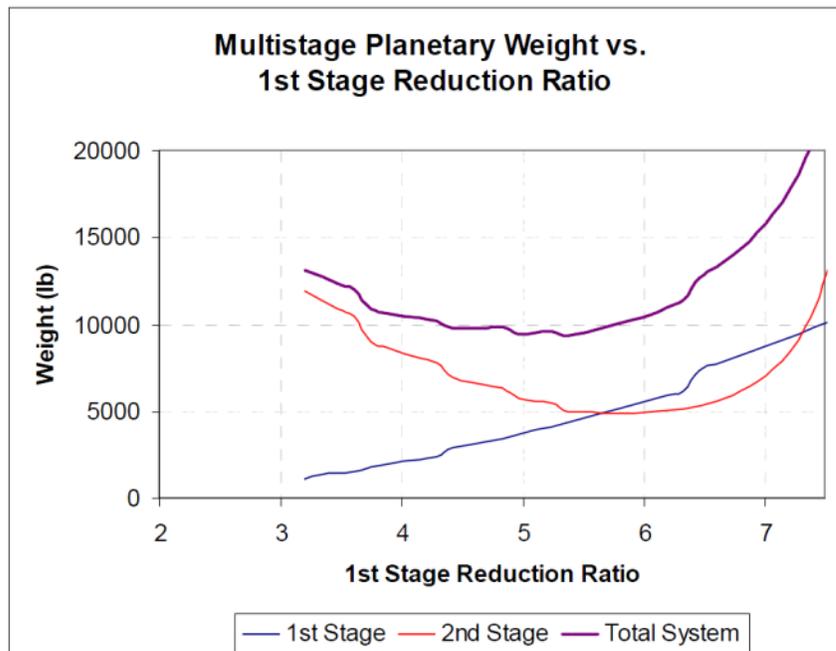


Figure 30. Multistage planetary weight minimization technique [47]

Transmission design in the conceptual stages has been limited to empirical weight estimation given by the ‘square – cube law’ [47, 54], which states that when an object undergoes a proportional increase in size, its new volume is proportional to the cube of the multiplier and its new surface area is proportional to the square of the multiplier (Figure 31).

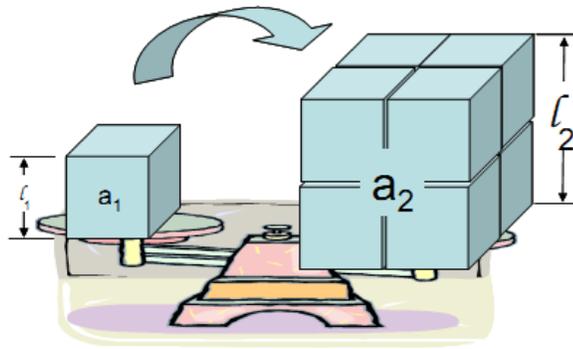


Figure 31. Square cube law block [12]

The square-cube relation is used to describe the relation between weight and torque of a gear, as in Equation 1.

$$W_2 = W_1 \left( \frac{T_2}{T_1} \right)^{3/2}$$

Equation 1. Weight - torque relation using square cube law [47]

This relationship gives a very good guideline to estimating gear weight. AMCP 706-201 predicts the weight to be proportional to torque to power of 1.43 [11].

$$W_2 = W_1 \left( \frac{T_2}{T_1} \right)^{1.43}$$

Equation 2. Weight - torque relation as per AMCP 706 -201 [11]

Bellocchio plotted the weight - torque relation for square-cube law, AMCP 706 – 201 gear stage and AMCP 706 – 201 shafting (Figure 32).

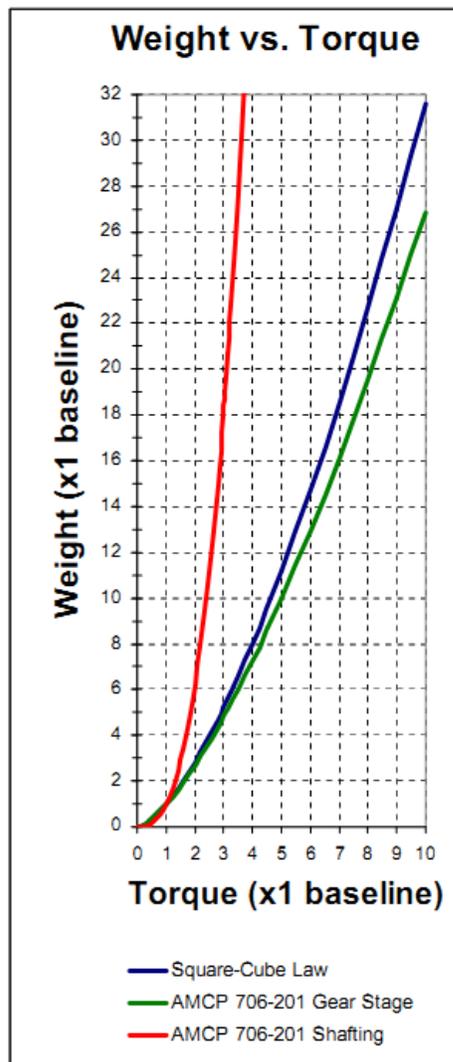


Figure 32. Weight – torque relation [47]

Another method used to estimate gear weight is based on the solid rotor volume method, where the volume is calculated as face width multiplied by square of pitch diameter (Equation 3), as presented by Willis [55]. The gear volume is also similarly calculated (Equation 4) using the following formulae:

$$Fd_p^2 = \frac{2T}{K} \left( \frac{m_G + 1}{m_G} \right)$$

Equation 3. Pinion solid rotor volume [55]

$$Fd_g^2 = Fd_p^2 . m_G^2$$

Equation 4. Gear solid rotor volume [55]

$$\sum Fd^2 = \frac{2T}{K} \left( \frac{m_G + 1}{m_G} \right) + \frac{2T}{K} \left( \frac{m_G + 1}{m_G} \right) m_G^2$$

Equation 5. Gear set solid rotor volume [55]

Where,

$F$	Pinion face width
$d_p$	Pinion pitch diameter
$d_g$	Gear pitch diameter
$T$	Torque
$m_G$	Reduction ratio
$K$	Surface durability factor

Stepniewski and Shinn [56] empirically formulated rotorcraft gearbox (Equation 6) and drive shaft weight (Equation 7) based on their study of Soviet and Western Helicopters. The total weight of the drive system is a sum of the two weights.

$$W_{gb} = 172.7 T_{mr_{gb}}^{0.7693} T_{tr_{gb}}^{0.079} n_{gb}^{0.1406}$$

Equation 6. RTL gear box weight formula [56]

Where,

$W_{gb}$  Total gearbox weight (lbs.)

$T_{mr_{gb}}$  Ratio of HP to main rotor RPM

$T_{tr_{gb}}$  Ratio of tail rotor HP to its RPM

$n_{gb}$  number of gearboxes

$$W_{dsh} = 1.152 T_{mr_{gb}}^{0.4265} T_{tr_{gb}}^{0.0709} L_{dr}^{0.8829} n_{dsh}$$

Equation 7. RTL shafting weight formula [56]

Where,

$W_{dsh}$  Total drive shafting weight (lbs.)

$L_{dr}$  Horizontal distance between rotor hubs (ft)

$n_{dsh}$  number of drive shafts excluding rotor shaft

The Boeing-Vertol weight formulae for main rotor and tail rotor, as studied by Stepniewski and Shinn [56] are:

$$W_{dsmr} = 250a_{mr} \left[ (HP_{mr} / rpm_{mr}) z_{mr}^{0.25} k_t \right]^{0.67}$$

Equation 8. Boeing-Vertol main rotor drive system weight [56].

Where,

$a_{mr}$  Empirically estimated adjustment factor

$z_{mr}$  number of stages in main rotor drive

$k_t$  Configuration factor ( $k_t=1$  for SMR)

$$W_{dstr} = 300a_{tr} \left[ 1.1(HP_{tr} / rpm_{tr}) \right]^{0.8}$$

Equation 9. Boeing-Vertol tail rotor drive system weight [56]

Where,

$a_{tr}$  Empirically estimated adjustment factor

Saribay et al. [57] studied the optimization of Intermeshing Rotor Transmission System Design. They calculated gear bending stress, contact stress and the allowable power carrying capacity using AGMA methods. They build multiple ‘cases’ to study the reduction ratio in different stages, shown in Figure 33. The cases define a particular reduction combination between stage 1 and 2.

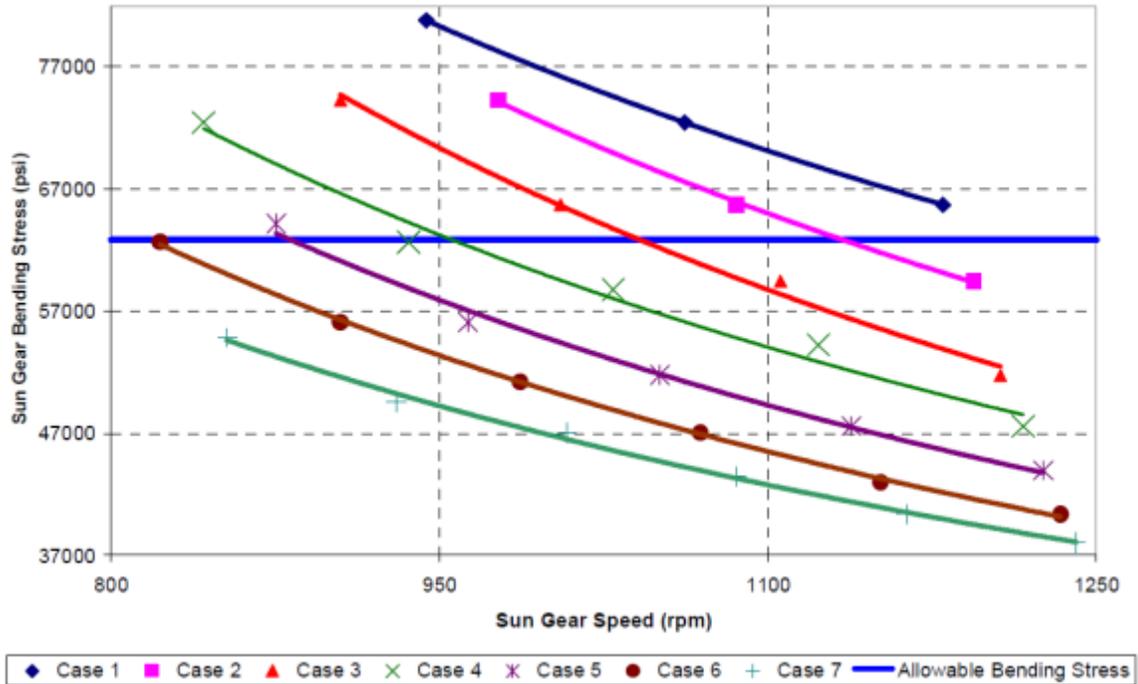


Figure 33. Sun gear bending stress against speed for multiple cases [57]

They then use an indirect method to calculate weight of the gear based on reduction ratio (Equation 10).

$$W = 0.0244 d_p^3 m_g^2$$

Equation 10. Gear weight relation [57]

In their study they enforce stress constraints through a cost function that penalizes a design if the calculated stress is high, regardless of being within the stress limit. Their equation for cost function (objective function) is given in Equation 11. They also assume the allowable stress numbers to be a constant value. The allowable stress must be allowed to vary based on loading cycles and pinion - gear material combination [48].

$$G = \sum_{i=1}^n a_i \Delta \omega_i + b_i \Delta w_i + c_i \Delta s_{ti} + d_i \Delta s_{ci}$$

Equation 11. Gear weight relation [57]

Where,

$a_i, b_i, c_i, d_i$       Weighting factors

$s_t$                       Bending stress

$s_c$                       Contact stress

Chong and Lee presented a volume minimization technique for gear trains using a genetic algorithm. They use a pseudo-objective function with an exterior penalty function to implement constraints [58]. In this study, they optimize a 2-stage gear train (Figure 34) using a typical GA process.

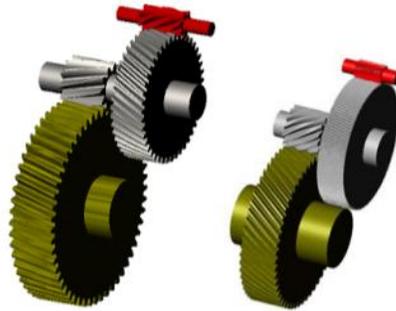


Figure 34. 2-Stage gear train optimization [58]

Figure 35 shows a list of equations implemented by them to address constraints. The first part of the objective function is the volume of the gear and the second, the penalty function. Their study requires the enforcement of many constraints to ensure configuration is within limits and feasible.

$$F_{objective} = \left(\frac{z_r m_n}{2 \cos \beta}\right)^2 b + \sum_{j=1}^p \gamma_j (\text{MAX}(G_j, 0))^2$$

$$\begin{array}{ll}
G_1 = \sigma_{Hs} - \sigma_{Hlim} & G_9 = b \cos \beta - 1.15 z_s m_n \\
G_2 = \sigma_{Fs} - \sigma_{Flim} & G_{10} = 0.25 z_s m_n - b \cos \beta \\
G_3 = v_{ts} - v_{tmax} & G_{11} = b \cos \beta - 1.15 z_p m_n \\
G_4 = 1.0 - \varepsilon_{\alpha s} & G_{12} = 0.25 z_p m_n - b \cos \beta \\
G_5 = \varepsilon_{\alpha r} - 2.5 & G_{13} = b \cos \beta - 1.15 z_r m_n \\
G_6 = 1.0 - \varepsilon_{\beta s} & G_{14} = 0.17 z_r m_n - b \cos \beta \\
G_7 = \varepsilon_{\beta r} - 6.0 & G_{15} = z_s m_n - 500 \cos \beta \\
G_8 = T_f - T_{flim} & G_{16} = (z_p + 2) - (z_p + z_s) \sin\left(\frac{\pi}{N}\right)
\end{array}$$

$$G_{17} = \cos^{-1} \left( \frac{a^2 + r_{a2}^2 - r_{a1}^2}{2ar_{a2}} \right) - \left( \cos^{-1} \left( \frac{r_{a2}^2 - r_{a1}^2 - a^2}{2ar_{a1}} \right) \frac{z_p}{z_r} + inv\alpha_w - inv\alpha_{a1} \right)$$

$$G_{18} = \frac{z_r}{z_p} \left( \sin^{-1} \sqrt{\frac{\left(\frac{\cos \alpha_{a1}}{\cos \alpha_{a2}}\right)^2 - 1}{\left(\frac{z_1}{z_2}\right)^2 - 1}} + inv\alpha_{a2} - inv\alpha_w \right) - \left( \sin^{-1} \sqrt{\frac{1 - \left(\frac{\cos \alpha_{a1}}{\cos \alpha_{a2}}\right)^2}{1 - \left(\frac{z_1}{z_2}\right)^2}} + inv\alpha_{a1} - inv\alpha_w \right)$$

$$G_{19} : \frac{z_r + z_s}{N} = \text{integer}$$

$$G_{20} : \frac{z_s}{N} \neq \text{integer}$$

Figure 35. Objective function and constraints for GA [58]

Padmanabhan et al. [59] studied gear train design using a GA. Their method also imposed multiple constraints to maintain gear teeth values as integers and their method does not account for face width. Yokota et al. [60] formulated an optimal weight design problem of a gear for a constrained bending strength of gear, torsional strength of shafts, and each gear dimension as a nonlinear integer programming problem and solved the same using an improved genetic algorithm. However, in their analysis, certain constraints were not satisfied and the converged solution was not the global optimum. Savsani et al. [61] studied the optimization for the gear train using particle swarm optimization and simulated annealing algorithms. Their method of handling design variables is non-conducive for implementation on larger optimization problems, requiring many constraints to be imposed.

The existing optimization techniques studied present a problem in scaling for implementation on larger gear trains. Methods to handle constraints and variables in a nonlinear design space need to be investigated.

An extensive library of literature exists that discuss design techniques and standards for gear trains [20, 62, 63]. Some general gear design handbooks have published information to develop sizing tools [64-66]. Gear technology and design methods are being investigated and improved [67]. New designs such as face gears are being developed and deployed; these designs offer superior performance, improved durability and torque carrying capability [68-71]. Kapelevich et al. [72], discuss methods to minimize bending stress by using root fillet geometry modifications. Root fillet modifications and Trochoid design, discussed by Math and Chand [73], offer designs for potential improvements. Complex methods to calculate AGMA geometry factor J were studied [48, 74]. These methods eliminate the necessity of using tables and interpolations that aren't very conducive to computing.

#### **2.4 Genetic Algorithm**

The Genetic Algorithm (GA) is studied to be an effective technique to handle nonlinear design space with discrete and integer variables. However, it is important to identify settings and tuning factors in GA that are problem specific, reliable and efficient. The method in which constraints are handled and their effect on convergence is of particular importance in realizing the fully-relational design framework.

The optimization of gear train is a highly constrained problem; a literature review to study various constraint handling methods employed in GAs was done. Penalizing strategy is a technique adopted to consider infeasible solutions in genetic search. Penalty technique perhaps is the most commonly applied technique used in constrained GA problems. The main issue with the penalty strategy, as identified by Glover [75], is how to design the penalty function  $p(x)$  which can effectively guide genetic search towards a favorable area of solution space. Several techniques have been proposed in the area of evolutionary computation; however, there is no general guideline on designing penalty function. Constructing an efficient penalty function is very problem-dependent.

Gen and Cheng [76, 77], in their survey of penalty techniques in GAs studied the following techniques:

1. Rejecting strategy
2. Repairing strategy
3. Modifying genetic operators strategy, and
4. Penalizing strategy

They discuss the advantages and disadvantages of each of these strategies. Rejecting strategy discards all infeasible chromosomes throughout whole evolutionary process. Repairing strategy depends on the existence of a deterministic repair procedure to convert an infeasible offspring into a feasible one. These strategies have the advantage that they never generate infeasible solutions but have the disadvantage that they consider no points outside the feasible regions. For highly constrained problem, infeasible solution may take a relatively big portion in population. Glover [75] suggests that constraint management techniques that allow movement through infeasible regions of the search space tend to yield more rapid optimization and produce better final solutions than do approaches limiting search trajectories only to feasible regions of the search space.

Yeniay [78] studied penalty functions for constrained problems and categorized them as following:

1. Methods based on penalty functions
  - a. Death Penalty
  - b. Static Penalties
  - c. Dynamic Penalties
  - d. Annealing Penalties
  - e. Adaptive Penalties

- f. Segregated GA
  - g. Co-evolutionary Penalties
2. Methods based on a search of feasible solutions
    - a. Repairing infeasible individuals
    - b. Superiority of feasible points
    - c. Behavioral memory
  3. Methods based on preserving feasibility of solutions
    - a. GENOCOP system
    - b. Searching the boundary of feasible region
    - c. Homomorphous mapping, and
  4. Hybrid methods

Multiple techniques have been researched to improve the GA functionality and to obtain satisfactory exploration and exploitation. Srinivas and Patnaik [79] recommend the use of adaptive probabilities for crossover and mutation to realize the twin goals of maintaining diversity in the population and sustaining the convergence capacity of the GA. Grefenstette [80] formulated the problem of selecting  $p_c$  (crossover probability) and  $p_m$  (mutation probability) as a sub-optimization problem. The disadvantage of Grefenstette's approach is that this could prove to be computationally expensive. If the probabilities are determined adaptively by the GA itself, the user and the algorithm are relieved of having to specify the values of  $p_m$  and  $p_c$ . Adaptive genetic algorithms have also been very effective in multiobjective problems, as researched by Bingul [81].

DeJong [82] introduced the idea of 'overlapping populations' and 'crowding' in his work. In the case of 'overlapping populations', newly generated offspring replace similar solutions of the population, primarily to sustain the diversity of solutions of the population and to prevent premature convergence and being overly exploitive. The technique, however, introduces a parameter CF (the crowding factor), which has to be

tuned to ensure optimal performance of the GA. The concept of ‘crowding’ led to the ideas of ‘niche’ and ‘speciation’ in GAs, as studied by Goldberg [83] for multimodal functions.

Srinivas and Patnaik [79] discuss two major characteristics of a GA: the first being the capacity to converge to ‘a optimum’, and the second being, its capacity to explore new regions of the design space in search of the global optimum. Increasing the value of  $p_m$  and  $p_c$  promote exploration at the expense of exploitation. They suggest using adaptive values for these probabilities:

$$p_c = k_1(f_{\max} - f') / (f_{\max} - \bar{f}), \quad k_1 \leq 1.0$$

$$p_m = k_2(f_{\max} - f) / (f_{\max} - \bar{f}), \quad k_2 \leq 1.0$$

Equation 12. Adaptive crossover and mutation probabilities [79]

Where,  $k_1$  and  $k_2$  are constants used to maintain  $p_c$  and  $p_m$  within tolerance,  $f'$  is the greater of the two mating parents and  $\bar{f}$  is the mean fitness value of the current population.

Cantu-Paz [84], in his study, describes the concept of parallel GAs and multiple sub-populations. Belkadi et al. [85] discuss the idea of migration between multiple sub-populations in a parallel GA. In parallel GA, sub-populations are isolated so the optimization progresses with greater diversity even when the algorithm is aggressive. These methods have implied benefits in enhancing diversity of the population without having detrimental effects on the exploitation.

## **2.5 Flexibility**

The early phases of design contain multiple sources of uncertainties in describing design, the decision making process in this phase exerts a critical effect upon all design

properties [86, 87]. Handling uncertainties and downstream changes, is therefore critical to the successful implementation of CE [88].

Saleh et al. [89] studied the concept of flexibility as relevant to the manufacturing, multidisciplinary design, and real options trading community. They claim that the notion of flexibility has been used in various fields but very few attempts have been made to formally define, quantify, and propose ways for achieving flexibility. With respect to life-span, they define flexibility as an attribute that offers a longer lifespan, as shown in Figure 36.

Price et al. [16] define a flexible design as one that is least sensitive to changing system objectives and the changing environment. The following figure (Figure 37), illustrates the concept of ‘flexible design’.

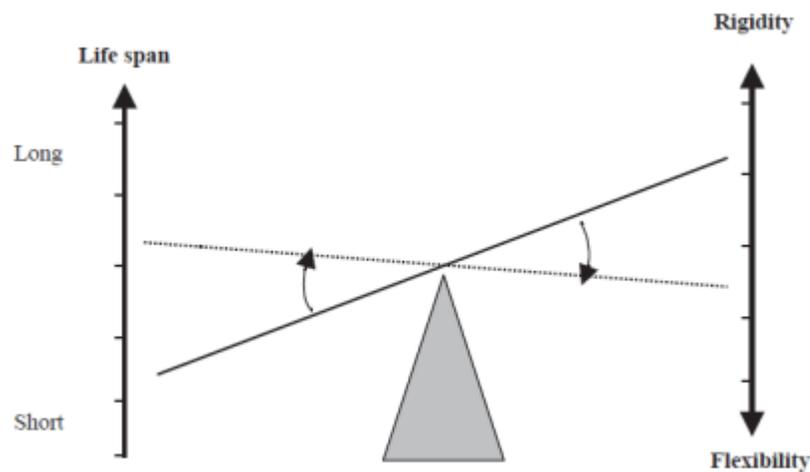


Figure 36. Simple model relating a system's life span and its flexibility [89]

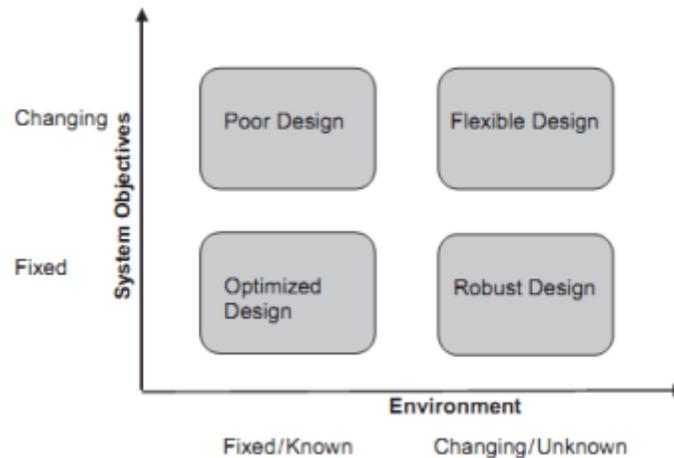


Figure 37. Flexible design in terms of system objectives and environment [16]

Flexibility is a word rich with ambiguity and is broadly defined as the ability to respond to change [89]. In the manufacturing community, different types of flexibility are defined based on the nature of change and the ability of the production system to accommodate this change. There is a great resource of literature on Flexible Manufacturing Systems (FMS); many topics are addressed in this field – ranging from the design of manufacturing cells and machine grouping, to the scheduling, loading, and control of FMS [90]. ‘Volume flexibility’ is defined as the ability of a production system to handle changes in volume demand on a weekly or hourly basis, of the same product, thus allowing the factory to operate profitably at varying required production levels. ‘Product mix flexibility’ is defined as the ability to manufacture a variety of products without major modification of existing manufacturing tools and setup. ‘Routing flexibility’ is defined as the ability to process a given set of parts on multiple machines through alternate routes. ‘Operation flexibility’ is defined as the ability to interchange the order of operations required to be performed on a given part, potentially allowing the ease of scheduling its production and decreasing production time [91, 92].

Flexibility in this field of study is not only viewed as a reactive capability, but is also regarded as one that offers a competitive advantage which not only allows an enterprise to respond to change, but also to create change and set a market niche for rapid production and innovation [93]. ‘Agility’ is another term related to the subject under study – to study the ability to respond to change. It was first introduced in manufacturing environments then broadened to encompass the extended enterprise [89]. It is often loosely defined, and used to characterize different things in a business environment. Oleson [94] defines agility as the “ability to respond with ease to unexpected but anticipated events.” He describes ‘agile strategic planning processes’, ‘agile automation’, and discusses the need for ‘agile business relationships’ with suppliers and customers. Similarly, Fricke et al. [95] define agility as the “property of a system to implement changes rapidly”, and flexibility as the “property of a system to be changed easily and without undesired consequences.” ‘Agility’ is thus used as a desired qualitative attribute for an enterprise to thrive in a competitive environment.

Research in the multidisciplinary design has addressed the issue of flexibility; the focus of these efforts has been in achieving ‘flexibility in the design process.’ Typical approaches have consisted of incorporating designers’ preferences with degrees of satisfaction in specifying design requirements. Thurston [96], for example, uses utility theory-based preference functions to express designers’ preference over single or multiple attributes. Wallace et al. [97] define specification functions to indicate the subjective probability that performance levels are achieved.

Fuzzy goals have been recommended to model the degree of satisfaction level [98]. Approaches such as this and other probabilistic methods have been developed to address this type of flexibility [89].

Saleh et al. [89] put forth the following questions:

1. How can one design for flexibility?
2. What are the design practices for embedding flexibility in design?
3. What are the tradeoffs associated with designing for flexibility.

Inoue et al. [44] studied the effect of uncertainty in describing a design by the use of a Preference Set-based Design (PSD) to identify a set of possible design solutions. The problem with the PSD method is that a large number of solutions need to be considered and the solution selection becomes sensitive to user preferences based on a preference rating and is highly dependent on the design space being continuous. The complexity of the whole product and the multidisciplinary aspect is not captured in the part preference rating. Hence its implementation for CE for aerospace has its obvious shortcomings. However, the Set-Based Concurrent Engineering (SBCE) method is a powerful model for efficient design and management of large scale operations [10, 99]. In SBCE, multiple functions define a broad set of solutions from their respective areas of expertise in the design space [9]. The broad set of each function corresponds to a kind of uncertainty. SBCE depends a great deal on making design decisions later in the design process. This notion is, in essence, prohibitive for application in complex aerospace systems.

Chen and Lewis [100] define their understanding of flexibility in the design process, as follows: “Our aim is to provide flexibility in the design process and to help further resolve the conflicts and disputes of rationality between the interests of multiple disciplines. By flexibility we mean that instead of looking for a single point solution in one discipline’s model, we look for a range of solutions that involve information passing between multiple players (disciplines). With this flexibility, the design freedom of individual disciplines could be significantly improved.”

DeLaurentis and Mavris [101] presented methods for design uncertainty modeling through robust design, modeling and simulation. They study robust design as a means to

obtain good design solutions in the presence of uncertainty. Their method to study propagation of uncertainty in multidisciplinary interactions is particularly useful. Mavris and Bandte [102] use a similar approach through metamodeling techniques to study constrained robust design.

Flexibility cannot be directly differentiated from robustness by the definition of ability to handle change. It is important to define ‘change’ and also what is meant by ‘ability’.

Saleh et al. [89] suggest three points to characterize ‘change’:

1. A time reference associated with the occurrence of change, i.e. when is the ‘change’ happening in the lifecycle of the system
2. A characterization of what is changing; for example, the system’s environment, the system itself, or the customer’s requirements.
3. An indication for providing metrics of flexibility, or the ability to rank different designs according to their flexibility.

The concept of flexibility and its potential for value is a well-accepted fact. However, as Trigeorgis [103] questions – “precisely how valuable is flexibility and how can its value be quantified?”

The literature survey of flexibility and its use in different fields of research help develop the background for understanding what is expected from a system in terms of change and response to change.

## **2.6 *Finite Element Analysis***

Finite Element Analysis (FEA) is a numerical technique to solve boundary value problems. It works by discretizing the analysis zone to several elements and approximates the solution based on partial differential equations: the discretized zone is

called the mesh. FEA harnesses computing capability to perform this analysis; the accuracy of the model is also proportional to the refinement of the mesh. The denser the mesh, the greater is the computing requirement. Although FEM has been around since 1960s, its real capability only emerged after computing performance matured [104]. One of the earliest mentions of using FEA for spur gears was by Coy et al. [105] in 1985. They identified the need to formulate rating based stresses and deflections as derived from structure stiffness. They claim the most powerful method for determining accurate stress and deflection information is the FEM. However, back then, the FEM based analysis was computationally expensive and the data handling cumbersome. They note that the capability of FEA is further limited by how the problem is setup; based on:

1. Number of elements
2. Representation of boundary conditions
3. Aspect ratio of solid brick elements.

Early analyses of gears using FEM were labor intensive. Some of the challenges faced earlier were with modeling the geometry and approximating the features. The next complication was with mesh refinement and building the right quality elements. In FEA, areas of interest where stresses or deflections are expected to be high are required to be meshed as densely as possible. As a consequence most of the FEA analysis on gears has been limited to a 2D problem. Analysis of spur gears as a 3D problem began in the early 1990s [106]. FEA methods have been verified across test data and have shown a high level of correlation, validating this approach [104].

According to AGMA Design Guidelines for Aerospace Gearing [48], modern finite element methods can be used if the pinion and gear are modeled as separate parts and the tooth load induced by torque applied to the shafts through the use of gap elements or with three dimensional contact modeling. The finite element technique is suitable for calculating load distribution because all of the factors which influence deflection and

manufacturing deviations can be evaluated. Also using FEA, accounts for variables that support stiffness and temperature effects. Accounting for these factors permits the load distribution to be determined.

Dynamic analysis of gears have demonstrated higher loads, as much as 50% as compared to static loads [106]. The problems of strength and dynamic loads, as well as resonant frequencies for such gearing, are now treatable with techniques such as finite element analysis [105].

Another area of application for FEA based analysis of gears is in vibration studies; the ability to analyze a given gearbox and modify its design, based solely on this analysis, in order to minimize its operating vibration level, requires the use of several finite element modeling techniques. These analyses define the excitation due to the gears, the response of the shaft support system to these excitations, the manner in which these shaft responses are transferred to the housing through their bearings, and the response to these various stimuli. In general, the approach involves the following analyses:

1. Modeling the gear teeth for local dynamic flexibility and kinematic loading
2. Natural frequency analysis of the gear flanks to determine the mode shapes and frequencies of these components;
3. Determination of the dynamic gear loads applied to the components
4. A detailed finite element model of the static gearbox structure
5. An analysis of the modes of the entire system

FEA for gears has been largely solved as plane stress problem, limiting its capability for Bevel and Helical gears [105]. Rao and Muthuveerappan studied FEA for a helical gear tooth. In their study, they discretize the tooth in 250 eight-noded isoparametric brick elements with 408 nodes, shown in Figure 38. Their FEA model

assumes the case to be static and they apply a progressively varying static load across the discretized contact line.

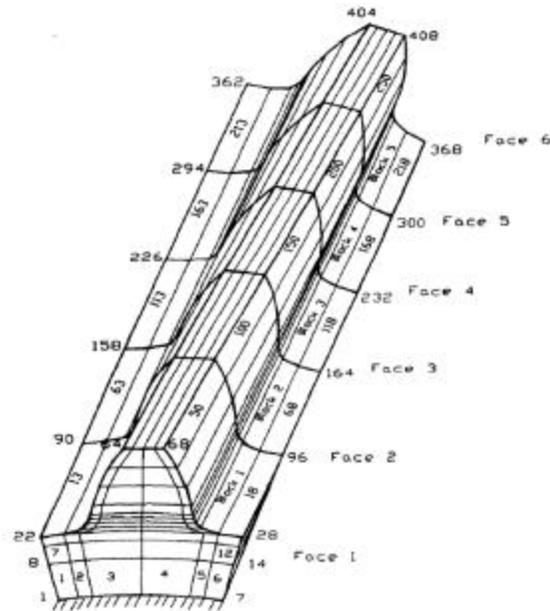


Figure 38. Finite element model of helical tooth [107]

Rameshkumar et al. [108] published their work on FEM based analysis for high contact ratio problems. Their problem was still modeled as a plane stress case and is seen in Figure 39.

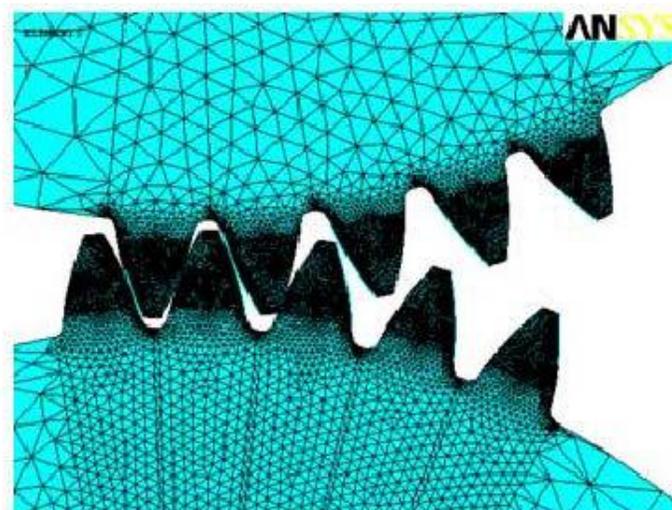


Figure 39. Plane stress model of high contact ratio gears [108]

Stoker et al. [104, 109] developed a program to parametrically model and analyze gears through FEA( Figure 40 and Figure 41). Their work primarily focused on studying stress and wear as a result of non-ideal loading. This study included a 2-D plane strain type of analysis with slight expansion to 3-D using APDL. They studied the load sharing phenomenon as a result of non-ideal loading. They conclude that the AGMA model underestimated the bending stress by about 44%. Possible reasons for this discrepancy being:

1. The analytical model only considers bending stress, while the numerical model considers bending, shear and axial stresses.
2. The contact force is not tangent to the pitch circle but perpendicular to the involute curve
3. The gear tooth is too short to be considered as a slender beam as Lewis bending theory assumes it to be.

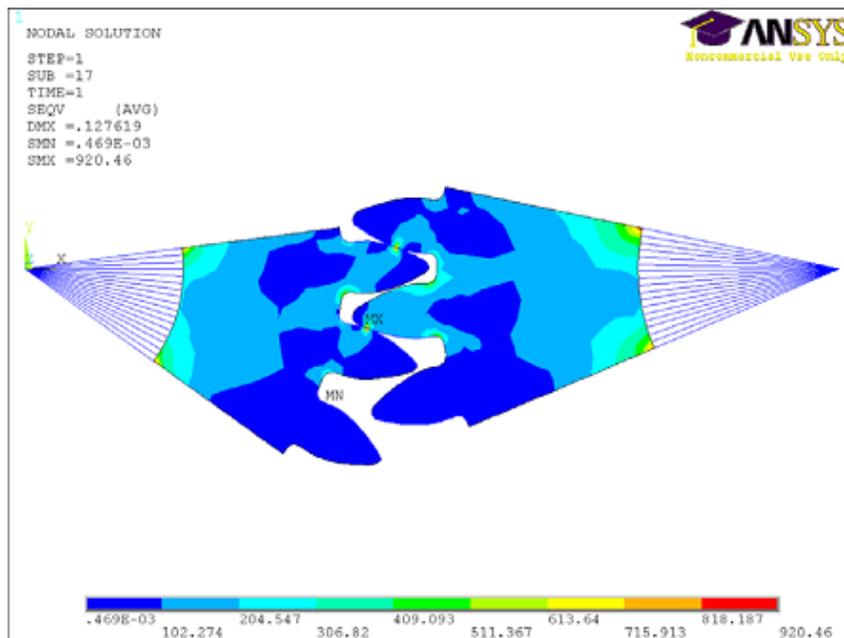


Figure 40. Plane strain analysis of spur gear [109]

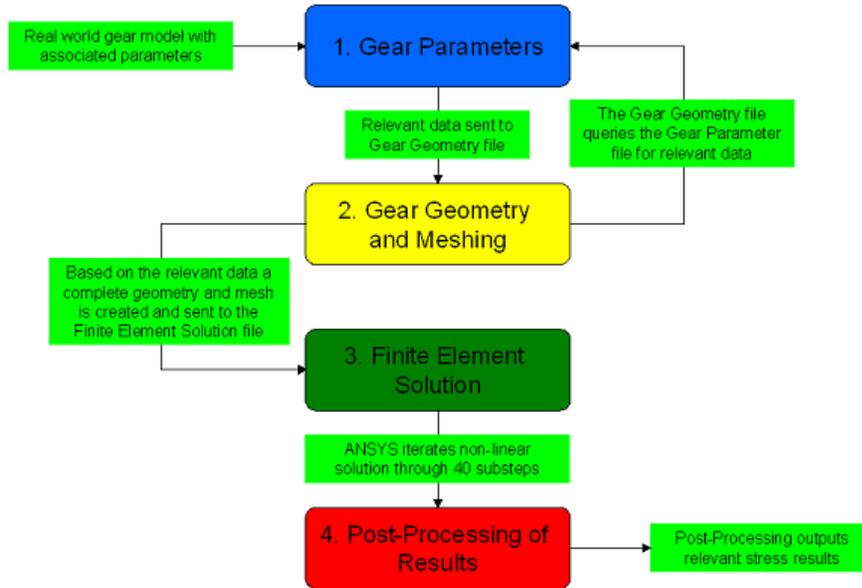


Figure 41. Parametric gear modeling and FEA setup [109]

Hassan, in his research paper [110] studies the contact stress of two spur gears in different contact positions, representing a mating pair during rotation. He uses APDL to model the contact problem (Figure 42). The finite element model recognizes possible contact pairs by the presence of specific contact elements. The contact elements are then interpreted with the model exactly where they are being analyzed for interaction. An eight node iso-parametric plane stress quadratic quadrilateral element was used to build the finite element model of the teeth. The type of contact was node to surface, the gear being the target element and the pinion being the contact element.

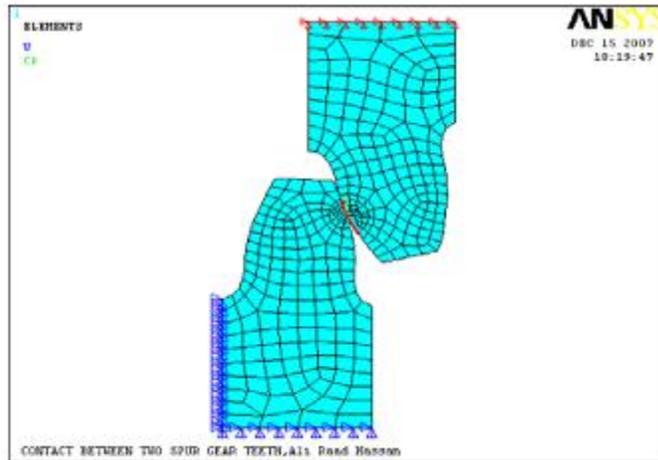


Figure 42. Contact model boundary conditions [110]

One of the advantages in finite element analysis is that the bending contact stress can be calculated at various rotational angles of the gear. Figure 43 shows the tensile and compressive bending stresses at the base of gear tooth as a function of rotational angles. It can be found that as the tooth rotates, the bending stress gradually increases and maintains almost a constant value for about ten degrees. At around eight degrees of rotation, the bending stress in both tension and compression drops significantly. This is due to three teeth being in contact simultaneously. Up until the point where the bending stress decreases, there are only two teeth in contact. For a few degrees of rotation, there are three teeth in contact, which reduces the bending stress by as much as 11%. Once the gear rotates further and only two teeth are in contact, the bending stress increases [104]. Load sharing and stress distribution can be studied using FEA for non-standard gears. However, the complexity of the analysis makes automation very challenging.

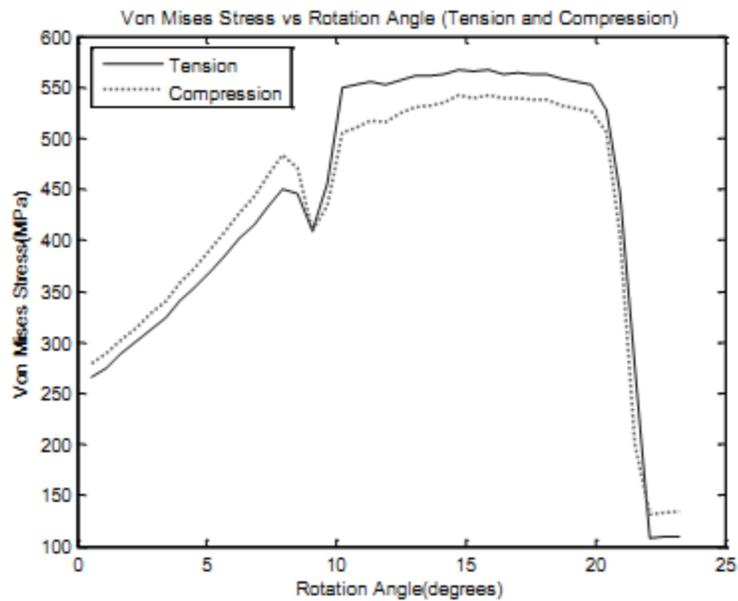


Figure 43. Bending stress at gear tooth base – tension and compression [104]

Li et al. [111] studied a single tooth loading independently as a bending and contact problem. When they studied the pure bending stress problem by applying a force on the tooth flank, the bending stress matched the predicted stress distribution. However, when they performed the contact analysis, they had to scale the torque such that the bending effect on the root fillet region matched that from the maximum single tooth loading case (Figure 44).

In a study performed by Kirov [112], AGMA and FEA formulations are compared and he concludes that FEA is superior to AGMA and should be used extensively. He also notes - FEA has its inherent errors and the AGMA calculations are empirical and proven by field experiments. He also notes that it is potentially difficult to make direct comparison between the two methods. There is a desire to model teeth as FEA problems and use that information in conjunction with AGMA sizing methods as application factors and stress-modifying factors [105].

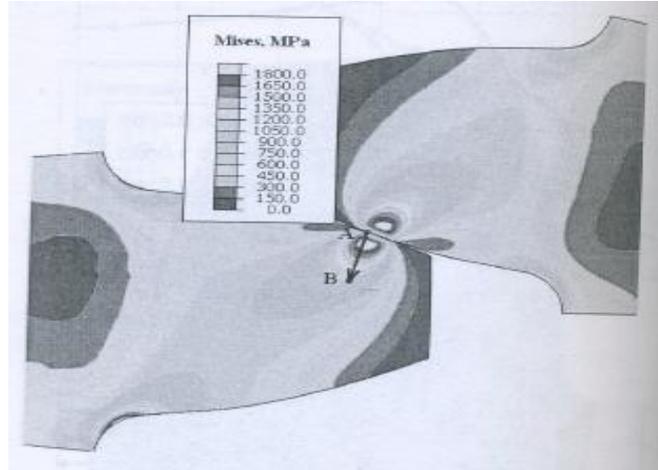


Figure 44. Stress distribution from dynamic contact analyses [111]

Gear web and non-standard gear profiles are estimated to produce about 30% in weight saving [66, 72]. The benefits of this weight reduction are huge and cumulate to a lot empty weight savings when applied to all the gears used in the transmission. Experimental gear studies also indicate that a 10% reduction in stress concentration can yield about 50% improvement in fatigue life [113].

Literature review of FEA has identified limitations, such as lack of combined bending and contact analysis, nonlinear 3-D analysis and also the need to eliminate load scaling so a consistent formulation can be used.

## 2.7 *Summary of Literature Research*

The systems engineering study of the aerospace design process have converged at the primary bottleneck being in interdisciplinary integration and knowledge transfer, change propagation, and balancing detail and fidelity. A broad level study of this problem is required and methods introduced that successfully study the aspects of systems integration, automation, and timely introduction of high fidelity analysis.

The advent of parametric and advanced CAD modeling enables multiple engineers to collaborate and use a common geometry database. Existing technology permits sharing of geometry and hierarchical change propagation. However, the level of propagation is restricted to geometry. It would be advantageous to design community to extend the scope of existing parametric geometry capability to influence design through engineering analyses. Change propagation that can trigger engineering analysis in an automatic fashion can further enhance early design tradeoff capability.

Empirical methods used in early design to determine drive system weight and efficiency do not allow for trade study of configurations and provide very little information on the volume and geometric arrangement. Existing methods use relations that give a rough approximation of the whole system weight and efficiency, which are useful in early study of the entire vehicle, but this does not allow for the sub-system design to come into play with the overall design. This is where a gap exists in design integration between sub-systems. The more information can be gathered earlier, greater design freedom can be achieved in design space exploration and concept selection. There is a need to introduce a drive system preliminary design methodology that can effectively close this gap.

A suitable optimization technique for gear trains is needed to automate and accurately optimize the system. Existing techniques are either overcomplicated or oversimplified. The complex optimization techniques impose many constraints for a simple system and make any scaling for a large drive system impractical. The simplified methods do not treat constraints well and require a graphical approach which is non-conducive for automation and design of larger drive trains.

## CHAPTER 3

### RESEARCH QUESTIONS, CONJECTURES AND HYPOTHESES

Literature survey and study of aerospace systems engineering have shown the need for an improved design framework at the sub-system level mapped with the customer requirements and the system objectives. Thus, integrating tools at varying levels of fidelity has to be achieved and seamless transfer of information between these tools established. Automation of the entire process while maintaining relations with associated systems has to be accomplished. Automation at the expense of fidelity and precision is undesirable and therefore automation methods used must be conscious of these requirements. It is of importance to study the balance between fidelity and detail in different stages of design and understand the value of information as it pertains to design decisions; and optimization techniques and analyses must consider computing technology and requirements.

- **Research Question #1:** *How can system integration be effectively performed at the sub-system level?*
- **Research Question #1a:** *What are the requirements of a framework and the logical steps involved in integrating a system with its associated systems to perform tradeoffs?*

***Conjecture #1:** Using the fully-relational design technique a subsystem can be effectively integrated with its associated systems; this greatly enhances the capability to perform tradeoffs and increases design cycle efficiency.*

This research question addresses the broad level objectives to improve knowledge available by integrating design process for more systems and the objective to enable design-for-change. This question directly focuses on the need to develop a framework flexible to interfaces that is fast and accurate with integration and automation capability (Research Objective 1)

The implementation of a Fully-Relational Design (FRD) system requires the development of common geometry database, specifying explicit relations, and multidisciplinary integration with automated information flow capability. It requires at the system level, the development of a generic, requirements-driven, design and optimization framework. FRD, as described above, is implemented here on a three-stage gear box and the requirements studied. This methodology is further expanded to a large scale single main helicopter drive system.

- **Research Question #2:** *Is a Genetic Algorithm suitable to optimize gear train optimization problem.*
- **Research Question #2a:** *How can Genetic Algorithms be improved to optimize gear train type constrained problems?*

***Conjecture #2:** Optimization of gear trains can be setup using a GA with a sub-optimization routine. The performance of a GA can be improved by including innovative methods such as Adaptive Crossover and Mutation Rates, Migration between sub-populations and introduction of random members. The problem of handling constraints in GA can be alleviated by investigating effective penalizing techniques.*

This research question addresses the research objective to improve knowledge available and to improve understanding of optimization techniques for gear train design.

A suitable optimization technique is required to effectively implement FRD. It is important to also investigate the expandability and performance improvement that can be derived from a genetic algorithm optimization technique.

Gear train optimization is a complex task requiring selection of teeth pairs with the same diametral pitch, helical/spiral angle, to match a given gear ratio. Literature review of current optimization techniques show the need for improved methods that are applicable for rotorcraft drive systems. Experiments can be performed on different constraint handling techniques to evaluate their performances. Constraint dependencies can be studied – since structural constraints are highly dependent on each other. Furthermore, a highly nonlinear design space can lead to premature convergence if the optimization method does not account for it. Techniques such as adaptive rates, migration between subpopulations, and introduction of random members can be studied, and their effects validated through experiments. The optimization methods and experiments are discussed in detail in Section 5.2 (page 132).

- **Research Question #3:** *How do geometric and spacing requirements affect gear train design?*

***Hypothesis #1:** Geometric location of input and output shafts of a given gear train and volumetric constraints of the housing influence the design. This interaction can be quantified and used to alter the optimal configuration.*

This Hypothesis is directly related to the Research Objective 5 to *develop a method to enforce geometry and space constraints through 'fully-relational' design.*

This hypothesis states that when the shaft locations are changed, for a simple gear train, and a volume constraint imposed on the system, the optimal configuration will change. This means that an initially optimal design is infeasible under certain variation in volume and input/output locations and another sub-optimal design is feasible in this requirement space. To test this hypothesis, two simple gear train designs are assembled within a geometry analysis system to test the implications of volume and input and output shaft locations.

**Substantiation criteria:** To study this hypothesis, experiments are setup to see if optimality of designs changes under geometric variation. This hypothesis is substantiated if the optimal design becomes infeasible under a given condition while an initially sub-optimal design remains feasible.

The geometry and spacing analysis is performed on a three-stage spur gear system and is discussed in Section 5.3 (page 154).

- **Research Question #4:** *How can design engineers select between distinct families of designs, early in design, without sacrificing capability in later stages of the design process?*

**Conjecture #3:** *Flexibility, studied as a metric in the evaluation of alternatives, in early design, helps the designer select a concept that has improved capability and adjustability to possible later changes in upstream information.*

Developing a method to select a ‘flexible’ configuration in conceptual design stages is an important research objective in this thesis that this research question and conjecture address (Research Objective 6).

Flexibility is defined here as the capability of a particular configuration to continue to perform well (i.e. not degrade in functionality or performance greatly), when variability is introduced in the system requirements space (propulsion, rotor, etc.). A more ‘flexible’ alternative can be selected by studying the variation in the objective function when influencing design parameters are changed. Literature research on the topic of uncertainty in design phase describes how *flexible* is different from *robust* design; the objective of a *robust* design is to be less sensitive to uncertainties in environmental factors and noise variables. Objective of *flexible* design is to obtain a design that performs well and is adaptable to varying requirements with respect to environment and around engineering requirements that influence the design.

- **Research Question #5:** *Can a sufficiently fast and accurate 3-D Finite Element Analysis method be used for advanced design of gears?*

*Conjecture #4: Proper formulation of gear contact analysis can be developed to obtain results that are consistent, fast and accurate over a broad range of gears without the need for load scaling and formulation modifications.*

Developing the suitable formulation for this problem involves studying mesh convergence, contact treatment, interface treatment, solution stabilization, and time step controls. A set of experiments can be setup to study the accuracy and consistency of the formulation in regards to various tuning parameters and algorithms. A consistent formulation should be able to model the bending and contact phenomenon together over a wide range, such that it can be integrated within a design framework. Numerical errors occur in FEA due to multiple reasons and they are very problem specific. Linear static solvers are almost immune to numerical errors but nonlinear dynamic solutions are very

sensitive to these errors. The formulation developed should be less prone to numerical errors and consistent results obtained when small modifications to the geometry are made.

- **Research Question #6:** *Is advanced analysis of gears using Finite Element Analysis a 'design refinement' or does it alter the overall optimal configuration? Is this level of fidelity required for preliminary sizing of the drive system?*
- **Research Question #6a:** *How does gear-web topology optimization process impact gear weight and the overall drive system design? Is the consequent weight saving information large enough to change the design selection?*

***Hypothesis #2:** Information obtained from topology optimization does not influence the preliminary design decision and can be treated as design refinement.*

According to *Hypothesis #2*, using topological optimization for the gear web, it is not possible for a slightly over-sized gear (for a given torque) to weigh less than a gear that is adequately sized for the torque. So the final weight (after topology optimization) of a slightly over-sized gear (a gear that has a torque carrying capacity greater than that required for the application, and hence weighs more) will remain more than that of a gear that was sized for the application. The consequence of this test is that the optimum gear configuration, in terms of weight, will not change as a result of performing topology optimization. Thus the information from topology optimization is only useful after a design is selected

**Falsifiability condition:** Challenging this hypothesis is straightforward. A set of experiments are performed to test the effect of topology optimization on closely related designs. If as a result of these experiments, the optimality of a design changes, this assertion is falsified.

Figure 45 describes *Hypothesis #2*, graphically. Here *design A* is designed for the given torque and selected on the constraint boundary and *design B* is a slightly oversized gear which weighs more and is rated for a higher torque. The hypothesis states that it is not possible, as a result of topology optimization, for *design B* to weigh less than *design A*, even though there is more room for improvement (more material can be removed) in *design B*. Since the design space is discrete, if after topology optimization, *design B* weighs less than *design A*, topology optimization cannot be considered a design refinement, instead must be used to select the optimum design.

Also tested as part of this hypothesis is if a design initially non-optimal that has a greater web volume (more potential excess material removal capability), weighs less than an adequately designed optimal gear. This part of the hypothesis is depicted in Figure 46. Here *design A* weighs less and is farther from the constraint boundary than *design B*. However, in this case *design B* has a larger design volume i.e. web region from where excess material can be removed.

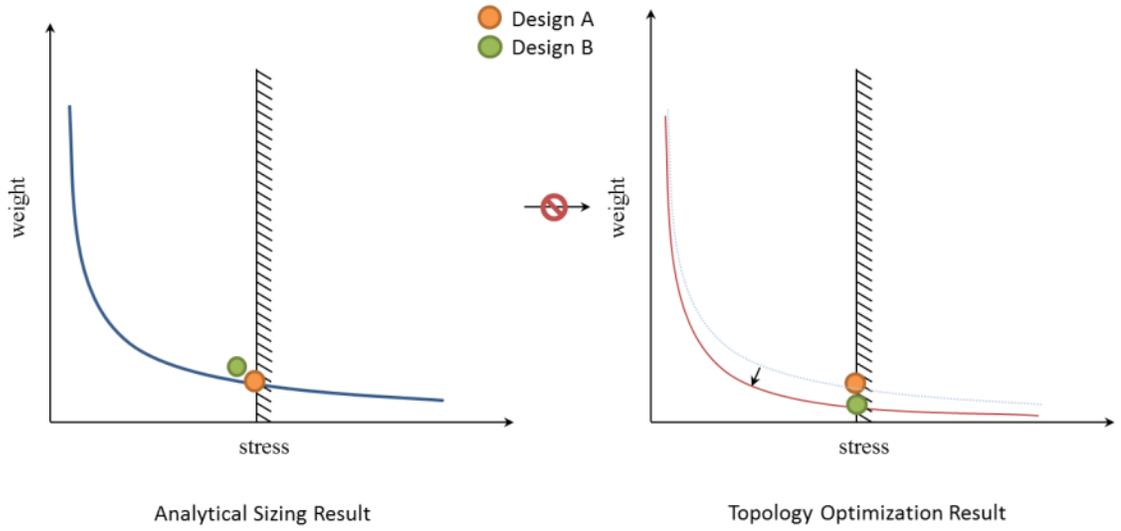


Figure 45. Hypothesis 2 – case 1

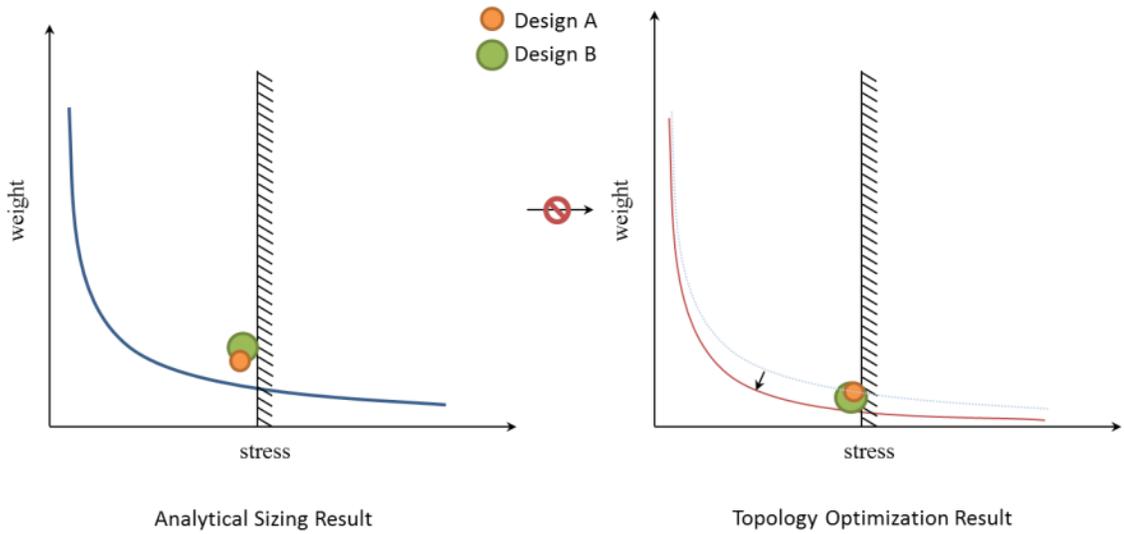


Figure 46. Hypothesis 2 – case 2

If this hypothesis is verified, the CAE-based optimization is a post-sizing design refinement that does not alter the sizing results. The required design process, as a consequence of the hypotheses being falsified, would need topology optimization and

advanced FEA based analysis to be performed within the preliminary sizing and design loop. The consequence of this hypothesis being verified and falsified is shown graphically in Figure 47.

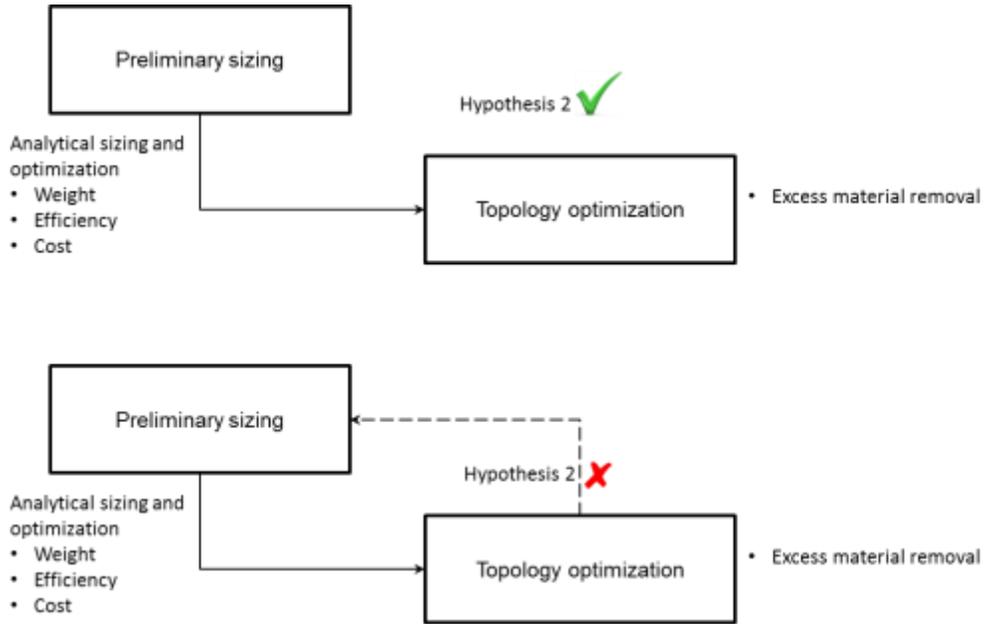


Figure 47. Consequence of Hypothesis 2

Research questions 5 and 6 address the objectives to introduce high fidelity analysis in a timely fashion (Research Objective 4). Conjecture 4 and Hypothesis 2 help improve knowledge available while closing the gap in high fidelity design in early stages (Research Objective 3).

## CHAPTER 4

### METHODOLOGY

#### 4.1 *Fully-Relational Design*

Relational design is a concept used in this thesis to address geometry unification between airframe, engine and transmission housing. Relational design is defined as a method of linking part and product designs within a product structure, with capabilities of parametric design and creation of parent/child relationships to control part behavior. Modern design practices of complex products such as rotorcraft involve hundreds of engineers, designers and experts of several design disciplines. This entire workforce may be scattered around the globe and comprise of multiple organizations. In case of an Original Equipment Manufacturer (OEM) – supplier collaboration, the OEM has to ensure that components and parts meet specific assembly tolerances. For this purpose, information regarding dimensions and attachment details are communicated between parent systems and associated components without giving out all proprietary information of associated parts.

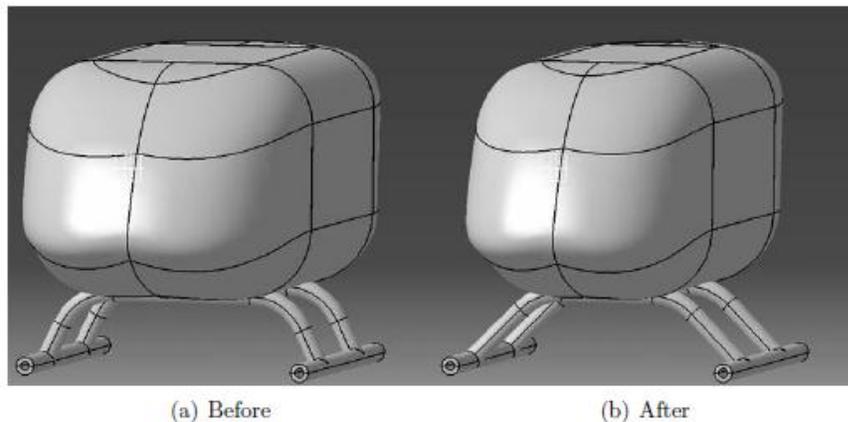


Figure 48. Relational design example [23]

Gunduz studied the relational design as a way to make parts adapt to changes to other parts. In the above figure (Figure 48), the fuselage and landing gear are modeled with reference to datum files. In this example, the fuselage undergoes a dimensional change in the form of narrowing of the width. Using relational design, the landing gear design is automatically adjusted to comply with this change.

Relational design defines a design that uses part-to-part links, thus creating relationships between part geometries. A change made to one part triggers subsequent changes to other parts. A relationship hierarchy is specified to define the order of part-to-part link. This is done in the form of CAD datum files that contain the following:

- Major Dimensions File (MDF): this supports structural arrangement and configuration of parts and subassemblies. It defines position and interfaces of detail parts. MDF is usually composed of points, lines and planes to be utilized as reference entities.
- Surface Definitions File (SDF): this defines theoretical shape of the part or subassembly. Several SDFs constitute source of shape definition for all child products and downstream processes.
- Product Relations Geometry (PRG): contains information regarding geometry that is used to define and coordinate interfaces between two or more parts/subassemblies.
- Installation Management File (IMF): defines the list of installed part instances, together with tolerance and annotation information for installing the parts [23, 24].

In the following example (Figure 49), the parent - child relation for fuselage and former member is described. Here the fuselage structure is the master geometry from which the former derives its guide profile from. Figure 50 shows the relational design

process: widening the fuselage geometry alters the former guide curve and the former geometry is automatically redesigned to conform to this change. However, in this example, the cross-section of the former member's geometry (Figure 51) remains unchanged. This is not ideal because the cross-section was designed based on some initial load specifications and it is not optimal anymore. In some cases this cross-section may not even be feasible because of stress or fatigue criteria.

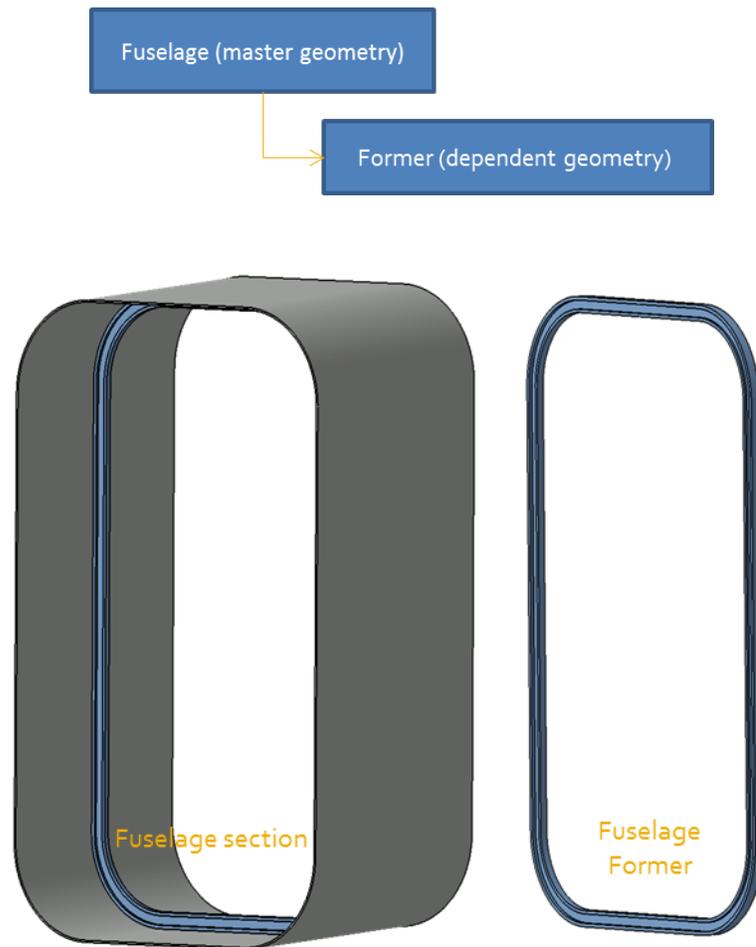


Figure 49. Fuselage – former relational design example

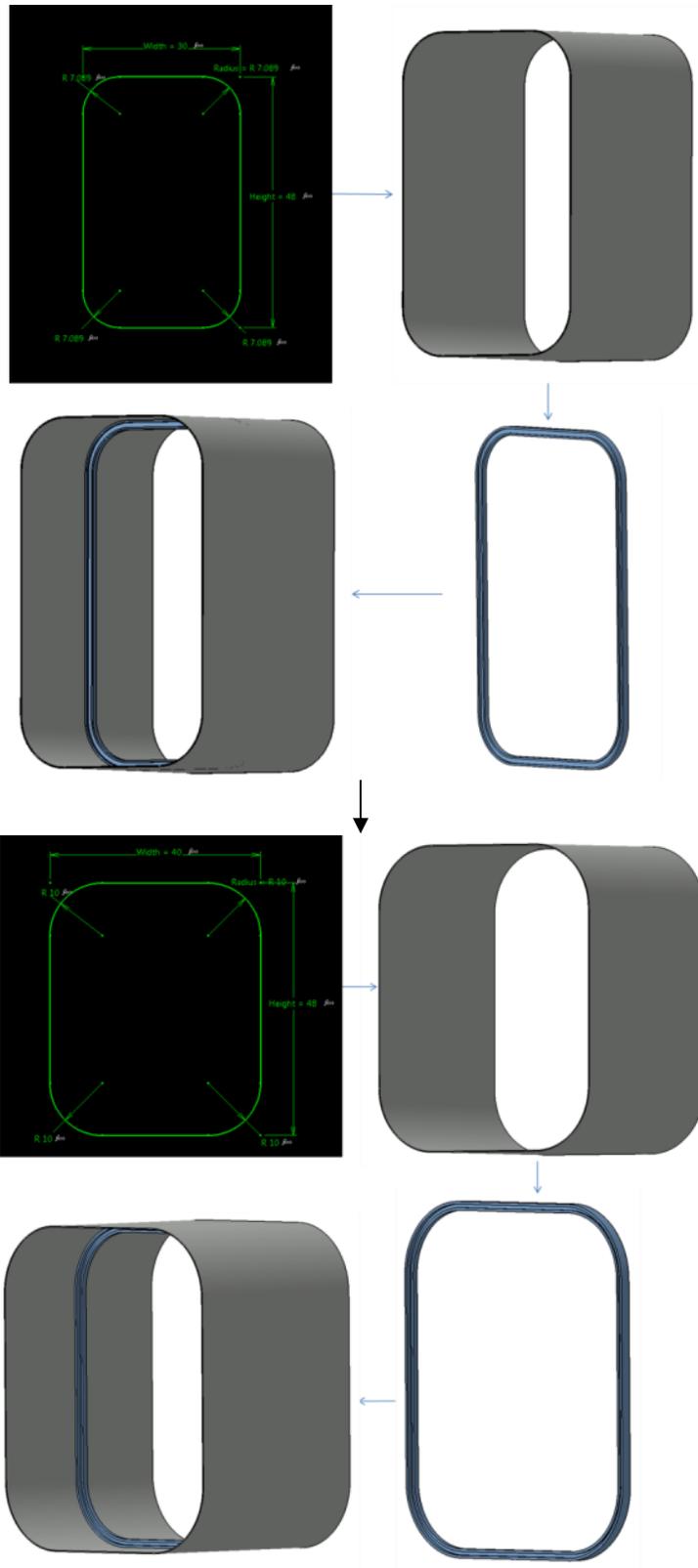


Figure 50. Relational design process for fuselage - former

To expand on the existing capability of relational design, a ‘Fully-Relational Design’ (FRD) design concept is proposed in this thesis. Implementing FRD involves geometry integration along with requirements, constraints and configuration management, where component design can be automatically updated by a series of information from various parent sources.

The FRD process for the above mentioned fuselage – former example is shown in Figure 52 and Figure 53. The inclusion of the ‘Analysis and Optimization’ element helps the designer process the change to obtain an overall optimum structure. The I-shaped section (Figure 51), in this example, gets parametrically changed based on the newly evaluated loads and conditions. It is even possible to completely change the cross-section, for example from I-shaped section to a C-shaped section, based on the analysis and optimization capability. However, it may not be necessary to perform the analysis or optimization for all changes.

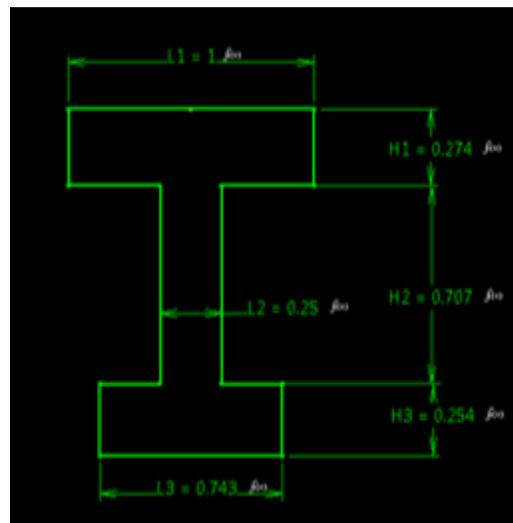


Figure 51. Former member cross-section

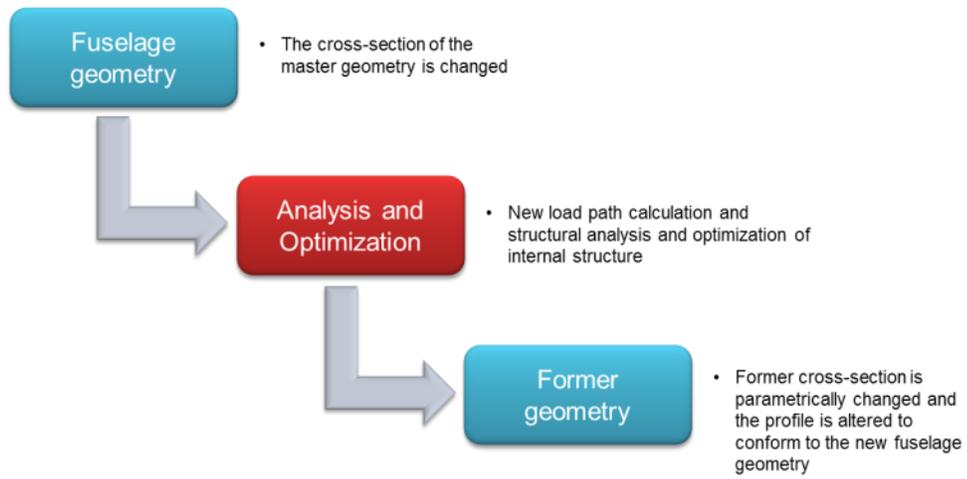


Figure 52. Fully-relational design schematic for fuselage - former

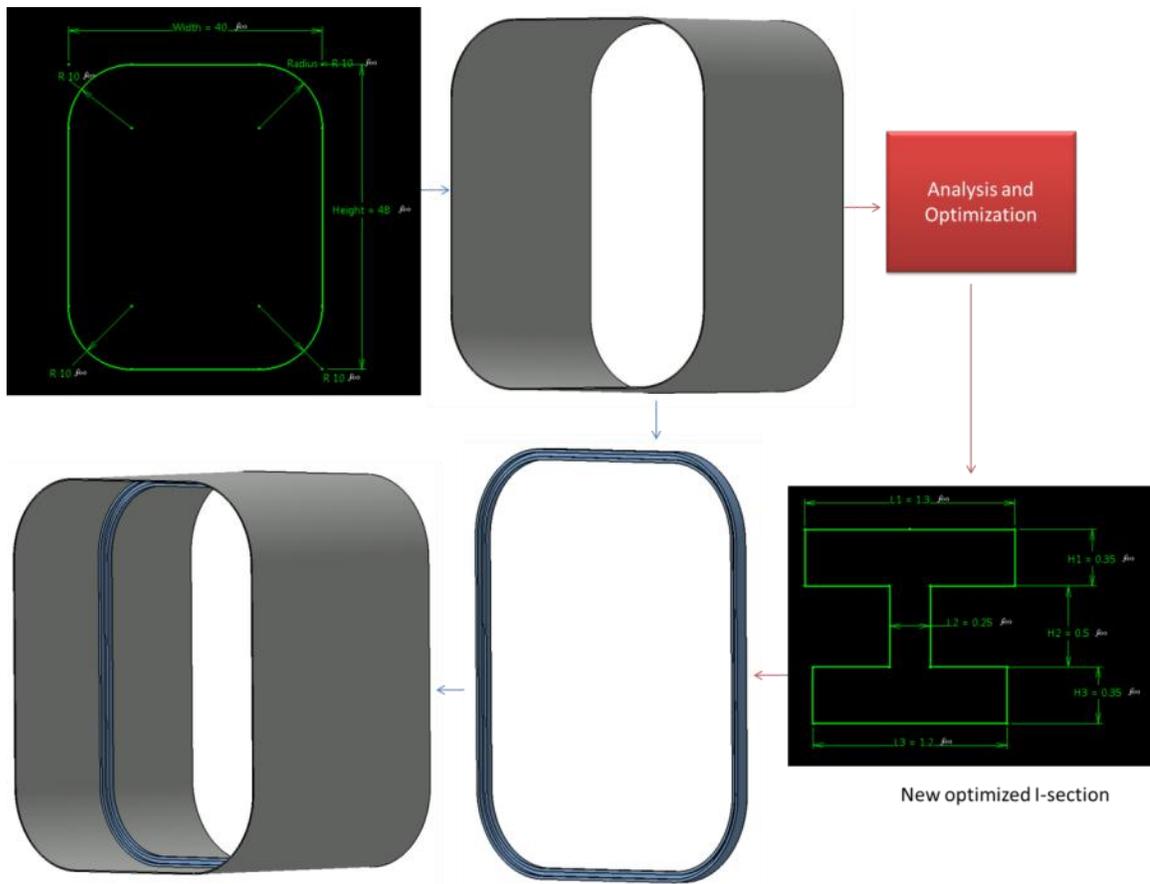


Figure 53. Fully-relational design process for fuselage - former

It is important to determine the type and magnitude of change that would warrant a rerun of analysis and this is something disciplinary experts must pre-specify. This will be largely dependent on computational capability to perform the analysis and the requirement of that level of fidelity. Implementation of automatically triggered analysis is only needed when the design has matured to a certain point. The logic for this process is described in Figure 54.

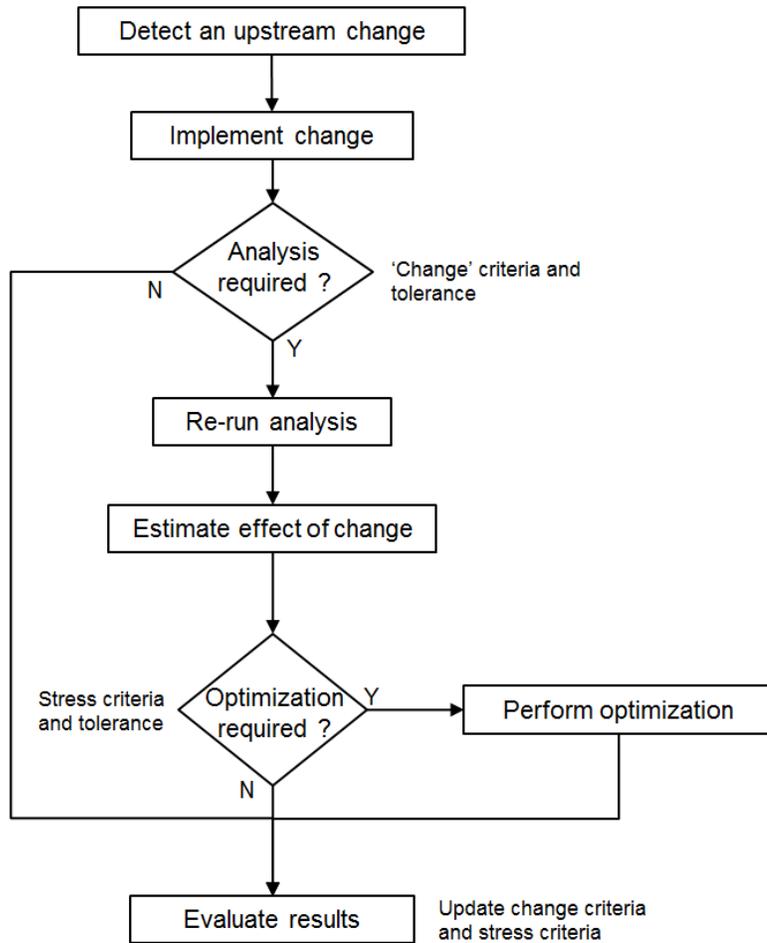


Figure 54. Full-relational design logic

The optimization, as with the analysis, is based on a tolerance and criteria. For example, if the analysis performed results in stresses that are within tolerance with

respect to the previous values, then a rerun of the optimization may not be required. This again must be determined by disciplinary experts and program a pre-specified criteria.

The following is an example explaining the concept of FRD with relevance to a helicopter drive system.

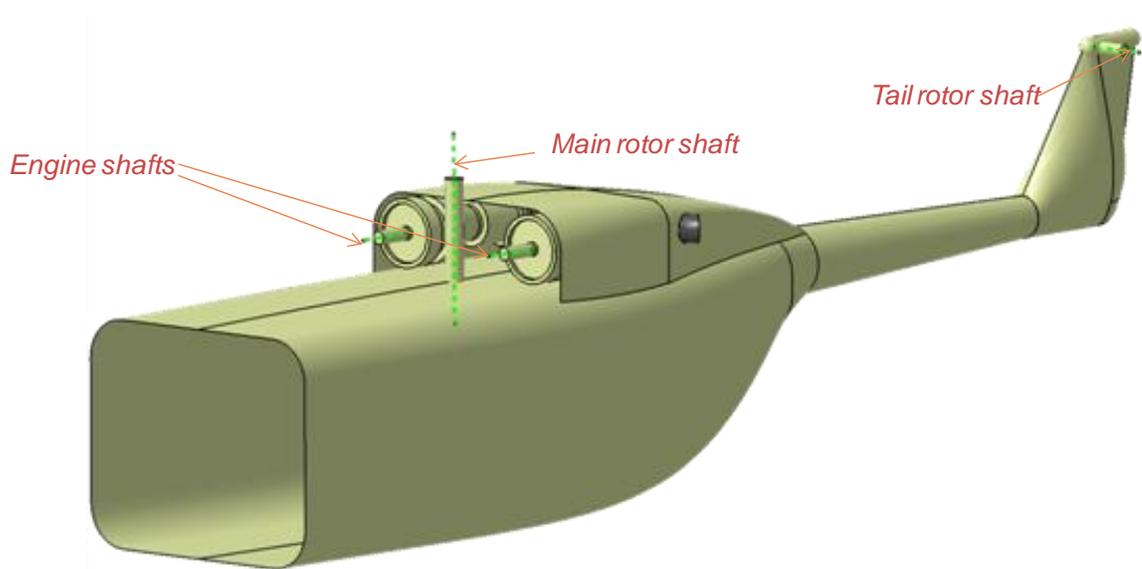


Figure 55. Single main rotor helicopter configuration

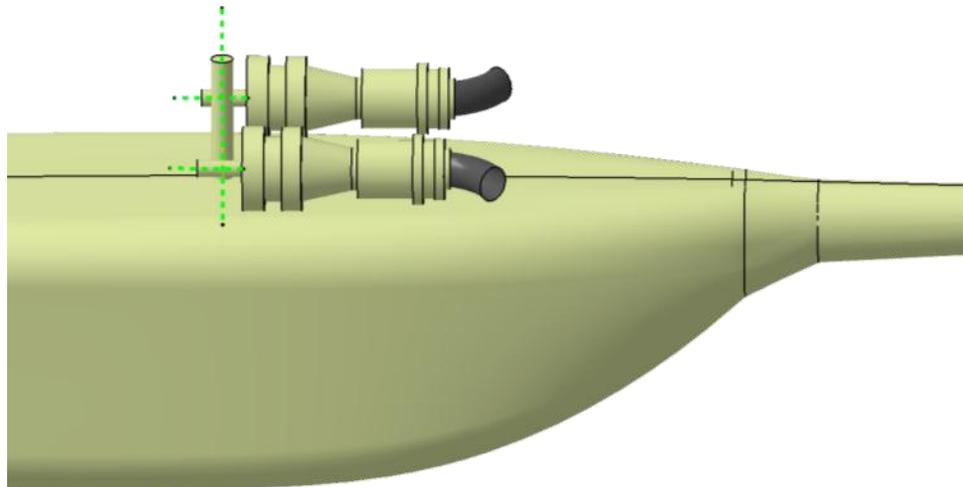


Figure 56. Initial helicopter fuselage and engine housing configuration

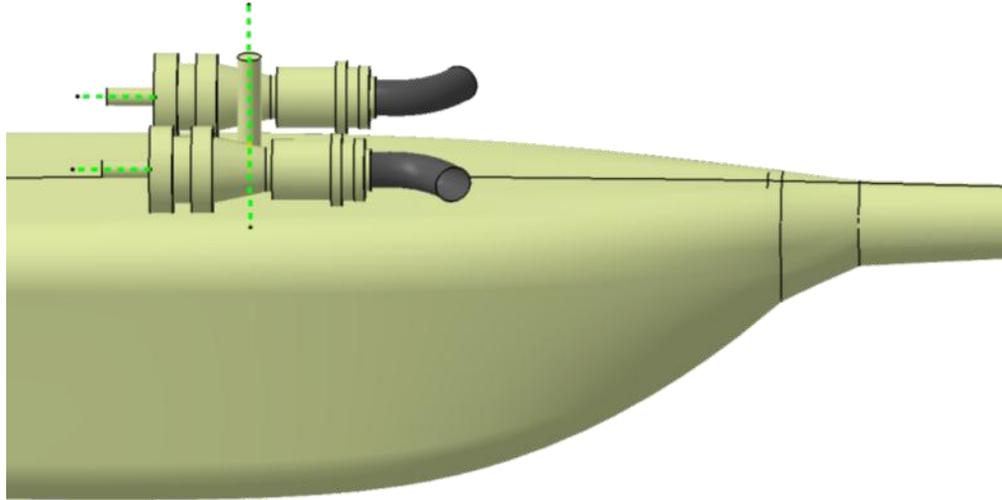


Figure 57. Parametrically offset product

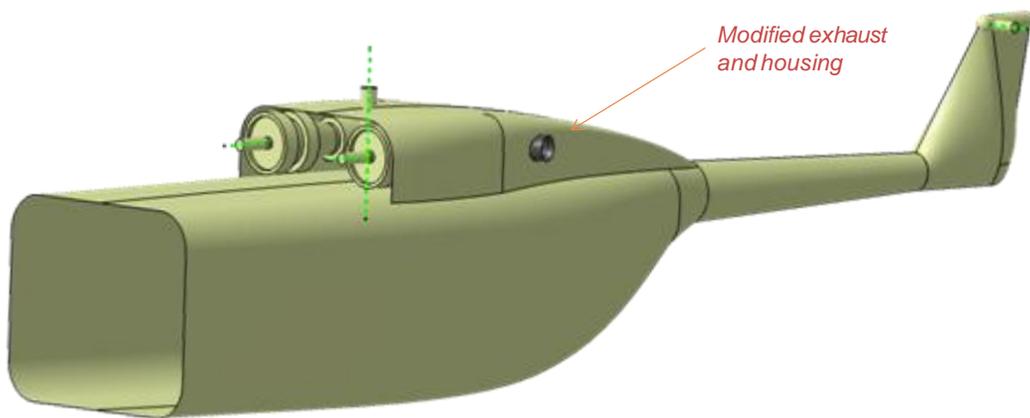


Figure 58. Modified fuselage and engine housing to comply with shift in rotor axis and engine location

In the above example, the rotor is shifted backwards and the engine moved forward; this sort of relocation is normal for longitudinal *CG* limit consideration. The shafts are aligned parametrically using datum planes. When this change takes place, the housing can be seen to automatically adapt to accommodate the new space and the exhaust pipe, extended to comply with the change. These specifications are not just used

to modify geometry but the framework uses the pertinent information to redesign the drive system based on these changes. This is done by enabling measurement based constraints and bevel gear angular requirements. This can also be used extensively in shaft design, and in detail stages to identify coupling requirements, and allocate space for pumps and accessories. Using geometrical constraints to alter part arrangement is described in Section 5.3 (page 154).

#### ***4.1.1 Requirements for Fully-Relational Design Implementation***

In order to implement fully-relational design, a logical system of information transfer and coherent method for design automation must be developed. Some of the important requirements of this methodology are design parameterization, information transfer hierarchy, fast and accurate optimization method, explicit geometric relations and decision processes.

The drive system is selected as a test bed for this methodology and the following sections describe the elements required for the implementation of FRD for it. First a sizing and analysis method for drive systems must be investigated and then a method to optimize it. Optimization is of particular importance; it must be fast enough for automation purposes and have high enough fidelity for preliminary design decisions. Additionally, the required level of fidelity must be investigated and FEA based methods tested for its applicability in FRD.

#### ***4.2 Gear Train Sizing***

Gear design needs to address failure modes so that the risks can be mitigated through design selection. Gear failure occurs in various modes: care taken during the design process can prevent such failures and a sound gear system can be achieved. Figure 59, below, delineates the different modes of failure. Design of gears must include a safety margin with the relevant loads and failure modes addressed.

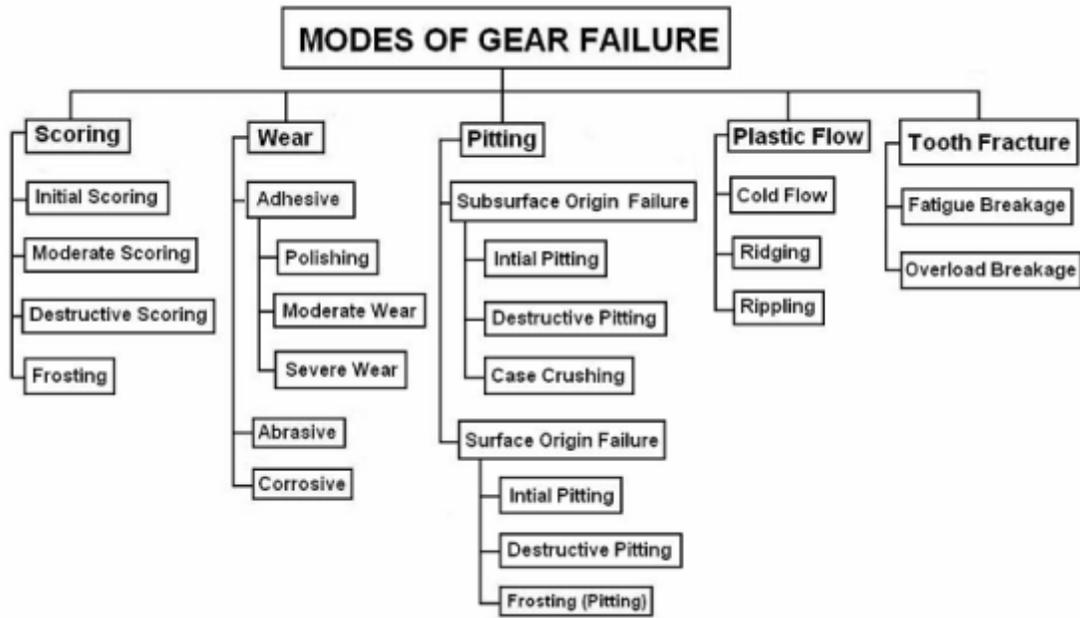


Figure 59. Modes of gear failure [114]

Aerospace gearing is primarily concerned with bending stress (Lewis stress), surface compression stress (Hertzian contact stress) and scuffing (scoring). Bending stress and its effects have the most serious consequences on gears, whereas pitting and scoring have durability implications [48].

In 1963 R.J. Willis published a paper showing how to pick gear ratios and gear arrangements for the lightest weight [64]. This formula was based on the solid rotor volume equation (Equation 3 and Equation 4). The weight equation using this formula is given as:

$$\sum (Fd^2 \times \text{weight constant}) = \text{weight, lb}$$

Equation 13. Weight using solid rotor volume

The weight constant for aerospace application is about 0.25 to 0.3. This assumes a limited life design and high stress levels.

The total weight of a planetary gear system - the sum of sun gear weight, number of planets multiplied with planet gear weight, and ring gear weight is given as:

$$\sum Fd^2 = Fd_s^2 + bFd_p^2 + Fd_r^2$$

Equation 14. Volume of planetary gear system

These formulae for estimating ratios become impractical for complex drive trains. A better weight minimization technique is required that uses accurate weight calculations for all types of gears. Using CAD models to update weight formulae helps improve the accuracy around weight.

Gear train (reduction ratio) optimization has been largely based on using series of graphs that aren't very conducive to computer programming. There is, hence, a requirement for a robust optimization technique that is more automation friendly. The gear train design section of this methodology will focus on using more computer modeling friendly algorithms for gear train weight minimization.

#### ***4.2.1 Bending Stress***

Bending stress is a concentrated stress at the base of the tooth. A gear tooth, according to Lewis' analysis, is considered a short cantilever beam with a point force acting on it at its free end. The highest point of stress concentration will occur at the base of the beam, or for gears, at the root fillet (Figure 60). When bending takes place, the base on loaded side of the tooth experiences a tensile stress whereas the other side experiences a compressive stress. Lewis' bending analysis corresponds to the tensile stress which is more severe in nature that can cause tooth fracture and breakage. The ability of a particular gear to resist this stress is called bending strength and is a function of the hardness and residual stress near the surface of the root fillet and at the core [48].

To determine failure, allowable bending strength/stress is de-rated by factors such as dynamic loading, overloading and reliability factors. This value is then compared to the calculated bending stress to rate the gear [47].

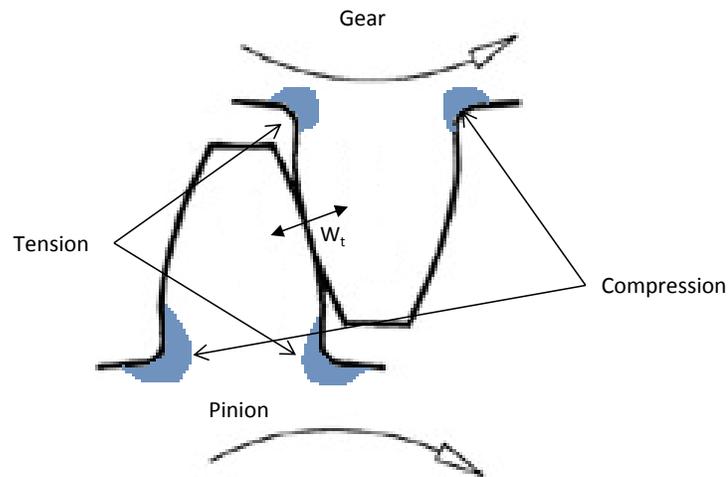


Figure 60. Illustration of bending stress

#### 4.2.2 Surface Contact Stress

Surface contact stress or compressive stress occurs due to the pressure generated in the contact region between the mating teeth. Some of the first works in the area of contact pressure between two deformable bodies was done by Archard [115] in 1953. This work led to much later refinements and improvements to the formulation of the contact problem. This theory expanded on the original work done by Heinrich Hertz – Hertz studied contact pressure between two deformable cylinders.

Contact stress in gears is come to be known as Hertzian stress and causes pitting which weakens the tooth surface by increasing local stress concentrations [47, 64, 104] (Figure 61). Gear teeth undergo compression and tension as the tooth rolls through the mesh with the mating tooth [48]. While the bending stress is dependent on the geometry

and shape of the gear tooth, contact stress is a function of the curvature and the surface and the material hardness and elasticity [116].

Over the life of the gear, repetitive cycles of loading progressively pits the surface until it eventually leads to fatigue failure. After initial pitting has set in, without corrective action to suppress pitting and sustained loading, destructive pitting sets in. Pitting spreads all over the tooth length, increases pressure on the unpitted surface, causing tooth failure. Allowable compressible strength measures the tooth surface's resistance to pitting. To increase compressive strength, aerospace gears are usually strengthened through carburizing, nitriding and case hardening. To determine failure, allowable compressive strength is de-rated by factors such as surface condition, hardness and dynamic factors, and this value is compared to the estimated contact stress.

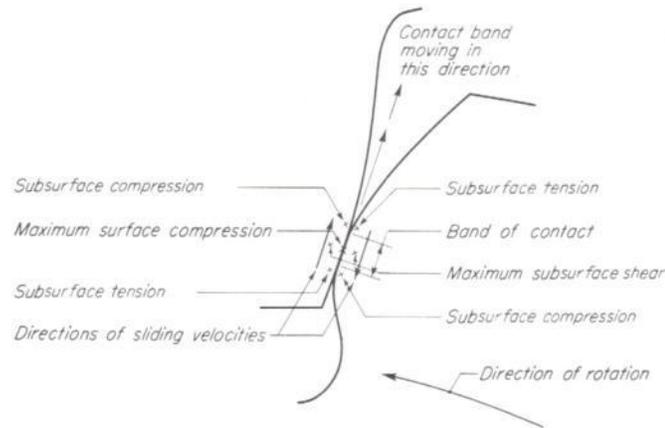


Figure 61. Stresses in region of tooth contact [64]

### 4.2.3 Scuffing Hazard

Scuffing or scoring is a lubrication failure in the contact region that occurs as a result of metal to metal contact. Scuffing is classified into 'initial', 'moderate' and

'destructive', and each stage is shown in Figure 62, Figure 63 and Figure 64, respectively.

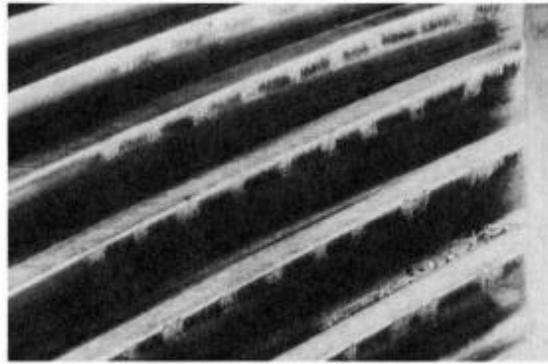


Figure 62. Initial scuffing [114]

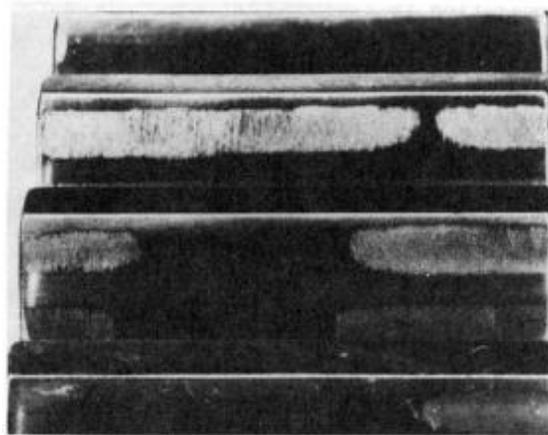


Figure 63. Moderate scuffing [114]

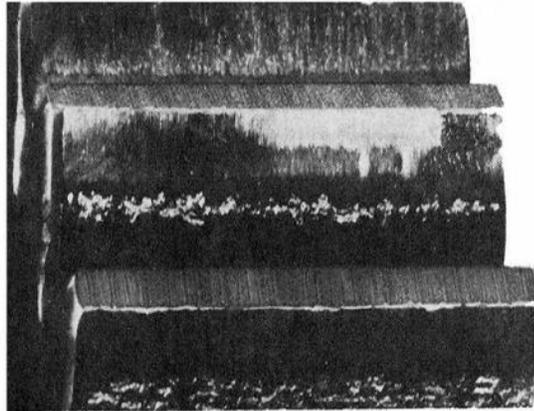


Figure 64. Destructive scuffing [114]

Initial scuffing occurs at the high spots left by previous machining. Lubrication failure at these spots leads to initial scuffing or scoring as shown in Figure 62. Once these high spots are removed, the stress decreases as the load is distributed over a large area. Scuffing is generally stopped if the load, speed and temperature of oil remain unchanged or reduced. Initial scuffing does not have progressive effect and can be completely avoided using the right corrective action. Moderate scoring occurs if load, speed or oil temperature increases after initial scoring has occurred. The scoring spreads to a larger surface area as shown in Figure 63. Destructive scoring occurs after additional loading, speed or oil temperature increases. This scoring is predominant over the pitch line region since elasto-hydrodynamic lubrication is the least in that region, shown in Figure 64.

#### ***4.2.4 Gear Rating***

As transmitted power increases, the bending stress increases linearly while compressive strength increase as the square root of transmitted power, shown in Figure 65. For the same gear geometry and design, compressive stress will be the higher stress in regions of lower transmitted power while bending stress often dominates the higher power regions. Gear rating and analysis is discussed in detail in Appendix A.1, page 228.

## Gear Tooth Stress vs Power

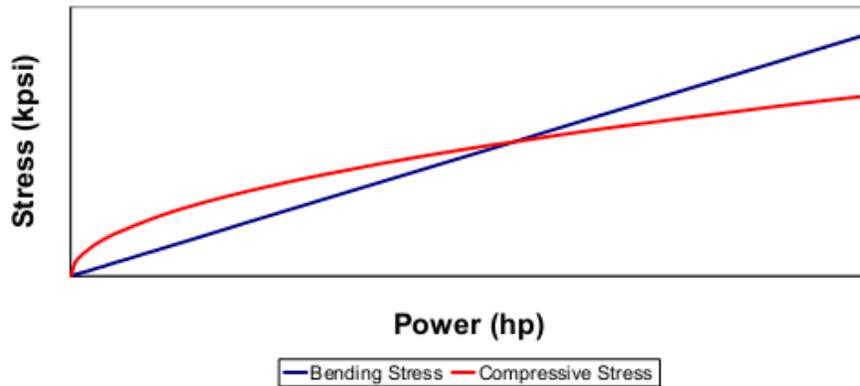


Figure 65. Bending and compressive stress vs. power [47]

Sizing codes are written in MATLAB to perform the bending and contact stress rating, and evaluate the scuffing hazard (Section A.1.c). The rating specifications obtained from AGMA standards [48-52, 63] are produced in Appendix A.1. Sizing for different gears is created as separate MATLAB functions. The three major sizing functions are:

1. Spur and helical gear sizing function - Appendix A.2.a.
2. Bevel and spiral gear sizing function - Appendix A.2.b.
3. Planetary gear sizing function - Appendix A.2.c.

The planetary sizing function is similar to the spur and helical sizing function, but is designed to evaluate reverse bending on planet gears and includes a section to compute sun, planet and ring gear teeth combinations for a given gear ratio.

Figure 66 shows the AGMA bending geometry factor  $J$  calculated for a spur gear mesh. For scuffing hazard analysis, flash temperature is calculated. Figure 67 shows the flash temperature as a function of the distance along the line of action. The code used to perform the scuffing analysis can be found in Appendix A.2.e.

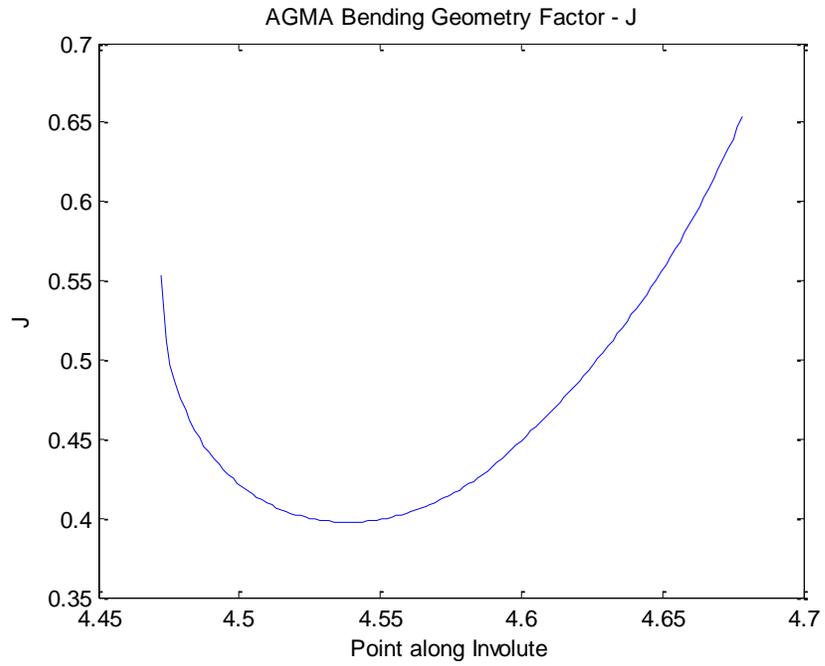


Figure 66. AGMA bending geometry factor J

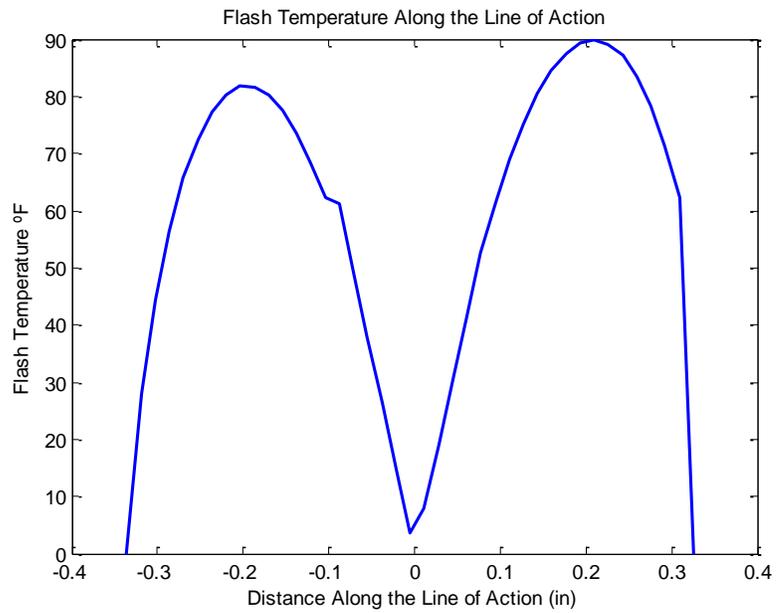


Figure 67. Flash temperature along line of action

A hunting ratio algorithm is used to obtain the combinations of teeth in a gear set, given the gear ratio. The code for the algorithm can be found in Appendix A.2.d. Hunting ratio precludes any particular combination of mating teeth to come into contact more or less frequently than other teeth. With a hunting ratio, any tooth on one member will, in time, contact all the teeth on the mating part. This tends to equalize wear and improve spacing accuracy. For example, if the pinion and gear have 21 and 76 teeth respectively, this ratio will hunt, since the factors of 21 are 7 and 3, and the factors of 76 are 2, 2 and 19 eliminating any common factors. A general rule of hunting ratio is that, tooth numbers should be selected such that there is no common factor between mating teeth. This applies to the number of teeth selected for the cutting tool that has a gear-like meshing action [64, 65].

In the hunting ratio, a tooth on one part has to get worn and wear all teeth on the other part, into a 'fit', with itself. Thus a 'full-fit' cannot occur until all pinion teeth are worn alike, all gear teeth are worn alike and the pinion-worn profile is a very close surface fit to the gear-worn profile [64]. Using a hunting ratio algorithm has the added advantage of limiting the total number of teeth combinations to be evaluated.

Gear materials used in this study and their properties are shown in Table 1.

Table 1. Gear materials

Description	Units	AISI 9310	VASCO X2M	PYROWEAR 53
AMS Spec		6265/6260		6308
Heat Treatment		C-H	C-H	C-H
Main Drive Application		Y	Y	Y
Accessory Application		Y		
High Temp. Application			Y	Y
Case Hardness	HRC	61	62	62
Core Hardness	HRC	37	40	40
Brinell Hardness	BH	632	647	647
Allowable Contact Stress	psi	244,897	250,145	250,145
Allowable Bending Stress	psi	52,102	51,990	51,990
Poisson's Ratio		0.292	0.3	0.292
Modulus of Elasticity		2.90E+07	2.96E+07	3.00E+07
Density	lb/in <sup>3</sup>	0.283	0.28	0.282

### 4.3 Optimization

Optimization entails the requirement to either minimize or maximize an objective function with or without constraints. The constraints and objectives are functions of design variables and a constrained optimization problem is generally denoted as follows:

Minimize:  $F(x)$ , *objective function*

Subject to:  $x \in X = \{x \in \mathbb{R}^n \mid g_i(x) \leq 0, h_j(x) = 0, \quad i = 1, \dots, m; \quad j = 1, \dots, k\}$ ,

*inequality and equality constraints*

This schematic gives the basic formulation of an optimization problem as developed by Schmit in 1960 for nonlinear problems. The development of computer

codes and programs has created a family of optimization methods known as numerical optimization. Numerical optimization techniques offer a logical approach to design automation, and many algorithms have been proposed in recent years. Some of these techniques, such as linear, quadratic, dynamic, and geometric programming algorithms, have been developed to deal with specific classes of optimization problems. A more general category of algorithms referred to as nonlinear programming has evolved for the solution of general optimization problems. Methods of numerical optimization are also collectively referred to as mathematical optimization techniques [117].

Optimization of the drive system requires minimizing weight and cost while improving efficiency, which is a multi-objective optimization problem. The design variables for this problem are the gear setup and gear design parameters. The constraints are of structural types that need to be imposed such that the gears meet the operation requirements and other system level constraints. An optimum solution is one that is ideally non-dominated with regards to the objectives. For a multi-objective problem, the non-dominated solution is not unique and there is more than one non-dominated solution. The family of non-dominated solutions is called a Pareto frontier, and the optimum is hence, a tradeoff based solution [118, 119].

Numerical optimization methods are broadly classified as:

1. Gradient based optimization
2. Heuristics based optimization

Each type of optimization method and the different types of techniques and algorithms that fall under them have their own advantages and disadvantages. Gradient based methods work very well in continuous design space and handle constraints well as long as the problem does not get highly nonlinear. Gradient based methods, when

executed well, can converge quickly and perform well without much ‘tuning’ and alteration and generally don’t require too many function calls.

Heuristic methods are more random in nature that can be easily formulated to run most problems and very good for handling discrete design space. Heuristic algorithms become more problem sensitive when it comes to fine-tuning. Although heuristic algorithms are generally much slower requiring more function calls, they adapt well to highly nonlinear problems and have the capability of directly imposing constraints [117, 120].

The design variables and their properties, identified for gear design are given in Table 2. The constraints for the design are discussed in Section 4.2 (Page 104).

Table 2. Optimization variables

Parameter	Symbol	Type	Range	Optimizer	Rules
Diametral Pitch	$P_d$	$P_1$	1, 1.25, 1.5, 1.75, 2, 2.5, 3, 4, 5, 6, 8, 10, 12, 14, 16, 18	GA	High $P_d$ for initial stages and low $P_d$ for final stages
Helical Angle	$\psi$	$P_1$	0, 10,20,30	GA	0 for Bevel
Gear Ratio	$m_g$	$P_1$	continuous	GA	Bevel: 2 - 5, Spur: 2 - 10, Planetary: 2- 10
Gear Material	$M$	$P_1$	AISI 9310, VASCO X2M, PYROWEAR 53	GA	
Number of Teeth	$N$	$P_2$	(range varies with $\psi$ )	SO	Hunting Ratio
Face Width	$FW$	$P_2$	continuous	SO	

For gear train design, the surplus of discrete variables in the design space and highly nonlinear constraints make heuristic methods more desirable. For these reasons, a Genetic Algorithm (GA) based method is proposed to perform the optimization.

### ***4.3.1 Genetic Algorithm***

Genetic algorithm is based on the theory of evolution. Due to its random nature, it cannot guarantee an optimum but has a better chance of finding a global optimum in the presence of many local optimums as would be expected in transmission optimization. The GA can be used to build a population of gear train designs and then select the ‘most fit’ designs, based on a predetermined criteria or an Overall Evaluation Criterion (OEC). The overall GA structure is shown in Figure 69. Each design combination is considered an individual in the population. The fittest parents are then ‘mated’ so that their combination produces a new population of designs that are potentially better. This continues until there is no improvement in several successive generations, the number of generations is specified as the convergence criteria. To increase the search capability, several offspring are also allowed to mutate or differ from either parent. Each design is represented by a binary string. Due to the nature of the binary string, the genetic algorithm works only on discrete values. Hence, the design space has to be discretized at the beginning; continuous variables such as gear ratios should be discretized based on tolerance limits. These tolerance limits specify the number of divisions that a continuous variable will take for the given upper and lower bound of that variable. All variables are then converted into binary strings and those strings are then assembled end to end to form one long binary string (or chromosome) that represents that design. Initially, a population of individuals is randomly generated. The overall string length is determined by converting all variables and their ranges into binary strings and piecing these end to end as specified above. The length of an individual,  $n$  is determined such that  $2^n$  is equal to number of combinations. A population of initial design combinations, typically 50-100 or so, are then randomly generated by randomly selecting 0 or 1 to fill each location in the string. The strings are then converted back into decimal numbers and these strings and decimal equivalents then move onto the selection stage [62, 117, 120].

There are many ways to determine which designs get selected into the parent population to mate and create the subsequent population. Two methods are used alternatively in this work. The first is the roulette wheel method and the second is the tournament selection method.

The Roulette wheel selection randomly selects individuals ( $P^k$ ) to be placed in the population over which genetic operations will occur. The selection is performed with a probability proportional to its fitness value [121], as shown in Equation 15.

$$S(P^k) = \bigcup_{i=1}^{N_{pop}} s(P^k)$$

where  $s(P^k) = p_j^k$  with probability

$$\Pr[s(P^k) = p_j^k] = \frac{f_j^k}{\sum_{n=1}^{N_{pop}} f_n^k}$$

Equation 15. Roulette wheel selection [121]

In the tournament selection method, two individuals are randomly selected and their translated decimal equivalents are used to perform crossover. In tournament selection, one design is compared with another and the fitter design progresses in the selection process of a tournament. Multiple tournaments are created where all individuals in the population compete. The successful individuals form the parent population [62, 122].

Once the parent population is chosen they are mated to create an offspring child generation. Crossover takes place with a certain pre-specified probability. If chosen and crossover permitted, that set will be ‘mated’, undergoing a two-point crossover [123]. In this crossover method, the size and location of the bit and its location to be crossed over is randomly selected. Thus, there is a chance that if the two numbers are sequential then

there will be uniform cross over (all bits swapped). Figure 68, below, illustrates a two point crossover between two points A and B. In this example,  $str_{size}$  is the total number of bits in the string.

$$k = Rand \in [0,1] \quad [A,B] = round(1 + (str_{size} - 1) * k)$$

$If A < B \rightarrow$	A	B	$\rightarrow$	A	B	$\rightarrow$	$If A > B \rightarrow$	B	A	$\rightarrow$	A	B	$\rightarrow$	A	B												
	0	1	0	1	0	1		0	1	1	0	1	1		0	1	0	1	0	1		1	0	1	1	1	0
	1	0	1	0	1	0		1	0	1	0	1	0		1	0	1	0	1	0		0	1	0	0	0	1

Figure 68. Two point crossover example [62]

Once the new generation is created from the parents, the individual members will be tested to see if they will undergo a random mutation based on a mutation probability. If mutation is permitted, a single bit in the string is randomly chosen. Mutation swaps this bit value, i.e. a 0 becomes 1 and vice versa.

Multiple-elitist strategy is an aggressive technique that permits a certain predetermined percentage of the fittest population to automatically progress to the next generation. This ensures that the best designs always get promoted making the optimization more aggressive and converge faster. The main disadvantage of this strategy is that it tends to be exploitive rather than explorative. When there is a greater chance for lower ranking designs to be used in the parent population, the algorithm naturally bounces around the design space more, but converges slower. This type of algorithm will quickly converge to an optimum, which if done too quickly without adequate exploration, can most times be a local optimum rather than global optimum, in a multimodal solution space. This method has an added advantage of speed, in that if an individual is randomly chosen more than once, the algorithm can recognize this, precluding the processing time of evaluation of fitness again i.e. run the analysis program

for it. If this occurs, then the analysis is performed for only the new individuals whose fitness have to be evaluated [119, 124, 125].

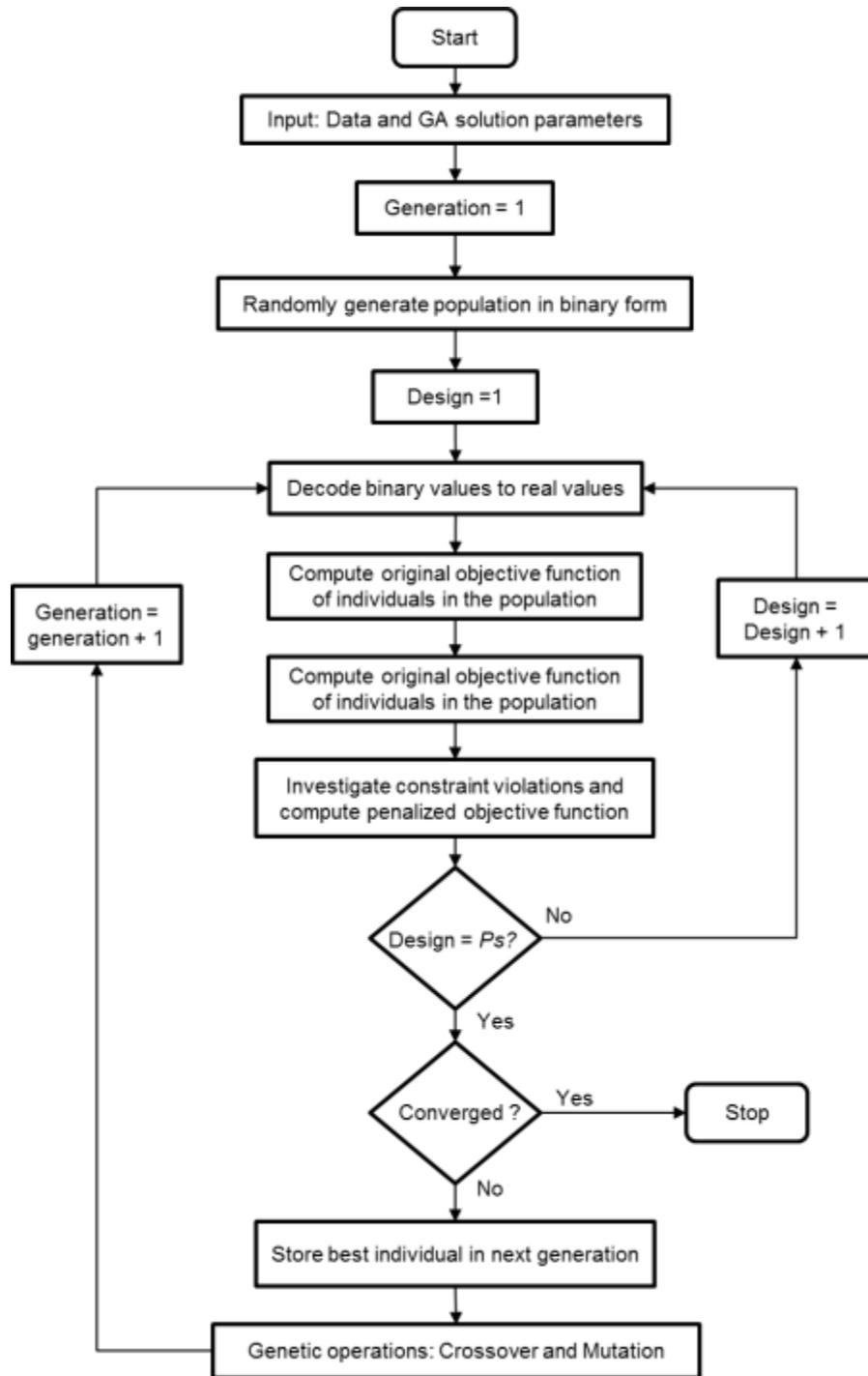


Figure 69. Genetic Algorithm Structure [123]

This is repeated until there is no change in the objective function of the best design in a child population for a predetermined number of generations. Once this occurs the optimization is considered to have converged.

Constraint handling in GAs is very problem specific. GAs are best suited for unconstrained optimization problems [123]. Gear train design, like other structural optimization problems, is a constrained optimization problem. Many methods have been studied and presented in literature as discussed in Section 2.4 (Page 68).

In this work, five different constraint handling penalty techniques are experimented. These are:

1. Rejection method ( or Death penalty) [126]
2. Static – linear [127, 128]
3. Static – nonlinear [123, 127, 128]
4. Dynamic – linear [129, 130]
5. Dynamic – nonlinear [129, 130]

The balance of exploration – exploitation is of particular importance in GA. ‘Exploration’ pertains to extent of search in the design space, and ‘exploitation’ pertains to quickness of convergence. A solution quick to converge may converge prematurely at a local optima, while a less exploitive will be computationally longer to converge at a solution. Quickness of convergence is of great importance for implementation within the FRD framework. The two answer the twin goals of maintaining diversity in the population and sustaining the convergence capacity of the algorithm, using adaptive probability rates enhances both these capabilities [79].

Exploration is required when designs start getting very similar; ‘similar’ is defined by having same or near same fitness values as result of similar design variable combinations. An aggressive GA with exploitive tendency will try to maintain a high

level of fitness through the entire population. This restricts exploration; however, aggressive cross-over and mutation will help in exploring new designs. So an effective adaptive GA is one that is able to select when to be aggressive and when to be more explorative. The formulation developed for cross-over and mutation in this study are given in Equation 16.

$$P_c = k_1 \frac{\bar{f}}{f_{\max}}$$

$$P_m = k_2 \frac{\bar{f}}{f_{\max}}$$

Equation 16. Adaptive cross-over and mutation rates

Where  $k_1$  and  $k_2$  are empirically estimated based on the population fitness standard deviation and ensure that the value of  $P_c$  and  $P_m$  are within their bounds. The values used are discussed in Section 5.2 (page 132)

Another method that has been discussed in literature to improve exploration capability is the introduction of a random member [131]. In this methodology, the random member is introduced and accepted regardless of its fitness but is used to replace the most frequently occurring member in the population, i.e. mode of the fitness.

A new concept that has not been discussed much in literature, introduced here is the use of parallel sub-populations. Sub-populations greatly benefit exploration without compromising on exploitation. The computing requirement per generation is proportionally higher but this greatly improves the probability of arriving at the global optimum. Sub-populations are only effective if the migration strategy is effective. The stochastic nature of GA requires a stochastic approach instead of a predefined predictable

strategy. The process and the implementation are discussed in detail in Section 5.2 (page 132). The code for this algorithm can be found in Appendix A.2.h.

#### **4.4 Flexibility**

The design process is subjected to many unknowns and fuzzy requirements. During early design phases, a large design space must be explored to get a set of feasible design solutions. The design process needs to be readily adaptive to changing conditions. Designing for change is related to design process flexibility.

When downstream changes occur, the performance of the design must not deteriorate and must be ‘flexible’ to change. Here the design is defined as a selected configuration. Change is defined as change in the requirements that were used to design the configurations. Flexibility here is concerned with the design lifecycle and not the product lifecycle.

In order to select a flexible design, a probabilistic modeling technique is proposed in this thesis. The probabilistic approach quantifies change and coupled with an optimization algorithm can be used to quantify the effect of change; this measures the degradation of functionality and performance. This technique also helps identify a concept that is likely to perform better in a changing environment and design space. The design space is a fuzzy environment defined by system requirements such as  $H_P$ , speed, fuselage space, etc. Applying distributions around the baseline values of these metrics will expose each concept’s capability. Being able to select a flexible design directly complements the modern SE approach. An example of the method used in this thesis is shown in Figure 70 in the form of a CDF plot for a minimization problem. The solid blue line (*design A*), in this example is shown to illustrate the performance of a design that may have had the lowest  $F(x)$  value for a point design space and environment, but

underperforms when the design space is opened up. The other design, shown in the form of a red dashed line (*design B*) indicates the flexible solution that underperformed, relatively, for the point design space but performs much better in the uncertain design space. At 25% probability, *design A* has a value of 4.5 while B has a value of 4. Similarly, at 75% probability, *design A* has a value of 6 while B has a value of 7. Figure 70 also shows the distribution function (PDF) for a system objective (top right corner).

In relation to rotorcraft drive system, ‘flexibility’ helps the design select between two configurations in a probabilistic fashion, not requiring a delay in decision. Design decision to select a split-torque vs. a planetary gear system for the main gearbox is one such decision that can be considered earlier if the performances of both configurations are tested in a varying requirements environment.

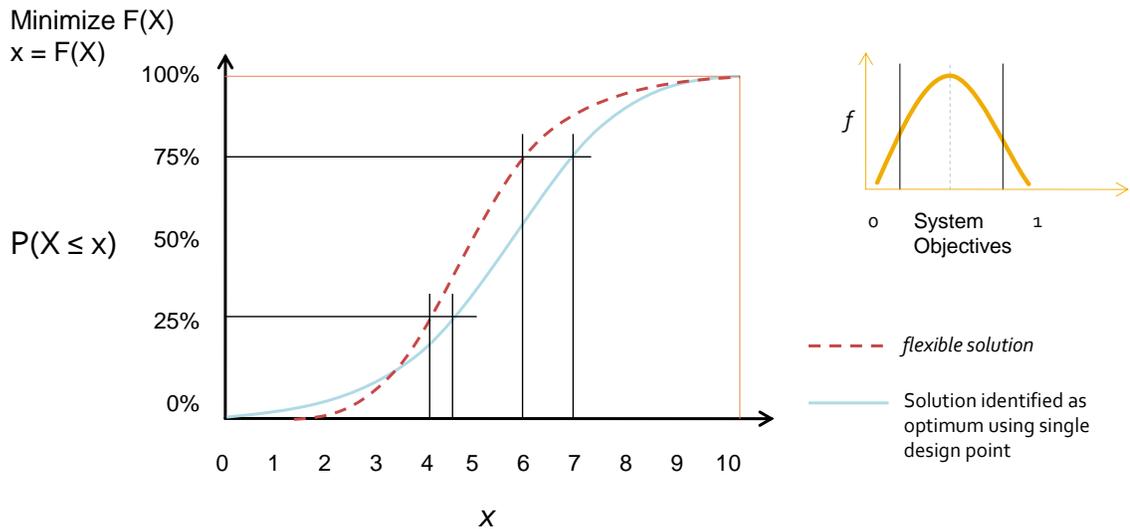


Figure 70. Quantification of flexibility

#### **4.5 *Topology Optimization***

Gear web can produce considerable weight savings in gears. Gear web structures have been empirically estimated through expensive bench tests. Finite element based Topology Optimization offers an inexpensive alternate method to study gear web and excess material removal.

Topology Optimization is a mathematical technique that produces an optimized shape and material distribution for a structure within a given design space. The optimization is performed subject to a given set of loads and boundary conditions such that the resulting geometry satisfies the prescribed conditions and performance. The domain is discretized using a finite element mesh and the solver calculates the material properties for each element. Using Topology Optimization, a structural designer can minimize weight and improve the structural properties of the structure by minimizing the compliance [132-134].

Topology Optimization may result in a design that cannot be manufactured or may have some undesirable features. This problem is overcome by enforcing manufacturing and feature constraints [135].

In this thesis, OptiStruct is used to perform the Topology Optimization. OptiStruct is a part of the HyperWorks (Altair Engineering Inc.) FEA suite. OptiStruct is a finite element optimization software that uses the RADIOSS solver to compute the required finite element responses. OptiStruct allows the user to model the structure, setup the problem, specify the objective function and constraints, and submit the job for processing [136].

The OptiStruct problem for the spur gear is shown in Figure 71. The design region is where excess material removal is desired and the regions that are required to be unaltered are the non-design regions.

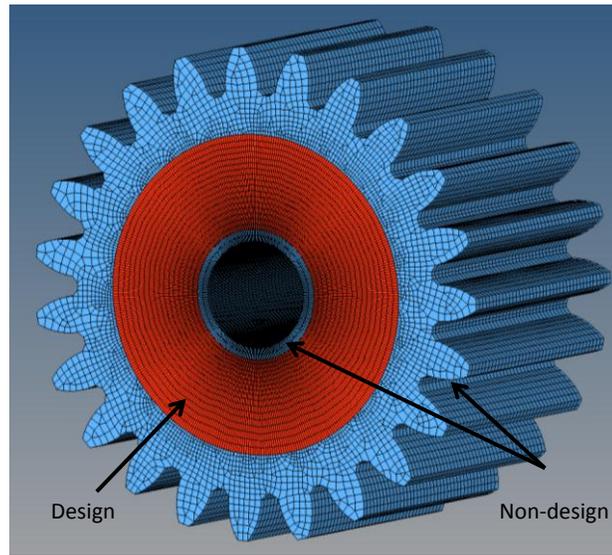


Figure 71. Topology optimization setup in HyperMesh

The optimization problem is defined as follows:

$$\text{Objective: } \min : F(x) = \text{Volume}_{\text{design}}$$

$$\text{Subject to: } x \in X = \left\{ x \in R^n \mid g_i(x) \leq 0, \quad i = 1, \dots, m \right\}$$

The constraints are stress limits and manufacturing constraints. Manufacturing constraints are imposed such that a symmetrical topology is obtained since the bending problem requires only one tooth to be loaded. Manufacturing constraints also prevent holes and certain small features from being formed. The implementation and problem setup are discussed in detail in Section 5.5.2 (page 197).

## CHAPTER 5

### IMPLEMENTATION AND RESULTS

#### 5.1 Design Framework

To study the overall methodology, a sound drive system design techniques must be developed. Based on literature survey and research done in the area of drive system and gear train design, the following design framework (Figure 72) was developed.

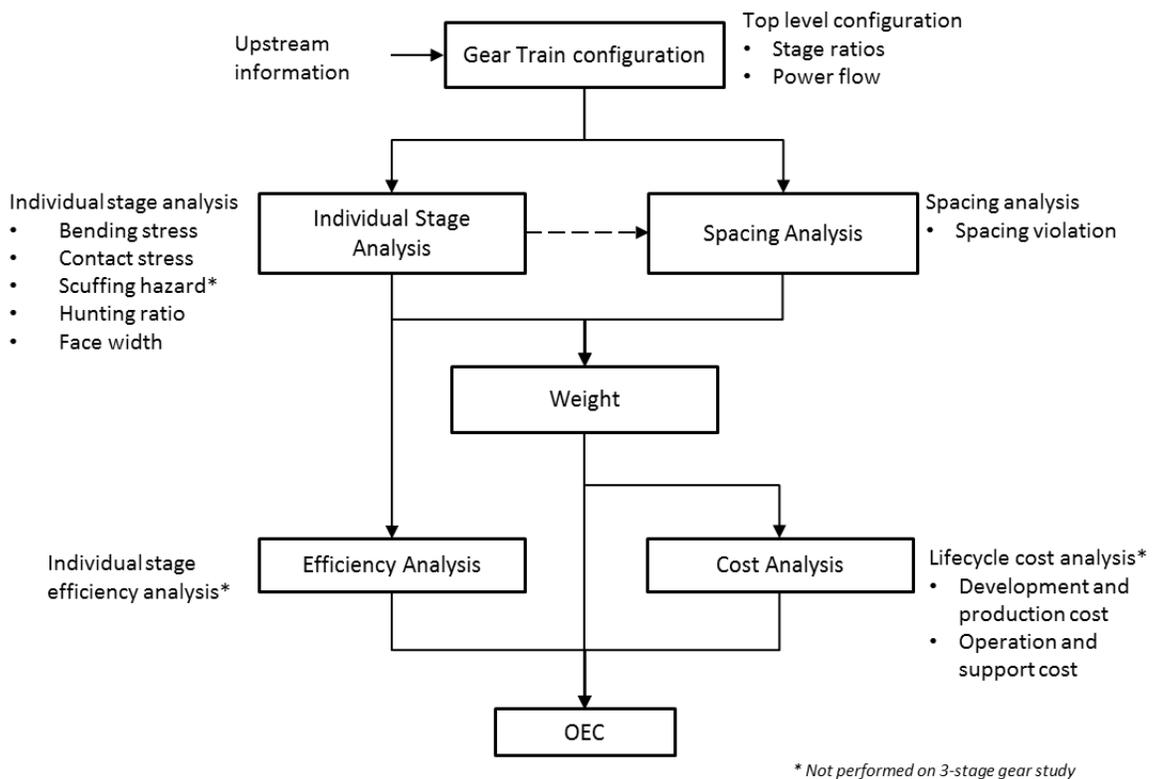


Figure 72. Drive system sizing and analysis framework

The process begins with the information required to perform the analysis, which are the system requirements. This information is then transferred on to perform the individual stage sizing and also the geometry spacing analysis. The methodology for gear

train sizing is discussed in Section 4.2 (page 104). The geometry and spacing analysis is discussed in Section 5.3 (page 154). The geometry and spacing analysis are implemented as a part of the FRD method only for the three-stage gear system. Efficiency and cost analysis are performed on the entire rotorcraft drive system and optimized using an OEC (Section 5.6). The code used for efficiency analysis can be found in Appendix A.2.e. The cost analysis and integration is discussed in Section 5.6.1 (page 217).

The FRD system is implemented in ModelCenter and is shown in Figure 73. ModelCenter is used as an integration tool to interface with the other tools. ModelCenter is a software environment designed for integrating originally unrelated software packages. It enables conducting complex design exploration tasks using a wide range of supported commercial analysis software or simple command line based executables. The design data is transferred from one program to another automatically, eliminating the need for manually converting output-input file formats of incompatible analysis tools. ModelCenter can be used as a passive environment, serving as the common medium for communication of programs.

The fundamental logic of the FRD system is shown in Figure 54. The vehicle sizing spreadsheet has the information that is required to size the drive system-  $Hp$ , input rpm and required output rpm. The housing CATIA geometry is a part that maintains parametric relation with the gear train geometry through the inbuilt database management system (ENOVIA) in CATIA, as was previously discussed in Section 4.1 (page 95).

ModelCenter is capable of automatically detecting change in the vehicle and geometry requirements; once change is detected, ModelCenter is setup to run the analysis decision logic. The analysis decision logic script determines whether the change is significant enough to perform the sizing and spacing analysis based on the predetermined criteria. If the change demands a rerun of analysis, then the sizing and spacing MATLAB function is executed. If the results from the analysis exceed the predetermined optimality

tolerance, then the optimization logic script automatically triggers the optimization MATLAB script. Once the optimization is performed, the gear train geometry is updated and a script is run to evaluate the results. The evaluation script is used to regenerate new criteria for analysis and optimization logic, and also returns the new results to the vehicle sizing spreadsheet.

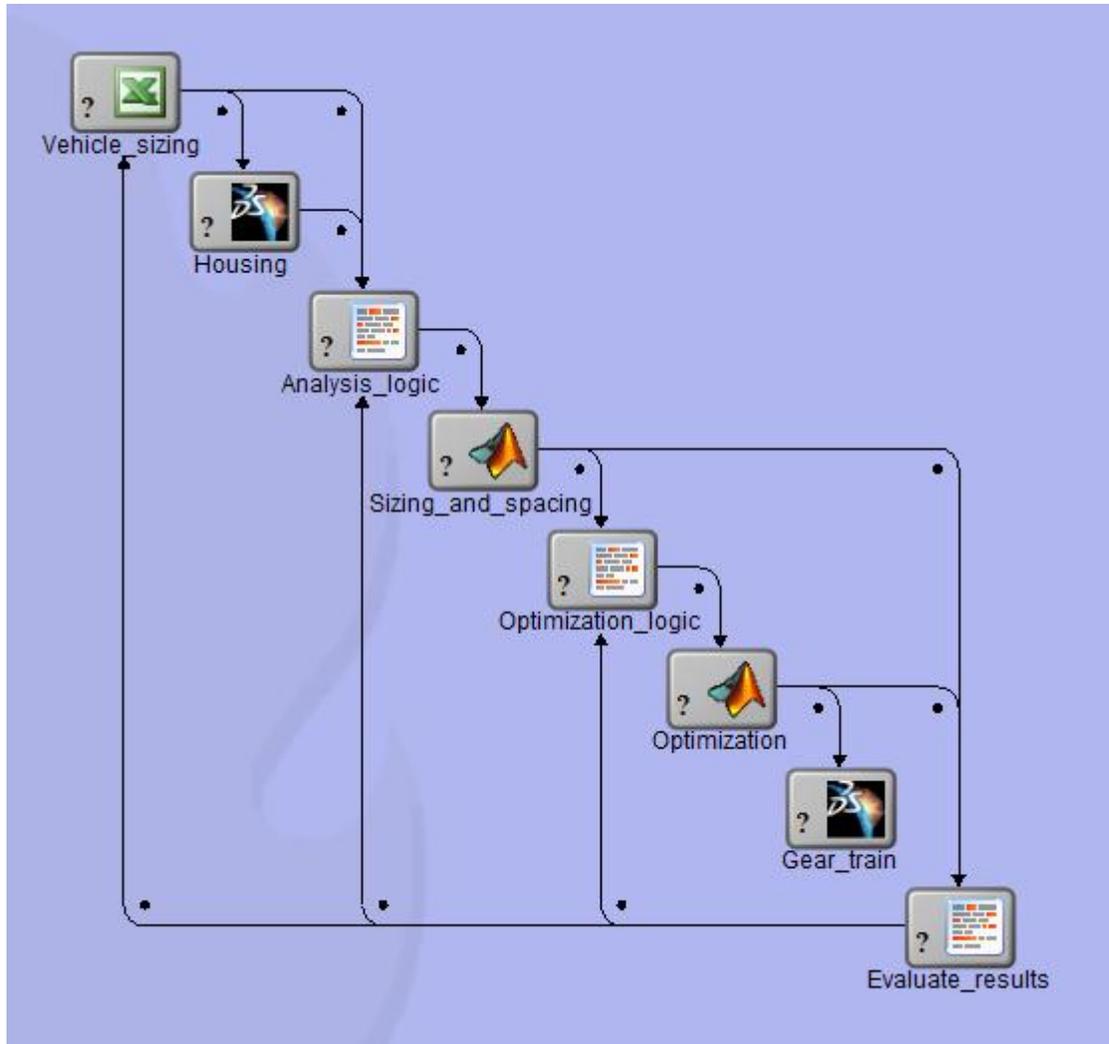


Figure 73. Fully-relational design implementation for three-stage gear system

### 5.1.1 Geometry integration

Geometry integration of the drive system follows the relational design methodology discussed in Section 4.1 (page 95). In order for relational design to be successfully implemented, all parts and products need to be parameterized. The geometry of a spur gear is shown in Figure 74. The gear geometry can be generated using mathematical calculations and completely parameterized. CATIA supports parametric design, through which gear geometry formulae and relations are created. Figure 75 shows the 2-D profile of a helical gear drawn in CATIA. This geometry is constructed by combining the involute curve and the trochoidal curve.

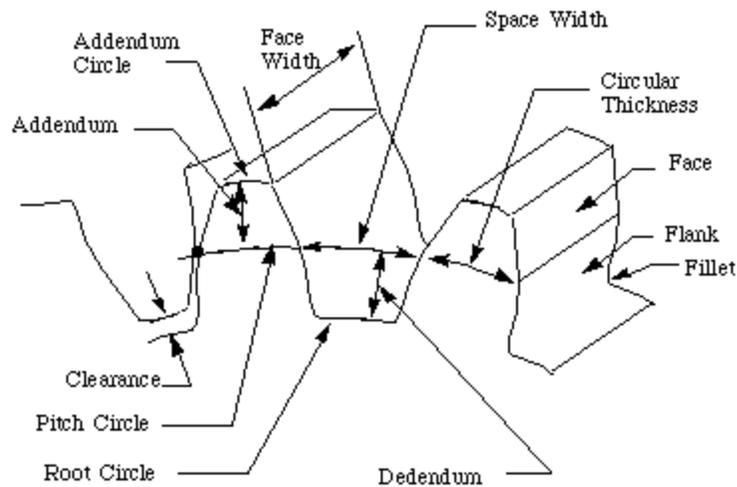


Figure 74. Spur gear geometry [65]

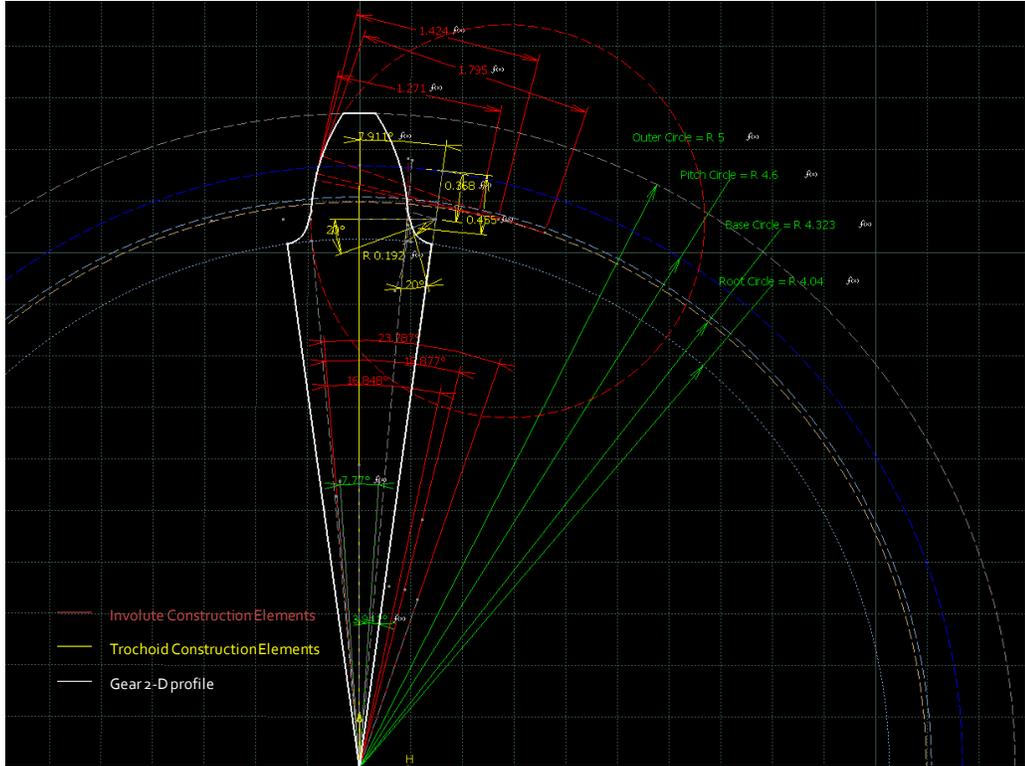


Figure 75. 2-D gear profile generated in CATIA

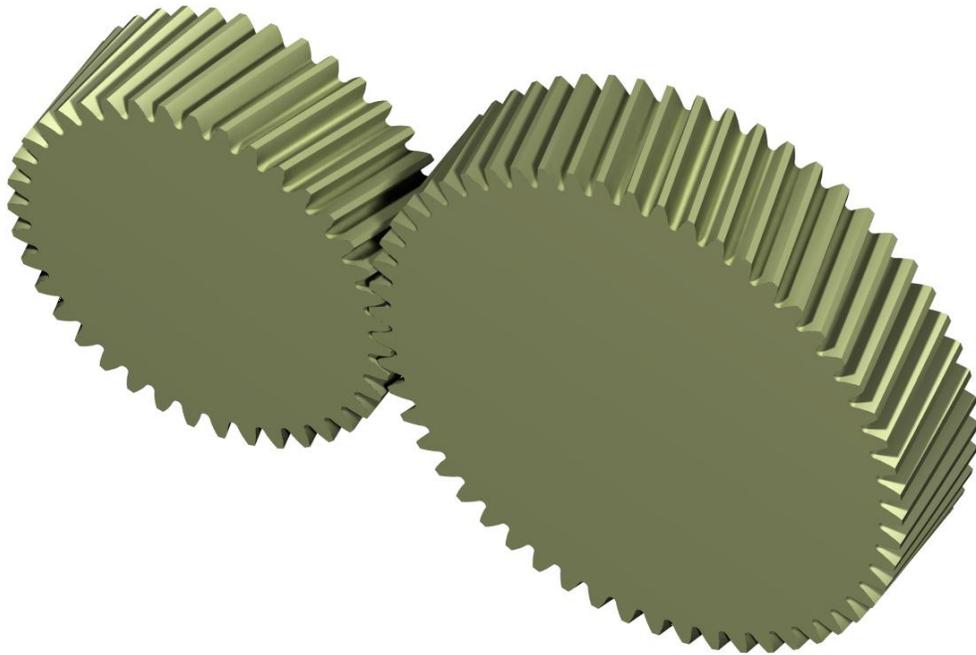


Figure 76. Helical gear pair generated in CATIA

Using the gear geometry shown in Figure 75, a pair of helical pinion and gear geometry is generated using relations and rules (Figure 76). The pitch circles of the pinion and the gear are drawn tangential to each other enabling ideal contact and arrangement of the pair. The geometry is completely modular and compatible with all gear parameters.

## 5.2 *Genetic Algorithm and Optimization*

To study the GA for the drive system, a program was written to optimize a three-stage and a four-stage reduction gear system for an input power of 500 HP at 35000 rpm. The required gear reduction of the system is 25:1 (output = 1400 rpm). The optimization of a gear train is a hierarchical problem. The gear teeth and face width are variables that depend on other variables such as diametral pitch, stage ratios, etc.; optimizing such a system poses unique difficulties.

The approach presented here treats the selection of number of teeth and face width as a sub-problem. For the gear train optimization problem discussed in the previous paragraph, the following variable handling method is used (Figure 77). The diametral pitch ( $P_d$ ) and gear ratio ( $m_g$ ) are used to generate the required pinion and gear teeth pair combination set based on the hunting ratio algorithm (discussed in Section 4.2, page 104). For a given pinion teeth, AGMA recommends a maximum limit on face width, not exceeding the pinion diameter [49]. The face width is discretized between the bounds of  $8/P_d$  to  $12/P_d$  in 10 steps. The analysis is performed on each discrete combination until a feasible design that meets all stress requirements is arrived at. If for all combinations, no feasible solution results, then the function returns the minimum violation information. This violation is used by the GA to penalize the combination of  $P_d$  and  $m_g$  that resulted in that design. Violation for the different stress constraints for the different stages is handled individually. The studies performed and the techniques used for handling the constraints within the GA are discussed in the following section.

The complication with this approach is the requirement to perform multiple analyses in a full factorial fashion, but nevertheless provides an efficient way of handling hierarchical design variables [137]. The requirement to include hunting ratio algorithm and dynamically calculate allowable stresses makes this approach far more efficient than those studied in literature. A combination of using the full-factorial sub-function with the GA requires that the constraints are handled efficiently. It is possible to implement a smaller optimization algorithm within the sub-function in order to avoid a full factorial search. A branch and bound method may help eliminate the requirement to run all cases. It is beyond the scope of the problem here to evaluate its efficacy.

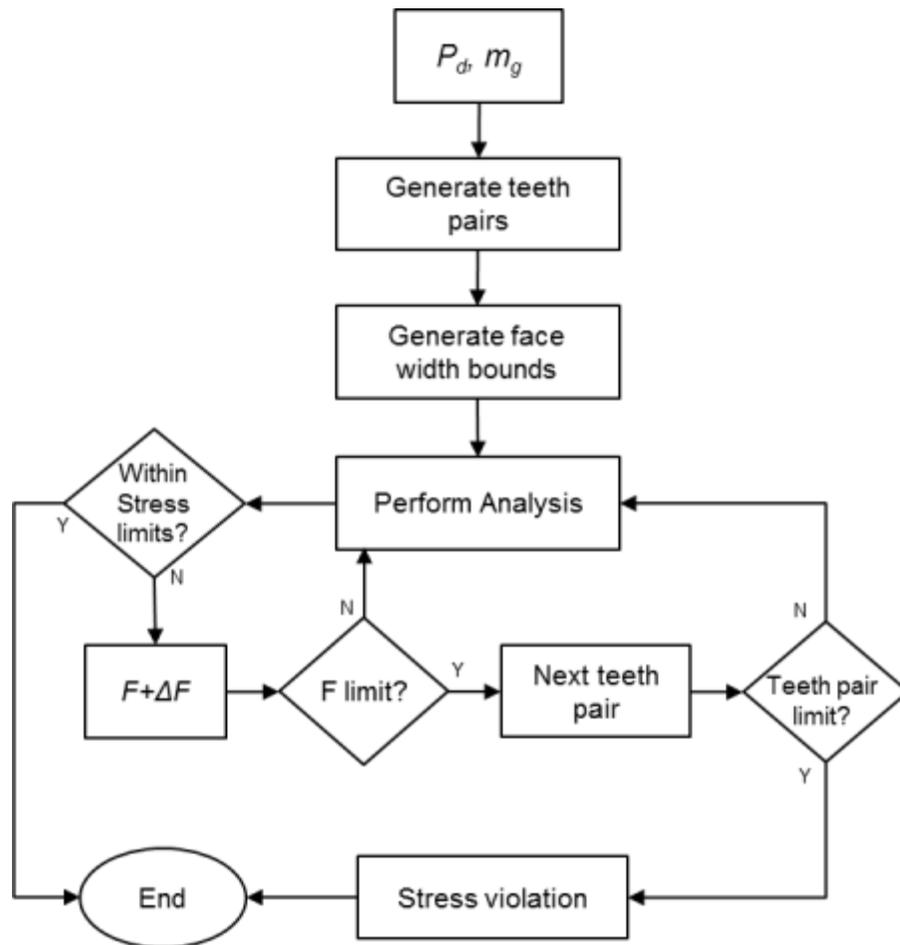


Figure 77. Gear stage sizing routine

The following constraint handling penalty techniques are studied here:

1. Rejection method ( or Death penalty)
2. Static – linear
3. Static – nonlinear
4. Dynamic – linear
5. Dynamic – nonlinear

In the rejection method, infeasible members are either filtered out or given a very low fitness. Since the GA here works with system weight as the objective, it is a minimization problem. In rejection, the ‘cost function’ (weight) is set to infinity. Thus, it has a poor fitness and theoretically has zero probability of being promoted to the next generation by being selected for crossover. The other method of rejection is to replace it with a feasible design. Although this method seems acceptable for a design space where there are very few constraints and unfit design members, it is cumbersome to program and have function for a large constrained problem: the algorithm ends up being sluggish trying to replace the unfit members. The ensuing question is then: what member the algorithm should replace the unfit ones with?

For these complexities, rejection is preferred to be implemented where it penalizes the fitness by setting the cost function to infinity. The obvious shortcoming of this technique is that it automatically loses information from members neighboring the constraint boundary. In a large constrained problem, this is undesirable. However, the algorithm does converge quickly, with an expected low success rate of finding the global optimum.

A simulation was performed on the rejection method with 50 runs. The simulation returned a success rate of 4% and average fitness of 0.971. The main problem with this method was premature convergence; the algorithm converged within 20 generations and

very easily gets stuck at a local optimum, because of aggressively penalizing all infeasible designs.

The global optimum was identified by a full-factorial algorithm that executed 4,194,304 function calls for all possible design combinations. The results are presented in Table 3.

Table 3. Three-stage full factorial optimization result

Parameter	Symbol	Global Optimum
Diametral Pitch	$Pd_1$	10
	$Pd_2$	6
	$Pd_3$	4
Gear Ratio	$mg_1$	2.81
	$mg_2$	3.38
	$mg_3$	2.62
Pinion Teeth	$Np_1$	21
	$Np_2$	21
	$Np_3$	21
Gear Teeth	$Ng_1$	59
	$Ng_2$	71
	$Ng_3$	55
Weight	$w_1$	11.95
	$w_2$	44.62
	$w_3$	94.94
Face Width	$F_1$	1.39
	$F_2$	1.33
	$F_3$	2.00
Total weight	$W$	151.509

While optimizing gear trains, certain combinations of design variables will lead to infeasible configurations. For example, for a given required gear ratio, combinations of gear ratio for stage 1 and 2 would lead to a gear ratio for stage 3 that is not within its bounds. This has been dealt as a constraint in the past; however, there is no need to penalize such a design because the information has particularly no use to the progress of

the gene pool. Rejecting such designs has potential advantages: 1) the space in the population is not occupied by a design that is not useful, and 2) the analysis program is not required to be run with an incompatible configuration.

Rejection technique is partially implemented in the algorithm using certain techniques depending on where the incompatible design occurs. When the initial population is randomly generated, a decision logic is placed to check if any of the members are incompatible. If they are incompatible, then the algorithm filters them by replacing them with a compatible configuration that is randomly generated. This can lead to a longer process if the design space has a large proportion of incompatible designs. However, starting the design process with compatible designs does have advantages for the genetic evolution progress. During crossover and mutation, new designs emerge –the way by which the GA explores the design space; when these designs are incompatible, it is not possible to filter them by replacing them with random designs because the GA would get structurally random. Instead, the design can be replaced by one of the original parents, in case of crossover, and replaced with the original member, in case of mutation. When a random member is introduced to replace the most frequently occurring design (mode), the design can either be filtered by replacing it with the first compatible design or by replacing it with the original string.

The static linear formulation is given in Equation 17. The magnitude of violation is normalized by the stress constraint value and multiplied by the penalty factor. Normalizing the constraint violation allows the algorithm to function consistently across a wide range, and the constraint is also non-dimensionalized so the penalty can be modified into a cost value. Similar formulation is observed for static nonlinear penalty (Equation 18), but the violation value is squared.

$$w_p = r_p (\max[o, g])$$

$$g = \frac{\sigma - \sigma_{\max}}{\sigma_{\max}}$$

Equation 17. Static linear penalty

$$w_p = r_p (\max[o, g])$$

$$g = \frac{(\sigma - \sigma_{\max})^2}{\sigma_{\max}}$$

Equation 18. Static nonlinear penalty

To test the impact of these penalty functions on the objective value, a simulation was run to obtain weight and stress information from 100,000 random designs. The gear stresses aren't independent, so a dependency check was performed to see how violations in pinion bending stress, gear bending stress and contact stresses were related. This is shown in Table 4. Stage 1 analysis returned no contact violations; this does not mean that no gear combination results in contact stress violation but this occurs because of the way the analysis program is setup to work. For given  $P_d$  and  $m_g$ , the sizing program only returns the minimum stress-violation configuration for number of teeth and face width. The alternate method was not tested for constraint dependency because this method is consistent with the manner in which the optimizer would work, and this information is directly relevant to the algorithm.

Since the constraints aren't independent of each other, it is important to keep the penalty factors at a moderate level to not over penalize designs, leading to loss of information from infeasible designs close to constraint boundary.

Table 4. Constraint dependency information

n = 100000	stage 1	stage 2	stage 3
Sbp violations	2440	18591	31051
Sbg violations	1575	17169	31051
Sc violations	0	589	15634
Sbp   Sc	-	100%	100%
Sbg   Sc	-	100%	100%
Sbg   Sbp	64.5%	92.4%	100%
Sc   Sbp	0%	3.2%	50.35%
Sc   Sbg	0%	3.4%	50.35%
Sbp   Sbg	100%	100%	100%

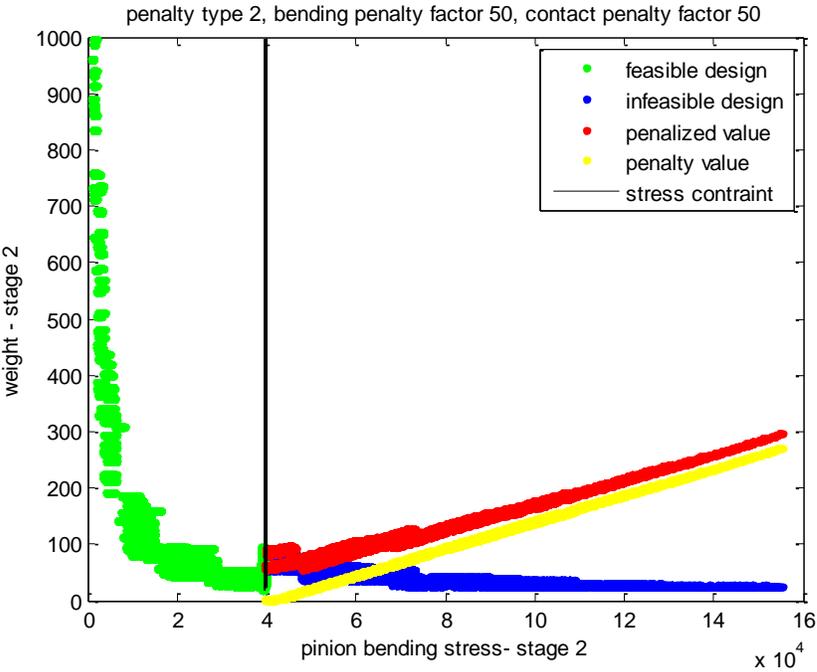


Figure 78. Static linear penalty – stage 2 ( $r_p = 50$ )

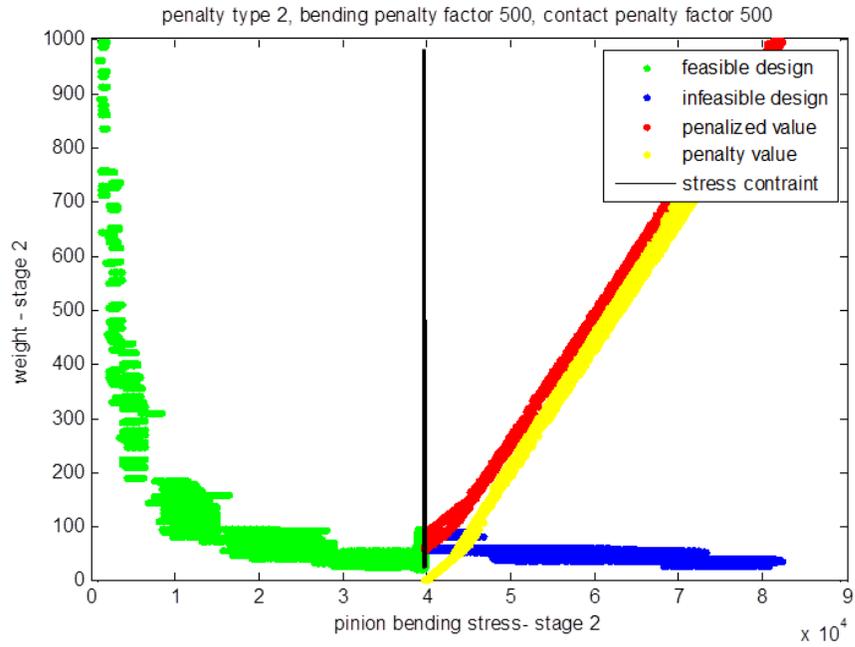


Figure 79. Static linear penalty – stage 2 ( $r_p = 500$ )

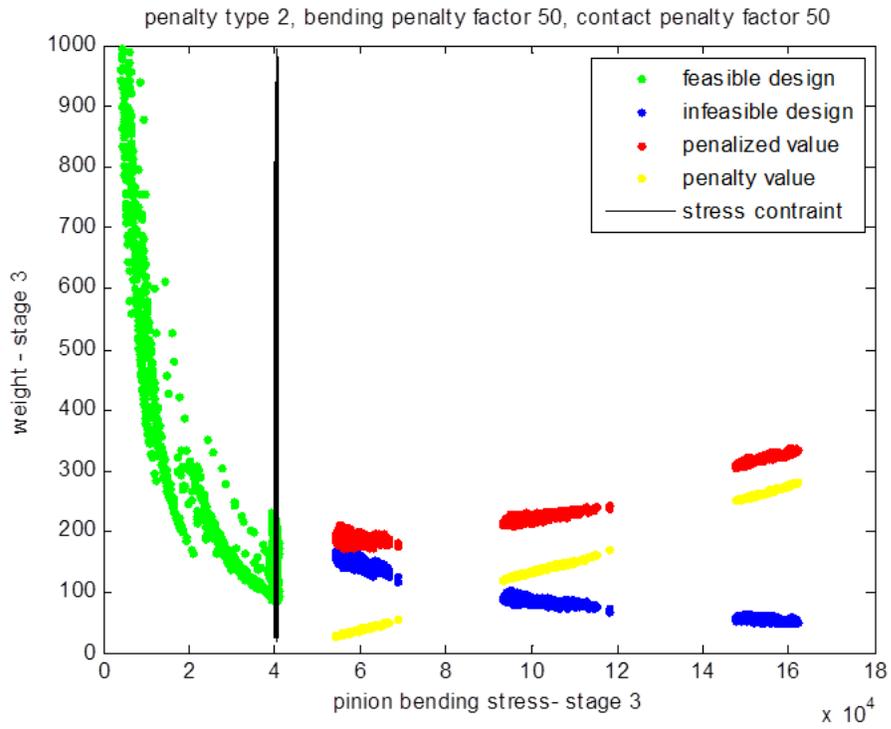


Figure 80. Static linear penalty – stage 3 ( $r_p = 50$ )

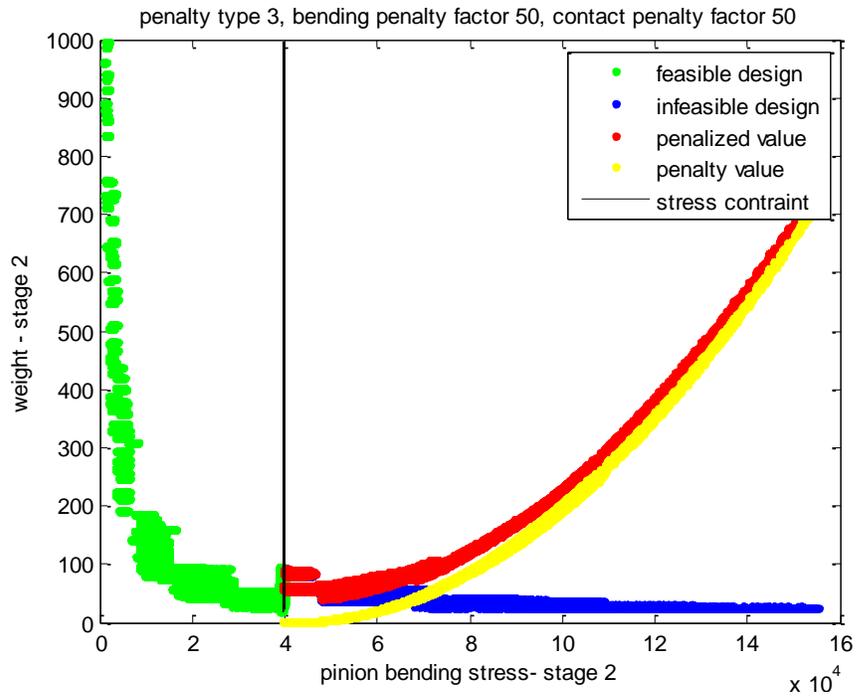


Figure 81. Static nonlinear penalty – stage 2 ( $r_p = 50$ )

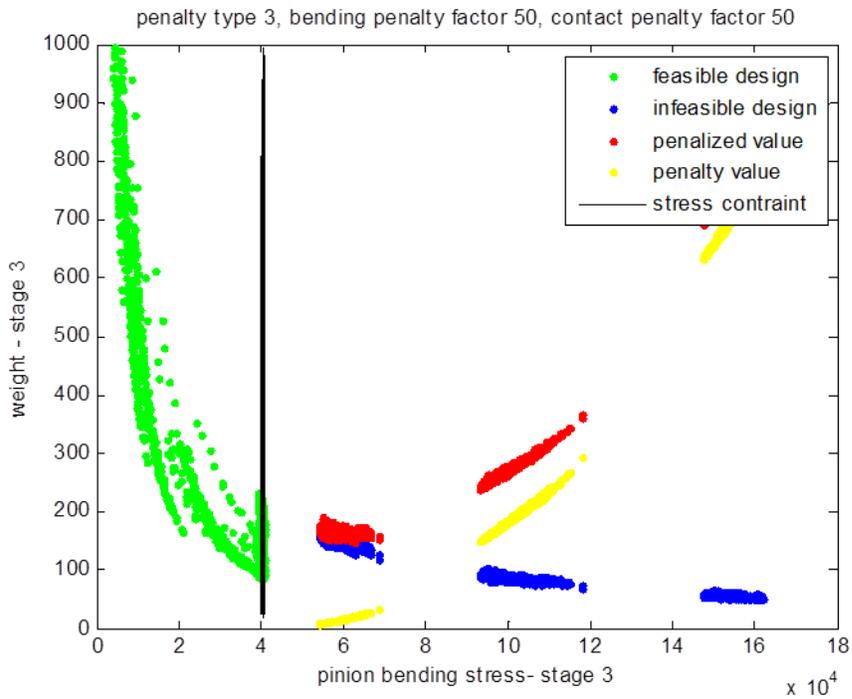


Figure 82. Static nonlinear penalty – stage 3 ( $r_p = 50$ )

The dynamic penalty functions are given in Equation 19 and Equation 20. Here the penalty function increases in value as the generations progress.  $B$  is an empirical constant that is assumed to be 0.5. Typical cost function distribution for this penalty type is shown in the following figures (Figure 83 - Figure 86). The dynamic penalty method is very sensitive to the penalty factors used (value of  $B$ ) and the expected convergence rate. If a longer convergence is expected, the dynamic penalty functions poorly because the penalty value increases. It tends to be more explorative in early stages (if a small penalty factor is used) and exploitive in later stages.

$$w_p = (B.gen).r_p(\max[o, g])$$

$$g = \frac{\sigma - \sigma_{\max}}{\sigma_{\max}}$$

Equation 19. Dynamic linear penalty

$$w_p = (B.gen).r_p(\max[o, g])$$

$$g = \frac{(\sigma - \sigma_{\max})^2}{\sigma_{\max}}$$

Equation 20. Dynamic nonlinear penalty

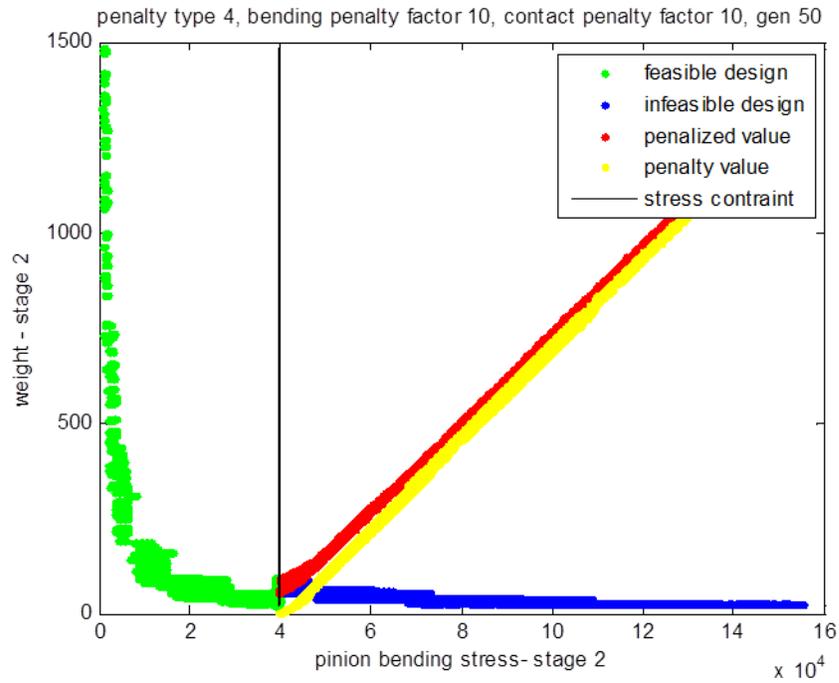


Figure 83. Dynamic linear penalty - stage 2 ( $r_p = 10$ , gen = 50)

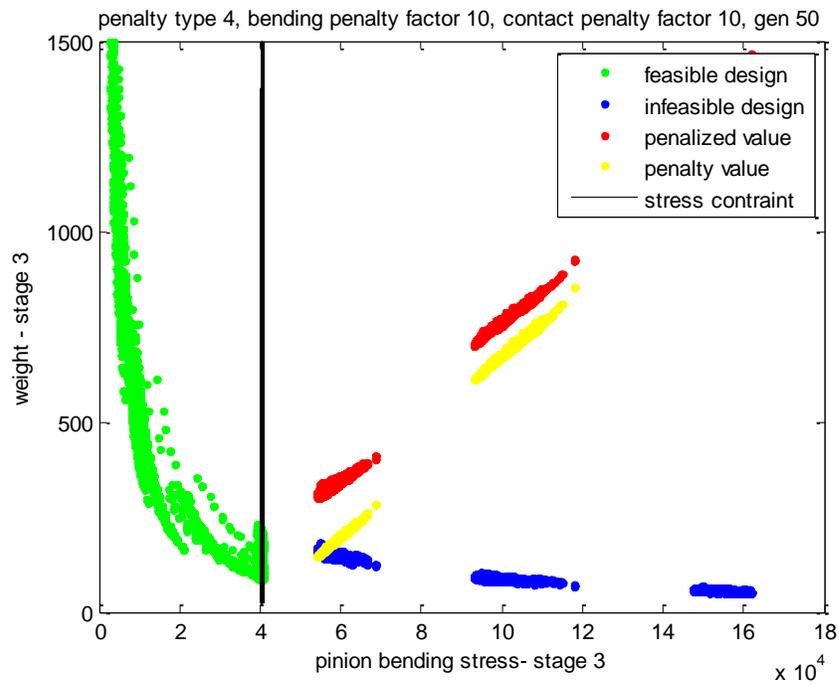


Figure 84. Dynamic linear penalty - stage 3, ( $r_p = 10$ , gen = 50)

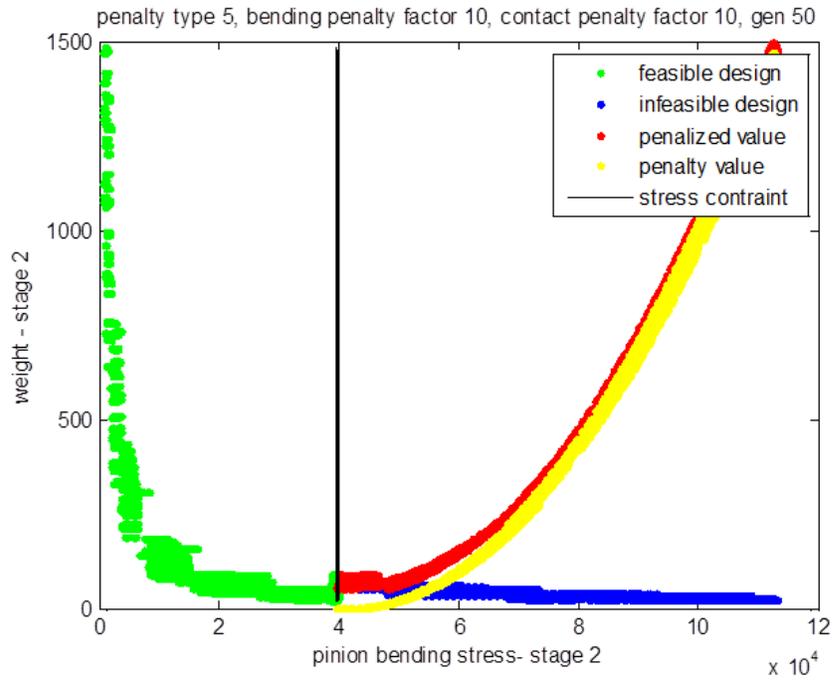


Figure 85. Dynamic nonlinear penalty - stage 2, ( $r_p = 10$ , gen = 50)

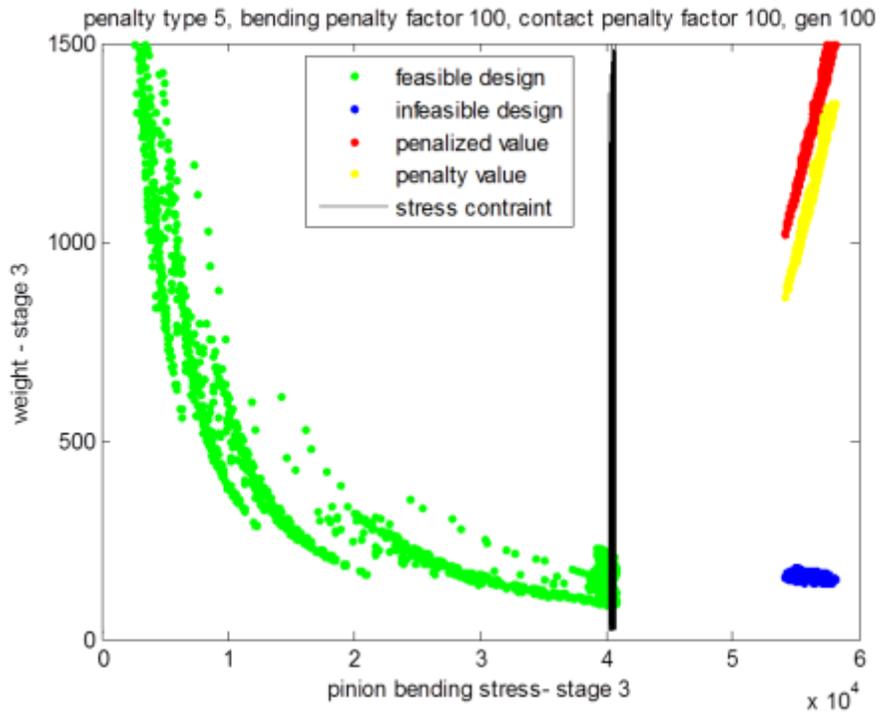


Figure 86. Dynamic nonlinear penalty - stage 3, ( $r_p = 100$ , gen = 100)

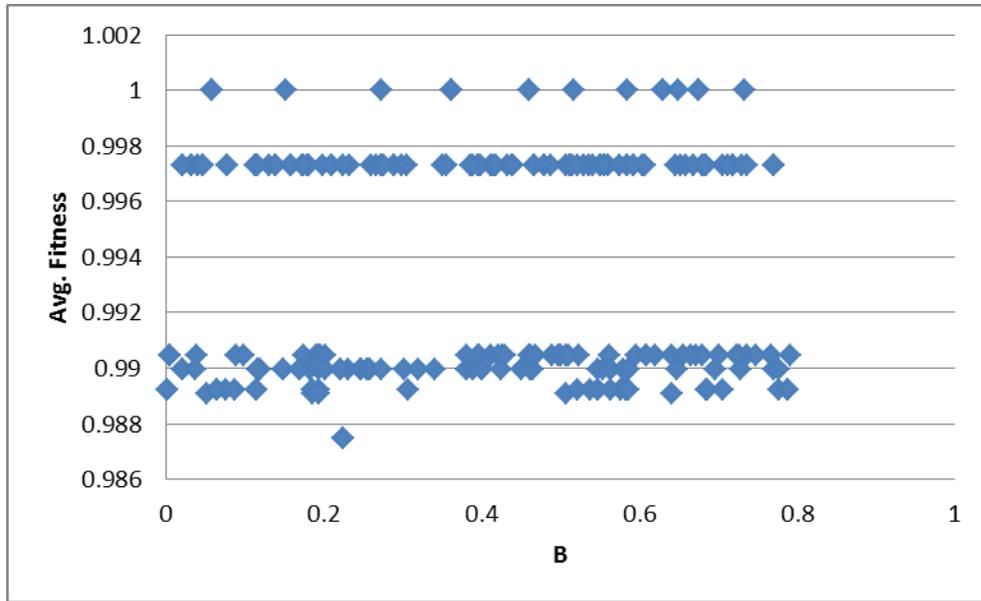


Figure 87. Average fitness vs. dynamic penalty factor  $B$

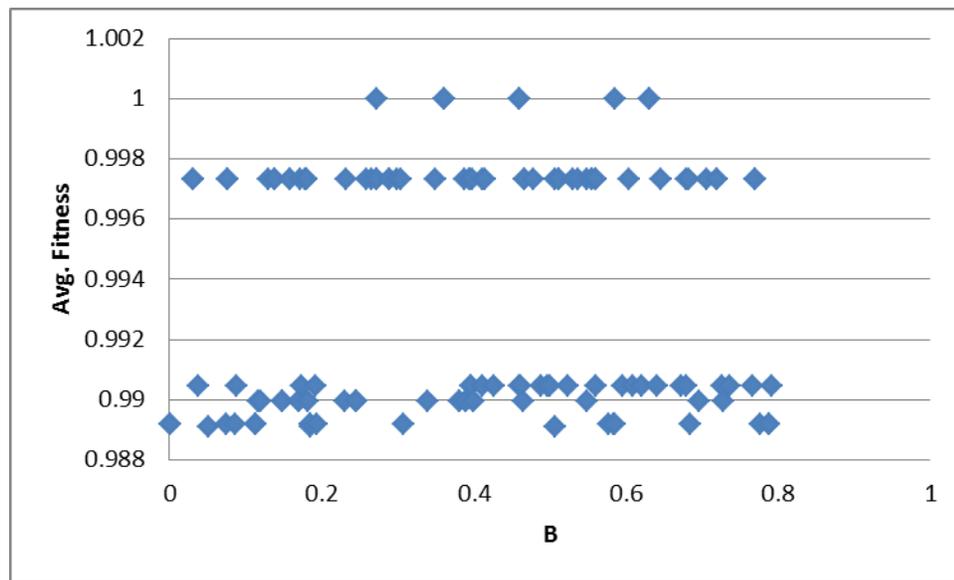


Figure 88. Average fitness vs. dynamic penalty coefficient  $B$  for dynamic linear

Figure 87 and Figure 88, show the success rate and average fitness for dynamic (linear and nonlinear) as value of  $B$  changes. The success of the linear type was less sensitive to  $B$ , while the nonlinear formulation tended to be more successful when the value of  $B$  was within the 0.3 to 0.7 range. Also observed was that the combination of

high penalty factor ( $r_p$ ) value and coefficient ( $B$ ) had a low success rate. Using a low penalty factor (50 to 100) along with mid-range value of  $B$  had a higher success rate ( $\approx 2:1$ ).

Summary of the different penalty techniques is shown in Table 5. These cases are not standard simulation for each penalizing technique but have varying factors such as penalty factor, method of parent selection, population size, number of sub-populations, etc.

Table 5. Summary of penalty techniques

Penalty Method	Success Rate	Avg. Normalized Fitness	Avg. Generations to Converge
Rejection	4%	0.971	20
Static linear	34.3%	0.992	5
Static non-linear	38.5%	0.994	6
Dynamic linear	10%	0.99	19
Dynamic non-linear	11%	0.992	15

### Parent Selection

Both the roulette wheel and tournament selection methods are used to select the breeding parents, and they can be alternately switched in the program. The tournament selection method has a higher likelihood of maintaining diversity in the population, since each member competes with only one other member and each member cannot be selected more than a maximum of two times. The roulette wheel method can sometimes tend to be more exploitive because it selects a design in relation to all other designs; therefore, it is possible for a design to be selected multiple times, especially if its fitness in relation to the other members is considerably larger. The roulette wheel selection does possess the

advantage of progressing fitter designs in a large diverse gene pool with a higher probability: so the mating pool in essence will have a higher fitness. The roulette wheel functions well when the population size is large enough. Simulation runs of the two methods for different penalty techniques indicated a higher success rate (3:1) for the roulette wheel selection method over the tournament selection method. The results are given in Table 6.

Table 6. Parent selection – simulation results

Selection Method	Success Rate	Avg. Normalized Fitness	Avg. Generations to Converge
Tournament	4%	0.992	8
Roulette wheel	12%	0.997	15

### **Adaptive Genetic Algorithm**

The methodology for adaptive crossover and mutation rates was introduced in Section 4.3.1 (page 117). A technique to estimate the empirical coefficients  $k_1$  and  $k_2$  in Equation 12, based on standard deviation of the population fitness was experimented here. The primary requirement of these coefficients is to keep the probability rates within favorable range. The other is to efficiently influence more exploration when the population diversity is decreasing. Based on experiments performed on the probability ranges, the following values for  $k_1$  and  $k_2$  were determined:

$$k_1 = \left\{ \begin{array}{ll} 0.9 & \sigma < 0.2c^* \\ 0.8 & \sigma < 0.3c^* \\ 0.7 & ow \end{array} \right\}$$

$$k_2 = \left\{ \begin{array}{ll} 0.15 & \sigma < 0.15c^* \\ 0.1 & \sigma < 0.2c^* \\ 0.05 & ow \end{array} \right\}$$

Equation 21. Coefficient values for adaptive crossover and mutation rates

Where  $c^*$  is cost (weight + penalty) of the fittest member of the previous generation.

Results of using these values are shown in Figure 89 - Figure 92. For the dynamic penalty case (Figure 90), as the generations progress, the magnitude of the penalty increases linearly, this causes the mean fitness of the population to drop significantly as can be seen in Figure 90. Values for  $k_1$  and  $k_2$  were determined to ensure that the probabilities remained within the reasonable range even when the mean population value dropped considerably as a result of aggressive penalizing techniques.

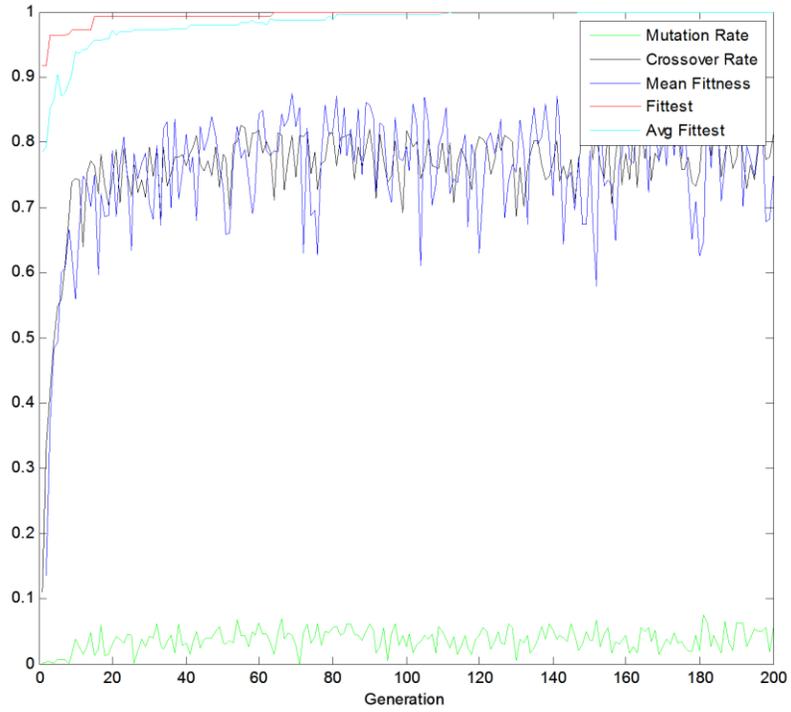


Figure 89. Adaptive GA results for static penalty

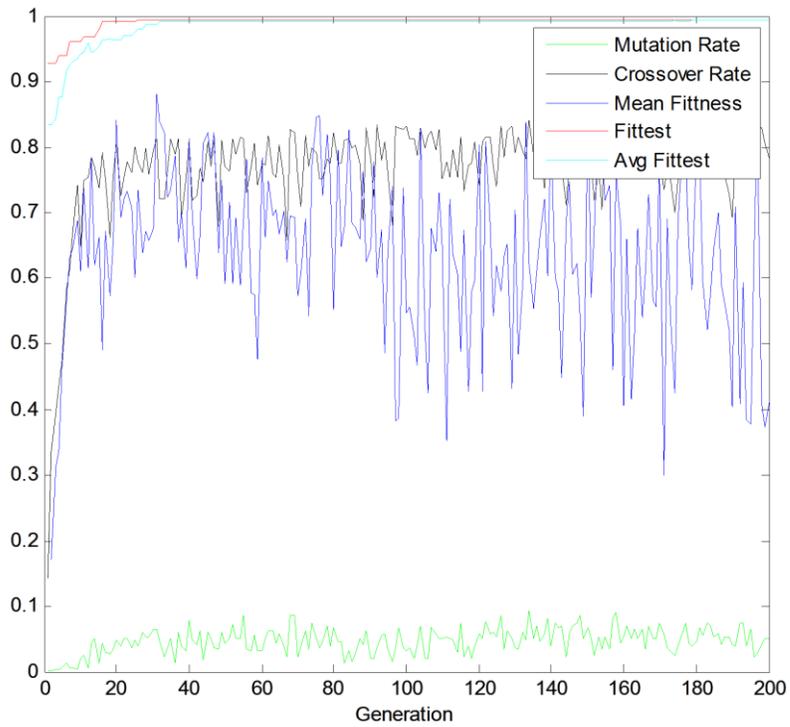


Figure 90. Adaptive GA results for dynamic penalty

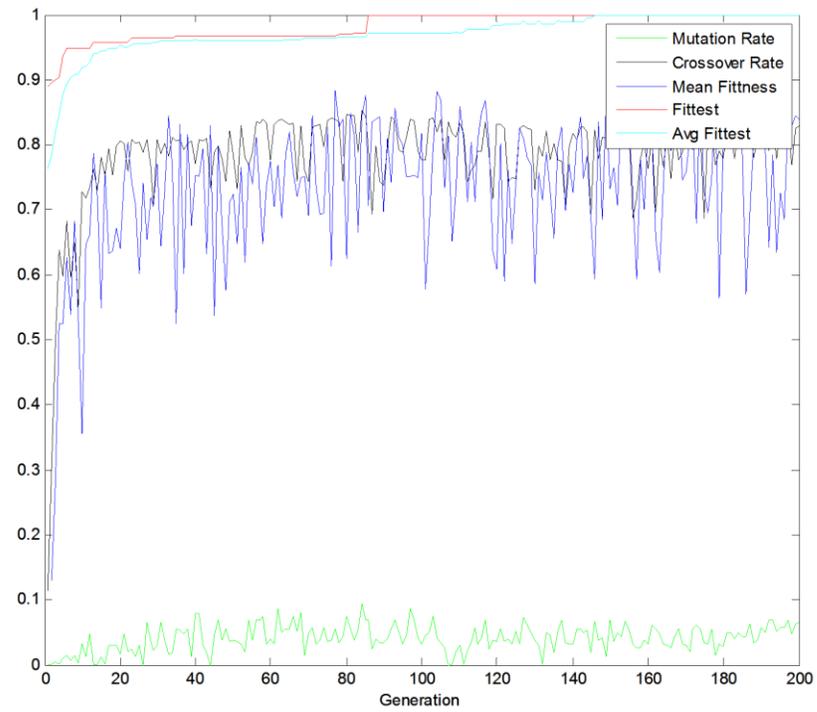


Figure 91. Adaptive GA results for static nonlinear penalty

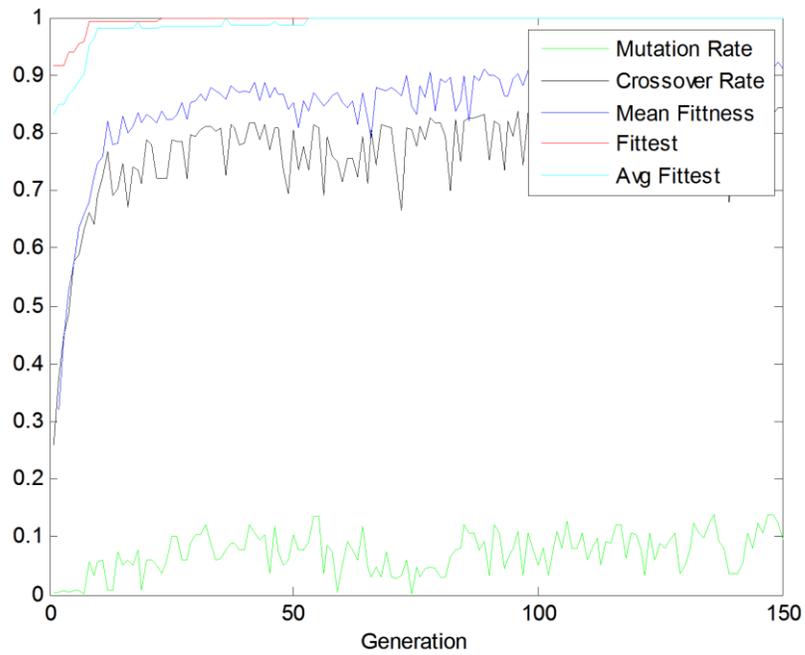


Figure 92. Adaptive GA results for four-stage gear train (static nonlinear)

Parallel GA and migration were introduced in Sections 2.4 and 4.3.1. A representation of parallel GA is shown in Figure 93. Here multiple subpopulations are setup; they are isolated from each other except through migration. Each sub-population works as a regular population in regards to parent selection, crossover, mutation and elitist list. At the end of each generation, before the elitist member in each sub-population is identified, the entire population goes through a migration process. The migration process has a pre-specified migration probability (set to a very low value of 0.01). The code for migration can be found in Appendix A.2.i. The migration process is as follows: for each population index (increasing from 1 to the size of population), if the migration probability is satisfied, a random permutation is generated. This random permutation determines which sub-population each member will migrate to, thus the migration is completely random.

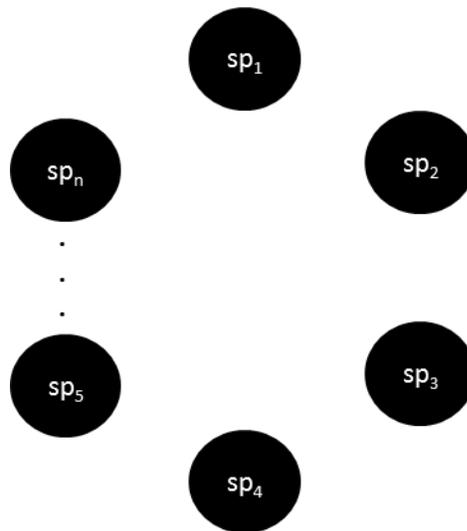


Figure 93. Parallel GA

Initial GA operations showed the algorithm very quickly converge at a local optimum. This is a regularly observed problem in multimodal problems. The penalizing aspect of imposing constraints makes the design problem a multimodal space. It is very

important, as a result, to avoid premature convergence and improve the diversity of the population. The success of finding the global optima and avoiding premature convergence improved considerably when sub-populations and limited migration was introduced. The results for simulations performed on sub-populations are shown in Figure 95. These simulations were performed with different sizes of populations, different penalty techniques, and both types of parent selection methods.

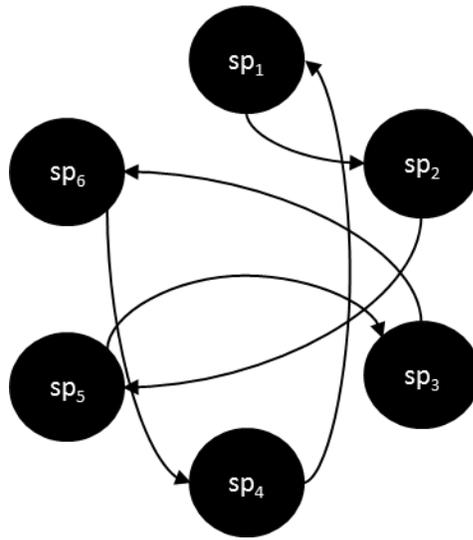


Figure 94. Migration in GA

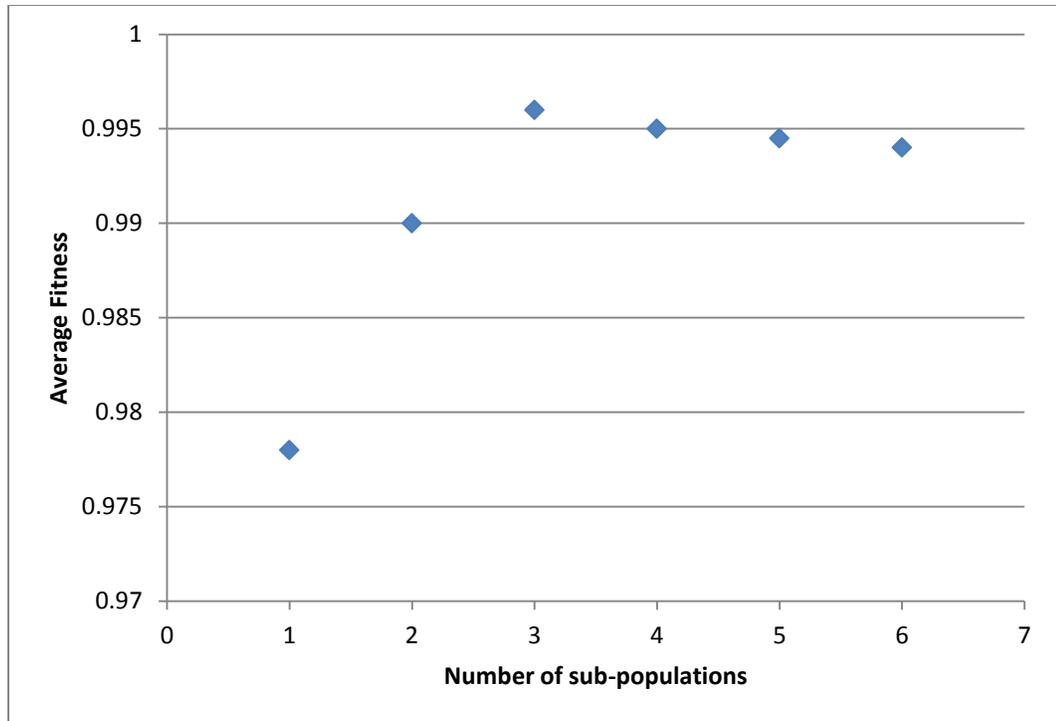


Figure 95. Average fitness vs. number of sub-populations

The complete GA flowchart for the three-stage and four-stage gear analysis is shown in Figure 96.

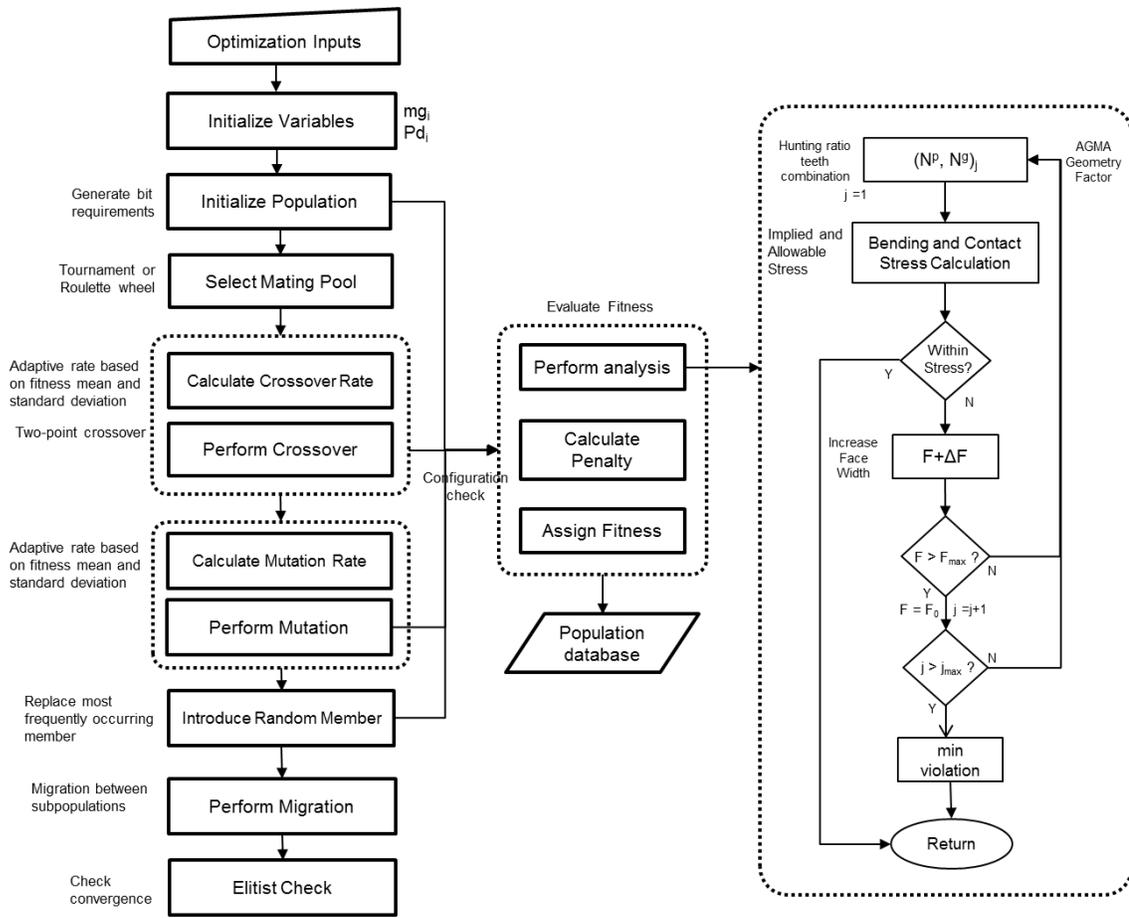


Figure 96. Genetic algorithm for gear train design

### 5.3 Spacing Analysis

The objective of this analysis is to study how volumetric constraints and input/output location information affect the design solution. To study Research Question #3 (Chapter 3), a spacing analysis program was developed in MATLAB. The spacing analysis program is used to study how a three-stage gear can be accommodated inside a cylinder. The model is shown graphically in Figure 97 and Figure 98.

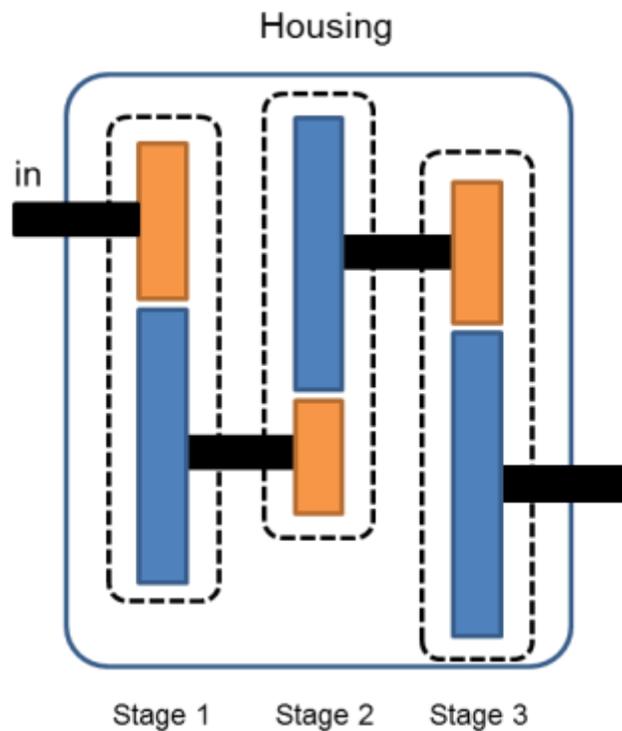


Figure 97. Three stage reduction drive inside a cylindrical housing

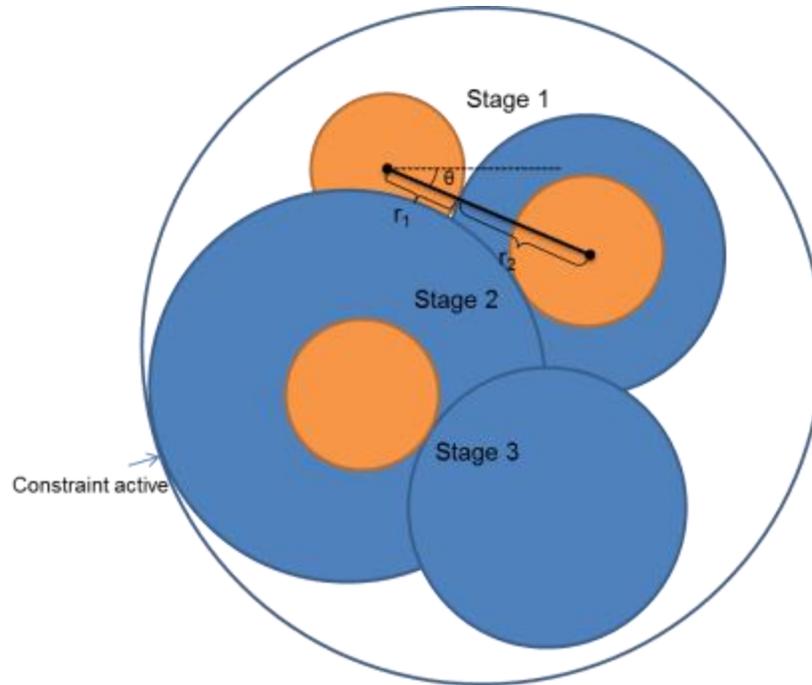


Figure 98. Three stage reduction drive inside a cylindrical housing (front view)

The algorithm for spacing analysis is presented Figure 99. The implementation of the model is shown in Figure 101. Here the part volume is altered by changing the stage 1 gear and stage 2 pinion orientations. Locations of pinion for stage 1 and gear for stage 3 are determined by the input and output shaft location information.

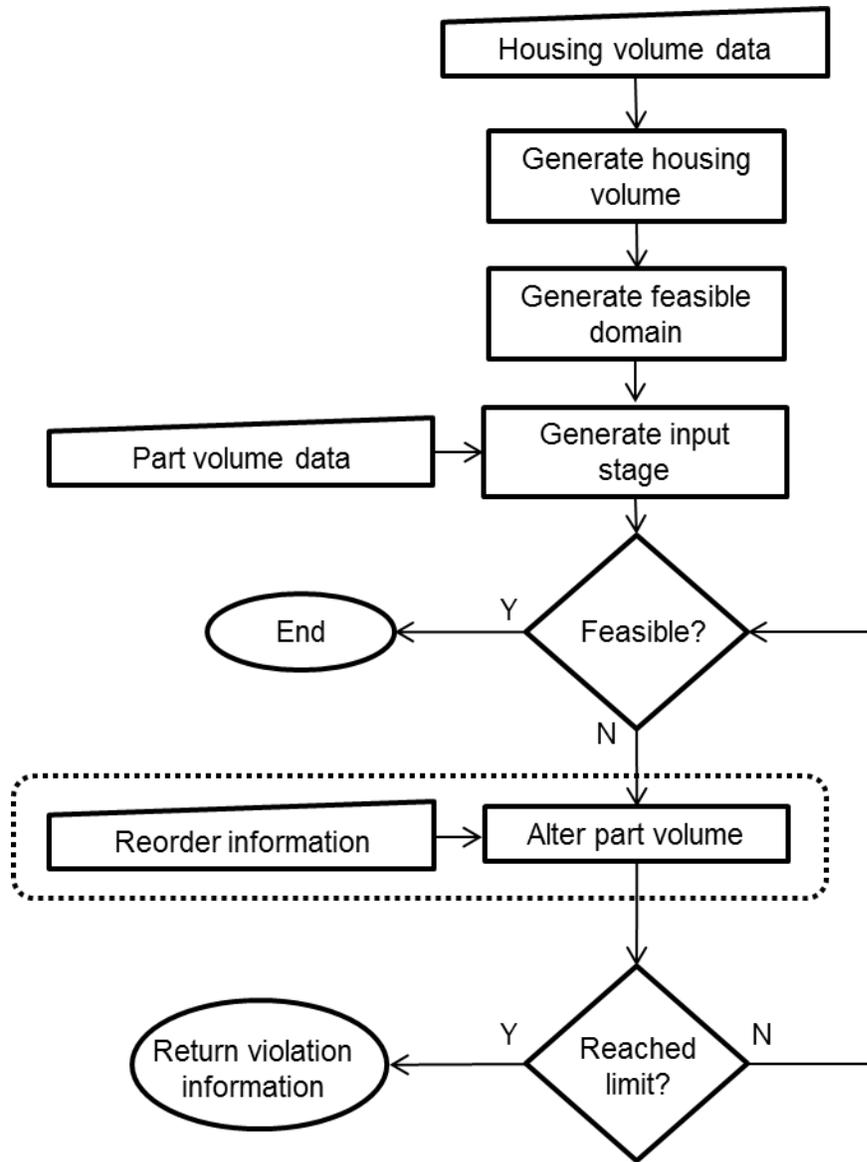


Figure 99. Spacing analysis algorithm

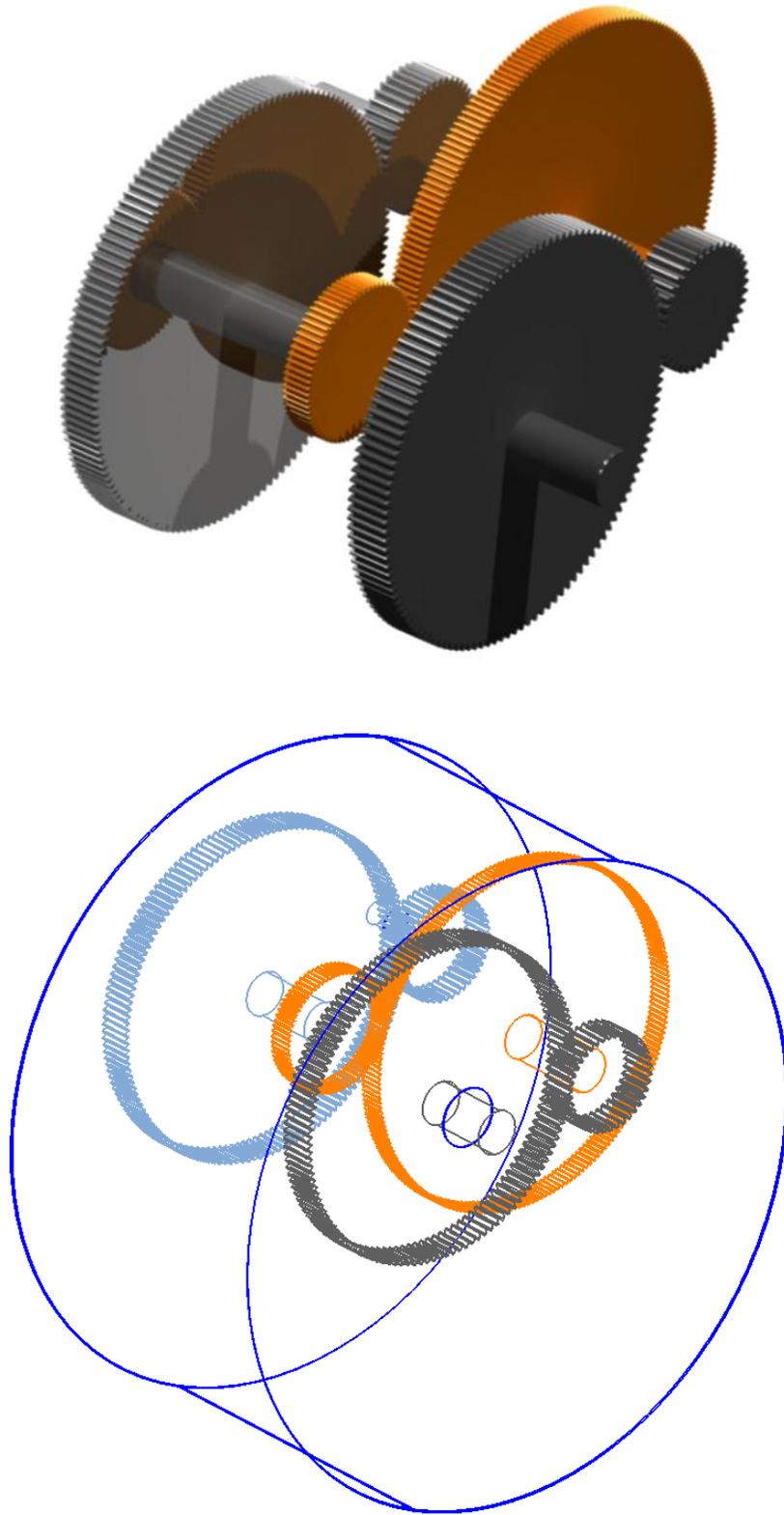


Figure 100. Three-stage gear spacing analysis problem

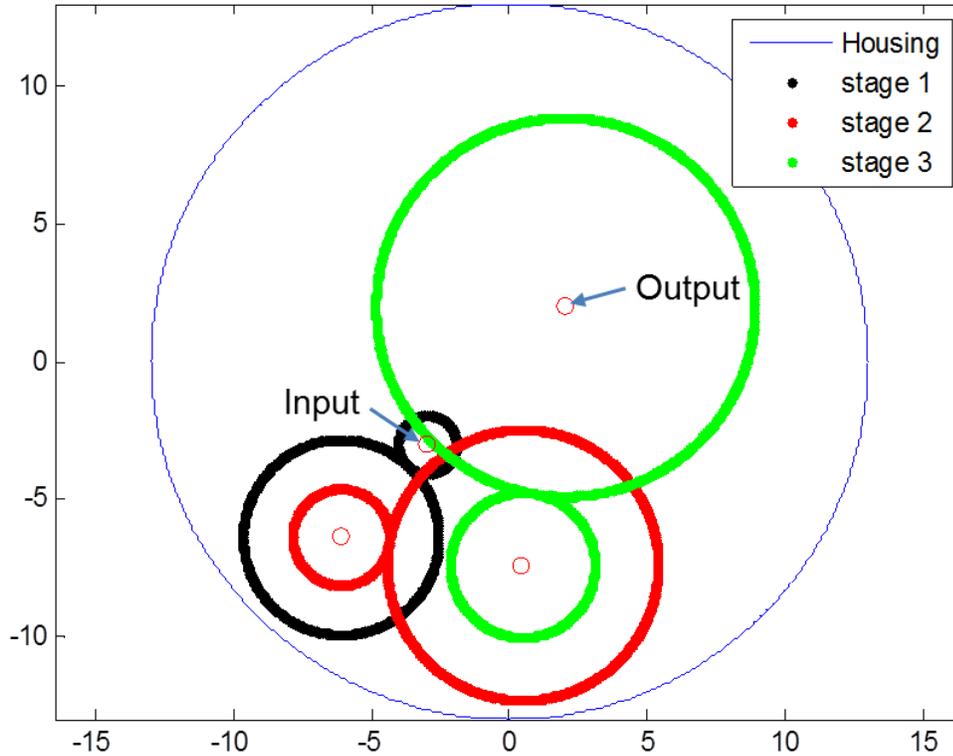


Figure 101. Three-stage gear and housing - spacing analysis algorithm implemented in  
MATLAB

To test *Hypothesis #1*, the analysis was performed on two designs shown in Table 7. Design A weighs less than design B. The two configurations have the exact same stage three gears; however, gear ratios for stage 1 and 2 are flipped for the two designs. Making stage one in design A smaller and stage two larger in relation to design B. The two designs are shown in Figure 102 for a housing cylinder of radius 13 in. The input and output shaft locations are offset from the center in this example.

Table 7. Spacing analysis designs

Parameter	Symbol	Design A	Design B
Diametral Pitch	$Pd_1$	10	10
	$Pd_2$	6	6
	$Pd_3$	4	4
Gear Ratio	$mg_1$	2.81	3.38
	$mg_2$	3.38	2.81
	$mg_3$	2.62	2.62
Pinion Teeth	$Np_1$	21	21
	$Np_2$	21	21
	$Np_3$	21	21
Gear Teeth	$Ng_1$	59	71
	$Ng_2$	71	59
	$Ng_3$	55	55
Weight	$w_1$	11.95	16.74
	$w_2$	44.62	40.24
	$w_3$	94.94	94.94
Face Width	$F_1$	1.39	1.39
	$F_2$	1.33	1.68
	$F_3$	2.00	2.00
Total weight	$W$	151.51	151.92

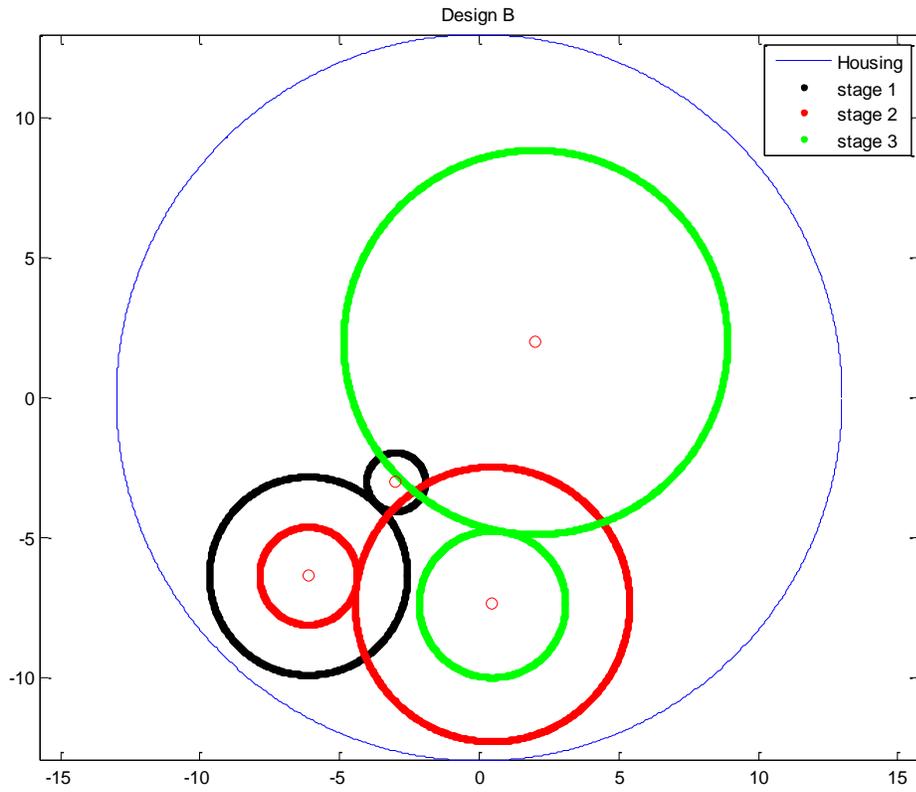
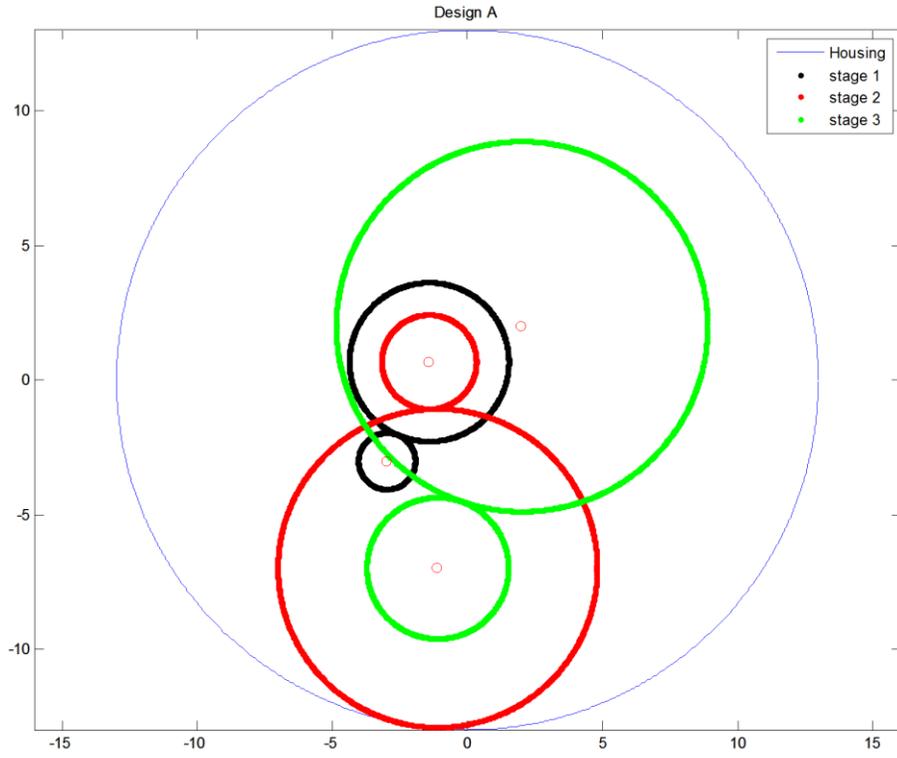


Figure 102. Spacing configuration for designs A and B with 13 in. radius housing

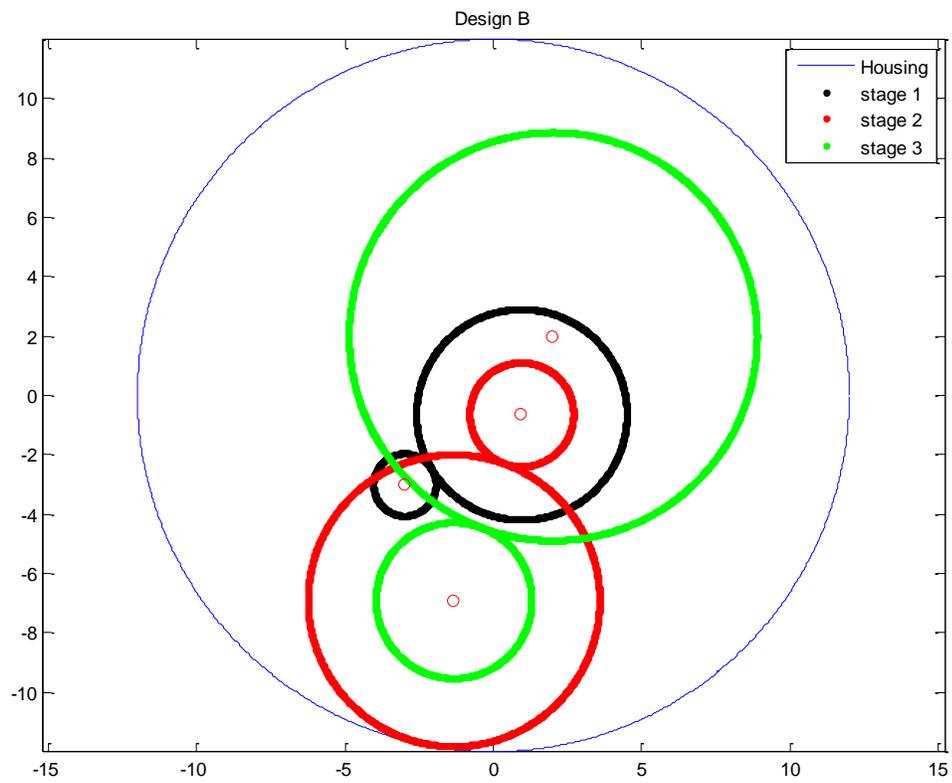
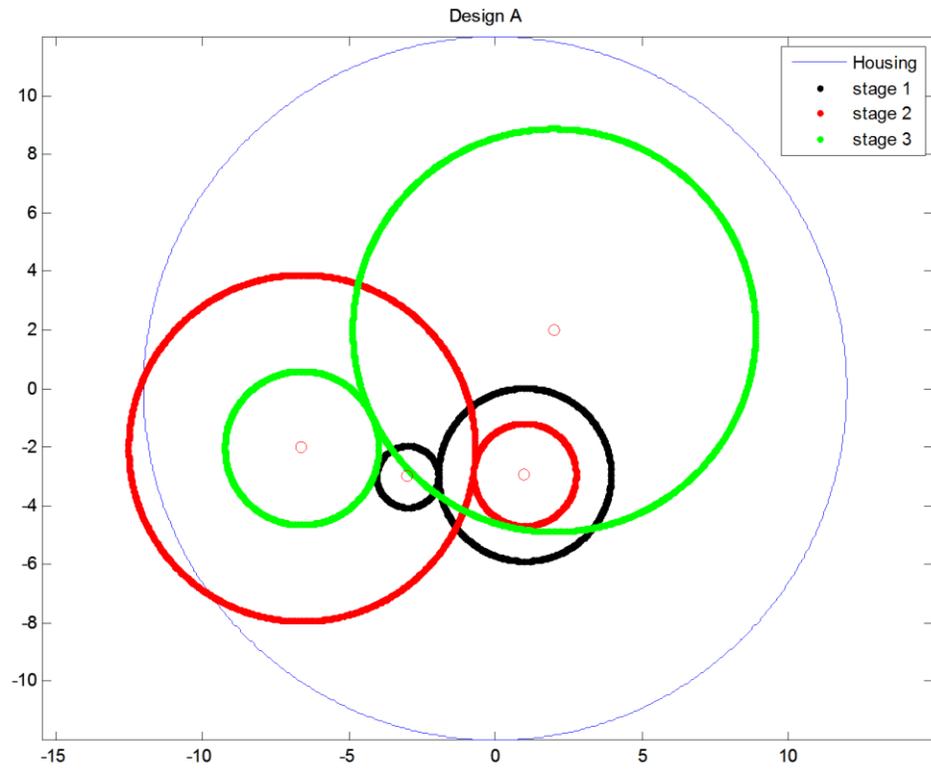


Figure 103. Spacing configuration for designs A and B with 12 in. radius housing

The optimization directly implemented without geometry analysis would result in *design A* as the optimal design. However, as can be seen in Figure 103, when the housing radius is reduced to 12 in, with the input and output locations held constant, *design A* exceeds the space restrictions making it infeasible. The minimum violation for this configuration of *design A* was 0.82 in. *Design B*, on the other hand fits within the given volume. Hence, *Hypothesis #1* is substantiated.

The geometric constraint is brought in to the optimization sequence as a penalty to alter the system weight using the following equation.

$$penalty = r_p \left( \frac{violation}{radius} \right)$$

Equation 22. Spacing penalty

For a violation of 0.82 in and radius 12 in, with a penalty factor ( $r_p$ ) of 100 yields a system weight of 158.34 lbs. This ensures that under the given geometric constraints, *design A* would be considered a sub-optimal design while performing the optimization.

#### 5.4 Flexibility of Design

To test design flexibility, the fully-relational design framework was used. The analysis was performed on the three-stage and four-stage configurations, shown in Figure 104 and Figure 105, respectively.

The baseline for each configuration was sized for an input power of 500 HP, input rpm of 35000, and required gear reduction ratio of 25:1. To test flexibility, these inputs are required to be changed and the effect quantified.

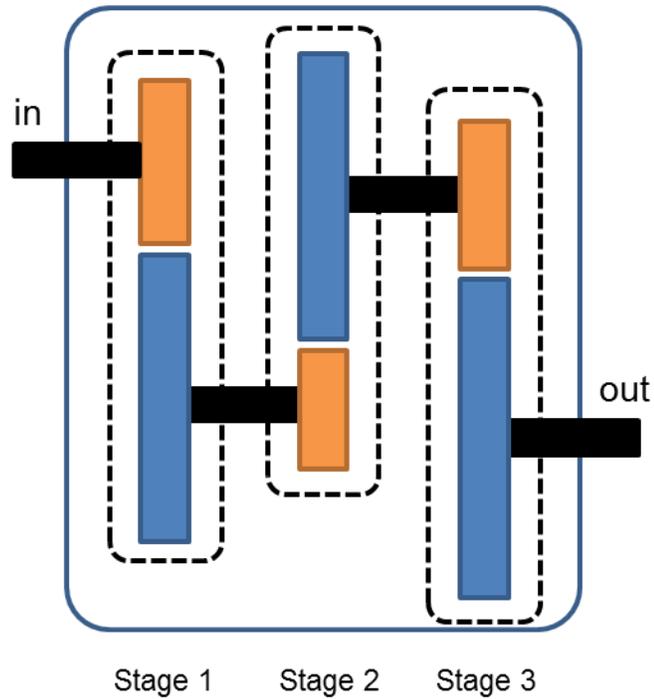


Figure 104. Three-stage gear train

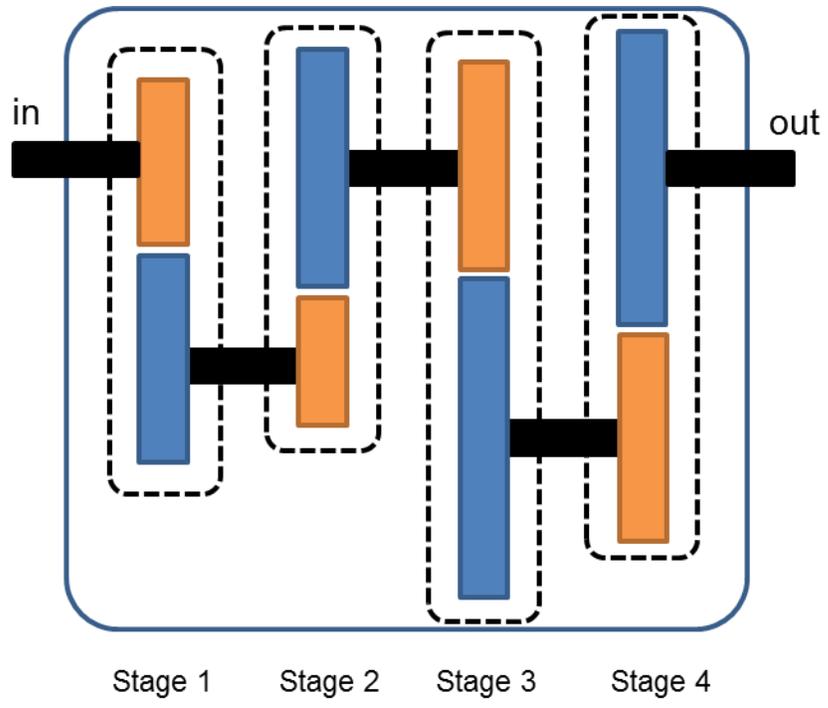


Figure 105. Four-stage gear train

Table 8. Three stage and four stage designs

Parameter	Symbol	Three stage	Four stage
Diametral Pitch	$Pd_1$	10	10
	$Pd_2$	6	8
	$Pd_3$	4	5
	$Pd_4$	-	4
Gear Ratio	$mg_1$	2.81	1.76
	$mg_2$	3.38	2.68
	$mg_3$	2.62	2.10
	$mg_4$	-	2.52
Pinion Teeth	$Np_1$	21	21
	$Np_2$	21	22
	$Np_3$	21	21
	$Np_4$	-	21
Gear Teeth	$Ng_1$	59	37
	$Ng_2$	71	59
	$Ng_3$	55	44
	$Ng_4$	-	53
Weight	$w_1$	11.95	5.6493
	$w_2$	44.62	20.02
	$w_3$	94.94	33.2419
	$w_4$	-	93.6636
Face Width	$F_1$	1.39	1.4316
	$F_2$	1.33	1.4737
	$F_3$	2.00	1.6
	$F_4$	-	2.1053
Total weight	$W$	151.509	152.5766

When this information is changed, each configuration is required to be optimized for this new information, requiring an optimization routine to be run many times. This is computationally expensive and not a viable option to perform a Monte Carlo simulation

to quantify flexibility. To alleviate the computation requirement, a metamodeling technique is implemented. Using Design of Experiments (DOE), a surrogate model can be built and used to perform a quicker simulation precluding the need for excessive optimization runs. DOE is a modeling and simulation technique that is used to build and study computationally intensive models. It is a process of making purposeful changes to inputs in order to observe the corresponding changes in the outputs. The inputs are known as design variables and the outputs are known as the responses. DOE uses a design set that tries to maximize the information and minimize the experimental effort [138]. Response Surface Methodology (RSM) is a technique used to build and optimize empirical models. Through the use of multivariate least squares regression, RSM approximates the output response to input parameters with a polynomial empirical equation. The regression equation is known as a Response Surface Equation (RSE). DOE is used to generate the input data set and obtain the regression data. RSM is generally used when the underlying physics of the analysis is not clearly known and to obtain an empirical approximation [139]. Using RSE allows rapid and efficient prediction of a much more complex and time consuming analysis, such as this gear train optimization routine. The alternate empirical model used to calculate the result is known as a surrogate model [47]. The polynomial equation that is used as the RSE is a second order approximation of the Taylor series given as follows:

$$R = b_o + \sum_{i=1}^k b_i x_i + \sum_{i=1}^k b_{ii} x_i^2 + \sum_{i=1}^{k-1} \sum_{j=i+1}^k b_{i,j} x_i x_j + \varepsilon$$

Equation 23. Response surface equation [140]

Where,

- $R$  Response
- $b_i$  Regression coefficient for the first order terms
- $b_{ii}$  Regression coefficient for the pure quadratic terms
- $b_{ij}$  Regression coefficient for the pure quadratic terms
- $x_{ij}$  Independent variables or factors
- $\varepsilon$  Associated error for neglecting higher order terms

### Case 1

A Central Composite Design – Orthogonal DOE table was generated using JMP for variation in power (400 to 600 HP), rpm (30000 to 40000) and gear ratio (20 to 30). The model fit for the three-stage and four-stage gear box can be found in Appendix A.4.

Table 9. DOE input table

Pattern	Power	rpm_in	mg_req
+++	600	37500	28
0	500	33750	25
0a0	500	27495	25
++-	600	37500	22
0	500	33750	25
---	400	30000	22
00A	500	33750	30
-+-	400	37500	22
+--	600	30000	22
+++	600	30000	28
0	500	33750	25
a00	333	33750	25
0	500	33750	25
0	500	33750	25
A00	667	33750	25
0	500	33750	25
0A0	500	40005	25
0	500	33750	25
0	500	33750	25
0	500	33750	25
--+	400	30000	28
-++	400	37500	28
00a	500	33750	20

The metamodel is used to perform a Monte Carlo simulation on the two configurations to evaluate their flexibility. The input distributions for this study are shown in Figure 106. A truncated normal distribution is used to ensure that values don't exceed the bounds.

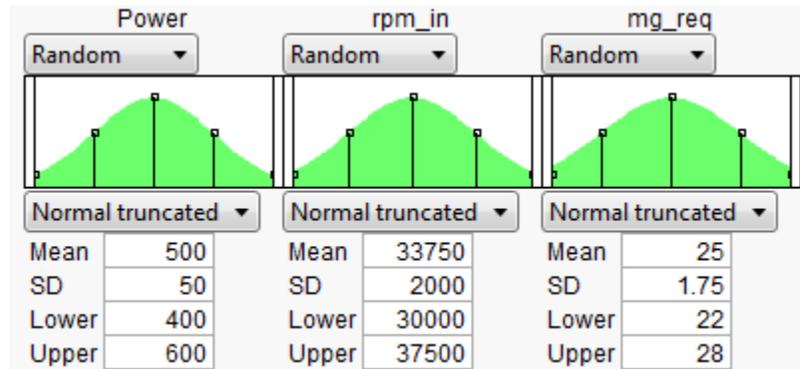


Figure 106. Distribution of input variables – case 1

Figure 107 and Figure 109 show the Probability Mass Function (PMF) and Cumulative Distribution Function (CDF), respectively, results for the simulation performed on the three stage gear system. Similarly, Figure 108 and Figure 110 show the simulation results for the four stage gear system.

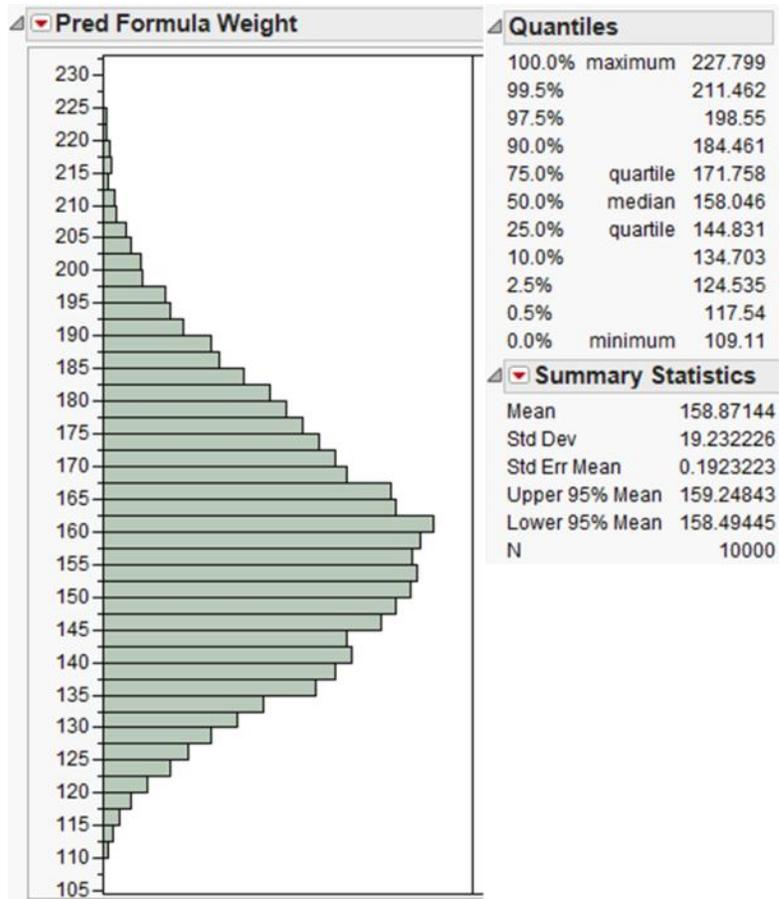


Figure 107. Three stage PMF result from simulation – case 1

Figure 107 shows the Probability Mass Function (PMF) and the summary for the simulation performed on the three stage gear system for case 1.

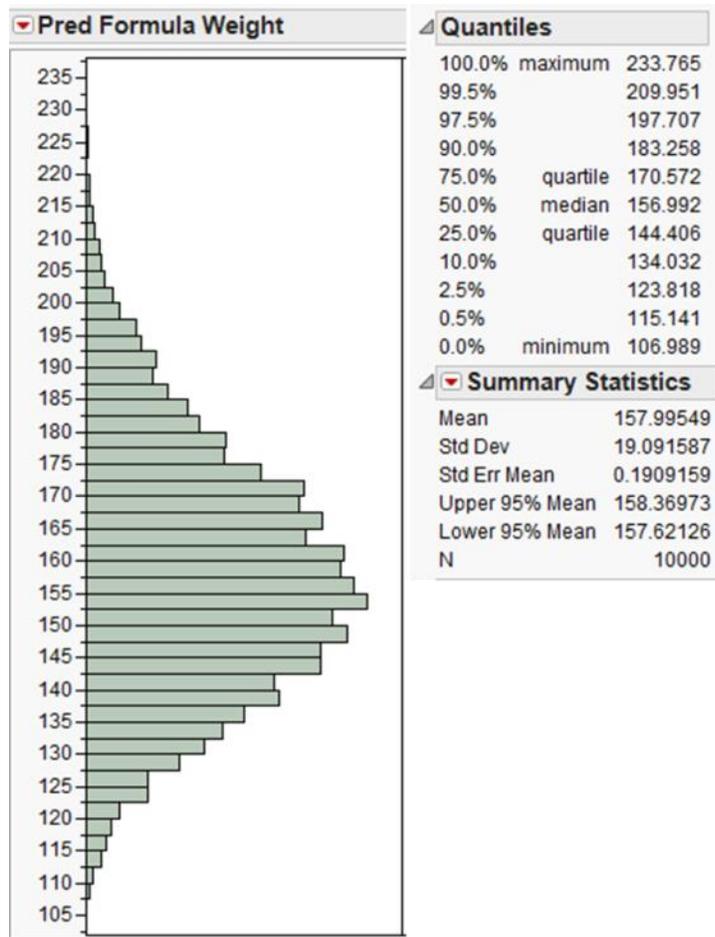


Figure 108. Four stage PMF result from simulation – case 1

Figure 108, shows the Probability Mass Function (PMF) and the summary for the simulation performed on the four stage gear system for case 1.

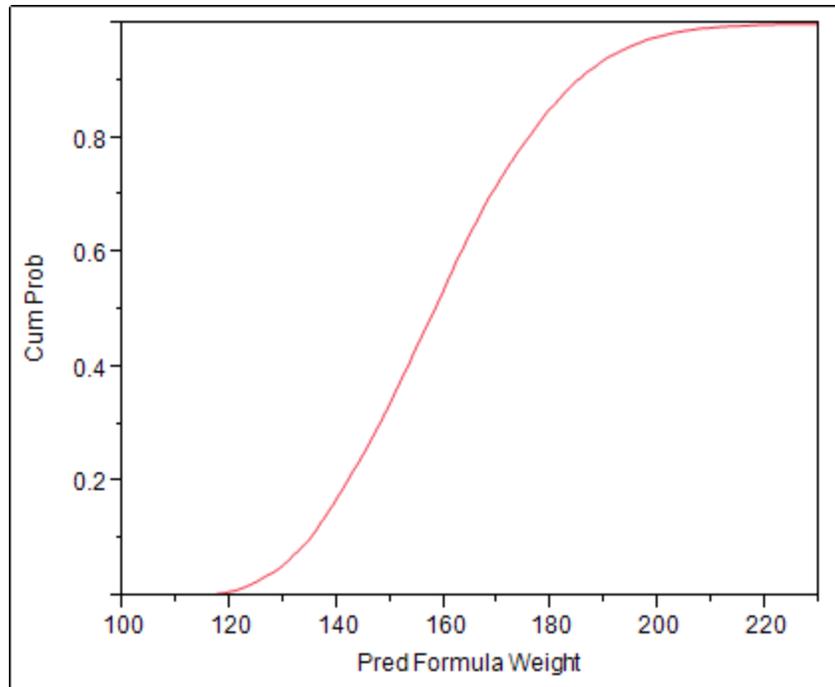


Figure 109. Three stage CDF result from simulation – case 1

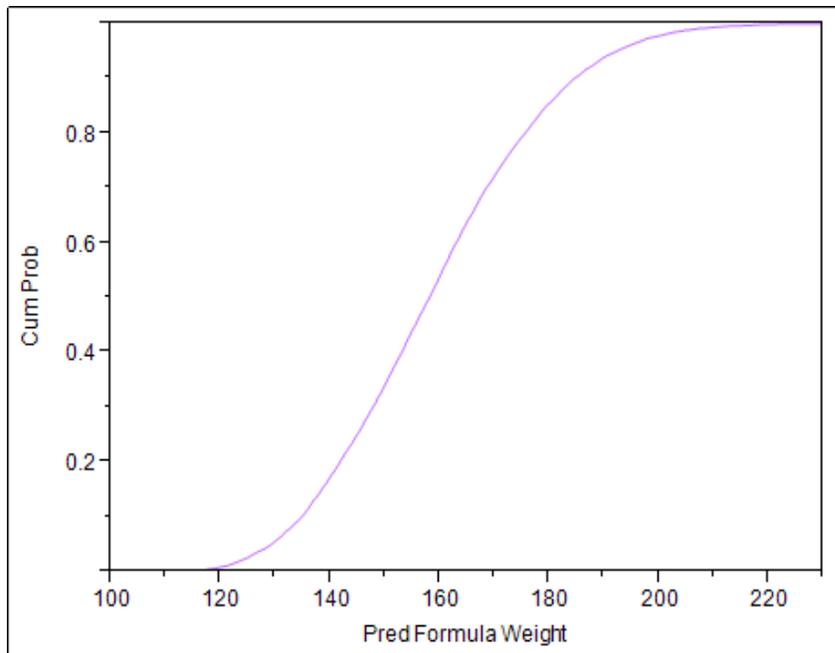


Figure 110. Four stage CDF result from simulation – case 1

The studies above provide a useful method to quantify the success of a particular configuration if it were required to be resized for a different set of requirements. Here the four stage configuration provides no flexibility advantage over the three stage design. With this study performed, the designer can go ahead with the design decision to select the three stage gear system for the given requirements.

### Case 2

To study flexibility and the effects of variability in requirements, another test case was studied with a larger bound on the input variables. Power is varied from 500 to 800 HP, speed from 25000 to 35000, and reduction ratio from 25 to 35. A new DOE was performed to build the RSEs for the two configurations for the new bounds. The model fit information can be found in Appendix A.4. For the input distributions shown in Figure 111, a Monte Carlo simulation with 10000 runs was performed for the three-stage and four-stage gear systems. The results are shown in Figure 112, Figure 113, Figure 114 and Figure 115. This time around, the four-stage gear system does show moderate advantages in 25% quartile and above. At 25% quartile the four-stage is 3 lbs. lighter and at the 75% quartile is 4.3 lbs. lighter. With the flexibility of both designs under extreme conditions assessed, the design can make a sound decision on the configuration to select.

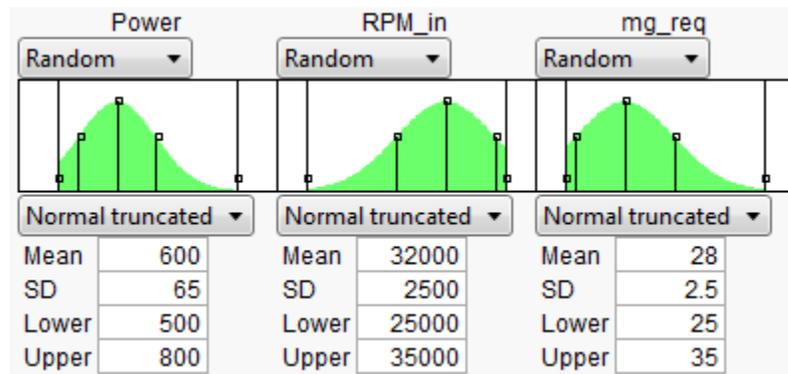


Figure 111. Distribution of input variables – case 2

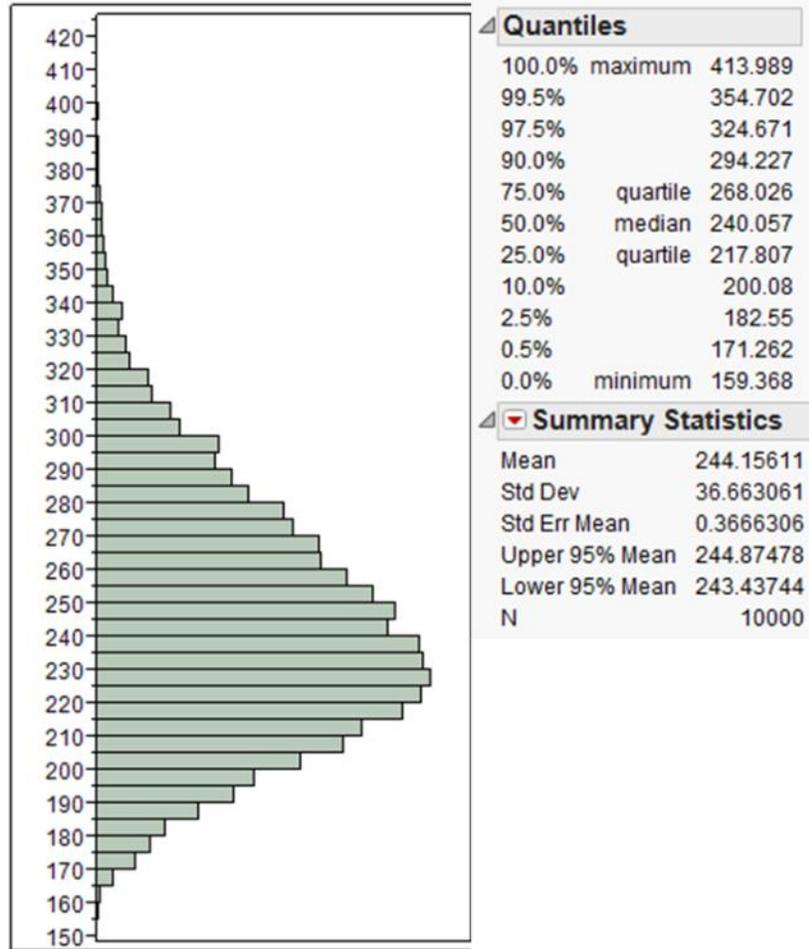


Figure 112. Three stage PMF result from simulation – case 2

Figure 112 shows the Probability Mass Function (PMF) and the summary for the simulation performed on the three stage gear system for case 2.

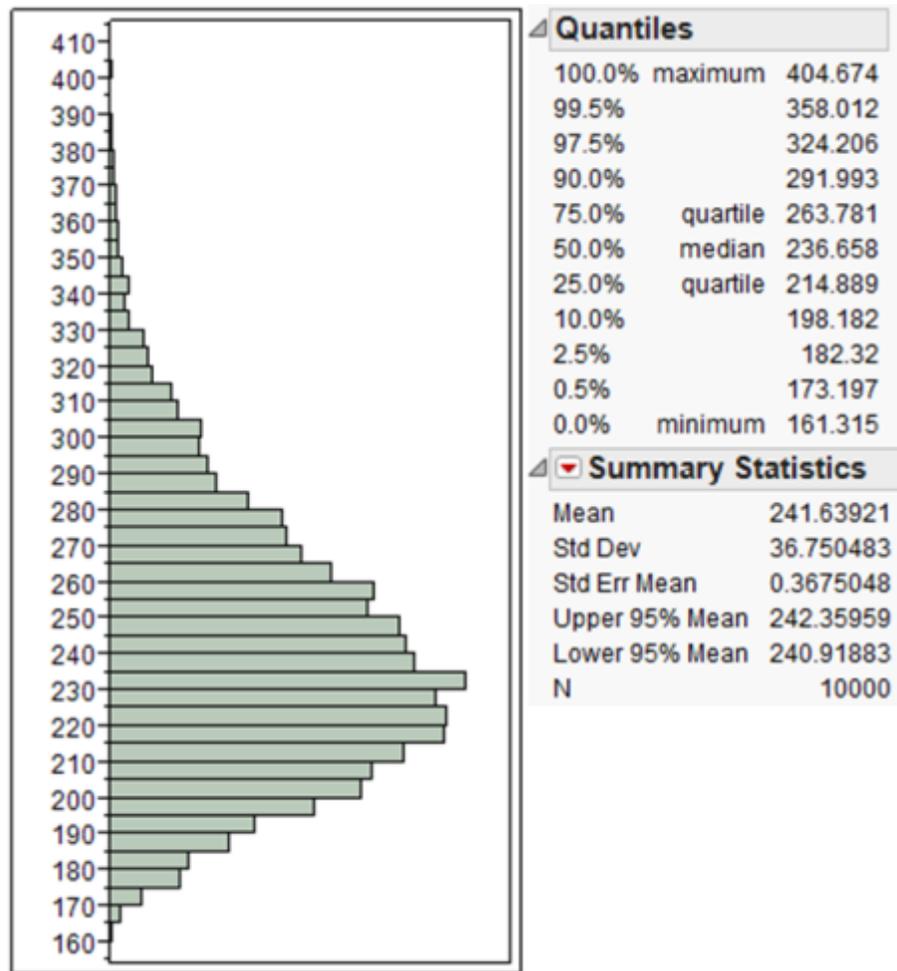


Figure 113. Four stage PMF result from simulation – case 2

Figure 113 shows the Probability Mass Function (PMF) and the summary for the simulation performed on the four stage gear system for case 2.

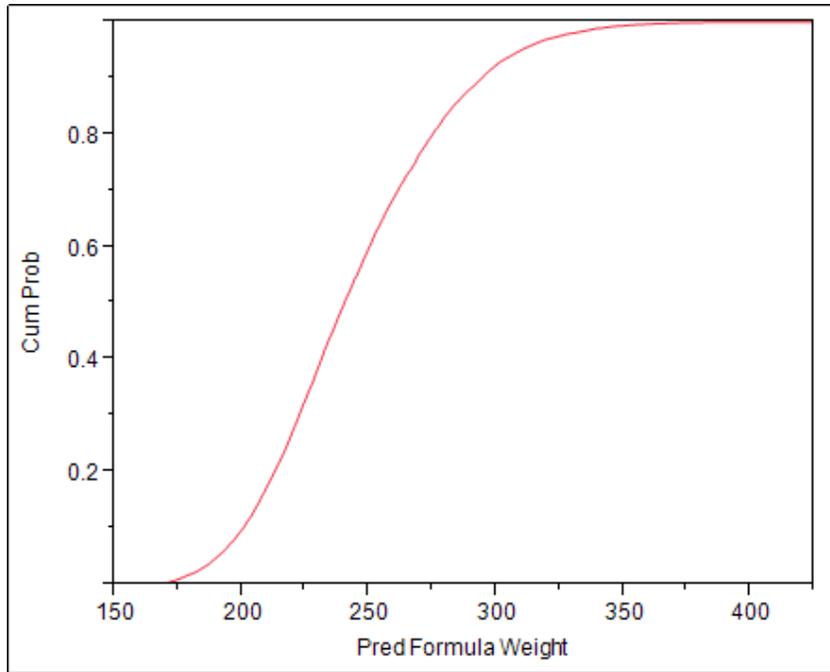


Figure 114. Three stage CDF result from simulation – case 2

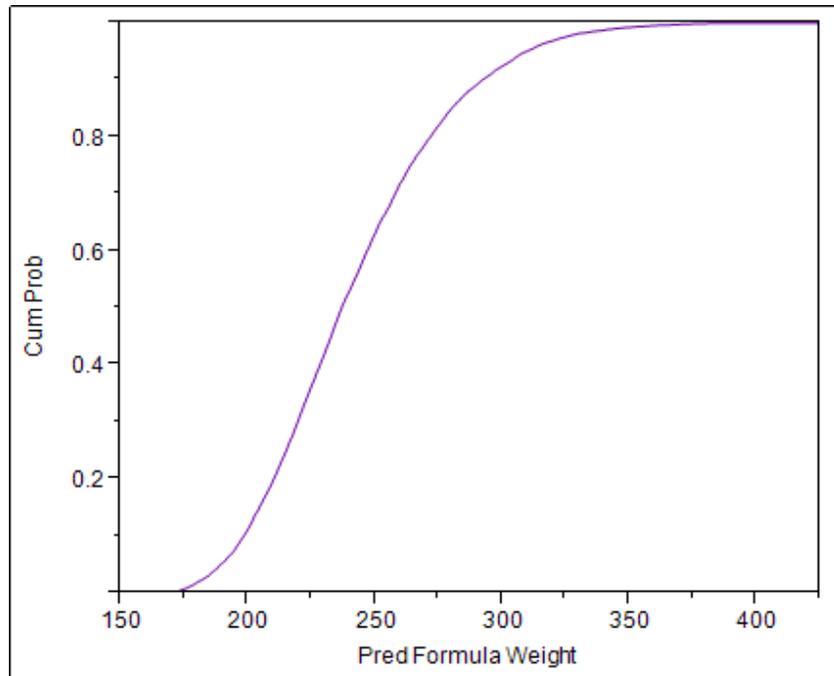


Figure 115. Four stage CDF result from simulation – case 2

## 5.5 *Finite Element Analysis*

Modern Finite Element Methods (FEM) can be used to evaluate three dimensional effects involving gears and is suitable for calculating load distribution. It is a more advanced technique than using rating functions as it is more application specific and can be used to study assembly and manufacturing deviations. FEA is used by industry in the detailed design stages to improve gear designs and apply more detailed load conditions. FEA is also widely used to perform multi-physics studies to understand the combined effects of structural and thermal loads. Vibration based studies are performed in the detailed stages to refine gear housing, bearings and sleeve designs [48].

The proposed study of FEA in this thesis is to study nonlinear contact formulations that can successfully model the gear contact and also study the effects of minor tooth modifications and their implied effects on contact and bending stresses.

FEA based topology optimization is proposed here as a method to study weight reductions through excess material removal in the web region. The background to topology optimization is discussed in Section 4.5 (page 125).

### 5.5.1 *Nonlinear Contact Analysis*

Modeling gear motion and contact requires explicit or transient nonlinear FEA. This process is illustrated in Figure 116, below. FEA preprocessing involves, obtaining geometry generated using a CAD tool, modeling contact, meshing, applying initial conditions for the transient analysis, applying boundary conditions and specifying the analysis settings. Post-processing is required to obtain the von-Mises nodal stress, eliminating hotspots or singularities arising from load application and segregating the nodal stress values in the contact and bending zones from the rest of the model. ANSYS is selected to perform the nonlinear contact analysis for the following reasons: ANSYS has a great user interface, and is extremely well-suited for nonlinear problems. ANSYS

also has a transient structural analysis module that can be used effectively for nonlinear analysis without performing a full explicit integration. The meshing tool is easily adaptable to automate mapped meshing for 3-d bodies as opposed to the standard tetra-mesh available in other tools. Post-processing in ANSYS can be easily automated by selecting the ‘faces’ of interest to find out maximum stress.

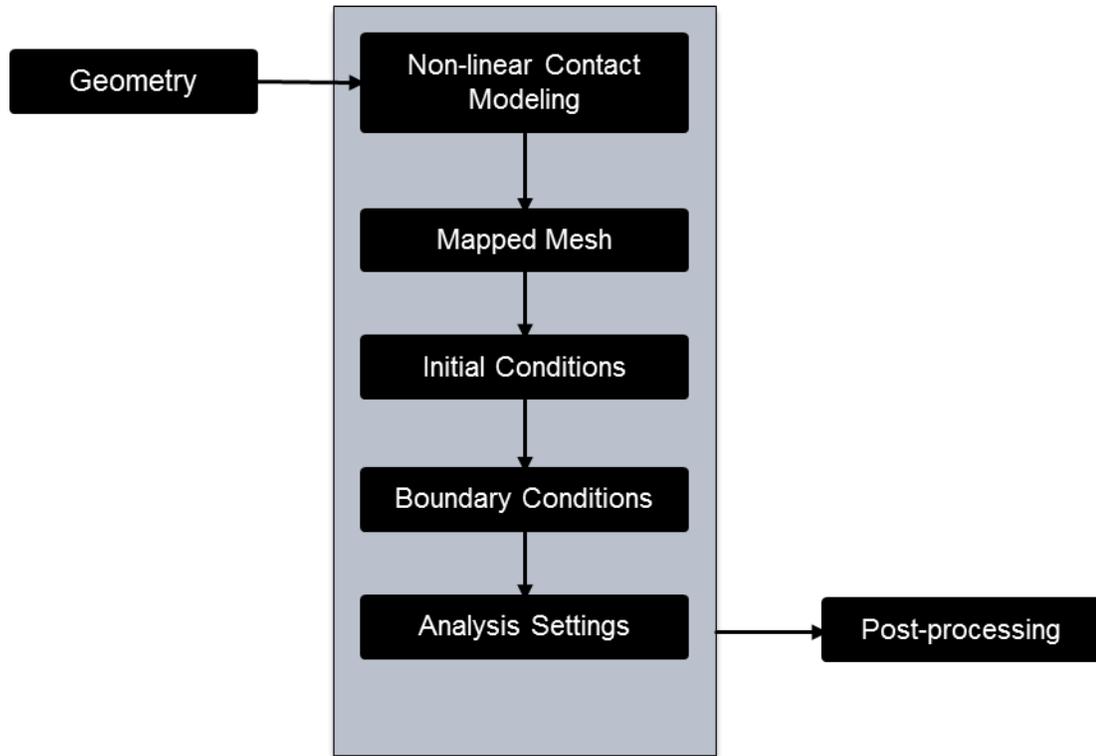


Figure 116. FEA process flow

### **Geometry**

FEA starts with generating the geometry. The geometry is generated in CATIA using mathematical relations as described in Section 5.1.1 (page 130). For the purpose of FEA, the geometry is modified to account for only the mating teeth i.e. 3 on both the pinion and gear, as shown in Figure 117. Also shown in Figure 117 is the transfer of geometry from CATIA to ANSYS.

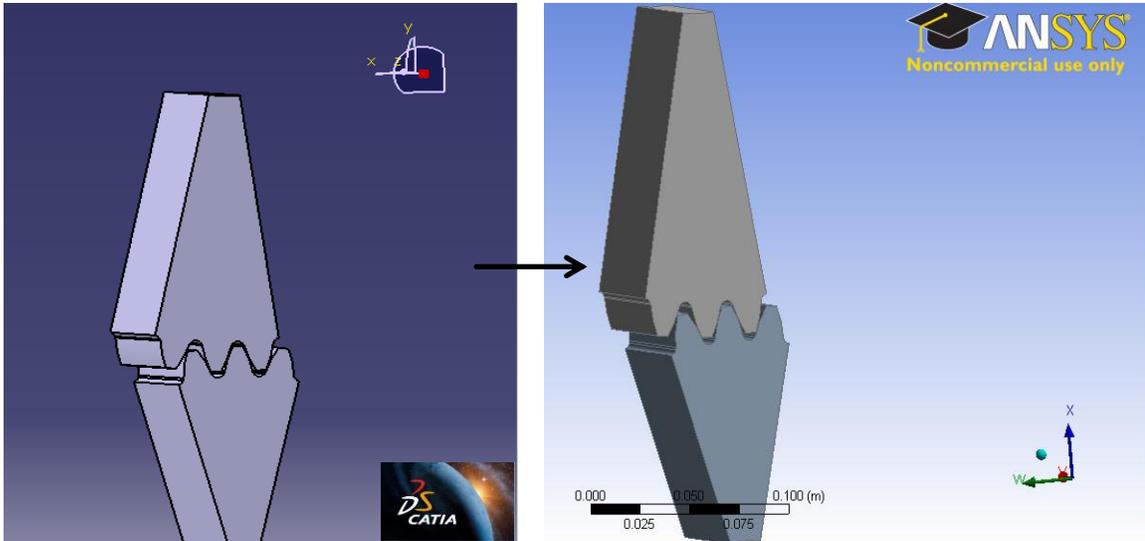


Figure 117. Gear geometry for FEA

### Meshing

Meshing, whether applied to FEA or CFD, has been referred to more as an art than a science [141]. It requires a lot of user interface and hence complete automation of high-fidelity FEA is difficult to achieve. Methods of auto-tetra meshing work well in certain cases, for developing rough estimates of the problem, but in complex cases, the auto mesh is usually non-ideal. Meshing requires strict sizing and shape enforcement in regions of load application, contact regions and regions where stresses are expected to be high. Mapped mesh can be applied to uniform geometries, and geometries that can be created by extruding or sweeping a 2-D surface. For gear geometry, mapped mesh is ideal whether applied to a spur gear or a helical gear. The size constraints get complicated for geometries such as bevel, spiral and face gears. Using mapped face meshing a more uniform mesh structure can be obtained.

The shape of the mesh generated using mapped meshing depends on the type of arrangement of vertex types on the face. The different types of vertices that are used to

control the behavior of the face mesh are shown in Figure 118. Table 10 gives a description of the vertex types and general rules of classification.

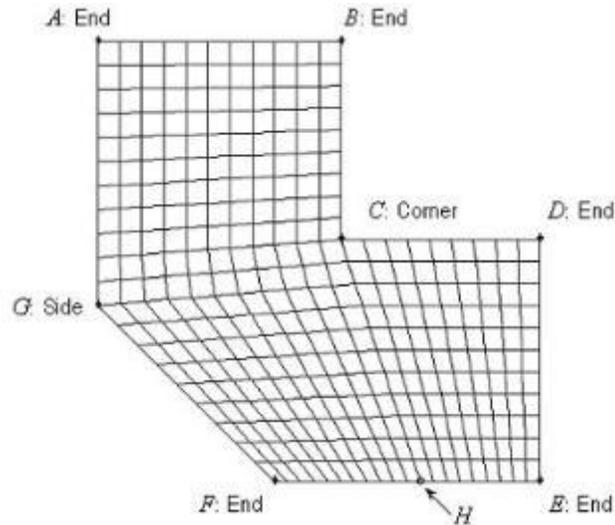
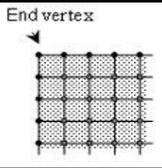
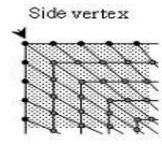
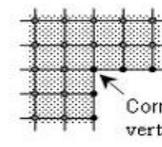


Figure 118. Vertex types [142]

In order map mesh, the face is divided into sub-mappable faces, as shown in Figure 119. The sub-maps are blended together using multizone mesh function to create quadrilateral face mesh and hexahedral 3-D mesh, shown in Figure 120. Size constraints can be applied on the tooth section to obtain mesh refinement in the region where contact is expected [142].

Table 10. Face-mapped mesh vertex types [143]

Vertex Type	Angle Range	Number of Elements Connected	Image
End	$0^\circ - 135^\circ$	1	
Side	$136^\circ - 224^\circ$	2	
Corner	$225^\circ - 314^\circ$	3	

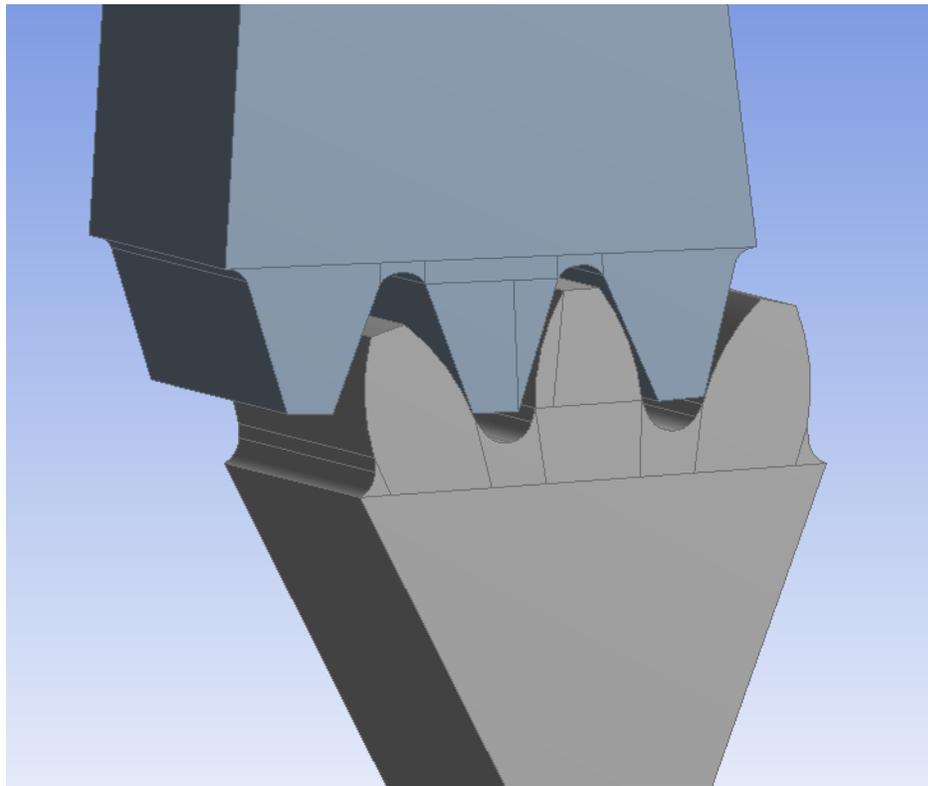


Figure 119. Face split into sub-mappable faces

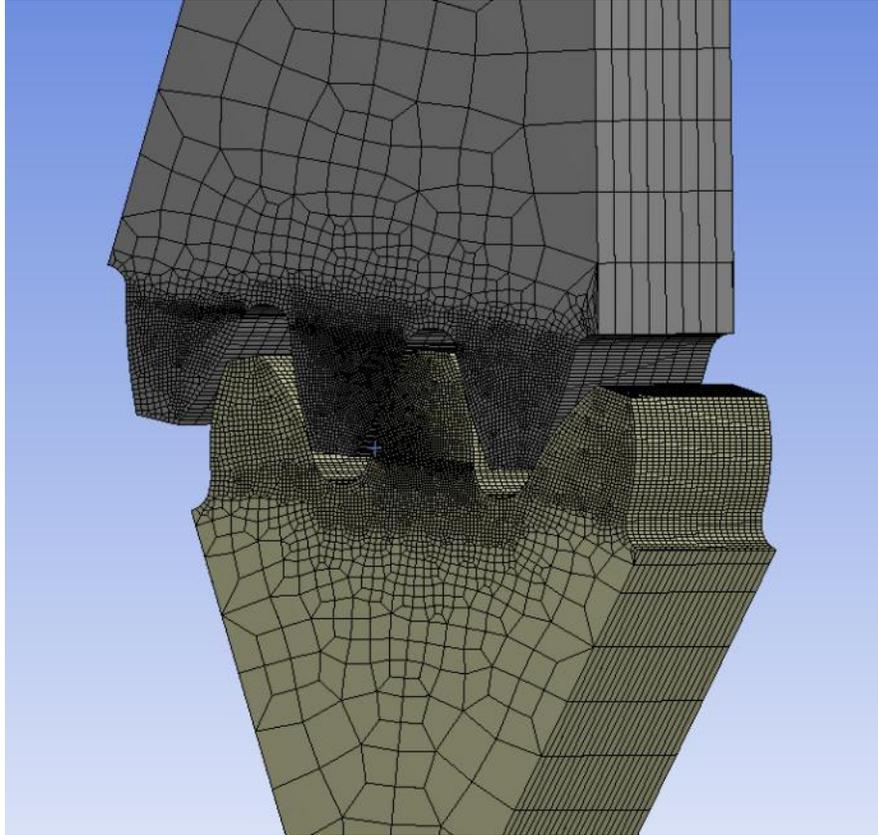


Figure 120. Mapped mesh

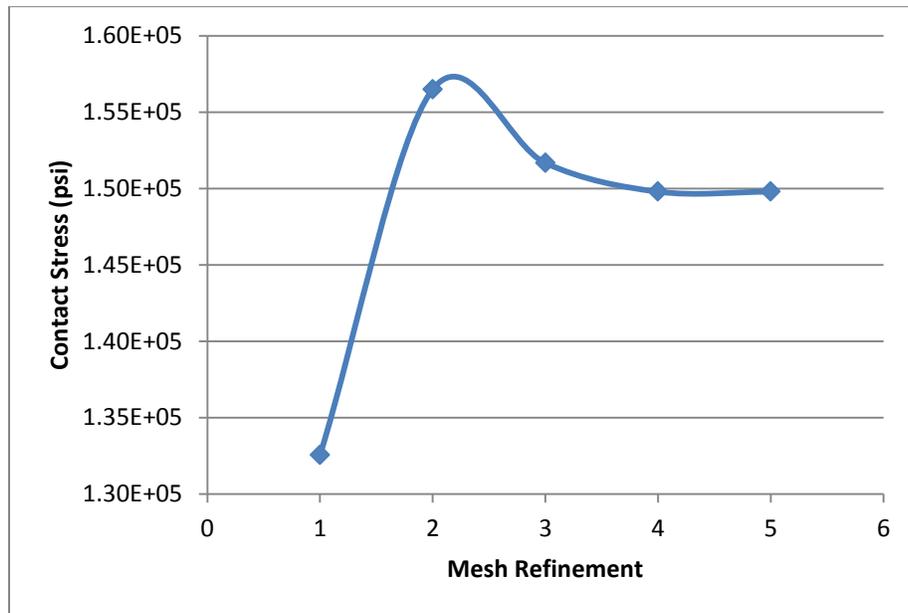


Figure 121. Mesh refinement for contact stress

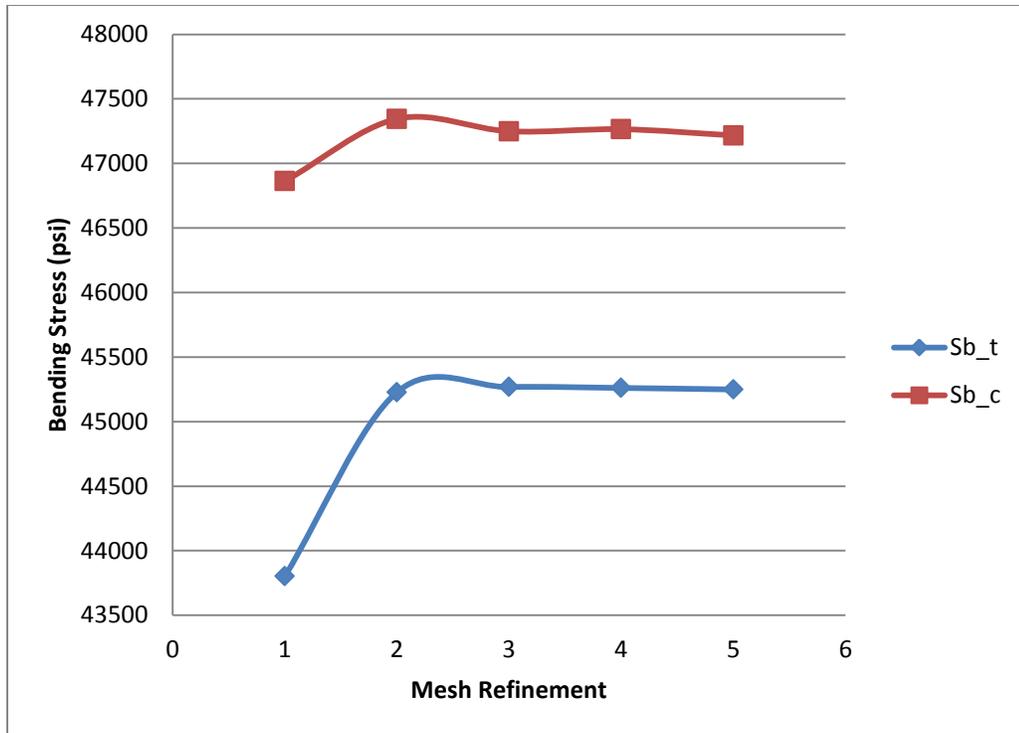


Figure 122. Mesh refinement for bending stress

The mesh is split into multiple zones and a mesh convergence study was performed. For accuracy, a very dense mesh is desired. However, a very dense mesh leads to high computational costs and results may not converge for nonlinear problems. So the right mesh density is preferred. Mesh refinement technique allows the user to refine the mesh up till the point where the percentage change in result to refinement is within tolerance. Here the mesh is refined till the contact stress change was less than 2% and the bending stress change was within 1%. To maintain a consistent mesh refinement level for the experiments, the mesh element size was normalized to the pitch diameter of the pinion. The contact elements are 0.175% of  $d_p$  and the root elements are 0.326% of  $d_p$ .

## **Nonlinear Contact Modeling**

ANSYS has the capability of performing nonlinear contact analysis. A nonlinear contact analysis models the deformation of the geometry as a function of time and iteration, to re-compute the stiffness matrix of the structure. For transient gear analysis, the following algorithm selections were studied to model the contact effectively.

To model the contact, the following functions are required to be setup.

1. Contact Type
2. Contact Formulation
3. Interface Treatment
4. Detection Method
5. Stabilization Damping, and
6. Time Step Controls

### **Contact Type**

Selecting the contact type is straightforward; ANSYS offers a frictional contact type where the user can specify a frictional coefficient. Here a frictional coefficient ( $\mu$ ) of 0.08 –to represent lubricated steel-to-steel contact– was selected.

### **Contact Formulation**

Contact formulation is the most important aspect of the contact modeling. This tells the ANSYS solver how to interpret the contact as it takes place. If no contact is modeled, penetration will take place.

ANSYS has the following four formulations:

1. Pure Penalty
2. Augmented Lagrange

3. MPC – Multi-Point Constraint contact
4. Normal Lagrange

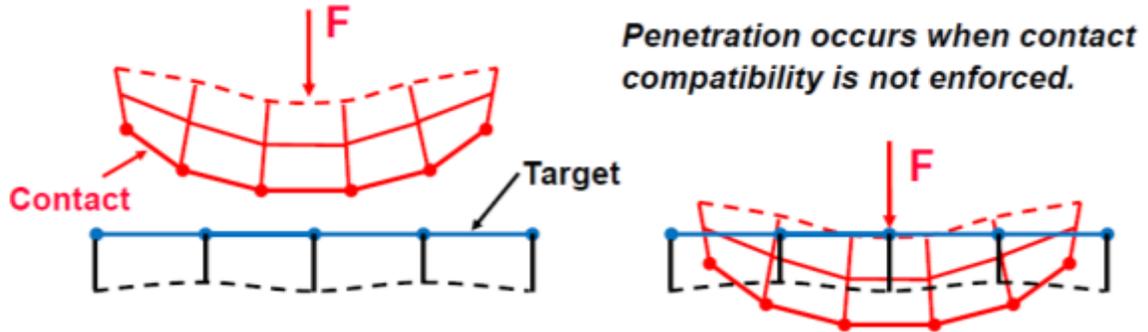


Figure 123. Contact modeling [144]

The Pure Penalty formulation is the default setting and is used to model nonlinear solid body contact of rigid bodies. The normal force exerted by the target on the contact face is calculated as:

$$F_{normal} = k_{normal} x_{penetration}$$

Equation 24. Pure penalty formulation

Where,  $k_{normal}$  is the contact stiffness.

The other penalty-based formulation is the Augmented Lagrange formulation where an extra term  $\lambda$  to the equation. This makes the penalty less sensitive to penetration. Compared to the Pure Penalty method, this method usually leads to better conditioning and is less sensitive to the magnitude of the contact stiffness coefficient. However, in some analyses, the Augmented Lagrange method may require additional iterations, especially if the deformed mesh becomes too distorted [144]

$$F_{normal} = k_{normal} x_{penetration} + \lambda$$

Equation 25. Augmented Lagrange formulation

The difference between Normal Lagrange and the other two penalty methods is shown in Figure 124.

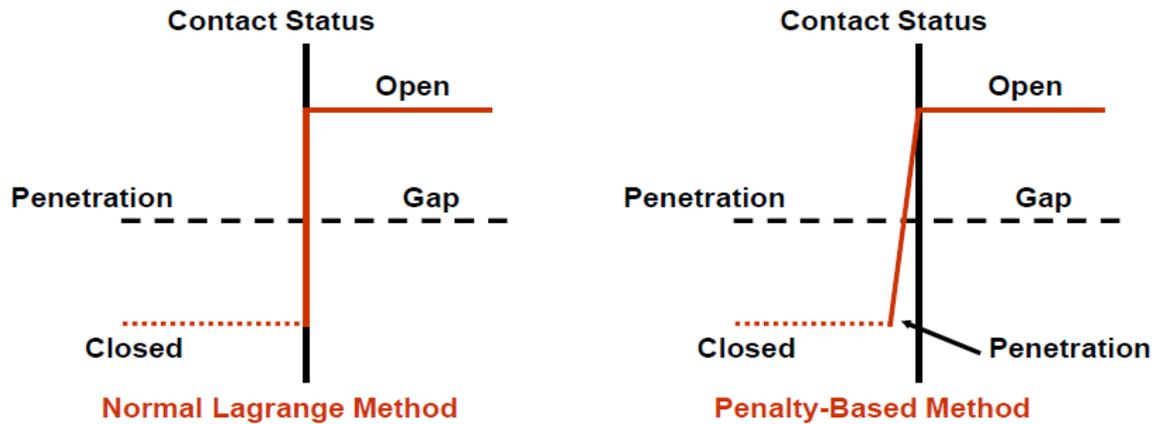


Figure 124. Normal Lagrange vs. penalty-based methods [144]

The Normal Lagrange formulation enforces zero penetration when contact is closed making use of a Lagrange multiplier on the normal direction and a penalty method in the tangential direction. Normal Stiffness is not applicable for this setting. Normal Lagrange adds contact traction to the model as additional degrees of freedom and requires additional iterations to stabilize contact conditions. It often increases the computational cost compared to the Augmented Lagrange setting.

A summary of the contact formulation types is given in Table 11.

Table 11. Summary of contact formulations [144]

Pure Penalty	Augmented Lagrange	Normal Lagrange	MPC
Good convergence behavior (few equilibrium iterations).	Additional equilibrium iterations needed if penetration is too large.	Additional equilibrium iterations if needed chattering is present.	Good convergence behavior (few equilibrium iterations).
Sensitive to selection of normal contact stiffness.	Less sensitive to selection of normal contact stiffness.	No normal contact stiffness is required.	
Contact penetration is present and uncontrolled.	Contact penetration is present but controlled to some degree.	Usually, penetration is near-zero.	No Penetration.
Useful for any type of contact behavior.			Only Bonded & No Separation behaviors.
Iterative or Direct Solvers can be used.		Only Direct Solver can be Used.	Iterative or Direct Solvers can be used.
Symmetric or Asymmetric contact available.		Asymmetric contact Only	
Contact detection at integration points.		Contact Detection at Nodes.	

Experiments were performed to study the effect of Augmented Lagrange vs. Normal Lagrange formulation. The Augmented Lagrange formulation being a penalty method enforces a very high normal force while allowing a small penetration. This decreases the contact pressure while increasing the bending load. Experiments showed the bending stresses to be twice as much as the analytical calculations; contact stresses were considerably low (~45% analytical value) because of the penetration.

Normal Lagrange formulation was able to model the contact appropriately without excessive bending loads. The contact pressure was modeled within desired tolerance and was selected as the best suited formulation for this contact problem.

### Detection Method

Detection Method allows the user to choose the location of contact detection used in the analysis in order to obtain a good convergence. ANSYS offers the following methods:

1. On Gauss Point
2. Nodal – Normal From Contact
3. Nodal – Normal To Target
4. Nodal – Projected Normal From Contact

ANSYS allows the use of Gauss integration points for detection while using penalty methods. In the ‘Nodal – Normal from Contact’ method, the contact detection location is on a nodal point where the contact normal is perpendicular to the contact surface and is on the target surface for ‘Nodal – Normal to Target’. The ‘Nodal – Projected Normal from Contact’ method is used for overlapping surfaces. It is desirable in the contact case of gears to have the node detection on the target element to model the motion while modeling the contact as a line. For these reasons, ‘Nodal – Normal to Target’ was used as the detection method.

### **Interface Treatment**

When nonlinear contact analysis is being performed, gaps develop between the contact surfaces. The methods used by ANSYS to cope with this are:

1. Adjust-to-Touch
2. Add Offset, Ramped Effects
3. Add Offset, No Ramping

In the ‘Adjust to Touch’ formulation any initial gaps are closed and any initial penetration is ignored creating an initial stress free state. Contact pairs are “just touching” as shown in Figure 125. This setting helps the user ensure that initial contact occurs even if any gaps are present. Without using this setting, the bodies may fly apart if any initial gaps exist. Although any initial gaps are ignored, gaps can still form during loading for the nonlinear contact types. The other two interface treatment methods are not compatible with the gear contact problem.

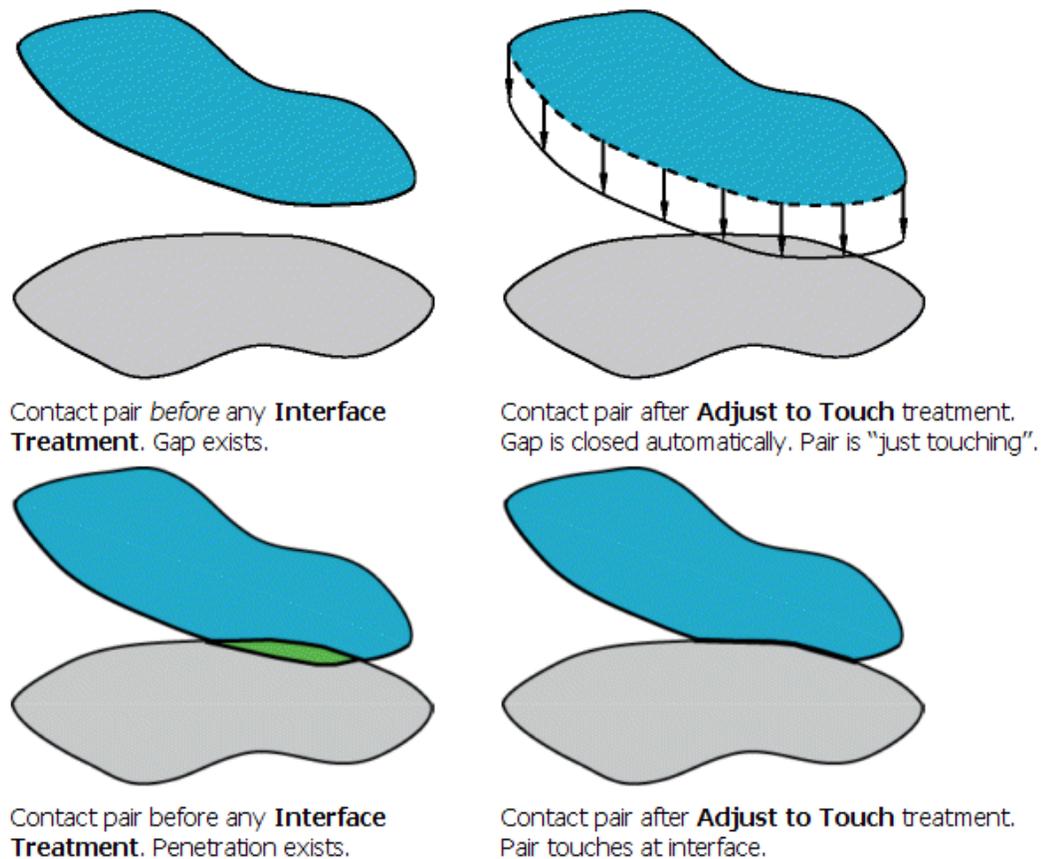


Figure 125. Adjust-to-touch formulation [144]

### Stabilization Damping Factor

The contact defined by the user may initially have a near open status due to small gaps, because of discretization, between the element meshes or between the integration points of the contact and target elements. As a consequence, the solver may not detect the contact during the analysis. This can cause a rigid body motion of the bodies defined in the contact. The stabilization damping factor provides a certain resistance to damp the relative motion between the contacting surfaces and prevents rigid body motion. This contact damping factor is applied in the contact normal direction and the damping is applied to each load step where the contact status is open. The value of the stabilization damping factor is required to be large enough to prevent rigid body motion yet be small enough to ensure convergence. The stabilization damping factor was varied between four

values of 0, 0.5, 1 and 1.5. When the value was set to zero, there were large rigid body motions. Higher values of 1 and 1.5 led to increased convergence requirements. A value of 0.5 was finally selected as an adequate damping factor to ensure fast convergence while preventing large rigid body motion.

### **Time Step Controls**

For nonlinear contact problems, ANSYS allows the user to specify how the time steps should be calculated. In the default mode, changes in contact behavior do not affect the time stepping. Two methods available that allow contact behavior to control the time stepping are:

1. Automatic Bisection
2. Predict for Impact

In the Automatic Bisection mode, contact behavior is reviewed at the end of each substep to determine whether excessive penetration or drastic changes in contact status have occurred, and the substep is reevaluated using a time increment that is halved (bisected). In the Predict for Impact mode, the formulation also predicts the minimal time increment needed to detect changes in contact behavior, and this option is recommended if impact is anticipated in the analysis. Since no impact is anticipated, in the case of gear contact, Automatic Bisection formulation is used for Time Step Controls.

A summary of the nonlinear contact modeling for the spur gear FEA is given in Table 12.

Table 12. ANSYS contact analysis setting

Function	Algorithm Selection
Contact type	Frictional - $\mu = 0.08$
Formulation	Normal Lagrange
Interface treatment	Adjust to touch
Detection	Normal – Normal to Target
Stabilization damping	0.5
Time step controls	Automatic bisection

Figure 126 shows the pinion and gear contact model. The pinion teeth are modeled as ‘contact’ body (red) and the gear teeth as ‘target’ body (blue). Each tooth contact model is controlled separately using 3 contact connections.

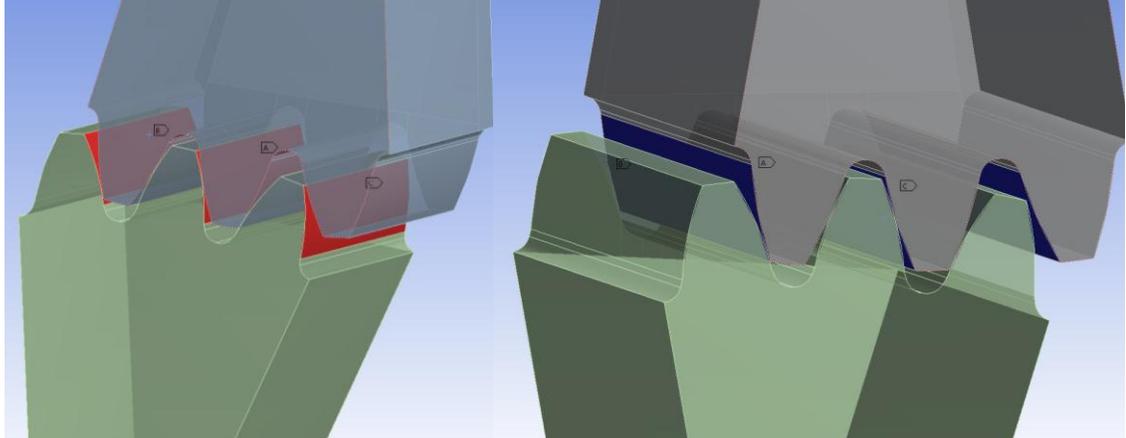


Figure 126. Pinion – gear frictional contact

## **Initial and Boundary Conditions**

A constant rotational velocity is applied to the pinion gear at the axis of rotation. A cylindrical support is applied to the surface of rotation of the gear and pinned to the axis using rigid elements. The cylindrical support is specified with free tangential motion, to allow rotation but is fixed in radial and axial direction. The pinion rotates about the cylindrical support by a prescribed angle while the gear is held stationary using a rigid cylindrical support. The torque is prescribed at the center of the pinion about the axis of rotation.

## **Results**

Four cases were selected to test the FEA formulation: these cases are shown in Table 13. *Case 1* is a low gear ratio, high torque combination. *Case 2* is a high gear ratio, low torque combination. *Case 3* is a scaled version of *Case 2* but the face width and torque are held constant; these two cases provide a good sampling to test the formulation consistency. *Case 4* is a combination with a slightly higher gear ratio for the same input torque as *1* and *2* but is a combination with a smaller face width. These cases were selected to represent a wide range of gears as well as to test consistency when small changes to the geometry and load were made. Results and their comparison with AGMA predicted values are shown in Table 13 (Figure 128, Figure 129 and Figure 130).

Table 13. FEA cases for ANSYS

Parameter	Unit	Case			
		1	2	3	4
Power	HP	2500	700	700	700
rpm	rpm	2788	3500	3500	3500
Q (lb_in)	lb-in	56514.85	12605.07	12605.07	12605.07
Pd	in-1	3.6	5	4.5	4.5
Np	-	39	23	23	25
Ng	-	55	104	104	63
mg	-	1.410	4.522	4.522	2.520
F	in	3.5	2.4	2.4	2.22
AGMA Pinion Bending Stress	psi	37323	48220	39046	38060
AGMA Contact Stress	psi	145560	174690	157230	157590
FEA - Pinion bending stress	psi	37807	48330	39274	35706
FEA - Contact Stress	psi	141000	166670	151670	164760
Bending stress variation	-	1.30%	0.23%	0.58%	6.18%
Contact stress variation	-	3.13%	4.59%	3.54%	4.55%



Figure 127. von-Mises stress contours for transient analysis

All cases resulted in satisfactory results with low bending stress variation (except Case 4) and contact stress variation < 5%. The discrepancy in Case 4 could possibly be

arising from either AGMA over predicting the bending stress for smaller face width gears or the FEA model under predicting the bending stress.

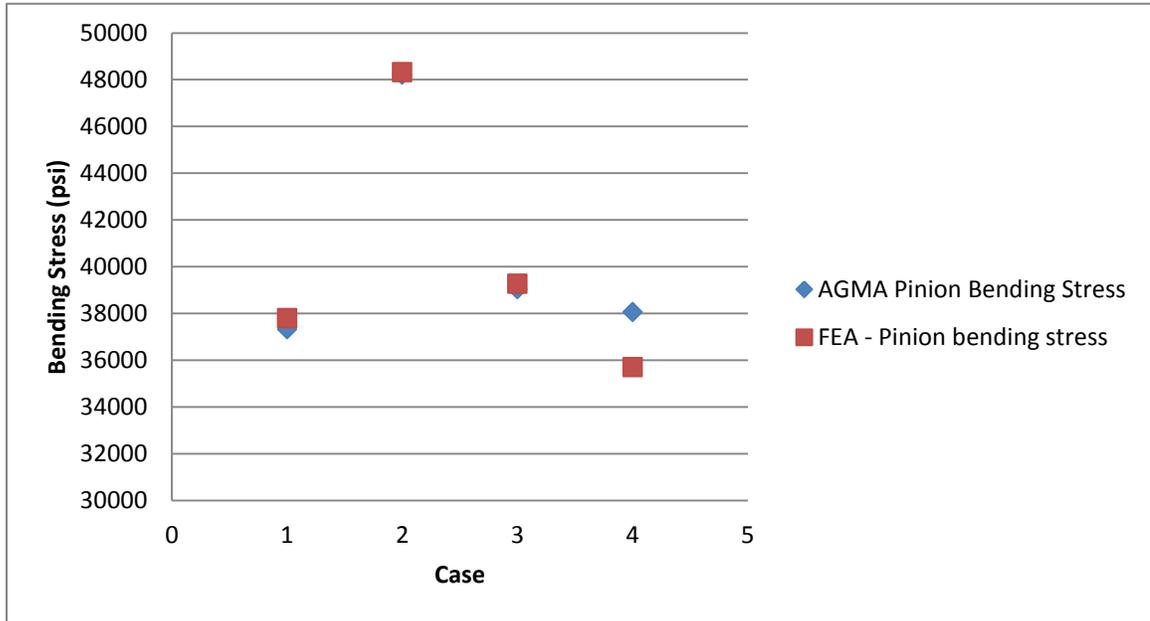


Figure 128. Pinion bending stress - AGMA vs. FEA

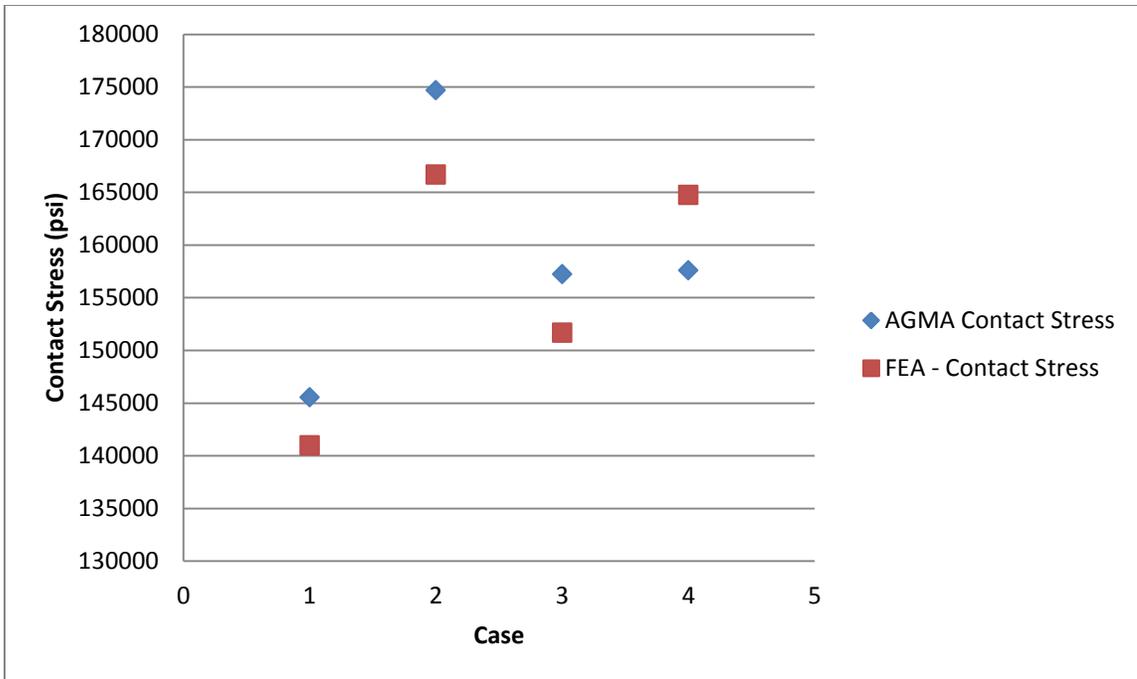


Figure 129. Contact stress – AGMA vs. FEA

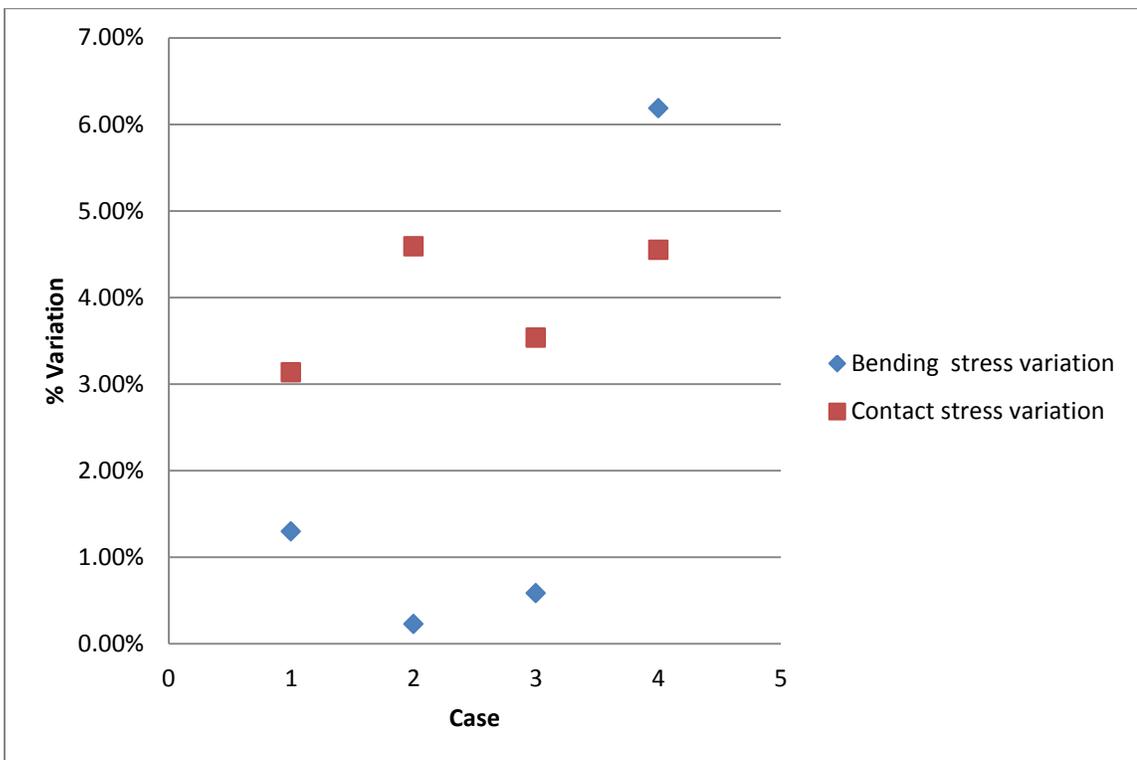


Figure 130. Percentage variation between AGMA and FEA

Another set of experiments were performed to test consistency of the FEA model on *Case 1* by varying the edge generating radius and addendum ratio. The edge generating radius alters the root fillet radius. The addendum ratio changes the ratio of the height of the pinion tooth with respect to the gear tooth. The ranges for these variables are determined to avoid undercut. The tooth thickness is not altered so the backlash provision remains constant. The results of these experiments are shown in Figure 131, Figure 132 and Figure 133. The contours show higher bending stresses for a higher edge generating radius and for low pinion addendum as expected from bench test studies [64]. A moderate increase in addendum ratio and a slight increase in edge generating radius offer improvements in bending stress on the tensile and the compressive side.

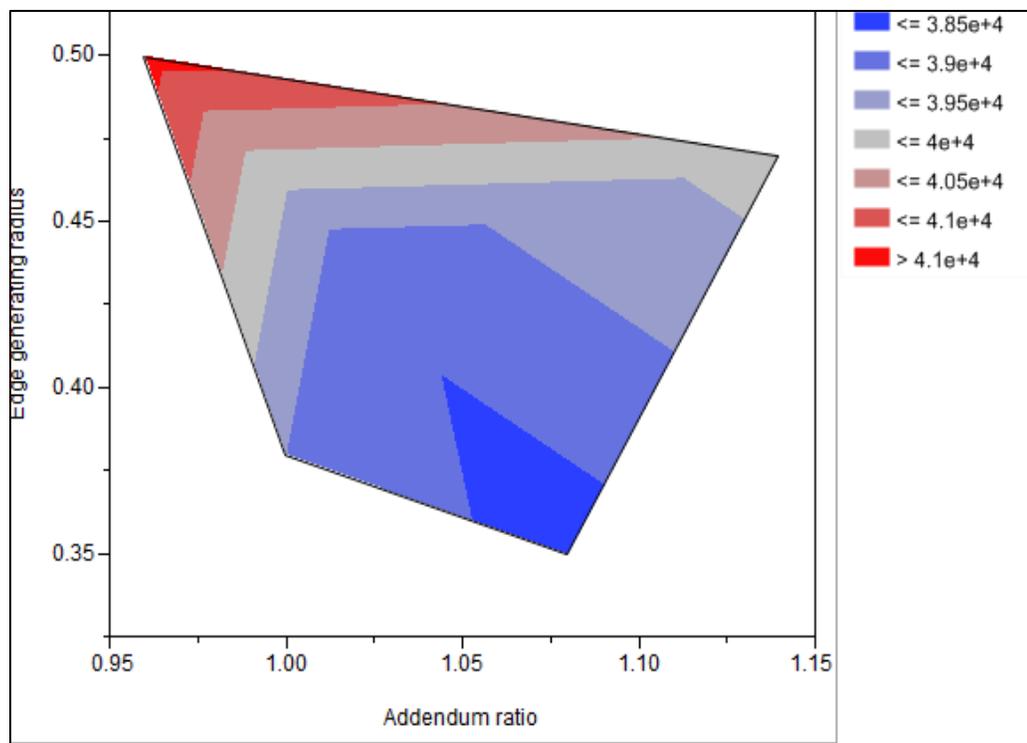


Figure 131. Contour plot for pinion bending stress - tensile

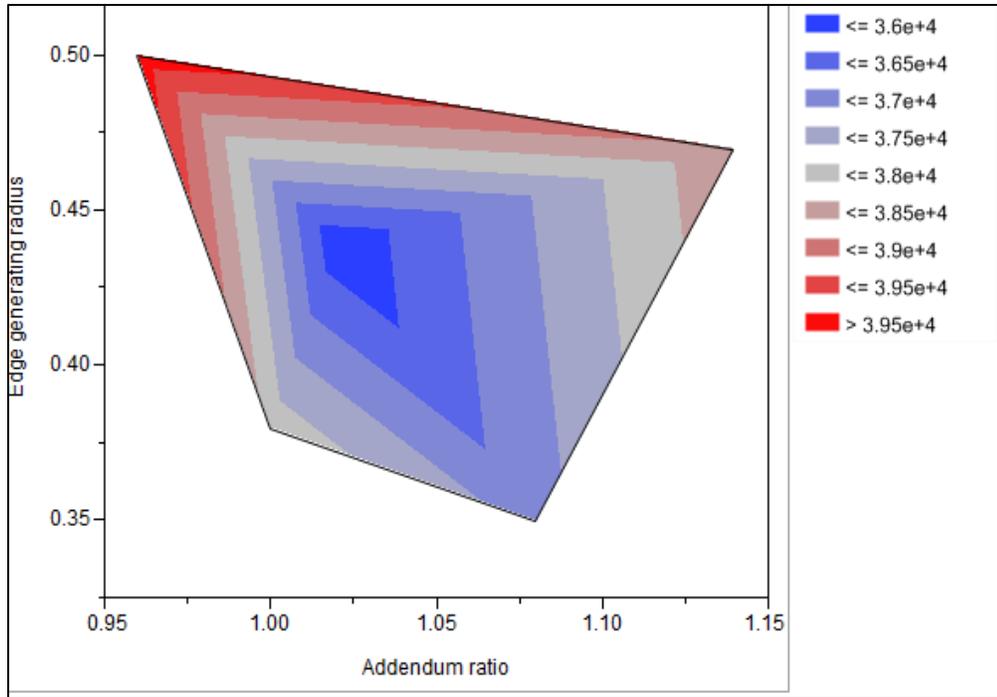


Figure 132. Contour plot for pinion bending stress - compression

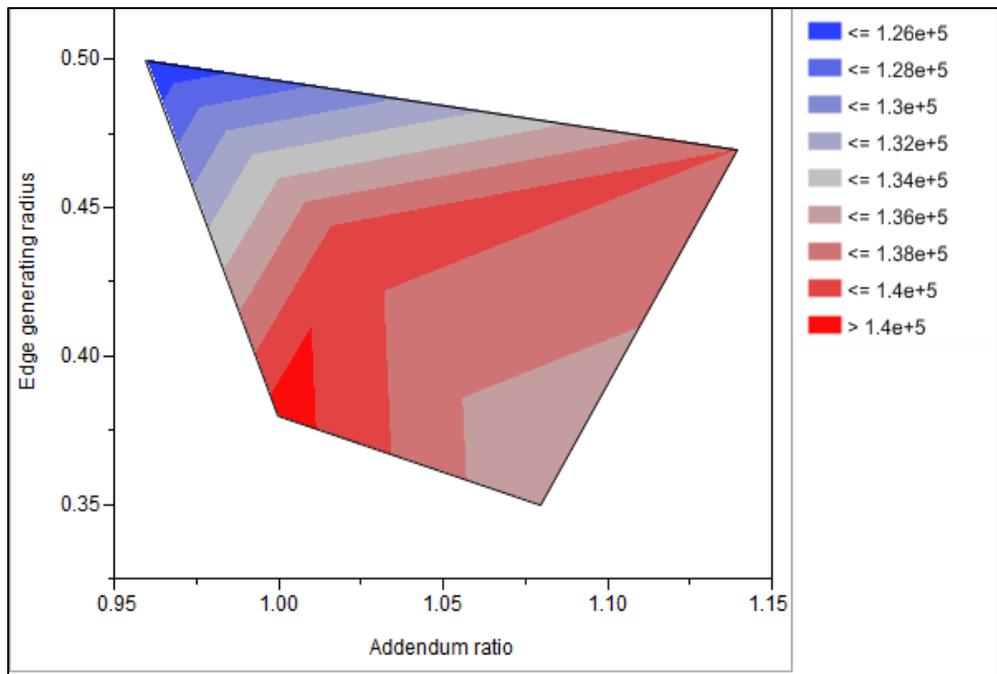


Figure 133. Contour plot for contact stress

### 5.5.2 Topology Optimization

The objective of using topology optimization and the methodology are presented in Section 4.5 (page 125). The implementation and testing of *Hypothesis #2* are presented here.

OptiStruct and HyperMesh are selected to perform the study of topology optimization for being the state-of-the-art commercial software program in this class of FEA and for its availability.

The gear geometry is built in CATIA and imported to HyperMesh. Topology optimization problem is setup in HyperMesh and the optimization is performed using OptiStruct. The gear model is divided in to multiple sub-regions. Each region is meshed separately through solid mapping of the 2-D face mesh. ‘Edge deviation’ algorithm was used to auto-mesh the face for the circular shaft-hole area and design region. The gear teeth area (face) was meshed using the ‘QI optimize’ algorithm with a target element size of 0.06 in (1% of  $d_p$ ) [145].

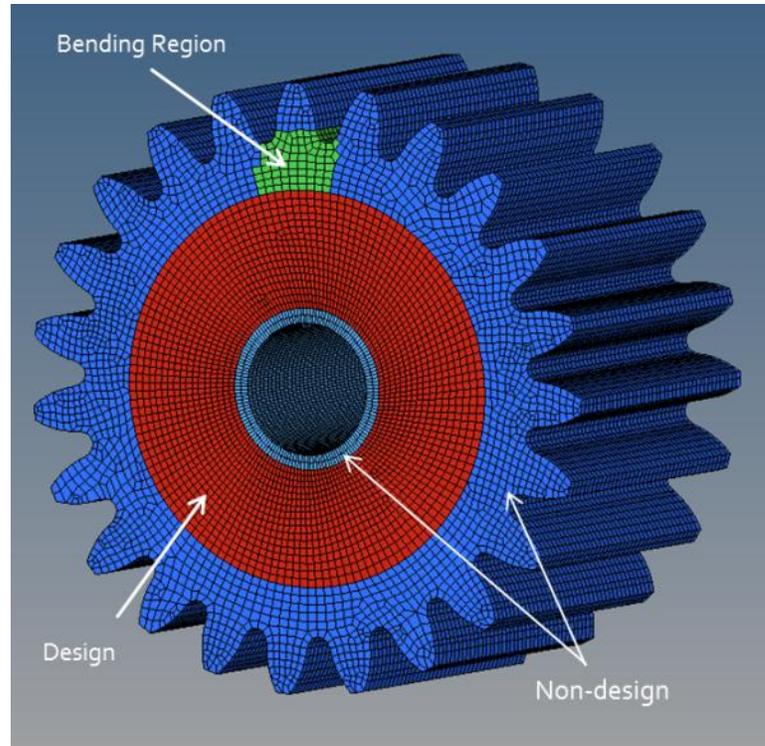


Figure 134. HyperMesh model of gear

The optimization problem for this operation is presented as follows:

$$\text{Objective: } \min : F(x) = \text{Volume}_{\text{design}}$$

$$\text{Subject to: } x \in X = \{x \in R^n \mid g_i(x) \leq 0, \quad i = 1, \dots, m\}$$

Where,

$$g_1(x) = (f \times \sigma_{\text{design}}) - \sigma^*$$

$$g_2(x) = \sigma_b - \sigma_b^*$$

$$g_3(x) \rightarrow \text{manufacturing constraint (draw)}$$

$$g_4(x) \rightarrow \text{manufacturing constraint (cyclic and symmetry)}$$

The objective is defined as design region volume, i.e. the region where the volume is to be minimized. Stress and manufacturing constraints are enforced to ensure that a reasonable factor of safety ( $f$ ) was maintained, and the overall results conformed to manufacturing processes and also gear design requirements. The draw constraint ensures

that cavities, that can't be manufactured, aren't created (such as the ones shown in Figure 135). The cyclic and symmetry constraint ensure that the topology is uniform, since only one tooth is loaded, and symmetrical on both sides. When fewer cyclic instances are imposed, irregular topology results are obtained, as shown in Figure 136 (24 teeth, 24 instances). This issue can be almost completely eliminated if the cyclic instances required are increased, as shown in Figure 137 (24 teeth, 96 instances – the number 96 is chosen as a multiple of the number of teeth). The problem, as setup in OptiStruct is given below:

Optimization Responses:

- Volume – Design region
- $\sigma_b$  - Tooth bending stress

Optimization Constraint:

- $\sigma_b < 35\text{ksi}$

Optimization Objective:

- Minimize volume (design region)

Design Variable Constraints:

- $\sigma_{\text{design}} < 20\text{ksi}$
- Cyclic constraint (number of instances = number of teeth)
- Split draw constraint

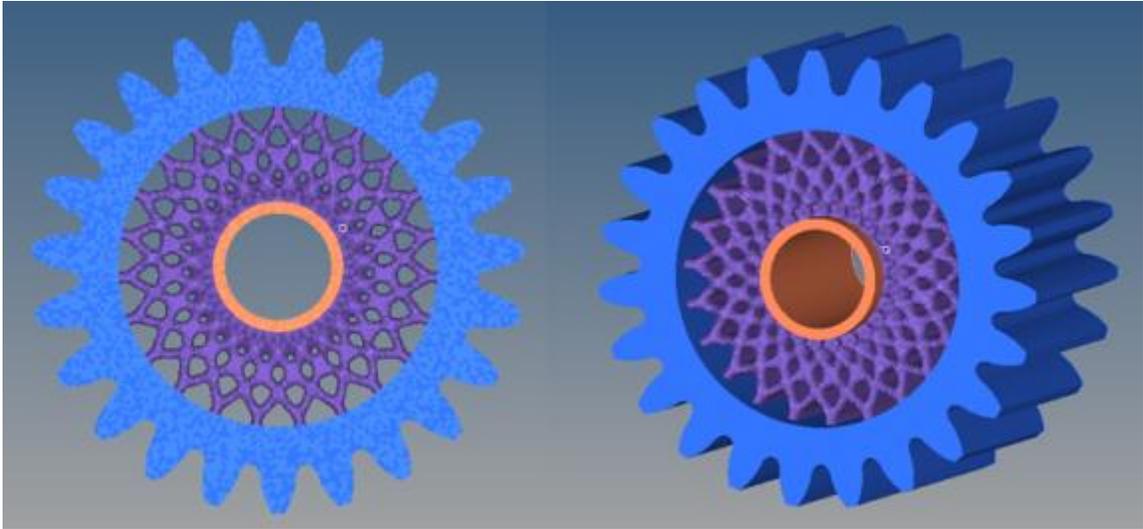


Figure 135. OptiStruct result without draw constraint

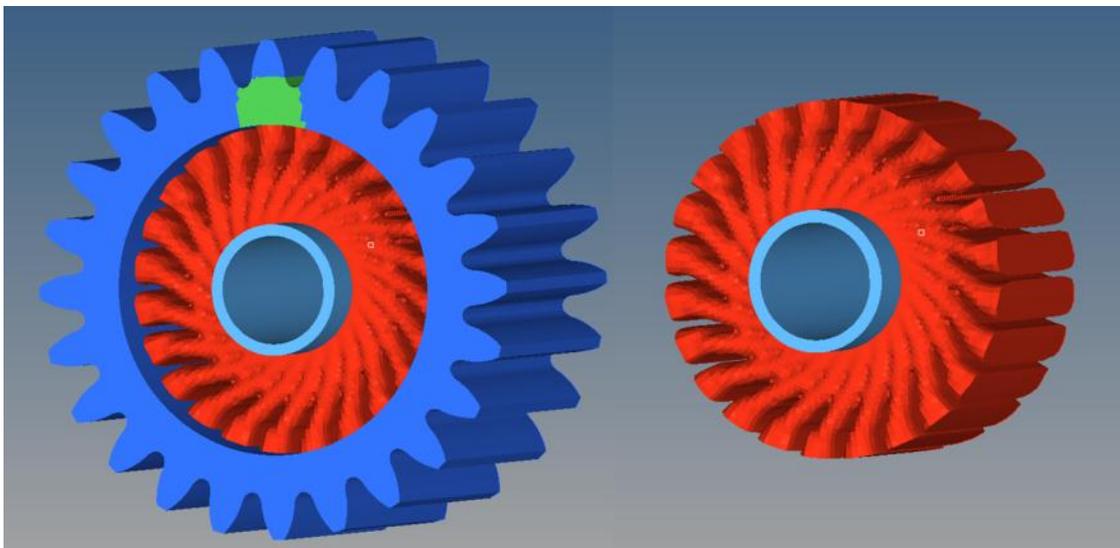


Figure 136. OptiStruct result few cyclic instances

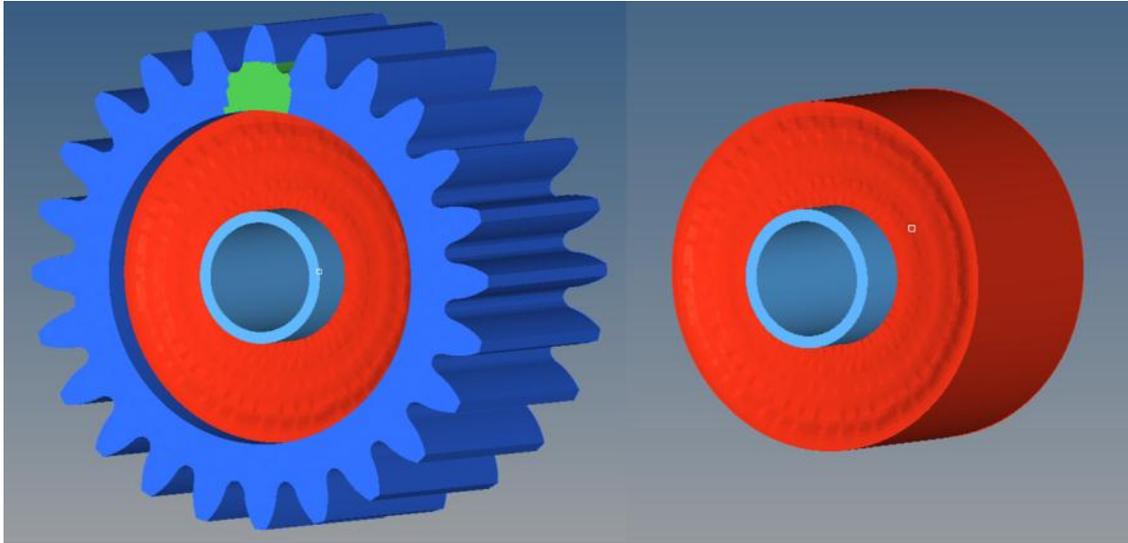


Figure 137. OptiStruct result with higher level cyclic constraint

To test Hypothesis #2, which states: *Information obtained from topology optimization does not influence the preliminary design decision*, two experiments are presented here.

The test for the first case is given in Table 14, and the problem represented graphically in Figure 138. Here *design A* is adequately sized for the torque and is on the bending stress limit of 35000 psi, while *design B* is a slightly oversized gear – it weighs more and can carry a higher torque and has slight margin between the allowable stress limit. The analytical sizing model and optimization routine would ideally return design A as the optimal design. However, the test here is to see if topology optimization alters the decision because it is possible to remove more material from the web region in an oversized gear to offset the original excess weight i.e. post topology optimization, *design B* weighs less than *design A*.

Table 14. Topology optimization - Case 1

Parameter	Symbol	Unit	Design A	Design B
Power	P	HP	700	
RPM	-	RPM	5000	
Gear ratio	$m_g$	-	3.3	
Diametral pitch	$P_d$	in <sup>-1</sup>	6	
Face width	F	in	2.4	
Pressure angle	$\phi$	deg	20	
Material	-	-	AISI 9310	
Number of teeth	$N^p$	-	23	24
Tangential load	$P_t$	lbf	4616.7	4424.4
Radial load	$P_r$	lbf	1680.4	1610.3
Bending Stress (AGMA)	$\sigma_b$	psi	35000	32951
Weight	W	lb	7.233	7.885

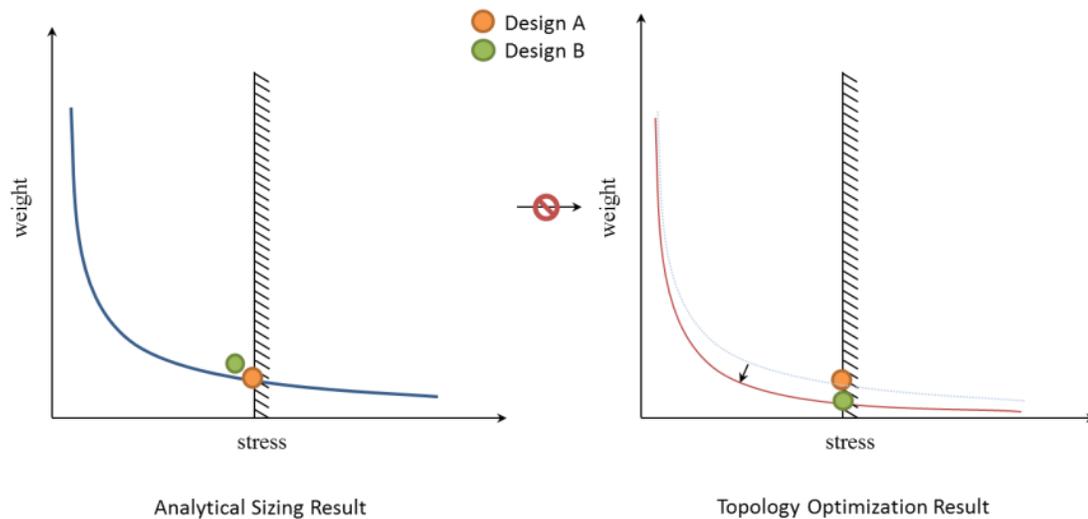


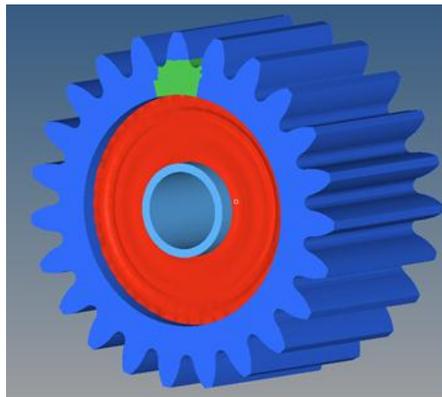
Figure 138. Topology optimization - Case 1

The results from this test are presented in Table 15. 18% weight reduction was obtained in the web region in *design A* while 26% was obtained for *design B*. The total

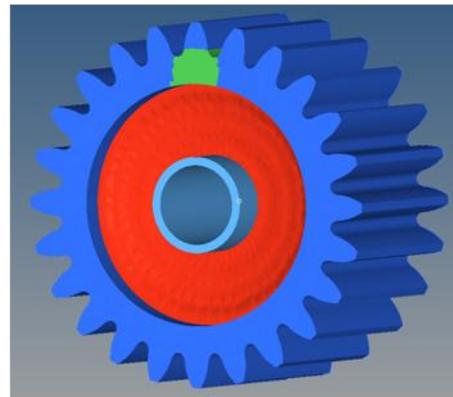
weight reduction of the gear for *design A* is 8% while that for *design B* is 12%. However, the total weight of *design A* was still lower than *design B*, substantiating the hypothesis.

Table 15. Topology optimization result - Case 1

Weight (lbs)	Design A			Design B		
	Before	After	% Difference	Before	After	% Difference
Gear Teeth	3.777	3.777	-	3.972	3.972	-
Web	3.243	2.674	18%	3.691	2.731	26%
Shaft	0.213	0.213	-	0.222	0.222	-
Total	7.233	6.664	8%	7.885	6.925	12%



Design A



Design B

Figure 139. Topology optimization result - Case 1

The experiment for the second case is given in Table 16, and the problem represented graphically in Figure 140. Here *design A* and *B* have different diametral pitch and different face width and *A* weighs less than *B*. The higher diametral pitch and more number of teeth on *B* mean that web region is a lot larger – percentage wise – than that *A*; this is shown in Table 17. AGMA recommends a rim height to tooth ratio to be not less than 1.2. Therefore, for *design B*, 57% of the overall weight is in the design region and

for *design A*, it is 48%. The test here is to see if topology optimization alters the overall weights to an extent that makes *design B* a better design in comparison to *A*.

Table 16. Topology optimization - Case 2

Parameter	Symbol	Unit	Design A	Design B
Power	P	HP	500	
RPM	-	RPM	7000	
Gear ratio	$m_g$	-	4.5	
Pressure angle	$\phi$	deg	20	
Material	-	-	AISI 9310	
Diametral pitch	$P_d$	$\text{in}^{-1}$	7	8
Face width	F	in	1.714	1.5
Number of teeth	$N^p$	-	25	31
Tangential load	$P_t$	lbf	2368.98	2183.39
Radial load	$P_r$	lbf	862.24	794.69
Bending Stress (AGMA)	$\sigma_b$	psi	29823	32300
Weight	W	lb	4.494	4.651

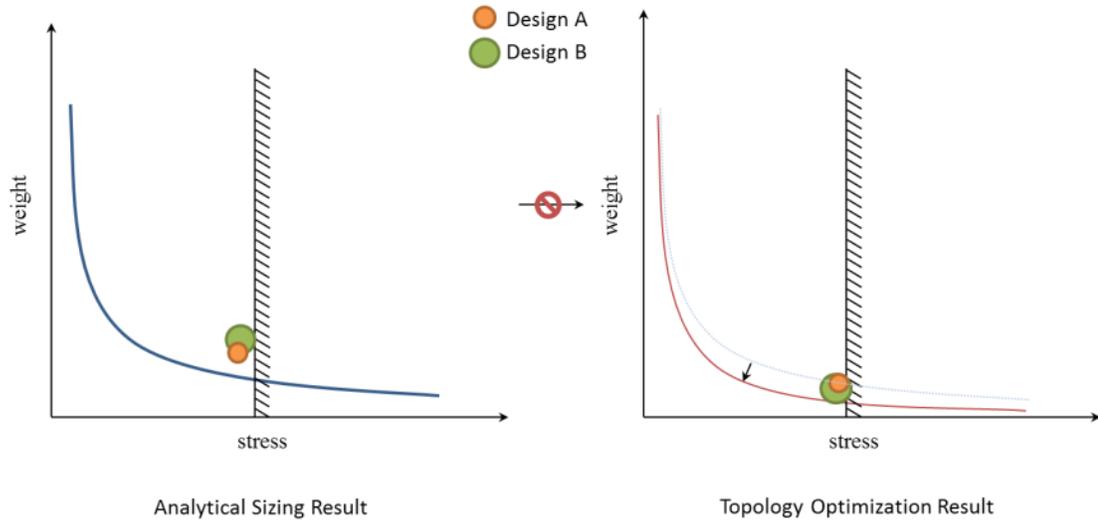


Figure 140. Topology optimization - Case 2

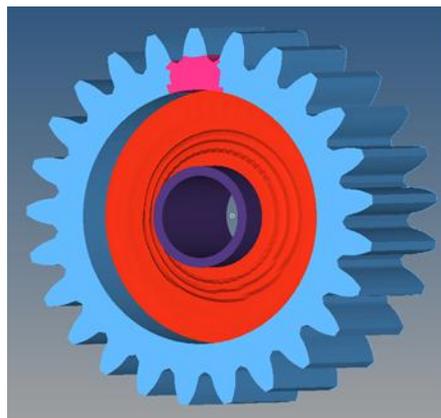
Table 17. Weight distribution in different regions – Case 2

Parameter	Unit	Design A	%	Design B	%
Teeth	lb	2.187	49%	1.876	40%
Design	lb	2.164	48%	2.64	57%
Shaft hole	lb	0.143	3%	0.135	3%

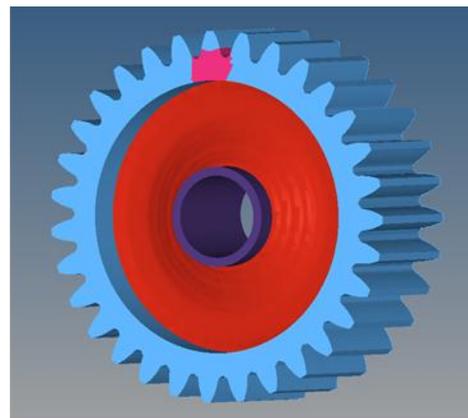
The results of topology optimization study for this case are shown in Table 18 and Figure 141. Although the percentage of material removed is greater in *A* than *B* (because *B* is closer to the stress limits), the overall weight reduction in *B* is greater than in *A*. However, the total weight of *A* is still lower than that of *B* and this process does not alter the dominance of *design A* over *B*, further substantiating the Hypothesis.

Table 18. Topology optimization results – Case 2

Weight (lbs)	Design A			Design B		
	Before	After	% Difference	Before	After	% Difference
Gear Teeth	2.187	2.187	-	1.876	1.876	-
Web	2.164	0.995	54%	2.64	1.35	49%
Shaft	0.143	0.143	-	0.135	0.135	-
Total	4.494	3.325	26%	4.651	3.361	28%



Design A



Design B

Figure 141. Topology optimization results - Case 2

## 5.6 Implementation of Methodology on Rotorcraft Drive System

The design framework proposed in this thesis falls in line with the generic IPPD methodology shown in Figure 13. The IPPD methodology is a suitable framework for design of complex aerospace systems. It has been used successfully as a guideline for transfer of information within the conceptual and preliminary design process.

The fully-relational design methodology for the drive system is shown in Figure 142. As was tested in the three-stage gear system (Section 5.1), the geometrical requirements and pertinent design parameters are linked with the drive system design and analysis block. The drive system analysis block is completely automated and geometrical relations are maintained through relational design.

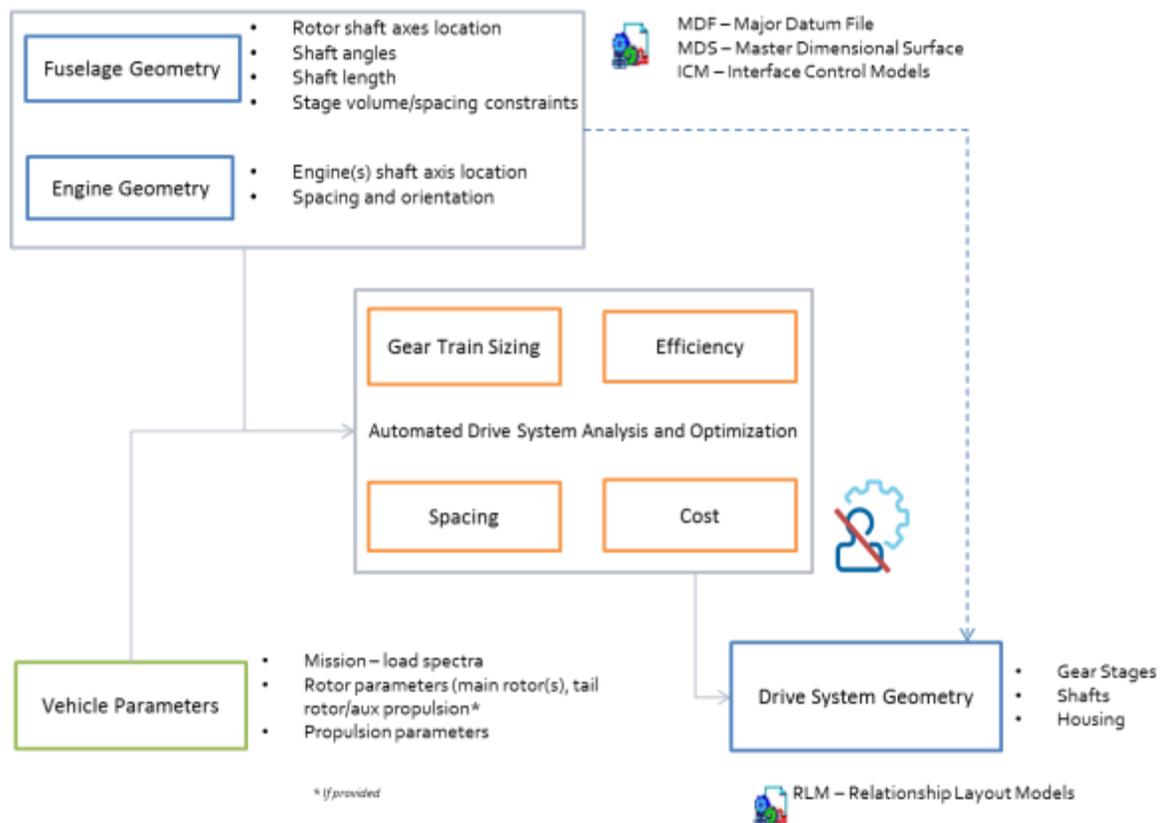


Figure 142. Fully-relational design for drive system

Figure 143 shows the outline of the framework proposed. Parent disciplines are reduced to system objectives and used within the conceptual development environment. These are then broken down into independent parameters that are used in the evaluation of alternatives. The top level optimizer uses the OEC to perform the Multi-Objective Optimization (MOO). This technique of using an OEC falls under Aggregated Objective Functions (AOF) type of MOO where the objective is an aggregate of multiple objectives with relative weighting coefficients. The sub-optimization is a part of the structural analysis code that is used to perform bending stress, compressive stress and scuffing hazard analysis and efficiency calculations, as discussed in Section 5.1 (page 127). The sub-optimization performs a semi-full factorial sweep of the number of teeth and face width and is explained in detail in Section 5.2 (page 132). The complete CAD geometry is automatically generated for each concept and the geometry is automatically integrated with the fuselage from which new spacing constraints and geometric requirements are derived. This part of the design loop is called vehicle synthesis.

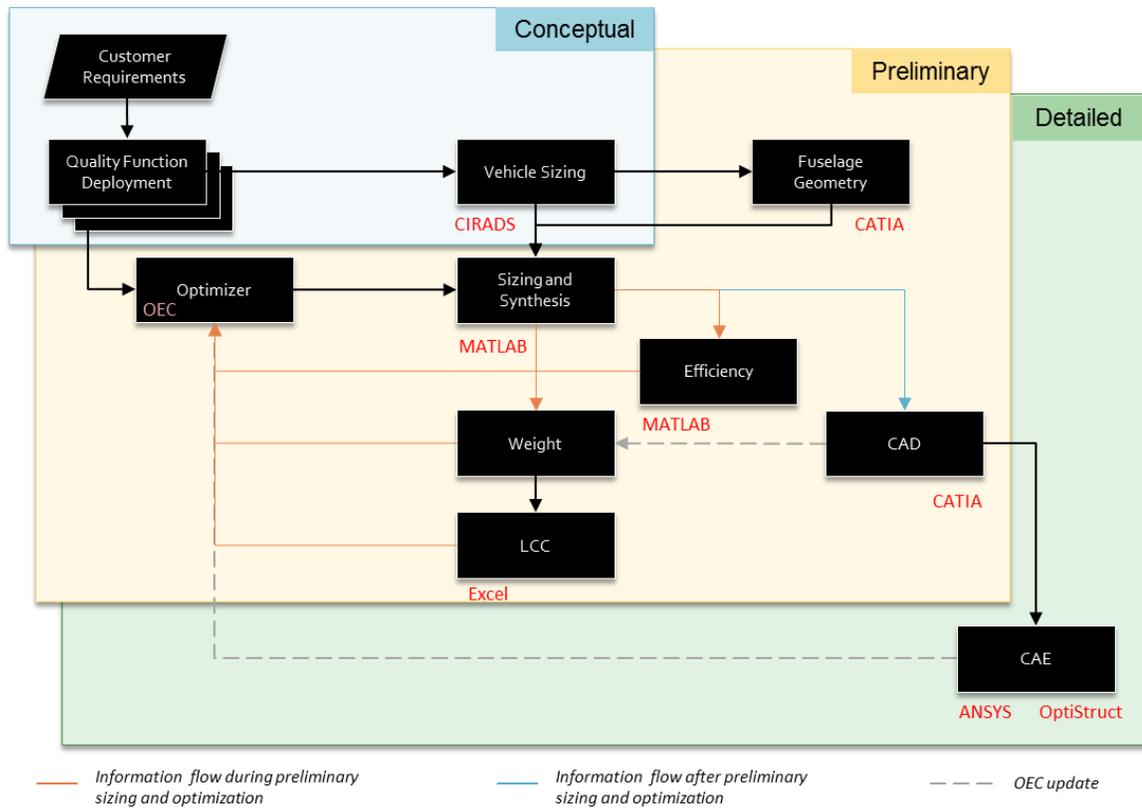


Figure 143. Drive system design framework

The modeling and simulation environment is broken down into three stages, namely conceptual, preliminary and detailed stages. The conceptual design stage starts with customer requirements that are obtained in the form of a Request for Proposal (RFP) for a contract or bidding process, or through a market survey in the case of a new product. The customer requirements are then translated to product attributes and engineering characteristics through a well-known method used in the engineering community, known as Quality Function Deployment (QFD). QFD can be viewed as a communication tool between management and engineering and is shown in Figure 144. The design process of the vehicle begins with the use of customer requirements, mission and specifications. The first step is to start translating this information into usable metrics and weightings through the use of the QFD matrix.

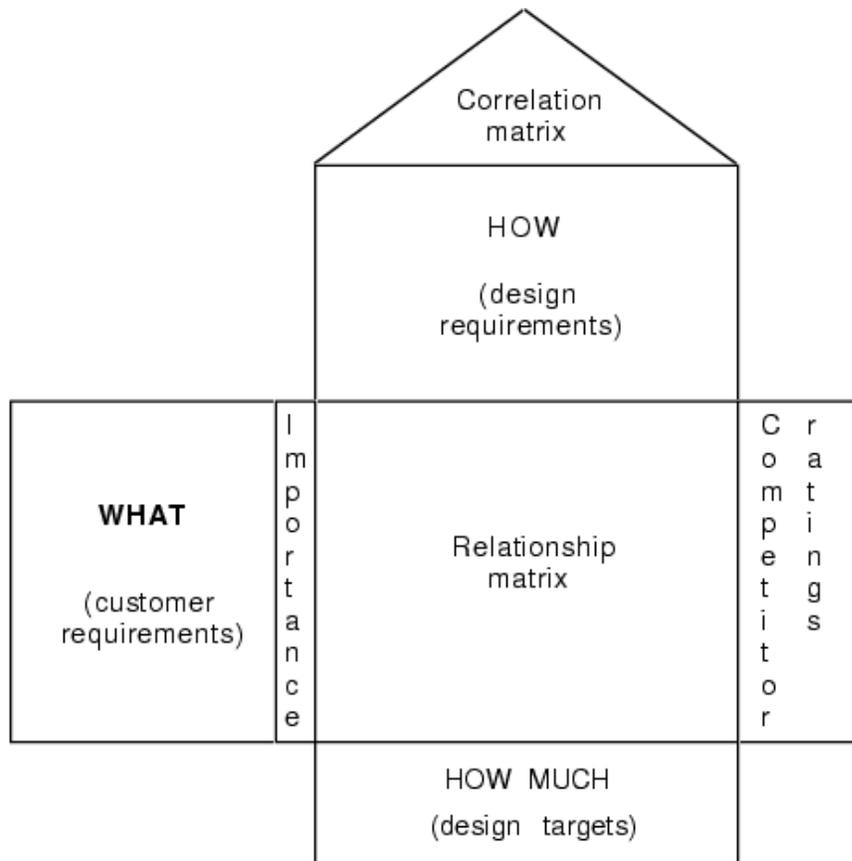


Figure 144. Quality function deployment matrix

There exist multiple levels of QFD; the top level in aerospace design is known as the vehicle level or system level. This level relates customer requirements and/or market based information to engineering characteristics. This is then deployed, as the name indicates to the next level, that is more disciplinary in nature, like rotor, engine, airframe, flight control system etc. This level is known as the sub-system level. Based on the engineering culture of the enterprise some sub-systems may include other sub-systems. For example, the landing gear may be included under the airframe category. In rotorcraft engineering, the drive system is usually categorized along with the engine, under propulsion. This sort of amalgamation of sub-systems works well when their objectives and requirements complement each other.

At the conceptual and preliminary level of design, the QFD is usually not deployed beyond sub-system level. Using the information generated by the QFD at the system level, the conceptual design stage targets for the vehicle can be generated. The drive system would require a deployment to the sub-system level. At the sub-system level QFD, the priority values are derived from system level technical matrix and the system level engineering characteristics are now the customer requirements. The drive system design is performed at the preliminary design stage and requires information generated from the sub-system level QFD, derived in the form of an Overall Evaluation Criterion (OEC).

CIRADS is used as the vehicle conceptual sizing tool. CIRADS stands for Concept Independent Rotorcraft Analysis and Design Software; it is a conceptual design tool that uses the  $R_F$  method to perform mission based vehicle design [146]. As an output from CIRADS, rotor RPMs and engine deck information is obtained. The integration of the drive system framework is done in ModelCenter and is shown in Figure 145. ModelCenter is used to integrate the drive system design tools. The capabilities of ModelCenter extend beyond automation and communication environment; it can be used to perform parametric design studies, DOE, RSM and optimization. The optimization tool consists of line search methods such as method of feasible directions, conjugate gradient method, sequential quadratic programming, and also GA. ModelCenter can control input and output parameters in CATIA, MATLAB, Excel and several other programs. Information flow within the modeling and simulation environment dictates the sequence in which the analyses are performed. Figure 145 shows the flow of information within the framework.

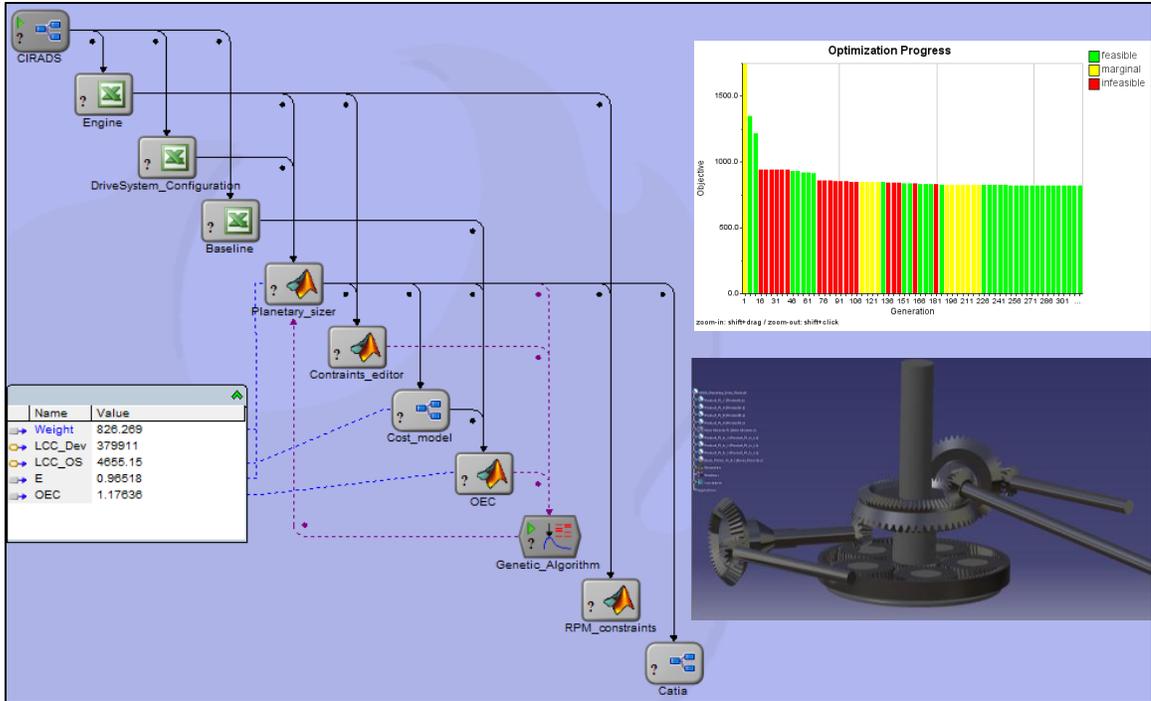


Figure 145. Integration of design tools in ModelCenter [32]

As part of the design framework, a cost analysis tool is integrated to obtain cost information to influence the design decision. The Bell PC based cost tool is integrated within the framework. LCC analysis and the Bell PC tool are discussed in Section 5.6.1 (page 217).

Digitized design platform and software integration capability through IPPD makes handling multiple concepts less expensive. Traditional design required early tradeoffs and concept selection without main requirements frozen. However, in reality, the requirements for a sub-system are fuzzy in nature and not deterministic; ill-informed decisions made earlier in design can have disastrous effects, since making these changes later on lead to higher cost implications. This requires a probabilistic approach in concept down-selection and flexible concepts so that more than one concept can be carried through preliminary design. The latter is of importance in cases where concepts are fundamentally different from one another. These concepts are made flexible enabling

subsequent design changes and efficient concept exploration. This modeling and simulation method digitizes most aspects of the concepts leading to longer concept retention, cost effectively, allowing designers to pursue multiple alternatives during the conceptual design stage. Figure 146 and Figure 147 show two concepts studied and implemented in this thesis. In the case of a planetary gear design, concepts could involve more than one reduction planetary stage and these can have different overall layouts [147]. In case of a split torque design concept, multiple options exist such as number of torque splits for each input and the option of double helical over helical to counter axial loads if a helical angle is desired [148]. These options along with newer concepts such as face-gears and variable speed transmission concepts result in multiple combinations that must be evaluated [12, 69, 70]. A few concepts may be eliminated using engineering judgment and historical information. The remaining concepts must be evaluated through a modeling and simulation environment. The flexibility testing methodology is very useful in evaluating an optimal concept in proceeding with a design decision. Selecting a design based on a probabilistic approach over a single point approach has a significant advantage in promoting efficiency of the product design lifecycle.

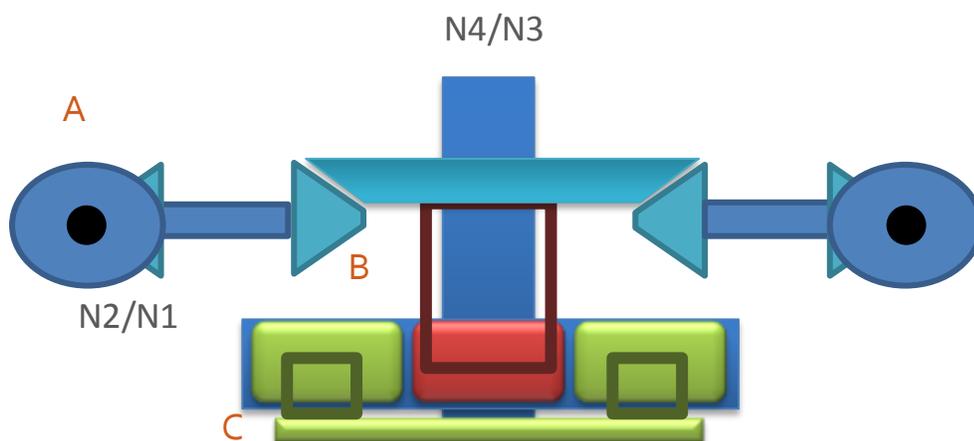


Figure 146. Planetary design concept (rear view)

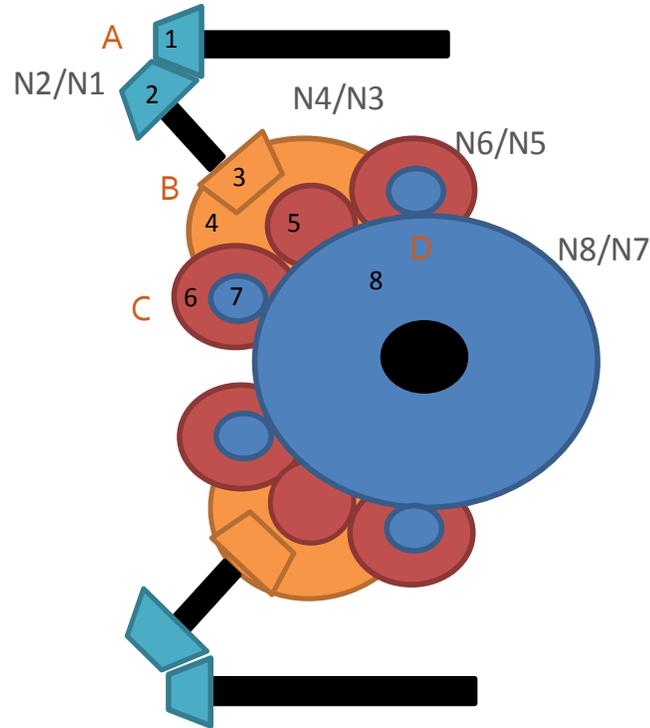


Figure 147. Split-torque design concept (top view)

Optimization of the drive system is a MOO problem that is performed by consolidating the objectives into one value function or OEC. This is known as an aggregated objective function method of solving MOO problems. The OEC for the drive system is shown in Equation 26. The customer requirements are deployed down to system level using a QFD to obtain weight ( $W$ ), efficiency ( $\eta_{tr}$ ) and lifecycle cost as engineering metrics of interest. The coefficients  $\alpha$ ,  $\beta$  and  $\gamma$  are the weightings that are obtained from QFD that describe how important each metric is in comparison to the other and they sum up to 1. The objectives are normalized with respect to a baseline value or during design iterations compared with a value from the previous iteration.

$$OEC_{drivesystem} = \alpha \left( \frac{W_{baseline}}{W} \right) + \beta \left( \frac{\eta_{tr}}{\eta_{trbaseline}} \right) + \gamma \left( \frac{LCC_{baseline}}{LCC} \right)$$

Equation 26. OEC for drive system

The optimization routine used is shown in Figure 148. The grey box indicates the GA and the blue box the sub-optimizer. The FFSO runs a full-factorial sweep of the number of teeth ( $N$ ) and face width ( $FW$ ). Full-factorial sweep is computationally expensive but guarantees a global optimum. The face width is discretized in 12 steps. Face width has size limits based on diametral pitch ( $P_d$ ) and pinion operating pitch diameter ( $d_p^p$ ) which are enforced while generating the discretized values. Teeth combinations for a given reduction ratio are estimated using the hunting ratio algorithm, discussed in Section 4.2 (page 104). Discretization and usage of hunting teeth reduce the computational cost of performing the full-factorial sweep. A penalty function is introduced along with the ‘sizer’ such that any violations of constraints results in a weight penalty determined by the penalty factor ( $r_p$ );  $r_p$  is empirically approximated to 50 for bending stress violations and 100 for contact stress violation, based on test runs to study the effect of the violations (Section 5.2, page 132). The penalty function enables the GA to search close to the constraint boundaries and obtain valid results from the sub-optimizer ensuring that the design is, hardly over-designed.

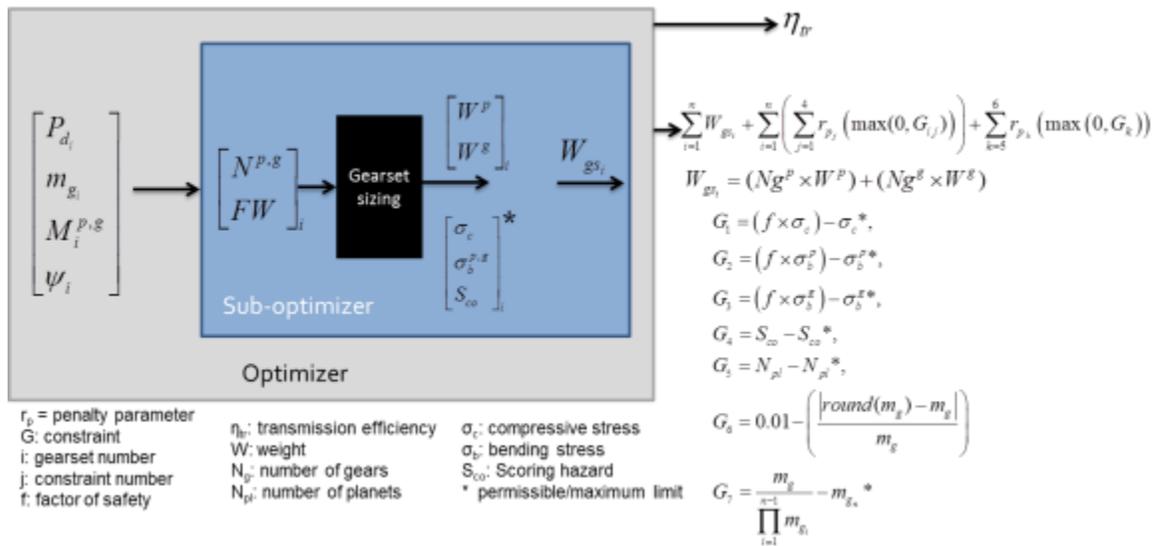


Figure 148. Optimization setup [149]

Additional constraints are added to the algorithm to ensure configurations are viable, shown in Figure 148. These constraints restrict maximum number of planets in the sizing for a given  $m_g$ , avoid whole number values of  $m_g$  to ensure hunting ratio can be complied with, and also avoid extreme combinations of gear-set reduction ratios.

A fuselage geometry based on the Sikorsky S-70 is developed and used as the baseline to implement spatial constraints, for transmission design and optimization. The CATIA design of the fuselage sections includes engine cowling, tail rotor cover, and vertical tail leading edge section. Figure 150 shows the schematic of the CATIA geometries. Using relational design, the entire geometry of the drive system can be generated, assembled and integrated automatically. Figure 149 below shows the automated assembly and integration process for a UH-60L based planetary drive system. This method uses the following sequence:

1. The individual gearsets are assembled first in sequence of stage (here, only the planetary gearset is shown)
2. The gearsets from each stage are assembled together
3. The transmission housing and casing is automatically generated and assembled with the drive system product
4. The product is integrated with the engine housing and fuselage geometry

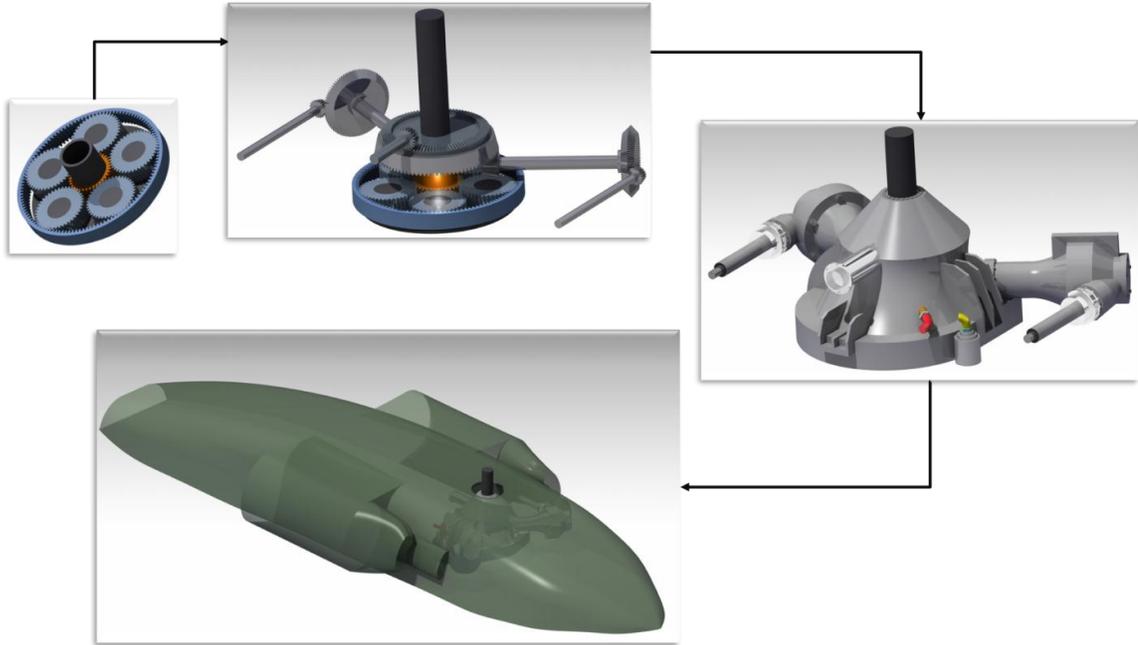


Figure 149. Geometry integration using CATIA [20]

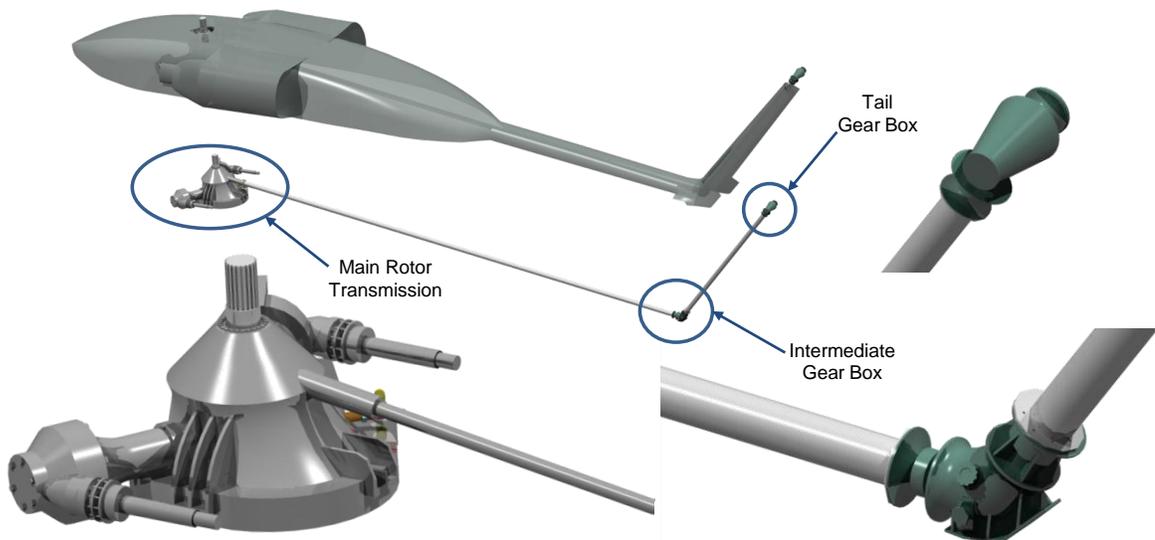


Figure 150. Drive system geometry in CATIA [149]

### ***5.6.1 Lifecycle Cost Analysis***

The Bell PC based cost tool is a weight-based empirical tool that can be used to calculate rotorcraft development, recurring production and operation and support cost [150]. Bell PC is a macro-enabled Microsoft Excel based tool that is integrated in ModelCenter to estimate the RDT&E, logistics and operation and support costs of the drive system. The drive system in Bell PC includes the main transmission, mast, tilt axis gearbox, mid-wing gearbox, free wheel unit, accessory gearbox, rotor brake, tail rotor 90 degree gearbox, tail rotor intermediate gearbox, tail rotor driveshaft, engine input driveshaft, interconnect driveshaft, tilt axis gearbox driveshaft, and combining gearbox (if not included with the engine installation). The cost parameters and model inputs can be found in Appendix A.7.

## CHAPTER 6

### CONCLUSION

The design of complex systems has become a more global, integrated problem. Complex systems comprise of various disciplines and analyses. Many design decisions are a compromise based on these disciplinary objectives. To facilitate design decisions in a concurrent manner, information transfer between these systems need to be streamlined and processes developed for improved integration. Amount of interfaces are subject to the extent of interaction anticipated. For most systems the level of interaction between subsystems and disciplines can vary greatly depending on the overall system objectives and the fidelity of the analyses. Efficient information transfer and change propagation is essential for multidisciplinary trade studies, needed to achieve the system's global objectives.

The Fully-Relational Design methodology allows for efficient multidisciplinary integration and change propagation through hierarchical parametric relations. The resulting design has the potential of being a better compromise. Digitized automation and efficient optimization are critical for the success of this methodology.

In the implementation of the IPPD based approach for the rotorcraft drive system, the following issues were addressed:

1. Different levels of interfaces required.
2. The corresponding technology logistics requirements.
3. Optimization of gear trains.
4. Uncertainty management and risk mitigation through flexible design.
5. Timeliness of introduction of higher fidelity of analyses.

## 6.1 *Review of Research Questions, Conjectures and Hypotheses*

- **Research Question #1:** *How can system integration be effectively performed at the sub-system level?*
- **Research Question #1a:** *What are the requirements of a framework and the logical steps involved in integrating a system with its associated systems to perform tradeoffs?*

*Conjecture #1: Using the fully-relational design technique a subsystem can be effectively integrated with its associated systems; this greatly enhances the capability to perform tradeoffs and increases design cycle efficiency.*

The concept of FRD was implemented on a three-stage design (Section 5.1) and then expanded to a rotorcraft drive system (Section 5.6). Certainly the most complicated aspect of implementing a FRD system is in understanding the interactions and developing sound interfaces to study the system's response to change. As the system's scope gets larger, the implementation gets more complex.

- **Research Question #2:** *Is a Genetic Algorithm suitable to optimize gear train optimization problem.*
- **Research Question #2a:** *How can Genetic Algorithms be improved to optimize gear train type constrained problems?*

*Conjecture #2: Optimization of gear trains can be setup using a GA with a sub-optimization routine. The performance of a GA can be improved by including innovative methods such as Adaptive Crossover and Mutation Rates, Migration between sub-populations and introduction of random*

*members. The problem of handling constraints in GA can be alleviated by investigating effective penalizing techniques.*

A novel approach to gear train optimization was developed to handle hierarchical design variables and nonlinear constraints. Techniques to improve the GA for the gear train optimization problem including constraint-handling techniques, and convergence and exploration techniques were tested.

- **Research Question #3:** *How do geometric and spacing requirements affect gear train design?*

*Hypothesis #1: Geometric location of input and output shafts of a given gear train and volumetric constraints of the housing influence the design. This interaction can be quantified and used to alter the optimal configuration.*

The geometry and spacing analysis is performed on a three-stage spur gear system and is discussed in Section 5.3 (page 154). The introduction of geometry and volume based design synthesis allows the designer to evaluate designs on an additional capability metric. To test this hypothesis, two discrete gear train designs with very close objective values were selected. The experiment indicated that when the geometry of the housing cylinder was changed, the initially optimal design violated the geometry restrictions while the sub-optimal design was still feasible, substantiating the hypothesis. Using a constraint violation measuring technique and penalizing method, this information was used in conjunction with the optimization to alter the selection of the optimal design.

- **Research Question #4:** *How can design engineers select between distinct families of designs, early in design, without sacrificing capability in later stages of the design process?*

*Conjecture #3: Flexibility, studied as a metric in the evaluation of alternatives, in early design, helps the designer select a concept that has improved capability and adjustability to possible later changes in upstream information.*

Flexibility was introduced as a design metric in the evaluation of concepts. The details of the process are discussed in Section 5.4 (page 163). Two concepts were developed to study the baseline reduction gearing. The optimized three-stage gear configuration weighed 151.51 lbs. and the optimized Four-stage gear configuration weighed 152.58 lbs; the difference being slightly over 1 lb. (with no inclusion of shaft weight etc.). It was of interest to see if the three-stage configuration functionality and performance would rapidly degrade when the requirements (power, speed and reduction ratio) are altered. Based on simulations performed on a metamodel, the four-stage gear system offered marginal improvement for this condition. Another set of simulation runs were performed on more extreme conditions by increasing the torque and reduction ratio range. Simulation runs were then run with mean values for the distribution higher than the baseline values to test the modeling and simulation method as well as quantify the flexibility. For this case, the four-stage gear system showed an appreciable benefit in capability over the three-stage. Information is thus made available to the designer to make a sound judgment in moving forward with the design into preliminary stage.

- **Research Question #5:** *Can a sufficiently fast and accurate 3-D Finite Element Analysis method be used for advanced design of gears?*

*Conjecture #4: Proper formulation of gear contact analysis can be developed to obtain results that are consistent, fast and accurate over a broad range of gears without the need for load scaling and formulation modifications.*

Experiments were setup to study the effects of various advanced contact formulation techniques in ANSYS. The following important contact formulation settings were studied to develop a nonlinear analysis model:

1. Contact Type
2. Contact Formulation
3. Interface Treatment
4. Detection Method
5. Stabilization Damping, and
6. Time Step Controls

Mesh requirements and transient analysis settings were assessed to develop this formulation. Contact analysis problem for four very different spur gear combinations were tested and results evaluated to be in agreement with AGMA predicted values.

Micro-parameter alteration to tooth shape in the form of addendum ratio and edge generating radius were performed and their effects studied to be consistent with literature. All contact analysis studies and the results are presented in Section 5.5.1 (page 176).

The formulation's consistency is an indication that this non-complex formulation can be integrated within a design framework with automation capability.

- **Research Question #6:** *Is advanced analysis of gears using Finite Element Analysis a ‘design refinement’ or does it alter the overall optimal configuration? Is this level of fidelity required for preliminary sizing of the drive system?*
- **Research Question #6a:** *How does gear-web topology optimization process impact gear weight and the overall drive system design? Is the consequent weight saving information large enough to change the design selection?*

***Hypothesis #2:** Information obtained from topology optimization does not influence the preliminary design decision and can be treated as design refinement.*

Two test cases were presented to study the effects of topology optimization to obtain a gear web. The hypothesis was tested to see if an optimal design selection remained optimal when topology optimization was performed on 1) a slightly oversized gear, and 2) a gear with a larger web volume. Results from these tests indicated that although there was more improvement in the suboptimal designs, through topology optimization, the initially optimal design still weighed less than they did before. The two cases tested are discussed in Section 5.5.2 (page 197). Results of the test substantiate the hypothesis.

## 6.2 *Review of Research Objectives*

A set of research objectives were established for this thesis (Section 1.3, page 45); this section summarizes the approach to realize each of these objectives.

1. Develop a framework flexible to interfaces, fast and accurate, with integration and automation capability.

A framework was developed using ModelCenter to integrate the design process and automate the information flow and analyses. The implementation and discussion are presented in Section 5.1 (page 127).

2. Improve understanding of optimization techniques for gear train design.

A significant gap was observed in the optimization techniques used for gear train applications. An improved understanding of the problem was developed through literature review of design and optimization of drive systems (Section 2.3, page 57). Applicability of the genetic algorithm for this problem was studied and a novel method developed to optimize gear trains. The optimization methodology and implementation are discussed in Section 5.2 (page 132).

3. Closing the gap in high fidelity design in early stages.

The Introduction of higher fidelity analysis using finite element methods was studied. Methods to study micro parameter change using fast nonlinear transient contact analysis and thin rimmed geometry using topology optimization was developed. Requirements for data transfer and information for these types of analyses were studied and discussed (Section 5.5, page176).

4. Understanding ‘when’ and ‘where’ high fidelity analysis information is required for design decisions.

Research Questions 6 and 6a discuss with specificity the timely introduction of higher fidelity analysis such as topology optimization. Higher fidelity analysis information is only required if it alters the design decision. It was shown, through testing Hypothesis 2, that topology optimization analysis does not alter the design decision (Section 5.5.2, page 197).

5. Develop a method to enforce geometry and space constraints through ‘fully-relational’ design.

Spacing and volumetric analysis were addressed in Research Question 3 and its impact studied through the testing of Hypothesis 1. A program was written to study the alignment of a three-stage gear train within a cylindrical housing. The program enforces volume constraints based on the dimensions of the casing cylinder and input and output shaft locations (Section 5.3, page 154). Furthermore, the space packing algorithm was integrated with the design framework using ModelCenter to influence overall design. Geometric information from the parent system (cylindrical casing) and input and output shaft locations from the CATIA geometry are integrated with the space packing algorithm as well as the gear train optimization algorithm. If the design changes, the CATIA geometry of the gear train is altered automatically, as discussed in Section 5.1 (page 127).

6. Develop a method to select a ‘flexible’ configuration in conceptual design stages.

A thorough literature review of handling uncertainty through ‘flexibility’ was performed. A method was developed to study flexibility as a metric of interest. Flexibility was used to compare a three stage vs. a four stage design configuration. This method and its applicability to conceptual design stages to pick a flexible design were studied.

### 6.3 *Contributions*

The following is a short note on the contributions of this research.

1. A sound literature research was done in the field of Systems Engineering and a few bottlenecks identified in the implementation of IPPD type CE in the area of aerospace MDAO. To improve the design process and enable the paradigm shift from serial approach to a more streamlined product-process driven approach, a novel method and enhancement to relational CAD design is introduced. To study complete sizing, synthesis and optimization, a fully-relational design methodology is presented; it is tested and implemented on a three stage gear system. This methodology is expanded on and the capability demonstrated on a full rotorcraft drive system.
2. A new, accurate and fast, optimization technique for gear trains is developed and tested. Multiple constraint handling techniques are tested and simulations performed to discuss the effects of different tuning factors. Methods to improve population diversity are introduced and tested.
3. A method to test systems for geometrical compatibility is introduced and used to study optimality of designs in the presence of geometric constraints.
4. Flexibility is studied as a design metric and used to evaluate design configurations.
5. FEA techniques for gears are studied as a way of introducing high fidelity design. The properness of introducing topology optimization is studied. It was concluded that topology optimization does not influence the design decision in selecting a design.

## **6.4 Future Work**

Future work is required in the following areas.

1. Full-scale implementation of fully-relational design including evaluation of sizing and optimization logic criteria for different systems. Higher fidelity of interfaces should be tested in a MDAO framework
2. Optimization technique needs to be tested on larger scale, and higher fidelity constraints including frequency harmonics need to be introduced.
3. Branch and bound type gear teeth and face width selection must be tested to improve speed of the optimization sub-routine.
4. Methods to study geometry and spacing analysis for more complex geometries need to be developed. An improved understanding of part movement in 3-D space is required.
5. FEA methods presented need to be tested and improved for helical, bevel gears, etc.
6. FEA methods need to be optimized to be automation friendly so they can be used in conjunction with AGMA sizing.
7. A combined topology optimization for pinion and gear would give an improved assessment of the effects of the process.

## APPENDIX

### A.1. Gear Rating

#### a. Bending Stress

$$\sigma_b = \frac{W_t P_d}{F J} K_o K_v K_s K_m K_b$$

Where

- $\sigma_b$  Bending stress (psi)
- $W_t$  Transmitted tangential load (lb)
- $P_d$  Diametral pitch ( $\text{in}^{-1}$ )
- $F$  Face width (in)
- $J$  AGMA geometry factor for bending
- $K_o$  Overload factor
- $K_v$  Dynamic factor
- $K_s$  Size factor
- $K_m$  Load distribution factor
- $K_b$  Rim thickness factor

#### Allowable Bending Stress

The allowable bending stress is adjusted for life, load cycles, thermal effects and reliability. The calculated bending stress has to be within this permissible stress limit.

$$\sigma_b \leq \frac{\sigma_{ab} Y_N}{S_F K_T K_R}$$

Where,

$\sigma_{ab}$  Allowable bending stress (psi)

$Y_N$  Stress cycle factor

$S_F$  Safety factor

$K_T$  Temperature factor

$K_R$  Reliability factor

**b. Contact Stress**

$$\sigma_c = \frac{C_p C_f}{I} \sqrt{W_t K_o K_v K_s \frac{K_m}{F d_p}}$$

Where,

$\sigma_c$  Contact stress (psi)

$C_p$  Elastic coefficient

$C_f$  Surface condition factor for pitting

$I$  AGMA geometry factor for pitting

$W_t$  Transmitted tangential load (lb)

$K_o$  Overload factor

$K_v$  Dynamic factor

$K_s$  Size factor

$K_m$  Load distribution factor

$F$  Face width (in)

$d_p$  Operating pitch diameter (in)

The entire component on the right hand side is collectively called the permissible bending stress limit denoted by  $\sigma_b'$ .

## Allowable Contact Stress

The allowable contact stress is adjusted for life, load cycles, thermal effects and reliability. The calculated contact stress has to be within this permissible stress limit.

$$\sigma_c \leq \frac{\sigma_{ac} Z_N C_H}{S_H K_T K_R}$$

Where,

$\sigma_{ac}$  Allowable contact stress (psi)

$Z_N$  Stress cycle factor

$C_H$  Brinell hardness ratio factor

$S_H$  Safety factor

The entire component on the right hand side is collectively called the permissible contact stress limit denoted by  $\sigma_c$  '.

### *c. Scuffing Hazard*

$$t_c = t_M + t_{fl}$$

Where,

$t_c$  Contact temperature

$t_M$  Bulk temperature

$t_{fl}$  flash temperature

$$t_{fl} = K \mu_M \frac{X_\Gamma w_{Nr}}{B_M (b_H)^{0.5}} \left| (v_{r1})^{0.5} - (v_{r2})^{0.5} \right|$$

Where,

- $K$  Numerical factor for frictional heat over the contact band
- $\mu_M$  Mean coefficient of friction
- $X_G$  Load sharing factor
- $W_{Nr}$  Normal unit load
- $V_{r1}$  Rolling velocity of pinion
- $V_{r2}$  Rolling velocity of gear
- $B_M$  Thermal contact coefficient
- $b_H$  Semi-width of Hertzian contact band

$$t_{oil} = \frac{t_{oil\ in} + t_{oil\ out}}{2} = t_{oil\ in} + \frac{\Delta T}{2}$$

$$t_M = -24 + 1.2t_{oil} + 0.56t_{fl\ max}$$

Where,

- $t_M$  Bulk temperature
- $t_{oil}$  Oil temperature (°F)
- $t_{fl\ max}$  Maximum flash temperature

$$t_{c\ max} = t_M + t_{fl\ max}$$

Where,

- $t_{c\ max}$  Maximum contact temperature

## A.2. MATLAB Codes

### a. Spur and Helical Gear Sizing Function

```
% Function to perform Spur and Helical Gear Sizing
% - Copyright 2013 Sylvester Ashok -
% Integrated Product Lifecycle Engineering Laboratory
% Georgia Institute of Technology

function [weight_p_op, weight_g_op, N_p_op, N_g_op, F_op, mg_op,
Sb_p_op, Sb_g_op, Sc_op, Sfb_p, Sfb_g, Sfc, P_scr]...
    = Spur_Helical_sizer(P, Q_p_in, mg, Pd, rpm_p, psi_deg, mat_p, mat_g,
No_p, No_g, idler, TBO, sp)
%% Correction Factors
Ka = 1.25;           % Application Factor for inconsistent loads
Kb_p = 1;           % Rim Thickness Factor*
Kb_g = 1;           % Rim Thickness Factor*
Kr = 1.2;           % Reliability Factor: 99.9% Reliability Rating
Ks = 1;             % Size Factor
Kt = 1;             % Temperature Factor
Ki = 1.42;          % Idler Factor

Ca = Ka;           % Application Factor for inconsistent loads
Cs = Ks;           % Size Factor
Cr = Kr;           % Reliability Factor

Sf = 1;            % Contact Safety Factor
Sh = 1;            % Bending Safety Factor
% ----- Mesh (internal/external) -----%
A_p = 1;
A_g = 1;
A = 1;

%% Acquiring Material Property
[Sbt_p, Sct_p, nu_p, E_p, rho_den_p, BH_p] = material_property(mat_p);
[Sbt_g, Sct_g, nu_g, E_g, rho_den_g, BH_g] = material_property(mat_g);
%% Bending Life Factor Kl
N_cycles = 60*TBO*rpm_p;           % Number of Cycles - Pinion
N_cycles_g = N_cycles / mg;
if N_cycles <= 10^3
    Kl = 3.5;
elseif N_cycles <= 1.2E6;
    Kl = 9.4518*(N_cycles^-0.148);
else
    Kl = 1.3558*(N_cycles^-0.0178);
end

if N_cycles_g <= 10^3
    Kl_g = 3.5;
elseif N_cycles_g <= 1.2E6;
    Kl_g = 9.4518*(N_cycles_g^-0.148);
else
    Kl_g = 1.3558*(N_cycles_g^-0.0178);
end
```

```

%% Surface Life Factor Cl

if N_cycles <=10^4 % Number of Cycles - Pinion
    Cl = 2.466;
elseif N_cycles <=10^7
    Cl = 2.466*(N_cycles^-0.056);
else
    Cl = 1.4488*(N_cycles^-0.023);
end

if N_cycles_g <=10^4 % Number of Cycles -
Pinion
    Cl_g = 2.466;
elseif N_cycles_g <=10^7
    Cl_g = 2.466*(N_cycles_g^-0.056);
else
    Cl_g = 1.4488*(N_cycles_g^-0.023);
end

%% Idler factor Ki; idler = 0 - no idler, 1 - pinion idler, 2 - gear
idler
if idler == 0;
    Ki_p = 1;
    Ki_g = 1;
elseif idler == 1;
    Ki_p = Ki;
else Ki_g = Ki;
end

%% Hardness Ratio Factor Ch
% ##Ch is applied only to gear
BH_ratio = BH_p/BH_g; % Brinell Hardness ratio for Pinion
and Gear
if BH_ratio < 1.2
    A_BH = 0;
elseif BH_ratio > 1.7
    A_BH = 0.00698;
else
    A_BH = 0.00898*BH_ratio - 0.00829;
end
Ch = 1 + A_BH*(mg - 1); % Hardness Ratio Factor

%% Working Stresses
Sfb_p = Sbt_p*Kl/(Kt*Kr*Sh);
Sfb_g = Sbt_g*Kl_g/(Kt*Kr*Sh);
Sfc_p = Sct_p*Cl/(Kt*Cr*Sf);
Sfc_g = Sct_g*Cl_g*Ch/(Kt*Cr*Sf);

Sfc = min(Sfc_p, Sfc_g);

%% Initializing
psi = deg2rad(psi_deg);
phi = deg2rad(20);
%% AGMA Geometry Factor for Bending
if psi ~= 0
    [J_s_p, J_s_g] = Geo_factor(psi_deg, 'spur');
    N_p_low = J_s_p(1,2);

```

```

else N_p_low = 21;
end

%% Initializing Arrays
F_array = linspace(8,16,12); % Face width array
teeth_p = H_ratio(mg, N_p_low); % Pinion teeth for Hunting ratio
th = length(teeth_p);
weight_g = zeros(th,12);
weight_p = zeros(th,12);
Sb_p = zeros(th,12);
Sb_g = zeros(th,12);
Sc = zeros(th,12);
N_p = zeros(th,1);
N_g = zeros(th,1);
wp = zeros(th,12);
wg = zeros(th,12);
wc = zeros(th,12);

%% Initializing
a_coeff = 1; % Addendum Coeff
b_coeff = 1.4; % Dedendum Coeff
a_p = a_coeff/Pd; % Addendum
a_g = a_coeff/Pd; % Addendum
b_p = b_coeff/Pd; % Dedendum
b_g = b_coeff/Pd; % Dedendum
w_p = a_p + b_p;
w_g = a_g + b_g;
Q_p_c = Q_p_in; % Compressive load is reduced for combining and
splitting
C_mc = 0.8; % Lead Correction factor for properly modified
leads
C_mt = 1; % Transverse load distribution factor
Ce = 1; % Mesh alignment correction factor = 1
% Mesh alignment empirical constants for precision enclosed gears
A_b = 0.0675;
B_b = 0.0128;
C_b = -0.0000926;

%% ~~~~ Analysis ~~~~
for i = 1: length(teeth_p)
    N_p(i) = teeth_p(i);
    N_g(i) = round (mg * N_p(i));
    R_p = N_p(i)/Pd/2/cos(psi); % Standard
    (reference) pitch radius in
    R_g = N_g(i)/Pd/2/cos(psi); % Standard
    (reference) pitch radius in
    phi_t = atan(tan(phi)/cos(psi)); % Transverse pressure
angle
    Rb_p = R_p*cos(phi_t); % Base radii
    Rb_g = R_g*cos(phi_t); % Base radii
    Q_v = 12; % Gear Quality Rating
    mg = N_g(i)/N_p(i); % Gear ratio
    C_r = R_g + A_p*R_p; % Operating center distance in
    pb = 2*pi*Rb_p/N_p(i); % Transverse base pitch
    pn = pi*cos(phi)/Pd; % Normal transverse pitch

```

```

    phi_r = acos((A_p*Rb_g + A_g*Rb_p)/C_r);      % Operating Transverse
pressure angle
    if psi ~= 0
        psi_b = acos(pn/pb); % Base helix angle
        psi_r = atan(atan(psi_b)/cos(phi_r));    % Operating helix
angle
    else
        psi_b = 0;
        psi_r = 0;
    end
    phi_nr = asin(cos(psi_b)*sin(phi_r));      % Operating normal
pressure angle
    C6 = C_r*sin(phi_r);                       % in
    dp_p = R_p*2;
    dp_g = R_g*2;
    Ro_p = N_p(i)/Pd/2/cos(psi) + A_p*a_p;
    Ro_g = N_g(i)/Pd/2/cos(psi) + A_g*a_g;    % Outside radius in
    phi_o_p = acos(Rb_p/Ro_p);                % Tip pressure angles
    phi_o_g = acos(Rb_g/Ro_g);                % Tip pressure angles
degrees
    C1 = A*C6-sqrt(Ro_g^2-Rb_g^2);             % SAP in
    C3 = C6/(mg+A);                           % Operating pitch point in
    C4 = C1 + pb;                              % HPSTC in
    C5 = sqrt(Ro_p^2 - Rb_p^2);                % EAP in
    C2 = C5 - pb;                              % LPSTC in
    Z = C5 - C1;                               % Length of line of contact in
    mp = Z/pb;                                 % Transverse contact ratio
    if psi == 0
        mF = 0;                                % Axial contact ratio
    else
        px = pi/sin(psi)/Pd;                  % Axial pitch
        mF = F/px;                            % Axial contact ratio
    end
    if mp > 2
        disp('error in mp')
    end
    nr = mp - floor(mp);                       % Fractional part
of mp
    na = mF - floor(mF);                       % Fractional part
of mF
    Fe = F;
    if psi == 0 && mp<2
        Lmin = Fe;                            % Minimum length of lines
of contact
    elseif psi > 0 && na<= (1-nr)
        Lmin = (mp*Fe - na*nr*px)/cos(psi_b);
    else
        Lmin = (mp*Fe-(1-na)*(1-nr)*px)/cos(psi_b);
    end
    Rml = 0.5*(min(Ro_p,Ro_g) + A*(C_r - max(Ro_p,Ro_g))); % Mean
radius of pinion
    do_p = 2*C_r/(mg+1);                       % Operating
pitch diameter
    % ----- Geometry factors -----
    if psi == 0;
        J_p = J_spur(Pd, N_p(i), N_g(i), 1, 1);
        J_g = J_spur(Pd, N_g(i), N_p(i), 1, 1);

```

```

else
    J_p = interp2(J_s_p(1,:), J_s_p(:,1), J_s_p, N_p(i), N_g(i)); %
interpolate from a table
    J_g = interp2(J_s_g(1,:), J_s_g(:,1), J_s_g, N_p(i), N_g(i)); %
interpolate from a table
end

R = N_p(i)/Pd/(2*cos(psi));
W_t = Q_p_in/R; %lb
W_t_c = Q_p_c/R; %lb
% ----- Dynamic factor (Kv) -----

wp = rpm_p*pi/30; % Rotational velocity of pinion
rad/s
v_t = 5*wp*R_p; % Operating pitchline velocity fpm
7,907.2
B_v = 0.25*(12-Q_v)^0.667;
A_v = 50 + 56*(1-B_v);
Kv = ((A_v+sqrt(v_t))/A_v)^B_v; % Dynamic Factor
Cv = Kv; % Dynamic Factor
for j = 1: 12
    F = F_array(j)/Pd; % Face width ranges from 8/Pd to
16/Pd
% ---- Load distribution factor Km (F in inches) ----
if F <= 1
    C_pf(1) = F/(10*2*R_p) - 0.025; % Pinion proportion factor
    C_pf(2) = F/(10*2*R_g) - 0.025;
elseif F > 1 && F <= 17
    C_pf(1) = F/(10*2*R_p) - 0.0375 + 0.0125*F;
    C_pf(2) = F/(10*2*R_g) - 0.0375 + 0.0125*F;
else
    C_pf(1) = F/(10*2*R_p) - 0.1109+0.0207*F-0.000228*F^2;
    C_pf(2) = F/(10*2*R_g) - 0.1109+0.0207*F-0.000228*F^2;
end
C_pm = 1.1; % Pinion proportion
modifier
C_ma = A_b + B_b*F + C_b*F^2; % Mesh alignment factor
C_mf = 1 + C_mc*(C_pf*C_pm + C_ma*Ce); % Face load
distribution factor
Km = C_mt*C_mf;
Cm = max(Km);

%% ////////// AGMA Bending Stress Calculations //////////
Sb_p(i,j) = W_t*(Pd/F/J_p)*Ka*Km(1)*Ks*(Kb_p*Ki_p)/Kv;
Sb_g(i,j) = W_t*(Pd/F/J_g)*Ka*Km(2)*Ks*(Kb_g*Ki_g)/Kv;

%% ////////// AGMA Contact Stress Calculations //////////
Cp = sqrt(1/(pi*((1-nu_p^2)/E_p)+((1-nu_g^2)/E_g))); %
Elastic coefficient for difference in p,g material
if psi == 0;
    rho_p = C2;
    mnp = 1;
elseif mF >1
    rho_p = sqrt(Rm1^2 - min(Rb_p,Rb_g)^2);
    rho_p_2 = sqrt((0.5*((dp_p/2+a_p)+(dp_p/2 - a_g)))^2 -
(dp_p/2*cos(phi))^2);
    mnp = F/Lmin;

```

```

end
% ----- Radius of curvature of the mesh geometry -----
rho_g = C6 - A*rho_p; % use '+ rho_p' for
internal gear
I = cos(phi_r)/((1/rho_p + A*1/rho_g)*do_p*mp);
% AGMA surface geometry factor for pitting resistance; use - 1/rho_g
for internal gear
Sc(i,j) = Cp*sqrt(W_t_c*Ca*Cv*Cs*Cm/(F*I*do_p)); %
Compressive stress

%% ----- Scuffing / Scoring Analysis -----
% P_scr - Probability of scoring; has to be less than 0.3
% t_b - Gear body temperature; has to be less than 300 °F
[P_scr, t_b] = Scuffing(P, Q_p_in, rpm_p, Pd, Km(1), Kv,
N_p(i), N_g(i), F, psi_deg, mat_p, mat_g);
%% --- Within Stress limits ?? ---
if Sb_p(i,j) <= Sfb_p && Sb_g(i,j) <= Sfb_g && Sc(i,j) <= Sfc
weight_p(i,j) = pi*F*rho_den_p*((R_p+ a_p - w_p)^2
+0.5*((R_p+a_p)^2 - (R_p+a_p-w_p)^2));
weight_g(i,j) = pi*F*rho_den_g*((R_g + a_g - w_g)^2 +((R_g
+ a_g)^2 - (R_g+a_g-w_g)^2)/2);
count = count +1;
else
% weight penalties for bending stress lconstraint violation
if Sb_p(i,j) > Sfb_p
wp(i,j) = (Sb_p(i,j) - Sfb_p)/Sfb_p*rp;
end
if Sb_g(i,j) > Sfb_g
wg(i,j) = (Sb_g(i,j) - Sfb_g)/Sfb_g*rp;
end
% weight penalties for contact stress constraint violation
if Sc(i,j) >Sfc
wc(i,j) = (Sc(i,j) - Sfc)/Sfc*rpc;
end
weight_p(i,j) = pi*F*rho_den_p*((R_p+a_p-w_p)^2
+((R_p+a_p)^2 - (R_p+a_p-w_p)^2)/2) + wp(i,j) + wc(i,j);
weight_g(i,j) = pi*F*rho_den_g*((R_g + a_g - w_g)^2 +((R_g
+ a_g)^2 - (R_g+a_g-w_g)^2)/2) + wg(i,j) + wc(i,j);
end
% weight penalties for Scuffing and Scoring constraint violation
if P_scr >= 0.3
weight_p(i,j) = weight_p(i,j) + rps* (P_scr - 0.3)/0.3;
end
end
end

%% ~~~~ Return ~~~~
weight_tot = No_p*weight_p + No_g*weight_g;
[min_weight_tot_1,i_2] = min(weight_tot,[],1);
[~,j] = min(min_weight_tot_1,[],2);
i_3 = i_2(j);
weight_p_op = weight_p(i_3,j);
weight_g_op = weight_g(i_3,j);
F_op = F_array(j)/Pd;
N_p_op = N_p(i_3);
N_g_op = N_g(i_3);
mg_op = N_g_op/N_p_op;

```

```

Sb_p_op = Sb_p(i_3,j);
Sb_g_op = Sb_g(i_3,j);
Sc_op = Sc(i_3,j);
end

```

### ***b. Bevel and Spiral Gear Sizing Function***

```

% Function to perform Bevel and Spiral Gear Sizing
% - Copyright 2013 Sylvester Ashok -
% Integrated Product Lifecycle Engineering Laboratory
% Georgia Institute of Technology

function [weight_p_op, weight_g_op, N_p_op, N_g_op, F_op, mg_op,
Sb_p_op, Sb_g_op, Sc_op, Sfb_p, Sfb_g, Sfc, P_scr] = ...
    Bevel_sizer(P,Q_p_in, mg, Pd, rpm_p, alpha, mat_p, mat_g, No_p,
No_g, TBO)

dp_p = N_p/Pd; % Pinion outer pitch diameter
dp_G = N_G/Pd; % Gear outer pitch diameter
% Size factor for pitting resistance
if F>= 0.5 && F<= 4.5
    Cs = 0.125*F + 0.4375;
elseif F<0.5
    Cs = 0.5;
else Cs = 1.0;
end

Ko = 1.25; % Overload factor - for speed increase 0.01*mg^2
Q_v = 11; % Transmission accuracy
vt = 0.262*dp_p*rpm_p;

% Dynamic factor Kv
Kv_B = 0.25*(12-Q_v)^0.667;
Kv_A = 60 + 56*(1-Kv_B);
Kv = (Kv_A/(Kv_A + sqrt(vt)))^-Kv_B;
vt_max = (Kv_A + (Q_v - 3))^2; % Max pitchline velocity
Km = 1 + 0.0036*F^2; % Load distribution factor
Cxc = 1.5; % Crowning factor

% Pitting resistance geometry factor
gamma = atan(sin(Sigma)/(N_G/N_p + cos(Sigma))); % Pinion pitch angle
Gamma = Sigma - gamma; % Gear pitch angle
Ao = 0.5*dp_G/sin(Gamma); % Outer cone distance
Am = Ao - 0.5*F; % Mean cone distance
k1 = 2.00; % Depth factor - Table 4
k2 = 0.125; % Clearance factor - Section 7.5
h = k1/Pd*(Am/Ao)*cos(psi); % Mean working depth
c = k2*h; % Clearance
hm = h + c; % Mean whole depth
m90 = sqrt(N_G/N_p*cos(gamma)/cos(Gamma)); % Equivalent 90deg ratio
c1 = 0.21+0.29/m90^2; % Mean addendum factor - Table 5
a_G = c1*h; % Gear mean addendum
a_p = h - a_G; % Pinion mean addendum
b_p = hm - a_p; % Pinion mean dedendum
b_G = hm - a_G; % Gear mean dedendum

```

```

AmG = Am; % Gear mean cone distance
Sigma_delta_S = atan(b_p/AmG) + atan(b_G/AmG); % Sum of dedundum angles
delta_p = atan(b_p/AmG); % Pinion dedundum angle
delta_G = Sigma_delta_S - delta_p; % Gear dedundum angle
gamma_o = gamma + delta_G; % Pinion face angle
Gamma_o = Gamma + delta_p; % Gear face angle
alpha_p = gamma_o - gamma; % Pinion addendum angle
alpha_G = Gamma_o - Gamma; % Gear addendum angle
a_op = a_p + 0.5*F*tan(delta_G); % Mean outer pinion addendum
a_oG = a_G + 0.5*F*tan(delta_p); % Mean outer gear addendum
b_op = b_p + 0.5*F*tan(delta_p); % Mean outer pinion dedendum
b_oG = b_G + 0.5*F*tan(delta_G); % Mean outer gear dedendum
k_prime = (N_G - N_p)/(3.2*N_G + 4*N_p); % Location constant
Pm = Ao/Am*Pd; % Mean transverse diametral pitch
p = pi/Pd; % Outer transverse circular pitch
pN = Am/Ao*p*cos(psi)*cos(phi); % Mean normal base pitch
pn = pN/cos(phi); % Mean normal circular pitch
p2 = pn/(cos(phi)*(cos(psi)^2 + tan(phi)^2)); % Mean normal circular
pitch
r = dp_p*Am/(2*cos(gamma)*Ao); % Mean transverse pinion pitch radius
R = dp_G*Am/(2*cos(Gamma)*Ao); % Mean transverse gear pitch radius
r_N = r/cos(psi)^2; % Mean normal pinion pitch radius
R_N = R/cos(psi)^2; % Mean normal gear pitch radius
r_bN = r_N*cos(phi); % Mean normal pinion base radius
R_bN = R_N*cos(phi); % Mean normal gear base radius
r_oN = r_N + a_p; % Mean normal pinion outside radius
R_oN = R_N + a_G; % Mean normal gear outside radius
Z_P = sqrt(r_oN^2 - r_bN^2) - r_N*sin(phi); % Length of mean normal
pinion addendum action
Z_G = sqrt(R_oN^2 - R_bN^2) - R_N*sin(phi); % Length of mean normal gear
addendum action
Z_N = Z_P + Z_G; % Length of action in mean normal section
mp = Z_N/p2; % Tranverse contact ratio
K_Z = F/Ao*(2-F/Ao)/(2*(1-F/Ao)); % Face contact ratio
mF = 1/pi*(K_Z*tan(psi) - K_Z^3/3*tan(psi)^3)*Ao*Pd; % Face contact
ratio
mo = sqrt(mp^2+mF^2); % Modified contact ratio
psi_b = acos(cos(phi)*sqrt(cos(psi)^2 + tan(phi)^2)); % Mean base
spiral angle
eta = sqrt(Z_N^2*cos(psi_b)^4 + F^2*sin(psi_b)^2);
rho_p = r*sin(phi)/cos(psi_b)^2; % Mean normal pitch profile radius of
curvature at pitch circle
rho_G = R*sin(phi)/cos(psi_b)^2; % Mean normal gear profile radius of
curvature at pitch circle
if psi > 0
    f_I = 0;
else f_I = Z_N/2 - pN; % Different for spiral gears
end
if psi > 0
    i = 1;
    I(1) = 100;
    sign = 1;
    factor = 1;
    while(true)
        i = i+1;
        eta_I = eta^2 - 4* f_I^2;

```

```

        zo = Z_N/2 + Z_N^2*f_I*cos(psi_b)^2/eta^2 +
F*Z_N*eta_I*k_prime*sin(psi_b)/eta^2 - Z_G;
        rho_1 = rho_p - zo; % Pinion profile radius of curvature at
point f1
        rho_2 = rho_G - zo; % Gear profile radius of curvature at point
f1
        rho_o = rho_1*rho_2/(rho_1+rho_2); % Relative radius of profile
curvature
        s = F*Z_N*eta_I*cos(psi_b)/eta^2; % Length of line of contact
if mo<2.0
        Ci = 2/mo; % Interia factor
else Ci = 1;
end
for k = 1:3
        if isreal(sqrt(eta_I^2 - 4*k*pN*(k*pN + 2*f_I))^3) == 1
            term_1 = (sqrt(eta_I^2 - 4*k*pN*(k*pN + 2*f_I))^3);
            continue
        else term_1 = 0;
        end
end
for k = 1:3
        % term_2 = (sqrt(eta_I^2 - 4*k*pN*(k*pN - 2*f_I))^3)
        if isreal(sqrt(eta_I^2 - 4*k*pN*(k*pN - 2*f_I))^3) == 1
            term_2 = (sqrt(eta_I^2 - 4*k*pN*(k*pN - 2*f_I))^3);
            continue
        else term_2 = 0;
        end
end
eta_I_prime = eta_I^3 + term_1 + term_2;
mNI = eta_I^3 /eta_I_prime^3; % load sharing ratio
I(i) = s*rho_o*cos(psi)*cos(phi)*Pd/(F*dp_p*Ci*mNI*Pm); %
Pitting resistance geometry factor
f_I = f_I + (-1^sign)*eta/(10*factor);

plot (i,f_I)
hold on
if I(i) > I(i-1)
        sign = sign + 1;
        factor = factor + 1 ;
end
if factor == 8;
        break
end
end
end
end
end

```

### c. Planetary Gear Sizing Function

```

% Function to perform Bevel and Spiral Gear Sizing
% - Copyright 2013 Sylvester Ashok -
% Integrated Product Lifecycle Engineering Laboratory
% Georgia Institute of Technology

function [weight_s_op, weight_r_op, weight_pl_op, weight_car_op,
N_s_op, N_r_op, N_pl_op, F_op, mg_op, Sb_s_op, Sb_r_op, ...

```

```

    Sb_pl_p_op, Sb_pl_g_op, Sc_s_pl_op, Sc_pl_r_op, Sfb_s, Sfb_r,
    Sfb_pl, Sfc_s, Sfc1, Sfc2]...
    = Planetary_sizer(P, Q_s, mg, Pd, rpm_s, psi_deg, mat_s, mat_r,
    mat_pl, No_pl, TBO)

%% Correction factors
Ka = 1.25;           % Application Factor for inconsistent loads
Kb_s = 1;           % Rim Thickness Factor*
Kb_pl = 1;          % Rim Thickness Factor*
Kb_r = 1;           % Rim Thickness Factor*
Kr = 1.2;           % Reliability Factor: 99.9% Reliability Rating
Ks = 1;             % Size Factor
Kt = 1;             % Temperature Factor
Ki = 1.2;           % Idler Factor
Kv = 1.1;           % Dynamic Factor
Km = 1;
Ca = Ka;            % Application Factor for inconsistent loads
Cs = Ks;            % Size Factor
Cr = Kr;            % Reliability Factor
Cv = 1.1;           % Dynamic Factor
Sf = 1.;            % Contact Safety Factor
Sh = 1.;            % Safety Factor
count = 0;
rp = 3e3;
rpc = 3e3;
%% Acquiring Material Property
[Sbt_s, Sct_s, nu_s, E_s rho_den_s, BH_s] = material_property(mat_s);
[Sbt_r, Sct_r, nu_r, E_r, rho_den_r, BH_r] = material_property(mat_r);
[Sbt_pl, Sct_pl, nu_pl, E_pl, rho_den_pl, BH_pl] =
material_property(mat_pl);

%% Bending Life Factor Kl
Noc_s = No_pl*rpm_s*TBO*60;
Noc_pl = 2*Noc_s/No_pl/(mg-2)*2;           % Number of Cycles -
Planets
Noc = max(Noc_s, Noc_pl);

if Noc <= 10^3
elseif Noc <= 1.2E6;
    Kl = 9.4518*Noc;
else
    Kl = 1.3558*(Noc^-0.0178);
end

%% Surface Life Factor Cl
if Noc <=10^4           % Number of Cycles -
Pinion
    Cl = 1.466;
elseif Noc <=10^7
    Cl = 2.466*(Noc^-0.056);
else Cl = 1.4488*(Noc^-0.023);
end

%% Hardness Ratio Factor Ch
% ##Ch is applied only to gear
BH_ratiol = BH_s/BH_pl;           % Brinell Hardness ratio for
Pinion and Gear

```

```

if BH_ratio1 < 1.2
    A = 0;
elseif BH_ratio1 > 1.7
    A = 0.00698;
else
    A = 0.00898*BH_ratio1 - 0.00829;
end
Ch1 = 1 + A*((mg-2)/2 - 1) ; % Hardness Ratio Factor

BH_ratio2 = BH_pl/BH_r; % Brinell Hardness ratio for
Pinion and Gear
if BH_ratio2 < 1.2
    A = 0;
elseif BH_ratio2 > 1.7
    A = 0.00698;
else
    A = 0.00898*BH_ratio2 - 0.00829;
end
Ch2 = 1 + A*((2*(mg-1)/(mg-2)) - 1); % Hardness Ratio
Factor

%% Modified Allowable Stresses
Sfb_s = Sbt_s*Kl/(Kt*Kr*Sh);
Sfb_r = Sbt_r*Kl/(Kt*Kr*Sh);
Sfb_pl = Sbt_pl*Kl/(Kt*Kr*Sh);
Sfc_s = Sct_s*C1/(Kt*Cr*Sf);
Sfc_r = Sct_r*C1*Ch2/(Kt*Cr*Sf);
Sfc_pl = Sct_pl*C1*Ch1/(Kt*Cr*Sf);

Sfc1 = min(Sfc_s, Sfc_pl);
Sfc2 = min(Sfc_r, Sfc_pl);

%% AGMA Geometry Factor for Bending

[J_s_p, J_s_g] = Geo_factor(psi_deg, 'spur');

%% Teeth and specifications
if (mg-2)/2 < 1;
    N_p_low = round(J_s_p(1,2)*2/(mg-2));
else
    N_p_low = J_s_p(1,2);
end
%% Initializing
psi = deg2rad(psi_deg);
phi = deg2rad(20);

% Initializing Arrays % Face width array
F_array = linspace(12,16,12);
teeth_p = H_ratio((mg-1),N_p_low);
th = length(teeth_p);
weight_s = zeros(th,12);
weight_pl = zeros(th,12);
weight_r = zeros(th,12);
weight_carrier = zeros(th,12);
Sb_s = zeros(th,12);

```

```

Sb_pl_p = zeros(th,12);
Sb_pl_g = zeros(th,12);
Sb_r = zeros(th,12);
Sc_s_pl = zeros(th,12);
Sc_pl_r = zeros(th,12);
N_s = zeros(th,1);
N_r = zeros(th,1);
N_pl = zeros(th,1);
wps = zeros(th,12);
wpp1 = zeros(th,12);
wpr = zeros(th,12);
wpc1 = zeros(th,12);
wpc2 = zeros(th,12);

% Addendum
a_p = 1/Pd;
a_g = 1/Pd;
w_p = 2.4/Pd;
w_g = 2.4/Pd;

a_r = 1/Pd;
w_r = 2.4/Pd;

tR = 1.2*2.4/Pd; % Rim thickness of Ring

%% Initializing
psi = deg2rad(psi_deg);
phi = deg2rad(20);

%% ~~~~ Analysis ~~~~
for i = 1: th
    N_s(i) = teeth_p(i);
    N_pl(i) = round(0.5*N_s(i)*(mg^-2));
    N_r(i) = N_s(i) + 2*N_pl(i);
    dp_s = N_s(i)/Pd;
    dp_pl = N_pl(i)/Pd;
    dp_r = N_r(i)/Pd;
    rp_s = dp_s/2;
    rp_pl = dp_pl/2;
    rp_r = dp_r/2;
    rr3 = rp_r - a_r;
    rr2 = rr3 + w_r;
    rr1 = rr2 + tR;
    %%
    R_s = N_s(i)/Pd/2/cos(psi); % Standard
    (reference) pitch radius in
    R_pl = N_pl(i)/Pd/2/cos(psi); % Standard
    (reference) pitch radius in
    R_r = N_r(i)/Pd/2/cos(psi);
    phi_t = atan(tan(phi)/cos(psi)); % Transverse pressure
    angle
    Rb_s = R_s*cos(phi_t); % Base radii
    Rb_pl = R_pl*cos(phi_t); % Base radii
    Rb_r = R_r*cos(phi_t);
    Q_v = 12; % Gear Quality Rating
    mg = N_g(i)/N_p(i); % Gear ratio
    C_r_1 = R_s + R_pl; % Operating center distance in

```

```

pb = 2*pi*Rb_s/N_s(i); % Transverse base pitch
pn = pi*cos(phi)/Pd; % Normal transverse pitch
phi_r = acos((Rb_g + Rb_p)/C_r_1); % Operating Transverse
pressure angle
if psi ~= 0
    psi_b = acos(pn/pb); % Base helix angle
    psi_r = atan(atan(psi_b)/cos(phi_r)); % Operating helix
angle
else
    psi_b = 0;
    psi_r = 0;
end
phi_nr = asin(cos(psi_b)*sin(phi_r)); % Operating normal
pressure angle
C6 = C_r_1*sin(phi_r); % in
dp_p = R_s*2;
dp_g = R_pl*2;
Ro_s = N_s(i)/Pd/2/cos(psi)+ a_p;
Ro_pl = N_pl(i)/Pd/2/cos(psi)+ a_g; % Outside radius in
phi_o_p = acos(Rb_p/Ro_p); % Tip pressure angles
phi_o_g = acos(Rb_g/Ro_g); % Tip pressure angles
degrees
C1 = A*C6-sqrt(Ro_g^2-Rb_g^2); % SAP in
C3 = C6/(mg+A); % Operating pitch point in
C4 = C1 + pb; % HPSTC in
C5 = sqrt(Ro_p^2 - Rb_p^2); % EAP in
C2 = C5 - pb; % LPSTC in
Z = C5 - C1; % Length of line of contact in
mp = Z/pb; % Transverse contact ratio
if psi == 0
    mF = 0; % Axial contact ratio
else
    px = pi/sin(psi)/Pd; % Axial pitch
    mF = F/px; % Axial contact ratio
end
if mp > 2
    disp('error in mp')
end
nr = mp - floor(mp); % Fractional part
of mp
na = mF - floor(mF); % Fractional part
of mF
Fe = F;
if psi == 0 && mp<2
    Lmin = Fe; % Minimum length of lines
of contact
elseif psi > 0 && na<= (1-nr)
    Lmin = (mp*Fe - na*nr*px)/cos(psi_b);
else
    Lmin = (mp*Fe-(1-na)*(1-nr)*px)/cos(psi_b);
end
Rm1 = 0.5*(min(Ro_p,Ro_g)+ A*(C_r - max(Ro_p,Ro_g))); % Mean
radius of pinion
do_p = 2*C_r_1/(mg+1); %
Operating pitch diameter
%%
if psi == 0;

```

```

        J_s = J_spur(Pd, N_s(i), N_pl(i), 1, 1);
        J_pl_g = J_spur(Pd, N_pl(i), N_s(i), 1, 1);
    else
        J_s = interp2(J_s_p(1,:), J_s_p(:,1), J_s_p, N_s(i), N_pl(i));
% Sun as pinion and Planets as gears
        J_pl_g = interp2(J_s_g(1,:), J_s_g(:,1), J_s_g, N_s(i),
N_pl(i)); % Sun as pinion and Planets as gears
    end

        J_pl_p = interp2(J_s_p(1,:), J_s_p(:,1), J_s_p, N_pl(i), N_r(i));
% Planets as pinion and Ring as gears
        J_r = interp2(J_s_g(1,:), J_s_g(:,1), J_s_g, N_pl(i), N_r(i));
% Planets as pinion and Ring as gears

    %Tangential Load

    W_t = ((Q_s)/(dp_s/2));
    Q_pl = Q_s*(mg-2)/2;
    W_t2 = (W_t*(mg-2)/2);
    %%
    for j = 1:12
        F = F_array(j)/Pd; % Face width ranges from 8/Pd to
16/Pd

        % ---- Load distribution factor Km (F in inches) ----
        if F <=2
            Km = 1.6;
        elseif F<=6
            Km = 1.7;
        elseif F<=9
            Km = 1.8;
        elseif F <20
            Km = 1.9;
        else Km = 2;
        end
        Cm = Km;

        %% ///// AGMA Bending Stress Calculations /////
        % Sun as pinion and Planets as gears
        Sb_s(i,j) = W_t/No_pl*(Pd/F/J_s)*Ka*Km*Ks*(Kb_s)/Kv;
        Sb_pl_p(i,j) = W_t2/No_pl*(Pd/F/J_pl_g)*Ka*Km*Ks*(Kb_pl*Ki)/Kv;

        % Planets as pinion and Ring as gears
        Sb_pl_g(i,j) = W_t2/No_pl*(Pd/F/J_pl_p)*Ka*Km*Ks*(Kb_pl*Ki)/Kv;
        Sb_r(i,j) = W_t2/No_pl*(Pd/F/J_r)*Ka*Km*Ks*(Kb_r)/Kv;

        %% ///// AGMA Contact Stress Calculations /////
        % Sun as pinion and Planets as gears
        dp_p = dp_s;
        dp_g = dp_pl;
        nu_p = nu_s;
        E_p = E_s;
        nu_g = nu_pl;
        E_g = E_pl;

```

```

        Cp = sqrt(1/(pi*((1-nu_p^2)/E_p)+((1-nu_g^2)/E_g))) ;
% Elastic coefficient for difference in p,g material

        if psi == 0;
            rho_p = sqrt((dp_p/2+(1/Pd))^2 - (dp_p/2*cos(phi))^2) -
pi/Pd*cos(phi);
            mnp = 1;
        else
            rho_p = sqrt((0.5*((dp_p/2+a_p)+(dp_p/2 - a_g)))^2 -
(dp_p/2*cos(phi))^2);
            Zpg = sqrt((dp_p/2 + a_p)^2 - (dp_p/2*cos(phi))^2) +
sqrt((dp_g/2 + a_g)^2 - (dp_g/2*cos(phi))^2) - (dp_p+dp_g)/2*sin(phi);
            mpg = Pd *Zpg/pi/cos(phi);
            nr = 1 - mpg;
            pn = pi/Pd*cos(psi);
            px = pn/sin(psi);
            phin = atan(tan(phi)*cos(psi));
            psib = acos(cos(psi)*cos(phin)/cos(phi));
            mF = F/px;
            na = 1-mF;

            if na<=(1-nr)

                Lmin = (mpg*F -na*nr*px)/cos(psib);
            else
                Lmin = (mpg*F-(1-na)*(1-nr)*px)/cos(psib);
            end

            mnp = F/Lmin;

        end

        % ----- Radius of curvature of the mesh geometry -----
        rho_g = (dp_g+dp_p)/2*sin(phi) - rho_p; %
use '+ rho_p' for internal gear
        I = cos(phi)/((1/rho_p + 1/rho_g)*dp_p*mnp); %
AGMA surface geometry factor for pitting resistance; use - 1/rho_g for
internal gear

        Sc_s_pl(i,j) = Cp*sqrt(W_t/No_pl/(F*I*dp_p)*Ca*Cm*Cs*Cv);
% Compressive stress
%%
% Planets as pinion and Ring as gears
        dp_p = dp_pl;
        dp_g = dp_r;
        nu_p = nu_pl;
        E_p = E_pl;
        nu_g = nu_r;
        E_g = E_r;

        Cp = sqrt(1/(pi*((1-nu_p^2)/E_p)+((1-nu_g^2)/E_g))) ;
% Elastic coefficient for difference in p,g material

        if psi == 0;
            rho_p = sqrt((dp_p/2+(1/Pd))^2 - (dp_p/2*cos(phi))^2) -
pi/Pd*cos(phi);
            mnp = 1;

```

```

else
    rho_p = sqrt((0.5*((dp_p/2+a_p)+(dp_p/2 - a_g)))^2 -
(dp_p/2*cos(phi))^2);
    Zpg = sqrt((dp_p/2 + a_p)^2 - (dp_p/2*cos(phi))^2) +
sqrt((dp_g/2 + a_g)^2-(dp_g/2*cos(phi))^2) - (dp_p+dp_g)/2*sin(phi);
    mpg = Pd *Zpg/pi/cos(phi);
    nr = 1 - mpg;
    pn = pi/Pd*cos(psi);
    px = pn/sin(psi);
    phin = atan(tan(phi)*cos(psi));
    psib = acos(cos(psi)*cos(phin)/cos(phi));
    mF = F/px;
    na = 1-mF;

    if na<=(1-nr)

        Lmin = (mpg*F -na*nr*px)/cos(psib);
    else
        Lmin = (mpg*F-(1-na)*(1-nr)*px)/cos(psib);
    end

    mnp = F/Lmin;

end

% ----- Radius of curvature of the mesh geometry -----
rho_g = (dp_g+dp_p)/2*sin(phi) + rho_p; %
use '+ rho_p' for internal gear
I = cos(phi)/((1/rho_p - 1/rho_g)*dp_p*mnp); %
AGMA surface geometry factor for pitting resistance; use - 1/rho_g for
internal gear

Sc_pl_r(i,j) = Cp*sqrt(W_t2/No_pl/(F*I*dp_p)*Ca*Cm*Cs/Cv);
% Compressive stress

%% ///// Scuffing / Scoring Analysis /////
% P_scr - Probability of scoring; has to be less than 0.3
% t_b - Gear body temperature; has to be less than 300 °F

% ---- Sun as Pinion, Planets as Gear ----
[P_scr_s_pl, ~] = Scuffing(P, Q_s, rpm_s, Pd, Km, Kv,
N_s(i), N_pl(i), F, psi_deg, mat_s, mat_pl);
% ---- Planets as Pinion/internal, Ring as
Gear/external ----
[P_scr_pl_R, ~] = Scuffing(P, Q_pl, rpm_s*(mg-2)/2, Pd,
Km, Kv, N_pl(i), N_r(i), F, psi_deg, mat_pl, mat_r);

%% Within Stress limits ??
if (Sb_s(i,j) <= Sfb_s && Sb_pl_p(i,j) <= Sfb_pl &&
Sb_pl_g(i,j) <= Sfb_pl ...
&& Sb_r(i,j) <= Sfb_r && Sc_s_pl(i,j) <= Sfc1 &&
Sc_pl_r(i,j) <= Sfc2)
    weight_s(i,j) = pi*F*rho_den_s*((rp_s + a_p - w_p)^2
+0.5*((rp_s + a_p)^2 - (rp_s + a_p - w_p)^2));
    weight_pl(i,j) = pi*F*rho_den_pl*((rp_pl + a_g - w_g)^2 +
0.5*((rp_pl + a_g)^2 - (rp_pl + a_g - w_g)^2));

```

```

weight_r(i,j) = pi*F*rho_den_r*(rr1^2 - rr2^2 + 0.5*(rr3^2-
rr2^2));
count = count +1;
weight_carrier(i,j) = pi*F/10*rho_den_pl*(rr2^2);

else
% Weight penalties for violating constraints
if Sb_s(i,j) > Sfb_s
wps(i,j) = (Sb_s(i,j) - Sfb_s)/Sfb_s*rp;
end
if Sb_pl_p(i,j) > Sfb_pl
wppl(i,j) = (Sb_pl_p(i,j) - Sfb_pl)/Sfb_pl*rp;
end
if Sb_pl_g(i,j) > Sfb_pl
wppl(i,j) = (Sb_pl_g(i,j) - Sfb_pl)/Sfb_pl*rp;
end
if Sb_r(i,j) > Sfb_r
wpr(i,j) = (Sb_r(i,j) - Sfb_r)/Sfb_r*rp;
end

% Compressive loading violation
if Sc_s_pl(i,j) > Sfc1
wpc1(i,j) = (Sc_s_pl(i,j) - Sfc1)/Sfc1*rpc;
end
if Sc_pl_r(i,j) > Sfc2
wpc2(i,j) = (Sc_pl_r(i,j) - Sfc2)/Sfc2*rpc;
end

weight_s(i,j) = pi*F*rho_den_s*((rp_s + a_p - w_p)^2
+0.5*((rp_s + a_p)^2 - ...
(rp_s + a_p - w_p)^2)) + wps(i,j) + wpc1(i,j);
weight_pl(i,j) = pi*F*rho_den_pl*((rp_pl + a_g - w_g)^2 +
0.5*((rp_pl + a_g)^2 - ...
(rp_pl + a_g - w_g)^2)) + wppl(i,j) + wpc1(i,j) +
wpc2(i,j);
weight_r(i,j) = pi*F*rho_den_r*(rr1^2 - rr2^2 + 0.5*(rr3^2-
rr2^2)) + wpr(i,j) + wpc2(i,j);
weight_carrier(i,j) =
((2*rp_s+rp_pl)^2)/3*No_pl*F/10*rho_den_pl;
end
end
end

%% ~~~~ Return ~~~~
weight_tot = weight_s + weight_r + No_pl*weight_pl + weight_carrier;

[min_weight_tot_1,ii] = min(weight_tot,[],1);
[min_weight_tot_2,j] = min(min_weight_tot_1,[],2);
i_2 = ii(j);
weight = min_weight_tot_2;
weight_s_op = weight_s(i_2,j);
weight_r_op = weight_r(i_2,j);
weight_pl_op = weight_pl(i_2,j);
weight_car_op = weight_carrier(i_2,j);
N_s_op = N_s(i_2);
N_r_op = N_r(i_2);
N_pl_op = N_pl(i_2);

```

```

F_op = F_array(j)/Pd;
mg_op = 1 + N_r_op/N_s_op;
Sb_s_op = Sb_s(i_2,j);
Sb_r_op = Sb_r(i_2,j);
Sb_pl_p_op = Sb_pl_p(i_2,j);
Sb_pl_g_op = Sb_pl_g(i_2,j);
Sc_s_pl_op = Sc_s_pl(i_2,j);
Sc_pl_r_op = Sc_pl_r(i_2,j);
end

```

#### ***d. Hunting Ratio Function***

```

% Function to generate teeth ratios
% - Copyright 2013 Sylvester Ashok -
% Integrated Product Lifecycle Engineering Laboratory
% Georgia Institute of Technology

function [Teeth_table] = H_ratio(mg,Np_min)
Np_max = floor(200/mg);
x = max - min + 1;
N2 = zeros(1,x);
N1 = linspace(Np_min,Np_max, x);
T_table = zeros(1,x);
j = 0;
for i = 1:x
    N2(i) = round(mg*N1(i));
    F_N1 = factor(N1(i));
    F_N2 = factor(N2(i));
    if isempty(intersect(F_N1, F_N2))
        j = j+1;
        T_table(j) = N1(i);
    end
end

Teeth_table = T_table(1:j);

End

```

#### ***e. Efficiency Analysis Function***

```

% Function to perform spur, helical and
% bevel gear efficiency analysis
% - Copyright 2013 Sylvester Ashok -
% Integrated Product Lifecycle Engineering Laboratory
% Georgia Institute of Technology

function[E] = Efficiency(mg, Pd, N_p, N_g, psi, type, angle)
%% corrections
bevel = type-1;
shaft_angle = deg2rad(angle);
psi = deg2rad(psi);
%% Initializing

```

```

ad = 1/Pd; % Addendum in
dp_p = N_p/Pd; % Pitch diameter of pinion in
dp_g = N_g/Pd; % Pitch diameter of gear in
phi = deg2rad(20); % Pressure angle (20°)
r_0 = dp_p/2 + ad; % Outside radius of pinion in
r = dp_p/2; % Pitch radius of pinion in
R_0 = dp_g/2 + ad; % Outside radius of gear in
R = dp_g/2; % Pitch radius of gear in
f = 0.035; % Average coefficient of friction
H_t = (mg + 1)/mg *(sqrt((r_0/r)^2 - cos(phi)^2) - sin(phi)); %
Specific sliding velocity at end of approach action
H_s = (mg + 1)*(sqrt((R_0/R)^2 - cos(phi)^2) - sin(phi)); %
Specific sliding velocity at start of approach action

%% ## Spur and Helical ##
if bevel == 0
    if psi == 0
        % Spur
        P_t = 50*f/cos(phi)*(H_s^2 +H_t^2)/(H_s + H_t);
    % Percent power loss
    else
        % Helical
        phi_n = atan(tan(phi)*cos(psi));
    % Normal pressure angle of helical gear
        P_t = 50*f*cos(psi)^2/cos(phi_n)*(H_s^2 +H_t^2)/(H_s + H_t);
    % Percent power loss
    end
else
    %% ## Bevel ##
    ad_g_bevel = 0.540/Pd + 0.460/(Pd*mg^2); %
    Addendum gear
    ad_p_bevel = 2/Pd - ad_g_bevel; %
    Addendum pinion
    gamma = atan(1/mg); %
    Pitch cone angle of bevel pinion
    Gamma = shaft_angle - gamma; %
    Pitch cone angle of bevel gear
    N_vg = N_g/cos(Gamma); %
    Number of virtual spur gear teeth
    N_vp = N_p/cos(gamma); %
    Number of virtual spur pinion teeth
    r_0 = 1/2*(dp_p + 2*ad_p_bevel*cos(atan(1/mg))); %
    Outside radius of large end of bevel pinion
    r = dp_p/2; %
    Pitch radius of large end of bevel pinion
    R_0 = dp_g/2 + 2*ad_g_bevel*cos(shaft_angle - atan(1/mg)); %
    Outside radius of large end of bevel gear
    R = dp_g/2; %
    Pitch radius of large end of bevel gear
    H_t = (mg + 1)/mg *(sqrt((r_0/r)^2 - cos(phi)^2) - sin(phi)); %
    Specific sliding velocity at end of approach action
    H_s = (mg + 1)*(sqrt((R_0/R)^2 - cos(phi)^2) - sin(phi)); %
    Specific sliding velocity at start of approach action
    P_t = 50*f*((cos(Gamma) + cos(gamma))/cos(phi))*((H_s^2 +
    H_t^2)/(H_s + H_t)); % Percent power loss

end

```

```

E = (100 - P_t)/100;           % Efficiency percentage
end

```

### ***f. Scuffing Hazard Analysis Function***

```

% Function to perform Scuffing Hazard Assessment
% - Copyright 2013 Sylvester Ashok -
% Integrated Product Lifecycle Engineering Laboratory
% Georgia Institute of Technology

function[P_scr,t_b] = Scuffing(P, Tq, np, Pd, Km, Kv, N_p, N_g, F,
psi_deg, mat_p, mat_g)
%% Type of Lubricant
lube_type = 4;
% 1 = Carb Steel MIL-L-7808
% 2 = Carb Steel MIL-L-6081
% 3 = Carb Steel MIL-L-23699
% 4 = VASCO MIL-L-23699

%% Oil flow design type
Oil_flow_design = 1;
% 1 = Recommended
% 2 = Manual
% 3 = Minimum
% 4 = Rule of Thumb

%% Type of tooth profile modification
T_p_m = 2;
% 1 = Unmodified
% 2 = Modified (pinion drives)
% 3 = Modified (gear drives)
% 4 = Modified (smooth mesh)

%% ----- THERMAL ELASTIC FACTOR -----
X_M = 1.75; % Thermal elastic factor (martensitic steels)

%% ----- BASIC GEAR GEOMETRY -----
A_p = 1; % Type of gear (internal=-1)
A_g = 1; % Type of gear (internal=-1)
A = 1; % Type of mesh (internal=-1)
phi = deg2rad(20);
psi = deg2rad(psi_deg);
phi_t = atan(tan(phi)/cos(psi)); % Transverse pressure angle
R_ref = N_p/Pd/2;
W_t = Tq/R_ref;
Fe = F;
mg = N_g/N_p; % Gear ratio
R_p = N_p/Pd/2/cos(psi); % Standard (reference)
pitch radius in
R_g = N_g/Pd/2/cos(psi); % Standard (reference)
pitch radius in
Cr = R_g + A_p*R_p; % Operating center distance
in

```

```

Rb_p = R_p*cos(phi_t); % Base radii
Rb_g = R_g*cos(phi_t); % Base radii
phi_r = acos((A_p*Rb_g + A_g*Rb_p)/Cr); % Operating Transverse
pressure angle
pb = 2*pi*Rb_p/N_p; % Transverse base pitch
pn = pi*cos(phi)/Pd; % Normal transverse pitch
if psi ~= 0
    psi_b = acos(pn/pb); % Base helix angle
    psi_r = atan(atan(psi_b)/cos(phi_r)); % Operating helix angle
else
    psi_b = 0;
    psi_r = 0;
end

phi_nr = asin(cos(psi_b)*sin(phi_r)); % Operating normal pressure
angle
a_p = 1/Pd; % Addendum
a_g = 1/Pd; % Addendum
Ro_p = N_p/Pd/2/cos(psi) + A_p*a_p; % Outside radius in
Ro_g = N_g/Pd/2/cos(psi) + A_g*a_g; % Outside radius in
phi_o_p = acos(Rb_p/Ro_p); % Tip pressure angles
phi_o_g = acos(Rb_g/Ro_g); % Tip pressure angles
sig_p = 13; % Surface Finish rms
sig_g = 13; % Surface Finish rms
S = mean(sig_p, sig_g); % Average Surface Roughness rms
mu_m = 0.06*50/(50-S); % Mean coeff of Surface Roughness

%% ----- ## Acquiring Material Property ## -----
[~, ~, nu_p, E_p, ~, ~] = material_property(mat_p);
[~, ~, nu_g, E_g, ~, ~] = material_property(mat_g);

%% ----- HERTZIAN CONTACT BAND -----
E_r = 2/(((1-nu_p^2)/E_p) + ((1-nu_g^2)/E_g)); % Reduced modulus of
elasticity psi
%% ----- DISTANCE ALONG THE LINE OF ACTION -----
C6 = Cr*sin(phi_r); % in
C1 = max(A*(C6-sqrt(Ro_g^2-Rb_g^2)), 0); % SAP in
C3 = C6/(mg+A); % Operating pitch point in
C4 = C1+pb; % HPSTC in
C5 = sqrt(Ro_p^2 - Rb_p^2); % EAP in
C2 = C5 - pb; % LPSTC in
Z = C5 - C1; % Length of line of contact in

%% ----- ROLL ANGLES -----
e1 = C1/Rb_p; % Roll angle at C1
e2 = C2/Rb_p; % Roll angle at C2
e3 = C3/Rb_p; % Roll angle at C3
e4 = C4/Rb_p; % Roll angle at C4
e5 = C5/Rb_p; % Roll angle at C5

%% ----- LUBRICATION ANALYSIS -----
eta_mesh = 0.995; % Mesh Efficiency
P_loss = P - eta_mesh*P; % Power Dissipated
Q = 42.4*P_loss; % Heat Generated Btu/min
Cp = 0.5; % Sp. heat of oil Btu/lb-F
M_rec = 4.7;
M_min = 3.1;

```

```

M_man = 15.2;
M_rot = 5;
% 1 = Recommended
% 2 = Manual
% 3 = Minimum
% 4 = Rule of Thumb
if Oil_flow_design == 1
    Oil_flow = M_rec;
elseif Oil_flow_design == 2
    Oil_flow = M_man;
elseif Oil_flow_design == 3
    Oil_flow = M_min;
elseif Oil_flow_design == 4
    Oil_flow = M_rot;
else disp ('lube error')
end
M = 7.5*Oil_flow;
del_T = Q/(Cp*M); % Temperature rise °F 30.0
t_in = 130; % Incoming Oil temperature °F
130
t_out = t_in+ del_T; % Outgoing Oil temperature °F
160
t_oil = t_in + 0.5*del_T; % Oil temperature °F 145.0

%% ----- CONTACT RATIOS -----
mp = Z/pb; % Transverse contact ratio
if psi == 0
    mF = 0; % Axial contact ratio
else
    px = pi/sin(psi)/Pd; % Axial pitch
    mF = F/px; % Axial contact ratio
end
if mp > 2
    disp('error in mp')
end
nr = mp - floor(mp); % Fractional part of mp
na = mF - floor(mF); % Fractional part of mF

if psi == 0 && mp<2
    Lmin = Fe; % Minimum length of lines of
contact
elseif psi > 0 && na<=(1-nr)
    Lmin = (mp*Fe - na*nr*px)/cos(psi_b);
else
    Lmin = (mp*Fe-(1-na)*(1-nr)*px)/cos(psi_b);
end

%% ----- GEAR TOOTH VELOCITIES AND LOADS -----
ng = np/mg; % Speed of member rpm
w_p = np*pi/30; % Rotational (angular) velocity rad/s
w_g = ng*pi/30; % Rotational (angular) velocity rad/s
vtr = 5*w_p*R_p; % Operating pitchline velocity fpm
Wtr_norm = 33000*P/vtr; % Norminal tangential load
Ka = 1.25; % Application/overload factor
CD = Ka * Kv * Km; % Combined derating factor
Wtr = Wtr_norm*CD; % Actual tangential load lb
WNR = Wtr/cos(psi_r)/cos(phi_nr); % Normal operating load lb

```

```

Wtra = Wtr/Lmin;           % Transverse unit load           lb
WNor = WNr/Lmin;          % Normal unit load              lb

%% ----- PROFILE OF RADII OF CURVATURE -----
l_Ci = zeros(100,1);
l_Ci(1) = C1;
j = 1;
while (true)
    j = j+1;
    l_Ci(j) = (C5-C1)/40 + l_Ci(j-1);
    if abs(l_Ci(j) - C5)/C5 <= 0.005
        break
    end
end
l_Ci = l_Ci(1:j);
l_e = l_Ci/Rb_p;
l_Gy = l_Ci/C3 - 1;
if l_Gy < -1
    l_Gy = -1;
end
l_r1 = Rb_p.*l_e;
l_r2 = C6 - A*l_r1;
l_rr = (l_r1.*l_r2)./(l_r2 + A*l_r1);
l_rn = l_rr/(cos(psi_b));
l_vr1 = w_p*l_r1;
l_vr2 = w_g*l_r2;
l_vs = abs(l_vr1-l_vr2);
l_ve = (l_vr1+l_vr2);
l_XT = zeros(j,1);
for i = 1:j
    % Load sharing factor (unmodified)
    if l_e(i)>= e1 && l_e(i)<e2
        l_XT1 = 1/3*(1 + (l_e(i) - e1)/(e2-e1));
    elseif l_e(i)>= e2 && l_e(i)<= e4
        l_XT1 = 1;
    elseif l_e(i)>e4 && l_e(i)<=e5
        l_XT1 = 1/3*(1 + (e5-l_e(i))/(e5-e4));
    else
        l_XT1 = 0;
    end

    % Load sharing factor (modified tooth profiles pinion driving)
    if l_e(i)>= e1 && l_e(i)<e2
        l_XT2 = 6/7*(l_e(i)-e1)/(e2-e1);
    elseif l_e(i)>= e2 && l_e(i)<= e4
        l_XT2 = 1;
    elseif l_e(i)>e4 && l_e(i)<=e5
        l_XT2 = 1/7 + 6/7*((e5-l_e(i))/(e5-e4));
    else
        l_XT2 = 0;
    end

    % Load sharing factor (modified tooth profiles gear driving)
    if l_e(i)>= e1 && l_e(i) <e2
        l_XT3 = 1/7 + 6/7*(l_e(i) - e1)/(e2-e1);
    elseif l_e(i) >= e2 && l_e(i) <= e4
        l_XT3 = 1;
    elseif l_e(i) >e4 && l_e(i)<=e5

```

```

        l_XT3 = 6/7*((e5-l_e(i))/(e5-e4));
else
    l_XT3 = 0;
end

%Load sharing factor (designed for smooth meshing)
if l_e(i)>= e1 && l_e(i)<e2
    l_XT4 = (l_e(i)-e1)/(e2-e1);
elseif l_e(i)>= e2 && l_e(i)<= e4
    l_XT4 = 1;
elseif l_e(i)>e4 && l_e(i)<=e5
    l_XT4 = ((e5-l_e(i))/(e5-e4));
else
    l_XT4 = 0;
end

% 1 = Unmodified
% 2 = Modified (pinion drives)
% 3 = Modified (gear drives)
% 4 = Modified (smooth mesh)s
if T_p_m == 1
    l_XT(i) = l_XT1;
elseif T_p_m == 2
    l_XT(i) = l_XT2;
elseif T_p_m == 3
    l_XT(i) = l_XT3;
else l_XT(i) = l_XT4;
end
end

l_bh = sqrt(8*l_XT*WNor.*l_rn/(pi*E_r));
l_XG = 0.51*sqrt(mg+A)*(abs(sqrt(1+l_Gy)-sqrt(1-A*l_Gy/mg))./...
    (((1+l_Gy).^0.25).*((mg-A*l_Gy).^0.25)));
l_tfl = 0.45*mu_m*X_M*l_XG.*(l_XT*Wtra).^0.75*sqrt(vtr)/(Cr)^0.25;

[U,j] = max(l_tfl);
%% ----- PROFILE OF RADII OF CURVATURE -----

Gi = l_Gy(j);           % Parameter on line of action
e = ((Gi+1)*C3)/Rb_p;   % Roll angle           degrees
rho_p = e*Rb_p;         % Transverse radii of curvature at general
contact point
rho_g = C6 - A*rho_p;   % Transverse radii of curvature at general
contact point
rr = (rho_p*rho_g)/(rho_g + A*rho_p);           % Transverse relative
radius of curvature in
rrc = mg/((mg+A)^2)*Cr*sin(phi)/cos(psi_b); % Normal relative radius of
curvature in
rn = rr/cos(psi_b);     % Equivalent radius of a
cylinder that represents the gear pair curvatures in contact along the
line of action in

%% ----- GEAR TOOTH VELOCITIES AND LOADS -----
vr_p = rho_p*w_p;
vr_g = rho_g*w_g;       % Rolling velocities    in/s
vs = abs(vr_p - vr_g); % Sliding velocity    in/s
ve = abs(vr_p + vr_g); % Entraining velocity in/s

```

```

%% ----- FLASH TEMPERATURE INDEX (DUDLEY/AMCP) -----

Zt = abs(0.0175*(sqrt(rho_p)-sqrt(N_p/N_g*rho_g))/...
        (cos(phi)^0.75*(rho_p*rho_g/(rho_p+rho_g))));% Geometry constant
W_te = Km*W_t; % Effective tangential
load lb
Fe = F; % Effective face width
in
s = sqrt(sig_p^2 + sig_g^2); % Mean Surface Finish
rms
t_b = t_out; % Gear body temperature
°F
t_flash = W_te^0.75/Fe*(50/(50-s))*Zt*sqrt(np)/(Pd^0.25); % Flash
temperature °F
t_f = t_b + t_flash; % Flash temperature
index °F
Low_risk = 300; % Low Risk of Scoring
High_risk = 350; % High Risk of Scoring
if t_b <= Low_risk
    Scr_risk = 1; % Risk of Scoring
Low
else
    Scr_risk = 2; % Risk of Scoring
High
end

%% ----- // SCUFFING SUMMARY //-----

t_fl_max = U; % Max Flash Temperature
°F
t_M = -24 +1.2*t_oil+0.56*t_fl_max; % Bulk temperature
°F
tcmax = t_M +t_fl_max; % Maximum contact
temperature °F

% 1 = Carb Steel MIL-L-7808
% 2 = Carb Steel MIL-L-6081
% 3 = Carb Steel MIL-L-23699
% 4 = VASCO MIL-L-23699

% Mean scuffing temperature °F
% Standard temperature deviation °F
if lube_type == 1
    mu_s = 366;
    st_s = 56.6;
elseif lube_type == 2
    mu_s = 264;
    st_s = 74.4;
elseif lube_type == 3
    mu_s = 391;
    st_s = 58.65;
elseif lube_type == 4
    mu_s = 459;
    st_s = 31;
else disp ('lube_type error')

```

```

end

tcmax;
if isreal(tcmax) ~= 1
    % disp('tcmax error')
    % disp(tcmax)
    P_scr = 1;
else
    P_scr = normcdf(tcmax,mu_s,st_s); % Probability of scoring hazard
end
% Scuffing Risk
% 1 = Low
% 2 = Medium
% 3 = High
if P_scr <0.1
    S_risk = 1;
elseif P_scr >0.3
    S_risk = 3;
else S_risk = 2;
end
S_f = (mu_s - t_oil) / (tcmax - t_oil); % Safety Factor
end

```

### ***g. Genetic Algorithm for Four Stage Gear Train***

```

% Genetic algorithm for FOUR stage spur gear train
% with adaptive crossover and mutation, and multiple
% subpopulations with migration capability, multiple
% elitist, mutation, random member generation, and
% exterior penalty functions, and tournament and
% roulette wheel selection methods
% - Copyright 2013 Sylvester Ashok -
% Integrated Product Lifecycle Engineering Laboratory
% Georgia Institute of Technology

% ~~ FOUR STAGE ~~

% Genetic Algorithm with multiple sub-popsulation and migration

% Genetic Algorithm I Inputs
conv_criteria = 150; % generations with same best Cost
gen_max = 10000; % maximum number of generations
P_c = 0.7; % crossover probability
AGA_c = 1; % Adaptive crossover rate switch
type_c = 2; % 1 = tournament, 2 = roulette wheel
P_m = 0.01; % mutation probability
AGA_m = 1; % Adaptive mutation rate switch
P_mi = 0.05; % migration probability
inf = 1/0; % infinity
n = 80; % population size; needs to be a factor of 4
n_sub_pop = 5; % number of subpopulations
el_no = 1; % number of elitist members to hold
mut_no = ceil(1*n); % number of members to mutate

```

```

penalty_type = 3;          % 1 = death penalty, 2 = static linear, 3 =
static nonlinear, 4 = dynamic linear, 5 = dynamic nonlinear
ppf = 30;                 % pinion bending penalty factor
gpf = 30;                 % gear bending penalty factor
cpf = 60;                 % contact penalty factor
vpf = 10;                 % volume penalty
vol_switch = 0;          % switch for spacing analysis
penalty_factor = [ppf, gpf, cpf, ppf, gpf, cpf, ppf, gpf, cpf, ppf,
gpf, cpf];
% Design Inputs
mg_req = 25;              % required gear ratio
P = 500;                  % power in HP
rpm = 35000;             % input rpm
IP = [P,mg_req,rpm];

% Design configuration constraints
mg_max = 3.8;             % max gear ratio per stage allowed
mg_min = 1.7;            % min gear ratio per stage allowed

% Variable Ranges
no_var = 7;              % number of variables
mg_lb = 1.65;            % gear ratio upper bound
mg_ub = 3.75;            % gear ratio lower bound
mg_length = 16;          % gear ratio bit length
Pd_values = [5, 6, 8, 10, 12, 14, 16, 18];
Pd_length = length(Pd_values);

bit(1:4) = log(Pd_length)/log(2);          % bit length for Pd variables
bit(5:7) = log(mg_length)/log(2);          % bit length for mg variables

bit_size = sum(bit);                          % total bit length

% Population database
% 1st column = ID
% 2nd column = Cost value; default = 0;
% 3rd column = constraints violated or not; default = 0, violated = 1
% 4th - 27st column = stress and allowable stress
% 28nd - 31st column = pinion teeth array
Pop_db = zeros(2^bit_size,31);
Pop_db(:,1) = 1:2^bit_size;

% Intializing for speed
child = zeros(n,bit_size+1);                % child in new population, either
product of xc or = parent
child_id = zeros(n,1);                       % child_id gc to dec
F_child = zeros(n,4);                        % weight array of child
Pd_array_child = zeros(n,4);                 % diametral pitch array decoded
mg_array_child = zeros(n,3);                 % gear ratio array decoded
Cost_child = zeros(n,1);                     % Cost (weight) of child
g_child = zeros(n,12);                       % constraint violation information
array; col 1 = Sb_p, 2 = Sb_g, 3 = Sc
g_ch = zeros(n,1);                           % constraint violation information
stress_child = zeros(n,24);                  % col 1 = Sb_p, 2 = Sfb_p, 3 =
Sb_g, 4 = Sfb_g, 5 = Sc, 6 = Sfc
sum_F_child = zeros(n,1);                    % sum of weight

```

```

cost_child = zeros(n,1);
new_pop = zeros(n, bit_size+1, n_sub_pop);
fn_call = 0;      % function call counter
conv = 0;        % convergence counter
gen = 1;         % generation number
mutation_count = 0;
F = zeros(n,4);
g = zeros(n,12);
N_p = zeros(n,4);
stress = zeros(n,24);
cost = zeros(n,1);
%% Initial Population
for l = 1:n_sub_pop
    % Generates 'n' chromosomes of length 'length'
    x = zeros(n,bit_size);
    y = zeros(n,no_var);
    Pd_array = zeros(n,4);
    mg_array = zeros(n,3);

    % Generation 1
    for i = 1:n
        for j = 1: bit_size
            x(i,j) = round(rand(1));
        end
    end

    %% Decode
    for i = 1:n
        [Pd_array(i,:),mg_array(i,:)] = de_code4(x(i,:),bit,X);
    end

    %% Filter
    % Filter for max, min and whole number mg_3 constraint
    % Initial population starts in feasible design space

    for i = 1:n
        mg_4 = mg_req/mg_array(i,1)/mg_array(i,2)/mg_array(i,3);
        if mg_4 > mg_max || mg_4 < mg_min || abs(mg_4 -
round(mg_4))/mg_4 < 0.02
            while (true)
                for j = 1 : bit_size
                    x(i,j) = round(rand(1));
                end
                [Pd_array(i,:),mg_array(i,:)] = de_code4(x(i,:),bit,X);

                mg_4 =
mg_req/mg_array(i,1)/mg_array(i,2)/mg_array(i,3);
                if mg_4 < mg_max && mg_4 > mg_min && (abs(mg_4 -
round(mg_4))/mg_4) > 0.02
                    break
                end
            end
        end
    end

    end

    %% Weight and Stress evaluation of members

```

```

    for i = 1:n
        [F(i,:),g(i,:),stress(i,:),N_p(i,:)] = GA_four_stage(P, rpm,
Pd_array(i,:), mg_array(i,:), mg_req) ;
        fn_call = fn_call + 1;
    end
    penalty = zeros(n,12);

    %% Constraint and Penalty estimation
    for i = 1:n

        if max(g(i,:)) > 0
            k = 1;
            for j = 1:12 % going through each constraint of the 12
                if stress(i,k) > stress(i,k+1)
                    penalty(i,j) = penalty_function(stress(i,k),
stress(i,k+1), penalty_type, penalty_factor(j), gen);
                end
                k = k+2;
            end
        end

        cost(i) = sum(F(i,:)) + sum(penalty(i,:));
    end

    new_gen(:,1:bit_size,1) = x; % new generation
    new_gen(:,bit_size + 1,1) = cost'; % new generation gets
cost information
    id = gc2dec(x);
    g2 = max(g, [], 2); % returns all designs with
stages with any constraint violated
    index_g = g2>0; % returns indices of violated
designs
    Pop_db(id,2) = sum(F,2); % insert weight information to
population database cost = weight + penalty
    Pop_db(id,3) = index_g; % insert constraint violation
information to population database
    Pop_db(id,4:27) = stress; % insert stress information to
population database
    Pop_db(id,28:31) = N_p;
end

for l = 1: n_sub_pop
    sort_gen = new_gen(:, :, l);
    [sort_fit, IX_el] = sort(sort_gen(:, bit_size + 1)); %
sorted cost and indices
    elitist(:, :, l) = sort_gen(IX_el(1:el_no), 1:bit_size+1); %
elitist for present generation
    Fittest(l, gen) = sort_gen(IX_el(1), bit_size+1); %
fittest in that generation
    el_id = fliplr(1:el_no); %
array to run check on elitist member; if there are 2 elitist, el_id =
[2 1]
end

error = 0;

```

```

%% ~~~~ Evolution begins here ~~~~
while (true)
    % new_gen --> new_sub_pop -->(selection)-> parent -(xc)-> child -
    (mutation)-> new_pop -(elitist update)-> new_gen
    % sum_F_child = sum(F_child)
    % cost_child = sum_F_child + penalty
    for l = 1:n_sub_pop
        new_sub_pop = new_gen(:, :, l);

        %% *** CROSSOVER ***
        % Select breeding parents
        if type_c == 1
            parent = tournament(new_sub_pop); % tournament selection
        else
            parent = roulette_wheel(new_sub_pop); % roulette wheel
selection
        end
        ptp = randperm(n); % parent picked to perform
crossover

        % ~~~~~ Adaptive GA ~~~~~
        % calculate P_m for AGA
        if AGA_c == 1
            if isnumeric(std(parent(:, bit_size+1))) == 1 &&
penalty_type ~= 1
                k1 = 0.7;
                if std(parent(:, bit_size+1)) < 0.2*Fittest(l, gen)
                    k1 = 0.9;
                elseif std(parent(:, bit_size+1)) < 0.3*Fittest(l, gen)
                    k1 = 0.8;
                end
                P_cc = k1*Fittest(l, gen)/mean(parent(:, bit_size+1));
            else
                P_cc = P_c;
            end

            P_cc_data(gen, l) = P_cc;
        else P_cc = P_c;
        end

        % Perform two point crossover
        j = 1;
        for i = 1: n/2
            % ~~~~~

            if rand(1) < P_cc % Perform crossover if
probability satisfied
                loc_1 = ceil(bit_size*(rand(1))); % location 1 for xo
                loc_2 = ceil(bit_size*(rand(1))); % location 2 for xo
                x_loc_1 = min(loc_1, loc_2);
                x_loc_2 = max(loc_1, loc_2);
                [child(j, :), child(j+1, :)] =
crossover(parent(ptp(j), :), parent(ptp(j+1), :), x_loc_1, x_loc_2); %
crossover fn
            else
                child(j, :) = parent(ptp(j), :); % child = parent,
otherwise

```

```

        child(j+1,:) = parent(ftp(j+1),:); % child = parent,
otherwise
    end
    j = j + 2;
end

%% New Population

for i = 1:n
    child_id(i) = gc2dec(child(i,1:bit_size)); % gray decoded
binary string
    [Pd_array_child(i,:),mg_array_child(i, :)] =
de_code4(child(i,:),bit,X); % Decode
    if Pop_db(child_id(i),2) == 0 % decode and evaluate
weight and stress only if cost has not been already evaluated.

        %FILTER
        % Filter for max, min and whole number mg_3 constraint
        % if a child is found ineligible, filtering will remove
it from current
        % gene pool - death penalty

        mg_4 =
mg_req/mg_array_child(i,1)/mg_array_child(i,2)/mg_array_child(i,3);
        if mg_4 > mg_max || mg_4 < mg_min || abs(mg_4 -
round(mg_4))/mg_4 < 0.02
            child(i,1:bit_size) = parent(i,1:bit_size); %
##replace it with the parent instead
            sum_F_child(i) = parent(i,bit_size+1); % change
fitness information
            child_id(i) = gc2dec(parent(i,1:bit_size)); %
change the child_id
            stress_child(i,:) = Pop_db(child_id(i),4:21);
            continue % ##CONTINUE AND DONT
EVALUATE cost
        end
        % EVALUATE COST
        fn_call = fn_call + 1;
        [F_child(i,:),g_child(i,:),stress_child(i,:),
N_p_child(i,:)] ...
            = GA_four_stage(P, rpm, Pd_array_child(i,:),
mg_array_child(i,:), mg_req) ;
            sum_F_child(i) = sum(F_child(i,:)); % total
weight
            g_ch(i) = max(g_child(i,:));
            Pop_db(child_id(i),2) = sum_F_child(i); % input
total weight to database
            Pop_db(child_id(i),3) = g_ch(i);
            Pop_db(child_id(i),4:27) = stress_child(i,:); % input
stresses to database
            Pop_db(child_id(i),28:31) = N_p_child(i,:); % input
pinion teeth information
        else
            sum_F_child(i) = Pop_db(child_id(i),2);
            g_ch(i) = Pop_db(child_id(i),3);
            stress_child(i,:) = Pop_db(child_id(i),4:27);
            N_p_child(i,:) = Pop_db(child_id(i),28:31);

```

```

        end
    end

    %% ## Constraint and Penalty estimation of new population ##
    penalty_child = zeros(n,12); % reset penalties to zero
    for i = 1:n
        % stress constraint check
        if g_ch(i) > 0
            k = 1;
            for j = 1:12 % going through each constraint of the 9
                if stress_child(i,k) > stress_child(i,k+1)
                    penalty_child(i,j) =
penalty_function(stress_child(i,k), stress_child(i,k+1), penalty_type,
penalty_factor(j), gen);
                end
                k = k+2;
            end
        end

        cost_child(i) = sum_F_child(i) + sum(penalty_child(i,:));
    end

    child(:,bit_size + 1) = cost_child;

    %% MUTATION
    if AGA_m == 1
        % calculate P_m for AGA
        if isnumeric(std(cost_child)) == 1 && penalty_type ~= 1
            k2 = P_m;
            if std(cost_child) < 0.1*Fittest(1,gen)
                k2 = 0.15;
            elseif std(cost_child) < 0.2*Fittest(1,gen)
                k2 = 0.1;
            end
            P_mm = k2*Fittest(1,gen)/mean(cost_child);
        else
            P_mm = P_m;
        end

        P_mm_data(gen,1) = P_mm;
    else
        P_mm = P_m;
    end

    [~,IX_fit] = sort(child(:,bit_size + 1), 'descend'); % sorted
    cost and indices - best to worst
    new_pop = child;
    for i = 1 : mut_no
        penalty_new_member = zeros(12,1); % reset
penalty to 0
        space_pen_new_member = 0;
        if rand(1) <= P_mm
            new_member = mutate(new_pop(IX_fit(i,:),:), bit_size);
            new_member_id = gc2dec(new_member(1:bit_size));
% gray decoded binary string
            if Pop_db(new_member_id,2) == 0
% check if weight and stress value has been already loaded

```

```

        % Decode
        [Pd_array_new_member,mg_array_new_member] =
de_code4(new_member,bit,X);

        %# FILTER
        % Filter for max, min and whole number mg_3
constraint
        % if the new_member is found ineligible, filtering
will remove it from current
        % gene pool. Replace with original string

        mg_4 =
mg_req/mg_array_new_member(1)/mg_array_new_member(2)/mg_array_new_membe
r(3);
        if mg_4 > mg_max || mg_4 < mg_min || abs(mg_4 -
round(mg_4))/mg_4 < 0.02
            new_member = new_pop(IX_fit(i),:); % ##replace
it with the original string
            continue %
##CONTINUE AND DONT EVALUATE weight and stress
        end

        %# EVALUATE COST
        fn_call = fn_call + 1;
        [F_new_member,g_new_member,stress_new_member,
N_p_new_member] =...
            GA_four_stage(P, rpm, Pd_array_new_member,
mg_array_new_member, mg_req) ;
        sum_F_new_member = sum(F_new_member); % sum
of weight

        g_nm = max(g_new_member);
        Pop_db(new_member_id,2) = sum_F_new_member; %
insert weight information to population database
        Pop_db(new_member_id,3) = g_nm;
        Pop_db(new_member_id,4:27) = stress_new_member;
        Pop_db(new_member_id,28:31) = N_p_new_member;
    else
        sum_F_new_member = Pop_db(new_member_id,2);
        g_nm = Pop_db(new_member_id, 3);
        stress_new_member = Pop_db(new_member_id,4:27);
        N_p_new_member = Pop_db(new_member_id,28:31);
    end

    ### Constraint and Penalty estimation of new member ##
    if g_nm > 0
        k = 1;
        for j = 1:12 % going through each constraint of the
9 total 3x3
            if stress_new_member(k) >
stress_new_member(k+1)
                penalty_new_member(j) =
penalty_function(stress_new_member(k), stress_new_member(k+1),...
                    penalty_type, penalty_factor(j), gen);
                end
                k = k+2;
            end
        end
    end
end

```

```

        cost_new_member = sum_F_new_member +
sum(penalty_new_member);
        new_member(bit_size+1) = cost_new_member;
        new_pop(IX_fit(i),:) = new_member; % update
new_generation with new_member
        mutation_count = mutation_count+1;
    end
end

%% ADD A RANDOM MEMBER
new_rand_member = zeros(1,bit_size+1);
for j = 1: bit_size
    new_rand_member(1,j) = round(rand(1));
end
new_member_id = gc2dec(new_rand_member(1,1:bit_size));
if Pop_db(new_member_id,2) ==0
    % Decode
    [Pd_array_new_member,mg_array_new_member] =
de_code4(new_rand_member,bit,X);

    %# FILTER
    % Filter for max, min and whole number mg_3 constraint
    % if the new_member is found ineligible, filtering will
remove it from current
    % gene pool. Replace with original string

    mg_4 =
mg_req/mg_array_new_member(1)/mg_array_new_member(2)/mg_array_new_membe
r(3);
    if mg_4 > mg_max || mg_4 < mg_min || abs(mg_4 -
round(mg_4))/mg_4 < 0.02
        new_rand_member = new_pop(1,:); % ##replace it with
the original string
        continue % ##CONTINUE AND
DONT EVALUATE weight and stress
    end

    %# EVALUATE COST
    fn_call = fn_call + 1;
    [F_new_member,g_new_member,stress_new_member,
N_p_new_member] =...
    GA_four_stage(P, rpm, Pd_array_new_member,
mg_array_new_member, mg_req) ;
    sum_F_new_member = sum(F_new_member); % sum of weight
    g_nm = max(g_new_member);
    Pop_db(new_member_id,2) = sum_F_new_member; % insert
weight information to population database
    Pop_db(new_member_id,3) = g_nm;
    Pop_db(new_member_id,4:27) = stress_new_member;
    Pop_db(new_member_id,28:31) = N_p_new_member;
else
    sum_F_new_member = Pop_db(new_member_id,2);
    g_nm = Pop_db(new_member_id,3);
    stress_new_member = Pop_db(new_member_id,4:27);
    N_p_new_member = Pop_db(new_member_id, 28:31);
end
penalty_new_member = zeros(12,1);

```

```

space_pen_new_member = 0;

% constraint violation
if g_nm > 0
    k = 1;
    for j = 1:12 % going through each constraint of the 9 total
3x3        if stress_new_member(k) > stress_new_member(k+1)
            penalty_new_member(j) =
penalty_function(stress_new_member(k), stress_new_member(k+1), ...
                penalty_type, penalty_factor(j), gen);
            end
            k = k+2;
        end
    end

    new_rand_member(1,bit_size+1)= sum(penalty_new_member) +
sum_F_new_member;
    md = mode(new_pop(:,bit_size+1)); % replace the most
commonly occuring member and replace it
    IX = find(new_pop(:,bit_size+1) == md,1);
    new_pop(IX,:) = new_rand_member;

%% ELITIST CHECK
if el_no ~= 0
    new_extended_list(1:n,:) = new_pop;

    % Check elitist constraint
    for i = 1: el_no
        id = gc2dec(elitist(i,1:bit_size,1));
        F_el = Pop_db(id,2);
        g_el = Pop_db(id,3);
        stress_el = Pop_db(id,4:21);
        space_pen_el = 0;
        penalty_el = 0;
        if g_el >0
            k = 1;
            for j = 1:9
                if stress_el(k) > stress_el(k+1)
                    penalty_el(j) =
penalty_function(stress_el(k), stress_el(k+1), penalty_type,
penalty_factor(j), gen);
                end
                k = k+2;
            end
        end
        if vol_switch == 1
            space_el = housing_GA(Pd_array_child(i,:),
mg_array_child(i,:), N_p_child(i,:), mg_req);
            if space_el~=0
                space_pen_el(i) =
penalty_function_space(space_el(i), penalty_type, vpf, gen);
            end
        end
        elitist(i,bit_size+1,1)= F_el + sum(penalty_el) +
space_pen_el;
    end
end

```

```

        new_extended_list(n+1: n+el_no,:) = elitist(:, :, 1);
        [~,IX_fit] = sort(new_extended_list(:,bit_size + 1));    %
re-sorted cost and indices - best to worst
        elitist_new_gen = new_extended_list(IX_fit(1:el_no),:); %
elitist for for present generation
        [~,IX_fit_2] = sort(new_pop(:,bit_size + 1), 'descend');
% sorted cost and indices - worst to best

        % Loading elitist values to new_gen
        for i = 1: el_no
            new_pop(IX_fit_2(i),:) = elitist_new_gen(i,:);
% new_pop gets updated with elitist values
        end
        elitist(:, :, 1) = elitist_new_gen;
    end
    new_gen(:, :, 1) = new_pop;                                % setting
new_gen = new_pop before doing elitist change

end

%% Migration
% migrate members between sub-populations
if n_sub_pop > 1
new_gen = migrate(new_gen, P_mi);
end
%% Update generation

gen = gen + 1;
disp('currently on gen')
disp(gen)

%% Convergence
for l = 1:n_sub_pop
    conv_gen = new_gen(:, :, l);
    [~,IX_fit] = sort(conv_gen(:,bit_size + 1));    % re-sorted
cost and indices - best to worst
    Fittest(l,gen) = conv_gen(IX_fit(1),bit_size+1);    % Fittest
in that generation
    ix = conv_gen(:,bit_size+1) ~= inf;
    perf(l,:) = [std(conv_gen(:,bit_size+1)), mean(conv_gen(ix,
bit_size + 1))];
end

if Fittest(:,gen) < Fittest(:,gen-1)
    conv = 0; % convergence set to 0, if fittest member has
changed
elseif Fittest(:,gen) == Fittest(:,gen - 1)
    conv = conv + 1; % convergence increased if fittest member has
not changed
end
measure(gen,:) = mean(perf);

if conv >= conv_criteria
    disp ('solution converged')
    break

```

```

end
if gen >= gen_max
    disp ('exceeded max generations')
    break
end
end
end

```

### ***h. Four Stage Gear Analysis Function***

```

% Function to analyze four stage gear
% - Copyright 2013 Sylvester Ashok -
% Integrated Product Lifecycle Engineering Laboratory
% Georgia Institute of Technology

function [weight, g, S_out, N_p_out, N_g_out, FW] = GA_four_stage(P,
rpm_in, Pd, mg, mg_req)
rpm = zeros(4,1);
rpm(1) = rpm_in;
% Initializing
weight = zeros(1,4);
g = zeros(1,12);
S_out = zeros(1,4*6);
N_p_out = zeros(1,4);
N_g_out = zeros(1,4);
FW = zeros(1,4);
j = 1;
k = 1;
for i = 1:3
% function calls the sizer for the ith stage and produces results of
[weight(1,i),g(1,j:j+2),S_out(1,k:k+5),mg(i), N_p_out(i),
N_g_out(i), FW(i)] = ...
    GA_spur_sizer(P, rpm(i), Pd(i), mg(i)) ;
rpm(i+1) = rpm(i)/mg(i);
j = j+3;
k = k+6;
end
mg(4) = mg_req/mg(1)/mg(2)/mg(3);
[weight(1,4),g(1,j:j+2), S_out(1,k:k+5),mg(4), N_p_out(4), N_g_out(4),
FW(4)]...
    = GA_spur_sizer(P, rpm(4), Pd(4), mg(4));
End

```

### ***i. Sub-Population Migration Function***

```

% Function to perform Migration
% between multiple sub-population
% - Copyright 2013 Sylvester Ashok -
% Integrated Product Lifecycle Engineering Laboratory
% Georgia Institute of Technology

```

```

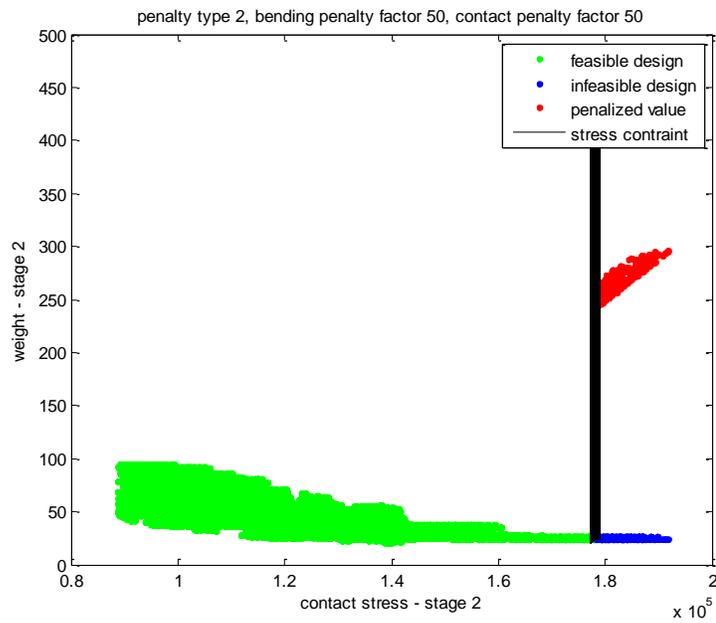
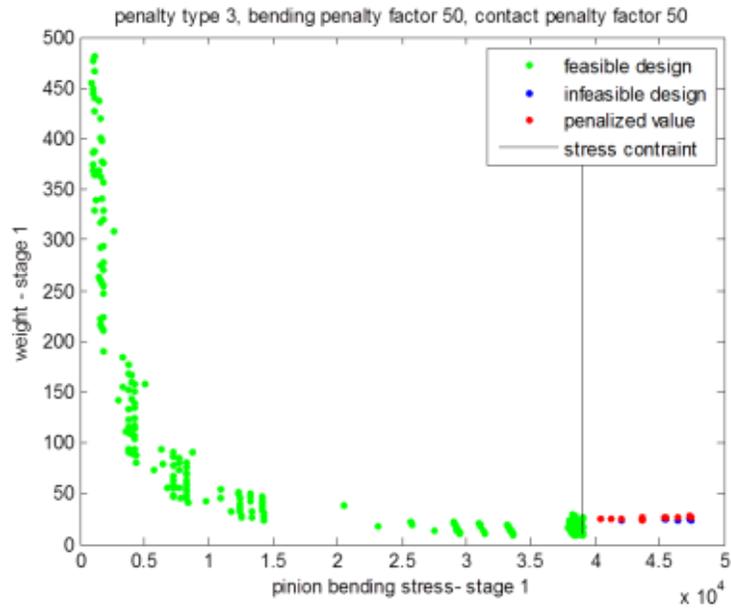
function [pop] = migrate(pop, P_mi)

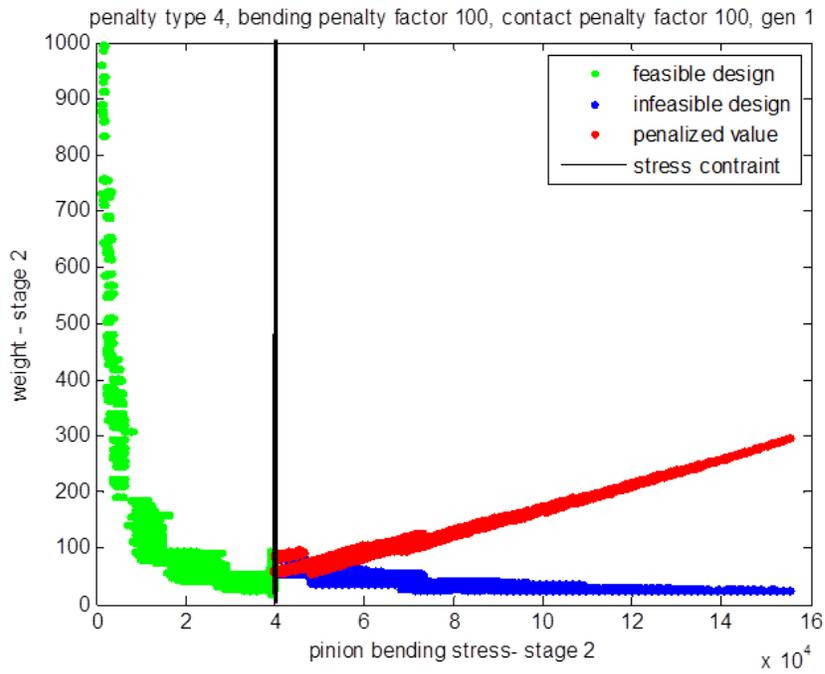
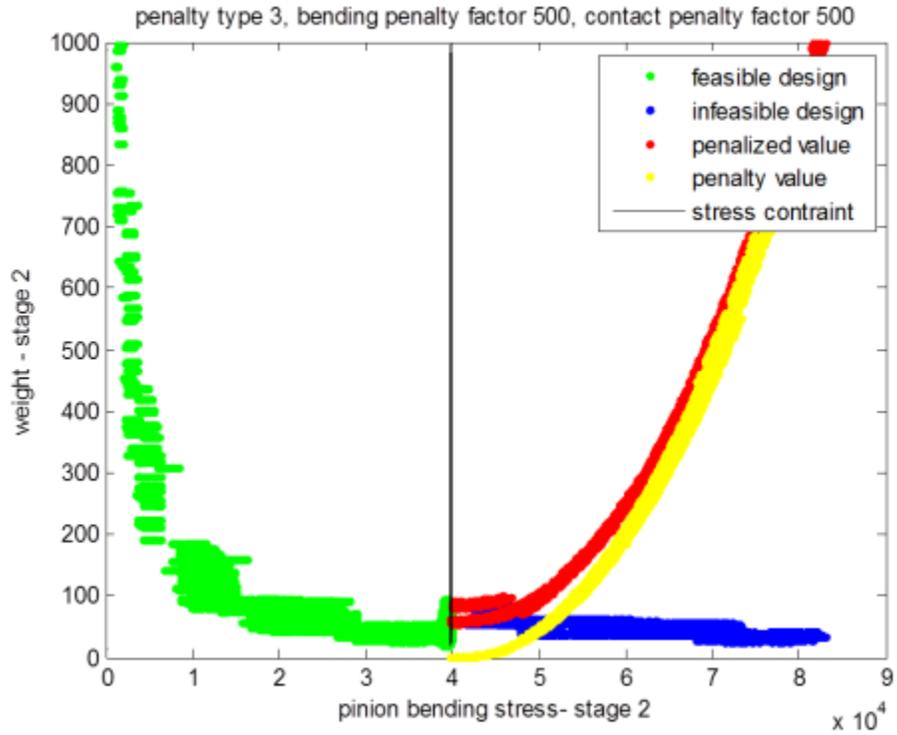
% pop: entire population from generation
% P_mi: probability of migration
% n: length of population
% n_sub_pop: number of sub populations

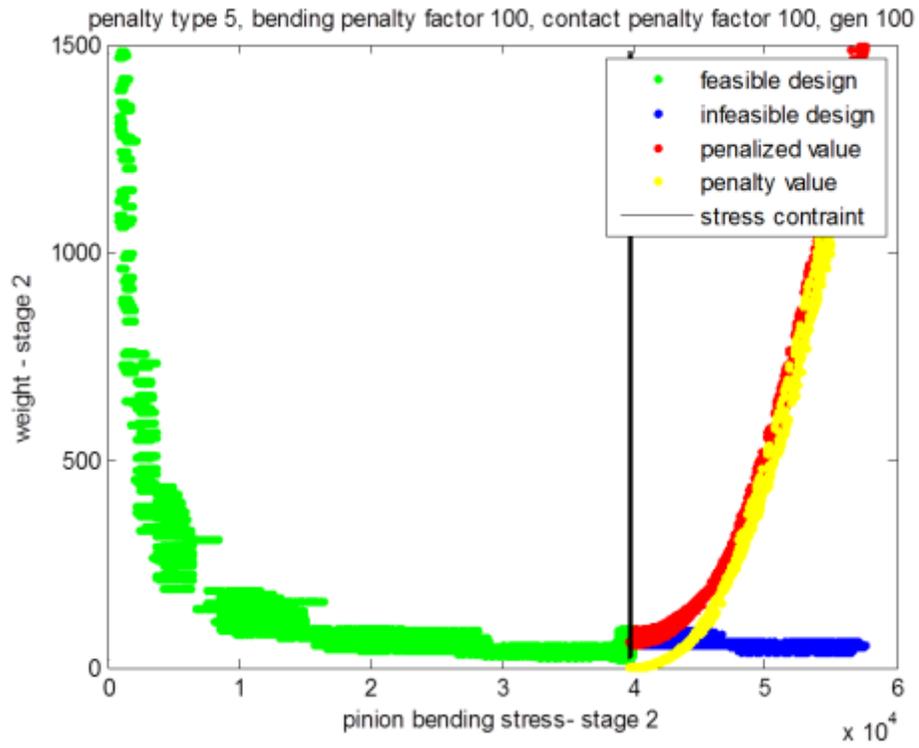
[n,~,n_sub_pop] = size(pop);
for i = 1: n
    a = rand(1);
    if a < P_mi
        b = randperm(n_sub_pop);
        for l = 1: n_sub_pop
            temp_new_gen(l,:) = pop(i,:,b(l));
        end
        pop(i,:,:) = temp_new_gen';
    end
end
end

```

### A.3. Additional Figures and Tables from Optimization Study



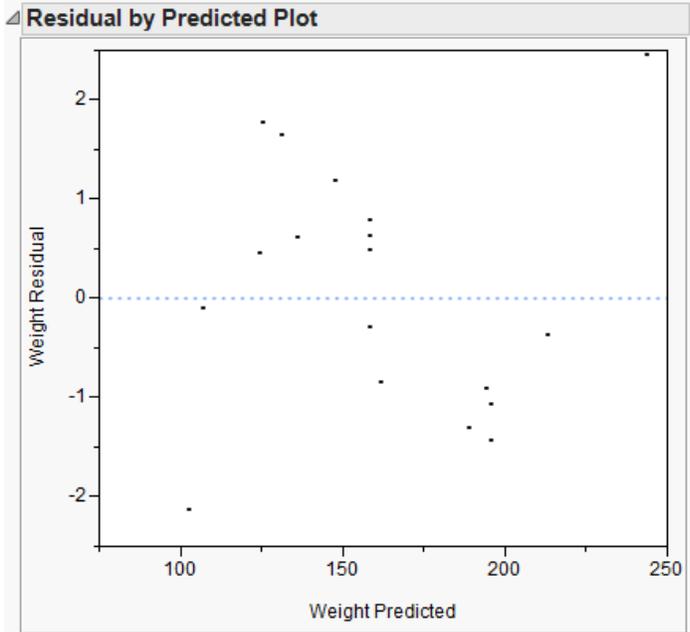
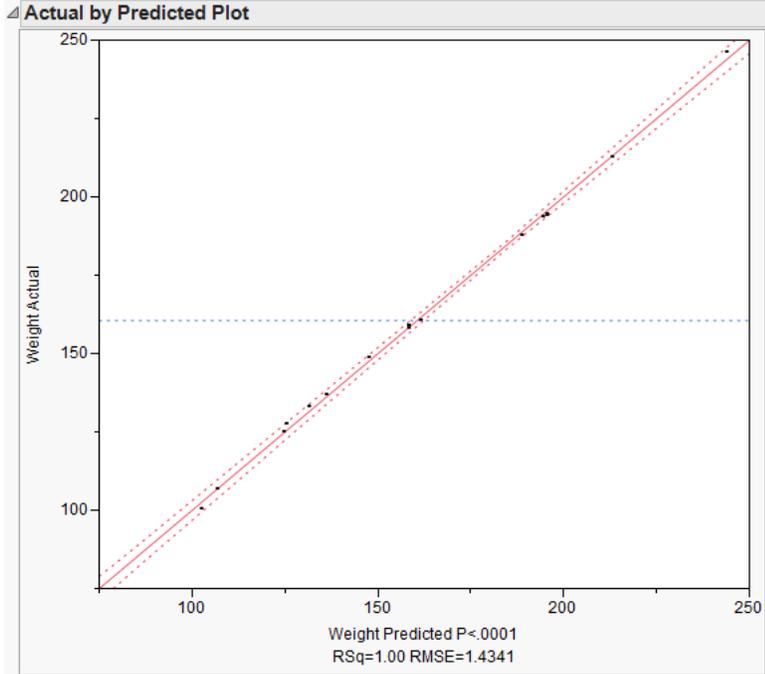




**A.4. Model Fit for Flexibility Study**

**a. Three-Stage Gear Train**

Case 1



### Summary of Fit

RSquare	0.998911
RSquare Adj	0.998156
Root Mean Square Error	1.434089
Mean of Response	160.557
Observations (or Sum Wgts)	23

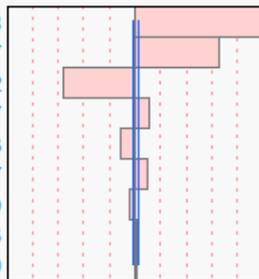
### Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob> t
Intercept	158.31632	0.477521	331.54	<.0001*
Power(400,600)	31.842392	0.389378	81.78	<.0001*
rpm_in(30000,37500)	-17.84016	0.389378	-45.82	<.0001*
mg_req(22,28)	20.938012	0.389378	53.77	<.0001*
Power*rpm_in	-4.527435	0.507027	-8.93	<.0001*
Power*mg_req	4.7513149	0.507027	9.37	<.0001*
rpm_in*mg_req	-1.772846	0.507027	-3.50	0.0039*
Power*Power	0.6050179	0.364462	1.66	0.1208
rpm_in*rpm_in	2.7601522	0.364462	7.57	<.0001*
mg_req*mg_req	0.4340558	0.364462	1.19	0.2550

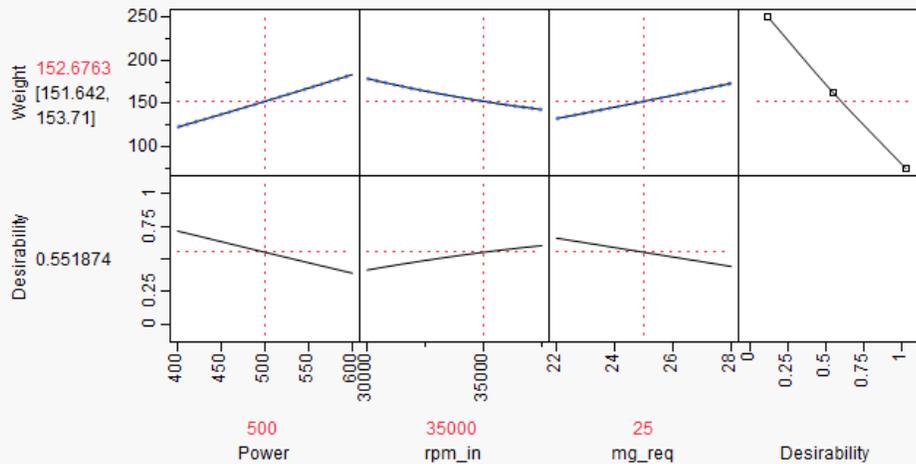
### Effect Tests

### Sorted Parameter Estimates

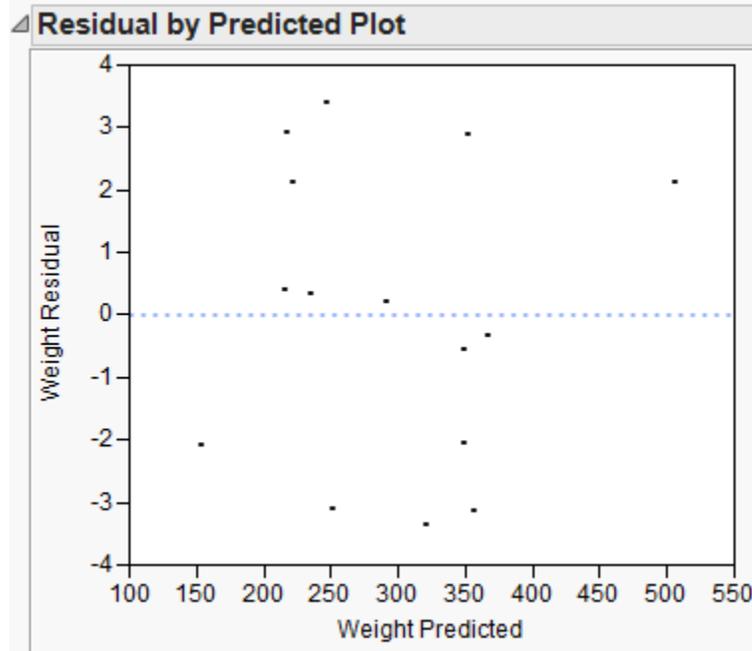
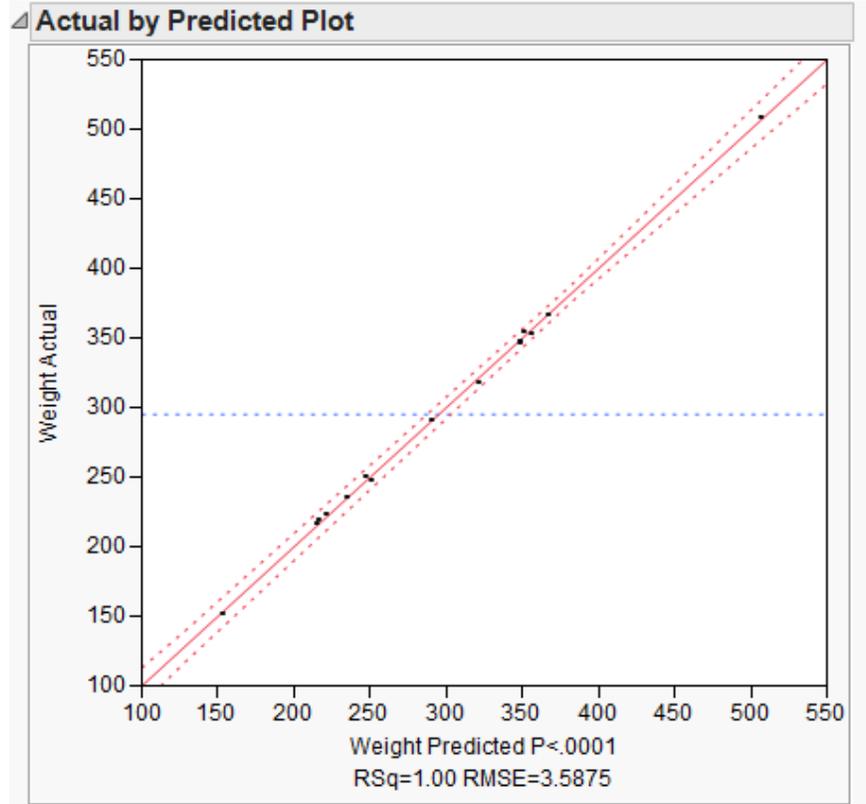
Term	Estimate	Std Error	t Ratio	Prob> t
Power(400,600)	31.842392	0.389378	81.78	<.0001*
mg_req(22,28)	20.938012	0.389378	53.77	<.0001*
rpm_in(30000,37500)	-17.84016	0.389378	-45.82	<.0001*
Power*mg_req	4.7513149	0.507027	9.37	<.0001*
Power*rpm_in	-4.527435	0.507027	-8.93	<.0001*
rpm_in*rpm_in	2.7601522	0.364462	7.57	<.0001*
rpm_in*mg_req	-1.772846	0.507027	-3.50	0.0039*
Power*Power	0.6050179	0.364462	1.66	0.1208
mg_req*mg_req	0.4340558	0.364462	1.19	0.2550



### Prediction Profiler



Case 2



Summary of Fit	
RSquare	0.999291
RSquare Adj	0.998228
Root Mean Square Error	3.587475
Mean of Response	294.9956
Observations (or Sum Wgts)	16

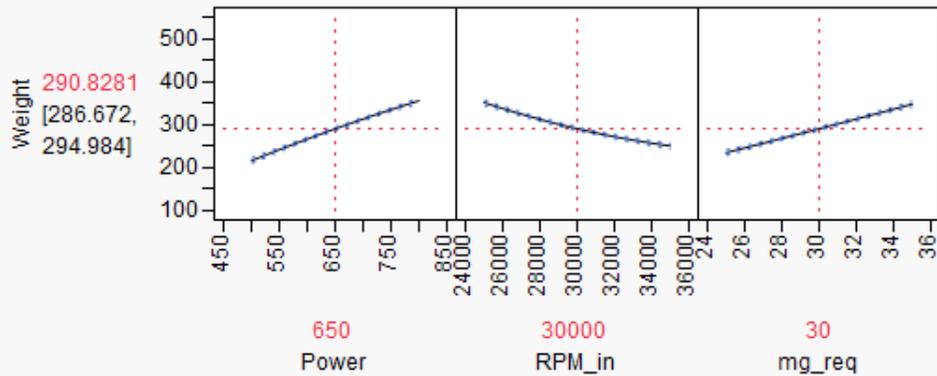
Parameter Estimates				
Term	Estimate	Std Error	t Ratio	Prob> t
Intercept	13.373905	101.621	0.13	0.8996
Power	0.5952927	0.146635	4.06	0.0067*
RPM_in	-0.015181	0.00563	-2.70	0.0357*
mg_req	9.4933577	5.62984	1.69	0.1427
Power*Power	-0.0002	9.82e-5	-2.04	0.0878
Power*RPM_in	-1.317e-5	1.691e-6	-7.79	0.0002*
RPM_in*RPM_in	4.172e-7	8.838e-8	4.72	0.0033*
Power*mg_req	0.0174769	0.001691	10.33	<.0001*
RPM_in*mg_req	-0.000378	5.073e-5	-7.46	0.0003*
mg_req*mg_req	0.0295465	0.088378	0.33	0.7495

▷ Effect Tests

▷ Residual by Predicted Plot

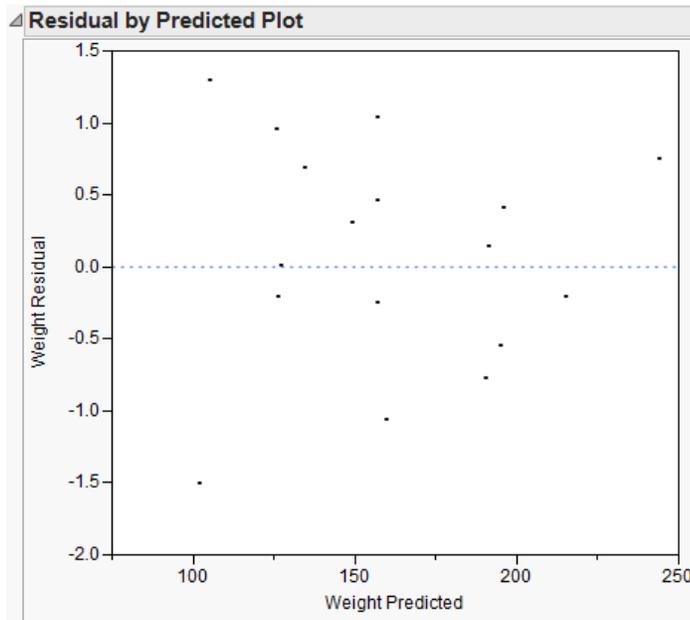
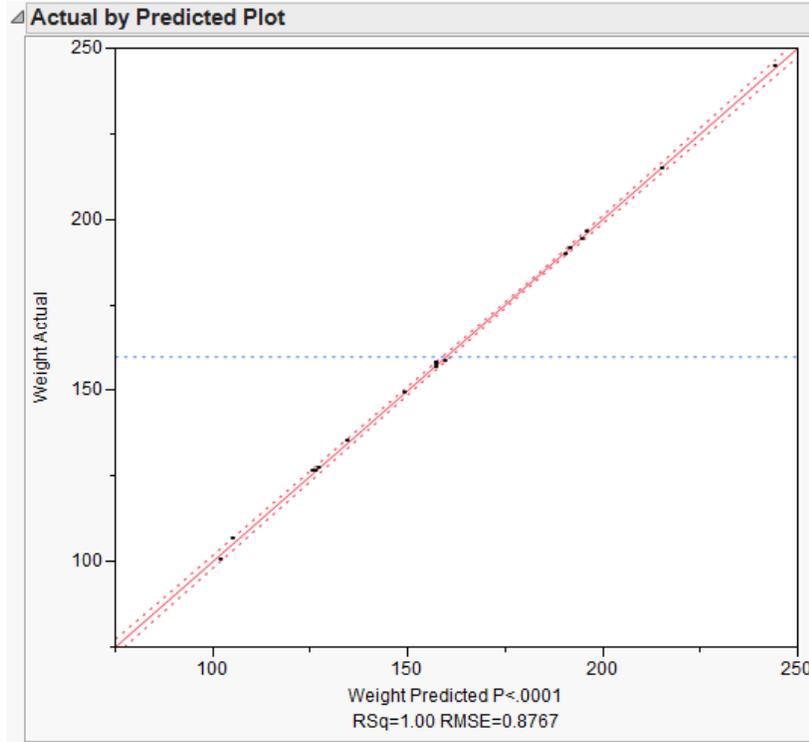
Sorted Parameter Estimates				
Term	Estimate	Std Error	t Ratio	Prob> t
Power*mg_req	0.0174769	0.001691	10.33	<.0001*
Power*RPM_in	-1.317e-5	1.691e-6	-7.79	0.0002*
RPM_in*mg_req	-0.000378	5.073e-5	-7.46	0.0003*
RPM_in*RPM_in	4.172e-7	8.838e-8	4.72	0.0033*
Power	0.5952927	0.146635	4.06	0.0067*
RPM_in	-0.015181	0.00563	-2.70	0.0357*
Power*Power	-0.0002	9.82e-5	-2.04	0.0878
mg_req	9.4933577	5.62984	1.69	0.1427
mg_req*mg_req	0.0295465	0.088378	0.33	0.7495

▷ Prediction Profiler



***b. Four-Stage Gear Train***

Case 1

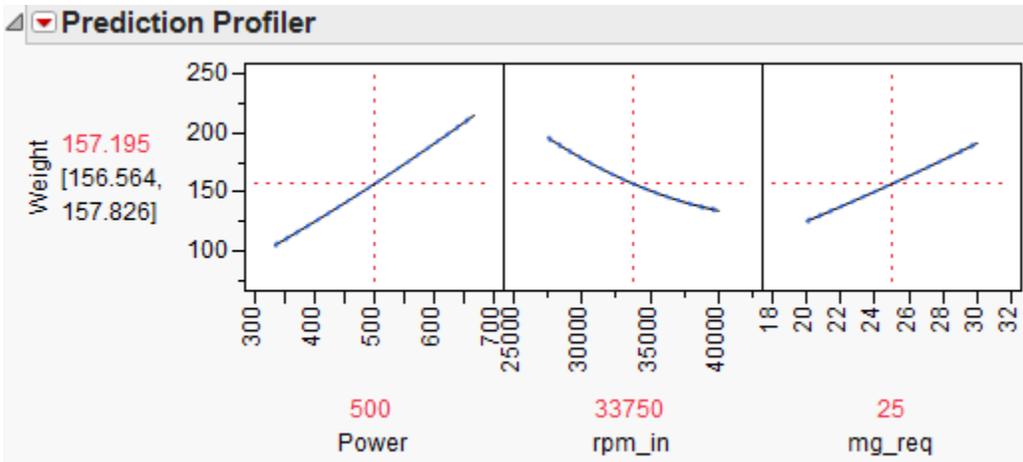


Summary of Fit	
RSquare	0.999603
RSquare Adj	0.999327
Root Mean Square Error	0.876711
Mean of Response	159.8489
Observations (or Sum Wgts)	23

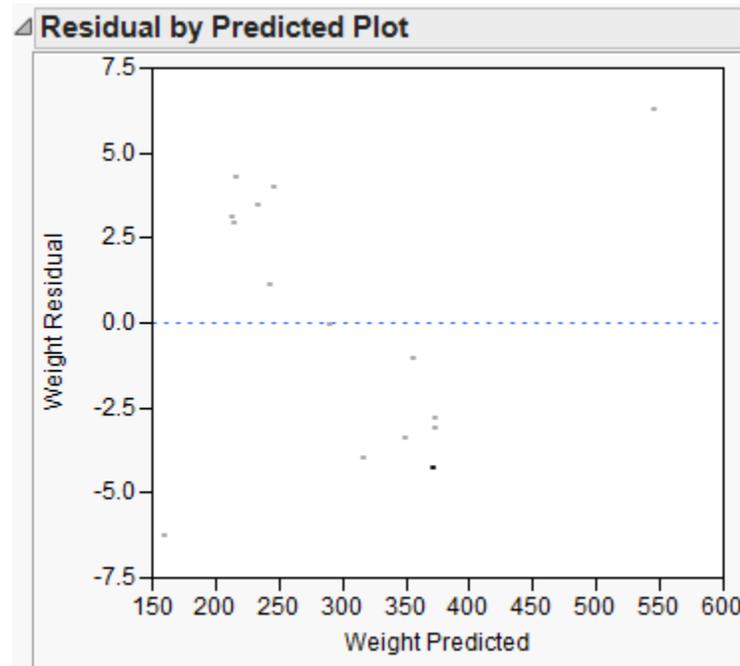
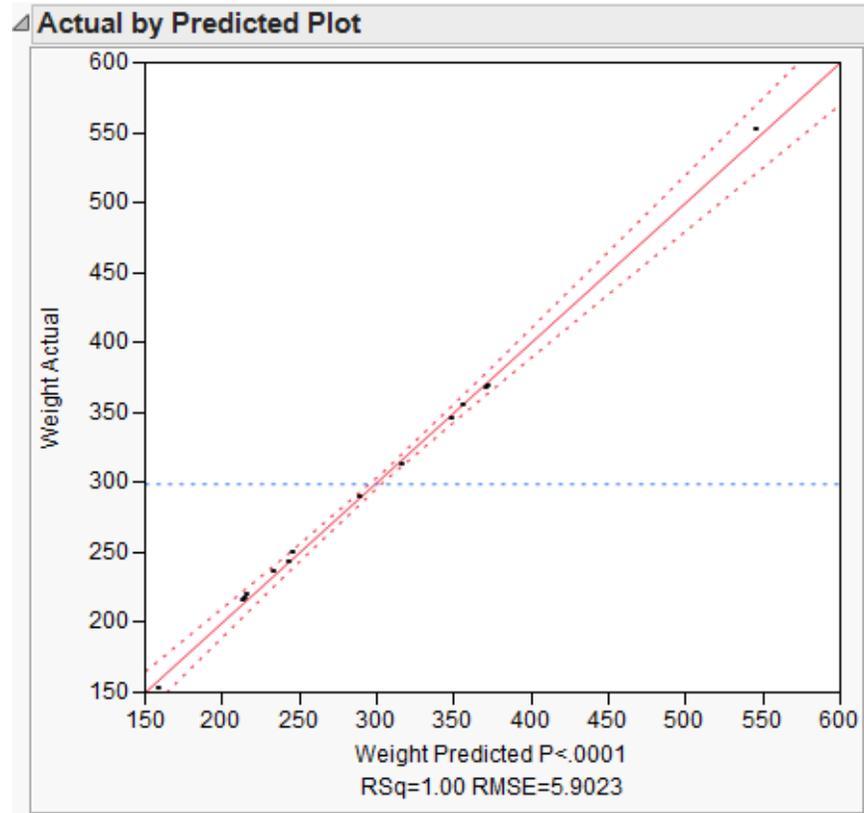
Parameter Estimates				
Term	Estimate	Std Error	t Ratio	Prob> t
Intercept	170.19549	38.70587	4.40	0.0007*
Power	0.169002	0.04411	3.83	0.0021*
rpm_in	-0.008929	0.001339	-6.67	<.0001*
mg_req	1.1109397	1.635254	0.68	0.5088
Power*Power	0.0001086	2.224e-5	4.88	0.0003*
Power*rpm_in	-1.113e-5	8.266e-7	-13.46	<.0001*
rpm_in*rpm_in	2.0655e-7	1.584e-8	13.04	<.0001*
Power*mg_req	0.0170798	0.001033	16.53	<.0001*
rpm_in*mg_req	-0.000175	2.755e-5	-6.34	<.0001*
mg_req*mg_req	0.0565724	0.024789	2.28	0.0400*

Effect Tests

Sorted Parameter Estimates				
Term	Estimate	Std Error	t Ratio	Prob> t
Power*mg_req	0.0170798	0.001033	16.53	<.0001*
Power*rpm_in	-1.113e-5	8.266e-7	-13.46	<.0001*
rpm_in*rpm_in	2.0655e-7	1.584e-8	13.04	<.0001*
rpm_in	-0.008929	0.001339	-6.67	<.0001*
rpm_in*mg_req	-0.000175	2.755e-5	-6.34	<.0001*
Power*Power	0.0001086	2.224e-5	4.88	0.0003*
Power	0.169002	0.04411	3.83	0.0021*
mg_req*mg_req	0.0565724	0.024789	2.28	0.0400*
mg_req	1.1109397	1.635254	0.68	0.5088



Case 2

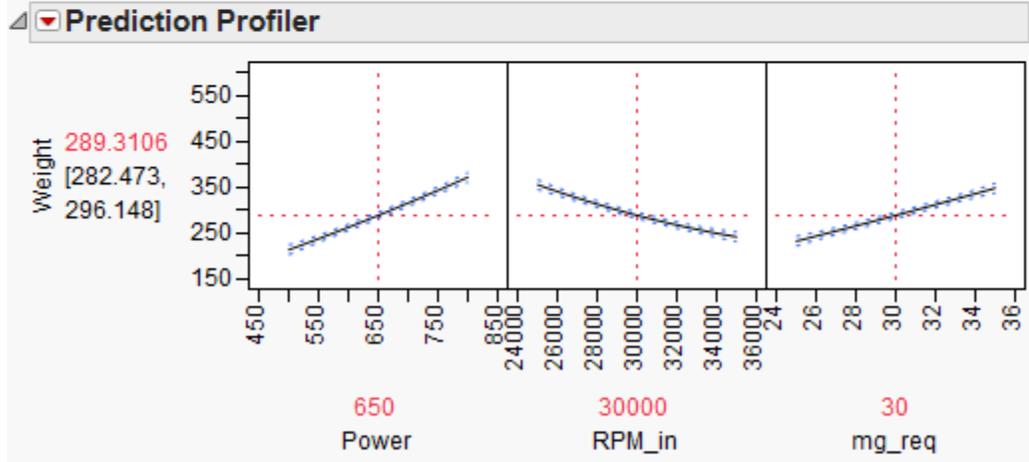


Summary of Fit	
RSquare	0.998449
RSquare Adj	0.996123
Root Mean Square Error	5.902263
Mean of Response	298.9569
Observations (or Sum Wgts)	16

Sorted Parameter Estimates				
Term	Estimate	Std Error	t Ratio	Prob> t
Power*RPM_in	-0.000024	2.782e-6	-8.59	0.0001*
Power*mg_req	0.0235402	0.002782	8.46	0.0001*
RPM_in*mg_req	-0.000474	8.347e-5	-5.67	0.0013*
RPM_in*RPM_in	3.9751e-7	1.454e-7	2.73	0.0340*
Power	0.3074697	0.241249	1.27	0.2496
Power*Power	0.0001766	0.000162	1.09	0.3163
mg_req	6.7901196	9.262448	0.73	0.4911
RPM_in	-0.005449	0.009262	-0.59	0.5778
mg_req*mg_req	0.0609157	0.145404	0.42	0.6898

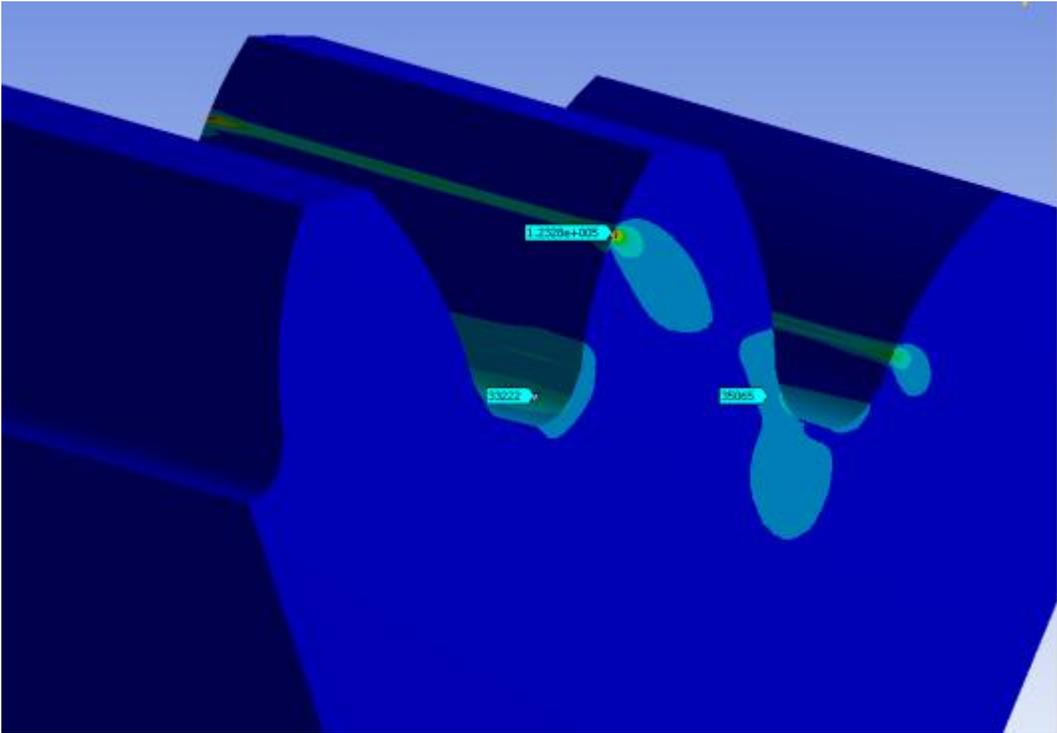
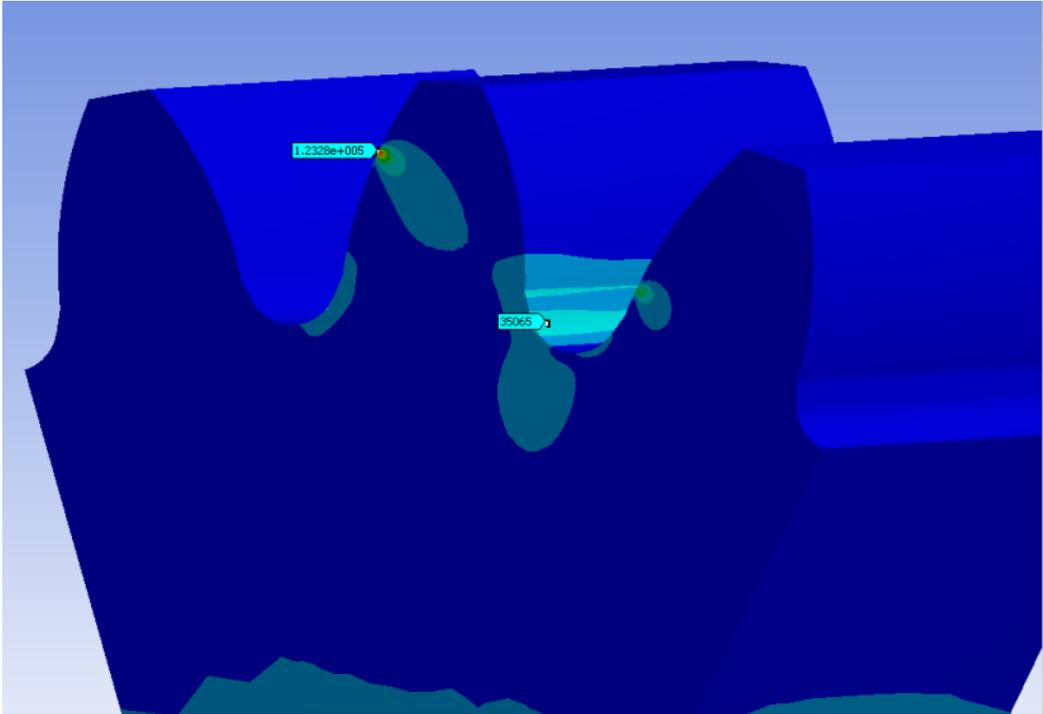
▷ Effect Details

▷ Response Surface

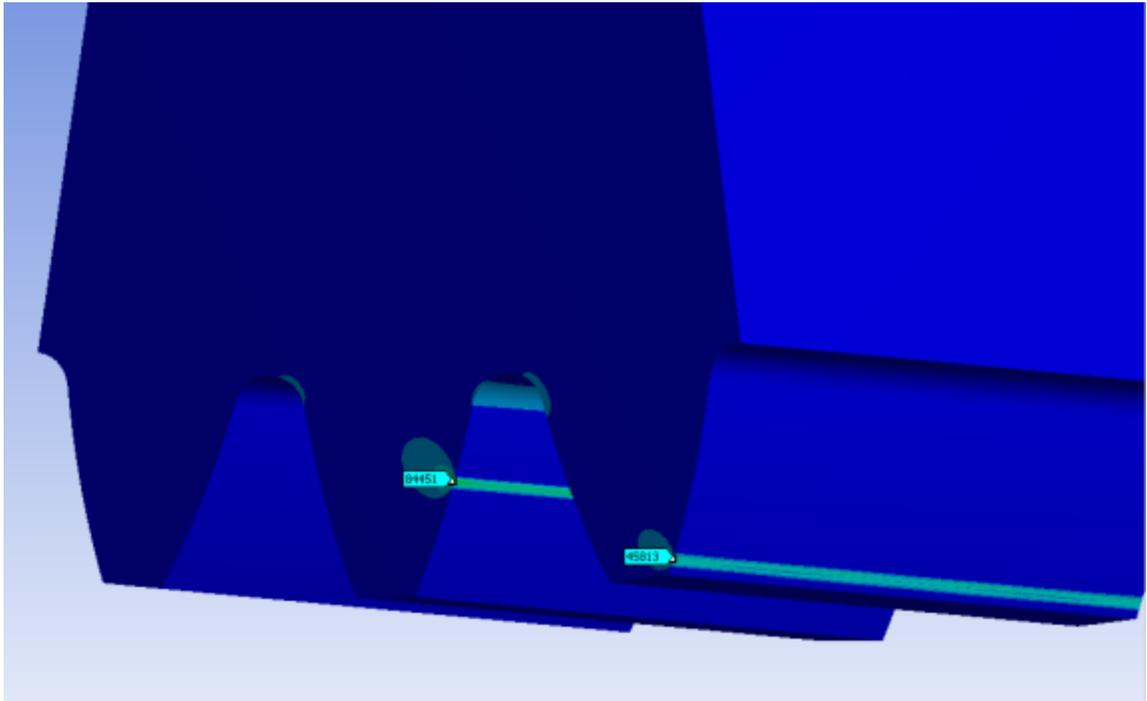


A.5. ANSYS Results

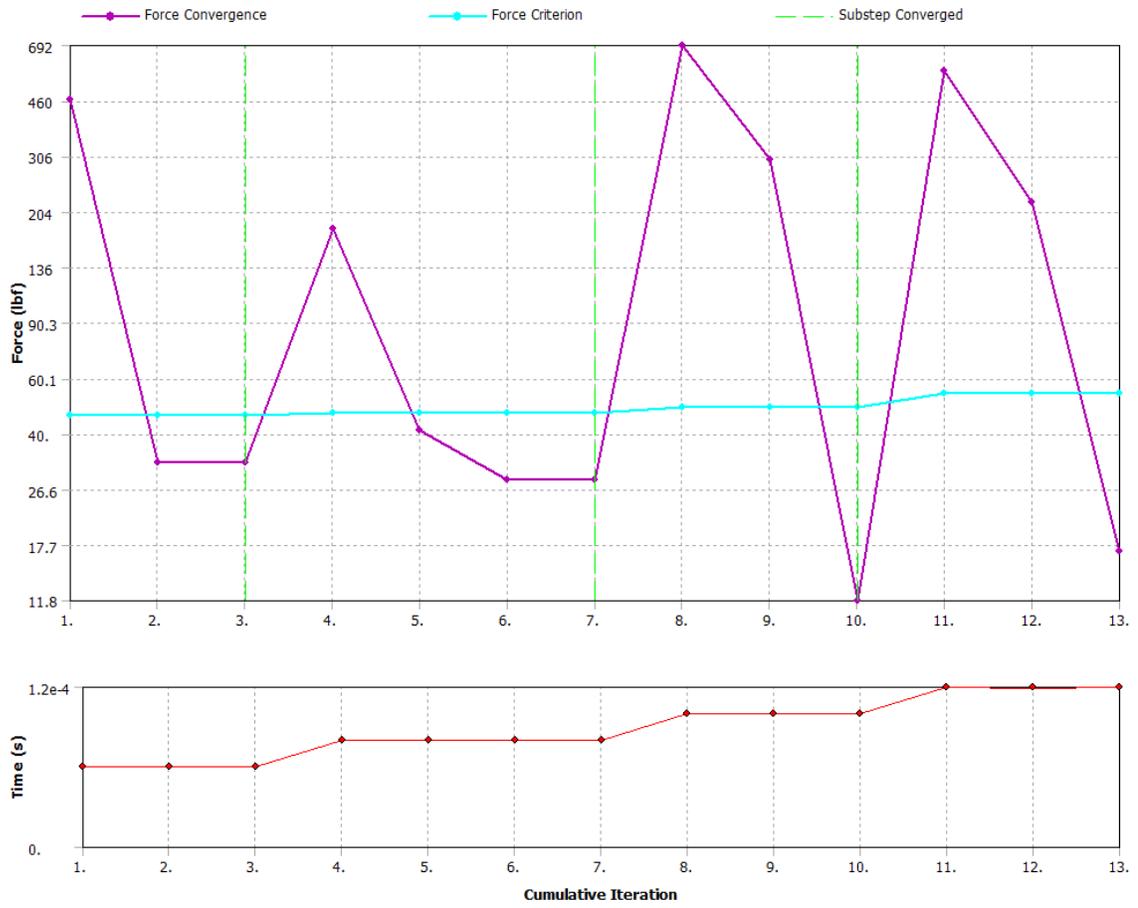
Pinion stress results (contact)



Gear stress results (Target)



## Solution convergence history

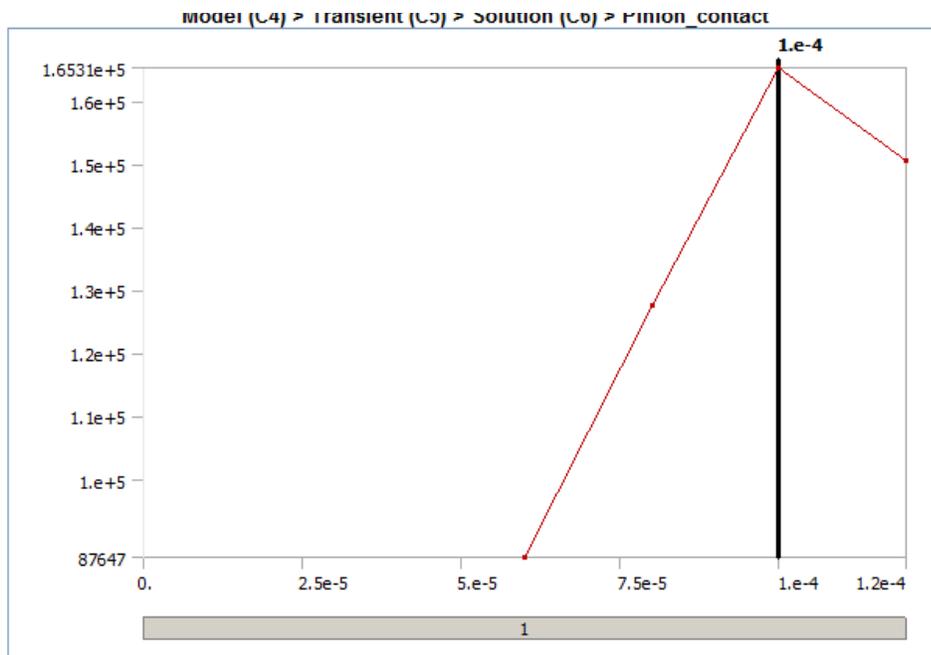
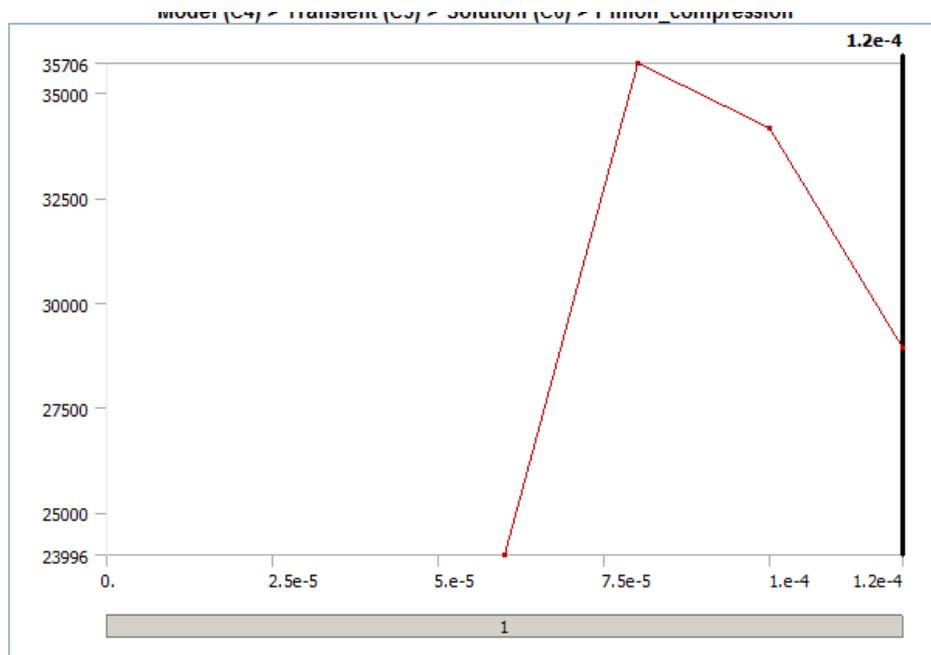


Model (C4) > Geometry > Parts

Object Name	<i>Spur_pinion2(Spur_pinion.1)</i>	<i>Spur_gear(Spur_gear.1)</i>
State	Hidden	Meshed
Graphics Properties		
Visible	No	Yes
Transparency		1
Definition		
Suppressed	No	
Stiffness Behavior	Flexible	
Coordinate System	Default Coordinate System	
Reference Temperature	By Environment	
Material		
Assignment	AISI 9310	
Nonlinear Effects	Yes	
Thermal Strain Effects	Yes	
Bounding Box		
Length X	1.8161 in	1.9915 in
Length Y	2.4266 in	3.9388 in
Length Z	2.22 in	
Properties		
Volume	6.0116 in <sup>3</sup>	12.442 in <sup>3</sup>
Mass	1.7049 lbm	3.5286 lbm
Centroid X	6.754e-007 in	0.26709 in
Centroid Y	1.879 in	4.4262 in
Centroid Z	1.11 in	
Moment of Inertia Ip1	1.2806 lbm-in <sup>2</sup>	5.2253 lbm-in <sup>2</sup>
Moment of Inertia Ip2	1.0081 lbm-in <sup>2</sup>	2.2243 lbm-in <sup>2</sup>
Moment of Inertia Ip3	0.8912 lbm-in <sup>2</sup>	4.5535 lbm-in <sup>2</sup>
Statistics		
Nodes	191715	172615
Elements	43227	38512
Mesh Metric	None	

Model (C4) > Connections > Contacts > Contact Regions

Object Name	<i>Frictional - Spur_pinion2(Spur_pinion.1) To Spur_gear(Spur_gear.1)</i>	<i>Frictional - Spur_pinion2(Spur_pinion.1) To Spur_gear(Spur_gear.1)</i>
State	Fully Defined	
Scope		
Scoping Method	Geometry Selection	
Contact	1 Face	
Target	1 Face	
Contact Bodies	<i>Spur_pinion2(Spur_pinion.1)</i>	
Target Bodies	<i>Spur_gear(Spur_gear.1)</i>	
Definition		
Type	Frictional	
Friction Coefficient	8.e-002	
Scope Mode	Manual	
Behavior	Asymmetric	
Suppressed	No	
Advanced		
Formulation	Normal Lagrange	
Detection Method	Nodal-Normal To Target	
Interface Treatment	Adjust to Touch	
Stabilization Damping Factor	0.5	
Pinball Region	Program Controlled	
Time Step Controls	Automatic Bisection	



## A.6. OptiStruct FEM code

No Grid, Element and Force data.

```
$$ Optistruct Input Deck Generated by HyperMesh Version : 11.0.0.47
$$ Generated using HyperMesh-Optistruct Template Version : 11.0.0.47
$$
$$   Template:  optistruct
$$ optistruct
$
FORMAT H3D
FORMAT HM
$$-----$
-----$
$$                               Case Control Cards
$
$$-----$
-----$
$$
$$ OBJECTIVES Data
$$
$
$HMNAME OBJECTIVES           1objective
$
DESOBJ(MIN)=1
$
$
$HMNAME LOADSTEP              1"Load"      1
$
SUBCASE           1
  SPC =           2
  LOAD =          1
  DESSUB =        2
$$-----$
$$ HYPERMESH TAGS
$$-----$
$$BEGIN TAGS
$$END TAGS
$
BEGIN BULK
$$
$$ Stacking Information for Ply-Based Composite Definition
$$

$HMNAME DESVARS              1Web
DTPL  1 PSOLID 1
+ STRESS 10000.0
+ PATRN   100.0   0.0   0.8571430.0   0.0   0.0
+        100
+ DRAW   SPLIT  0.0   0.0   0.857   0.0   0.0   0.0
$$
$$ OPTIRESponses Data
$$
DRESP1  1      Volume  VOLUME  PSOLID
1
```

```

DRESP1 2          Stress  STRESS  PSOLID          SVM
2
$$
$$ OPTICONSTRAINTS Data
$$
$
$HMNAME OPTICONSTRAINTS          1Bending Stress
$
DCONSTR          1          2          37500.0

DCONADD          2          1
$$
$$ DESVARG Data
$$ GRID Data
$$ SPOINT Data

```

\$ RBE2 Elements - Multiple dependent nodes

```

$
$$-----$
-----$
$$ HyperMesh name and color information for generic components
$
$$-----$
-----$
$HMNAME COMP          1"Shafthole"          3 "Non-Design" 5
$HWCOLOR COMP          1          27
$
$HMNAME COMP          2"DesignArea"          1 "Design Area" 5
$HWCOLOR COMP          2          30
$
$HMNAME COMP          3"GearTeeth"          3 "Non-Design" 5
$HWCOLOR COMP          3          54
$
$HMNAME COMP          4"Bending Region"          2 "Bending
Region" 5
$HWCOLOR COMP          4          34
$
$HMNAME COMP          24"lv110000.1"
$HWCOLOR COMP          24          43
$
$HMNAME COMP          25"Rigids"          4 "Rigids" 1
$HWCOLOR COMP          25          5
$$
$$ PMASS Data
$$
$HMNAME PROP          4"Rigids" 1
$HWCOLOR PROP          4          5
PMASS          40.0
$$
$$ PSOLID Data
$$
$HMNAME PROP          1"Design Area" 5
$HWCOLOR PROP          1          30
PSOLID          1          1
$HMNAME PROP          2"Bending Region" 5
$HWCOLOR PROP          2          33

```

```

PSOLID          2          1
$HMNAME PROP           3"Non-Design" 5
$HWCOLOR PROP           3          54
PSOLID          3          1
$$
$$  MAT1 Data
$$
$HMNAME MAT           1"Steel" "MAT1"
$HWCOLOR MAT           1          5
MAT1             13.0+7           0.3          0.268
$$
$$$$-----
-----$
$$ HyperMesh Commands for loadcollectors name and color information $
$$$$-----
-----$
$HMNAME LOADCOL           1"Force"
$HWCOLOR LOADCOL           1          22
$$
$HMNAME LOADCOL           2"Constraint"
$HWCOLOR LOADCOL           2          5
$$
$$  SPC Data
$$
SPC                2  456757  1234560.0
$$  FORCE Data
ENDDATA

```

### **A.7. Bell PC Cost Model**

The following is a description of the inputs in Bell PC specific to the drive system [150].

- Main transmission inputs

Type: Numeric

Choices: This value is the number of engine inputs to the main transmission. Single engine helicopters and tiltrotors have one input. If the power combining function for a multi-engine helicopter is performed by the main transmission and not by a combining gearbox, this value will be equal to the number of engines. For a multi-engine helicopter, a value of one in this cell will automatically add a combining gearbox to the configuration. It will be added to the drive system, if not included in the powerplant configuration above.

- Main transmission configuration

Type: List Box

Choices: Flat pack or planetary: This cell affects prototype cost only. The flat pack transmission uses a bull gear to transfer power and has fewer gears than planetary types. Although the flat pack is less costly, it weighs more than the planetary type. This cell applies to helicopter configurations only.

- Will the drive system have a rotor brake?

Type: List Box

Choices: Yes or No: This cell affects prototype cost only. A “Yes” input will add a rotor brake to the configuration.

- Will the aircraft have an intermediate (42-degree) tailrotor gearbox?

Type: List Box

Choices: Yes or No: This cell is only active for a helicopter configuration. “Yes” input adds an intermediate gearbox to the configuration.

- Will the aircraft have an accessory gearbox?

Type: List Box

Choices: Yes or No: “Yes” input adds an accessory gearbox to the configuration.

- Number of Accessories?

Type: Numeric

Choices: This cell becomes active if “Yes” is selected in the accessory gearbox cell above and accounts for the number of accessories being driven by the accessory gearbox.

- Mast Type?

Type: List Box

Choices: Straight or Flanged: This cell is active for the helicopter configuration. A flanged mast increases system complexity.

- Tailrotor driveshaft material or interconnect driveshaft material

Type: List Box

Choices: Metal or Composite: The tailrotor driveshaft material is active for a helicopter configuration and the interconnect driveshaft material is active for the tiltrotor configuration. This input only affects the prototype cost.

- Main transmission or proprotor gearbox TBO (FH),
- Tailrotor gearbox or tiltaxis gearbox TBO (FH),
- Intermediate gearbox or midwing gearbox TBO (FH),
  - This cell is active for tiltrotors (midwing gearbox) and helicopters configured with an intermediate gearbox.
- Combining gearbox TBO (FH), and
  - This cell is active for multi-engine helicopters configured with a combining gearbox
- Accessory gearbox TBO (FH)
  - This cell is active for aircraft configured with an accessory gearbox.

Type: Numeric input

Choices: The Time Between Overhaul (TBO) affects both prototype and operating and support costs. The model assumes that gearbox overhaul time is a function of gearbox weight. As a result, prototype cost will increase as gearbox TBO increases, but operating and support cost per flight hour will decrease. If zero is input into this cell, gearbox maintenance becomes on-condition, and drive system unscheduled maintenance cost increases to offset the absence of a designated scheduled gearbox overhaul requirement.

- Percent New Design

Type: Numeric

Choices: Value between 0% and 100%. This is an assessment of the amount of new design. A value of 100% represents a “clean sheet of paper”. A value of 5% is used to allow for simple drawing changes to document the new configuration. This variable

affects engineering design man-hours and is only active if “Parametric” is selected for the engineering design man-hour source.

- Technology Factor

Type: Numeric

Choices: This is an assessment of unknowns in the new design due to the introduction of new technology or difficulties that may be encountered with system integration. The default value of 1.0 will not change the system man-hour result. This is a linear adjustment; a value of 1.25 will increase the estimate by 25%. The upper risk level is also increased. This variable affects engineering design man-hours and is only active if “Parametric” is selected for the engineering design man-hour source.

- Will an existing transmission bench test stand be used?

Type: List Box

Choices: Yes or No: “No” input in this cell adds the cost of a transmission bench test stand to the bench test module.

- Labor and Burden Rate Designation

Type: List Box

Choices: This cell uses the values from the “Rate Input” worksheet. The rate titles from the “Rate Input” worksheet will appear in the list box. Different rates may be applied to each system to model teaming development programs or if certain systems are subcontracted. The rate structure in the first column of the “Rate Input” worksheet is used as the default rate if nothing is specified in this cell.

## REFERENCES

1. Chae, H. G., Gunduz, M. E., Sirirojvisuth, A., Liu, H. Y., and Schrage, D. "Development of a Digital Framework for Integrated Product/Process Development (IPPD)," *AHS 2nd International Forum on Rotorcraft Multidisciplinary Technology*. Seoul, Korea, October 2009.
2. Woyak, S. "CAD Fusion: Bridging the Handoff from Conceptual to Preliminary Design," *White Paper*. ModelCenter, Phoenix Integration, 2007.
3. Hart, P. B. "A PLM Implementation for Aerospace Systems Engineering Conceptual Rotorcraft Design," *Master of Science Thesis, Aerospace Engineering*. Georgia Institute of Technology, Atlanta, GA, May 2009.
4. "<http://www.cbo.gov/ftpdocs/88xx/doc8865/Chapter1.5.1.shtml>," *accessed on December 01, 2011*.
5. Germanowski, P. J., Stille, B. L., and Strauss, M. P. "Technology Assessment for Large Vertical Lift Transport Tiltrotor." NASA/CR-2010-216384, May 2010.
6. Tai, J. C. M. "A Multidisciplinary Design Approach to Size Stopped Rotor-Wing Configurations Using Reaction Drive and Circulation Control," *PhD Thesis, School of Aerospace Engineering*. Georgia Institute of Technology, Atlanta, GA, 1998.
7. Orr, S. A., and Narducci, R. P. "Framework for Multidisciplinary Analysis, Design, and Optimization with High-Fidelity Analysis Tools." NASA/CR-2009-215563, February 2009.
8. Sobieszczanski-Sobieski, J., and Haftka, R. T. "Multidisciplinary Aerospace Design Optimization: Survey of Recent Developments," *Structural Optimization* Vol. 14, 1997, pp. 1-23.
9. Singer, D. J., Doerry, N., and Buckley, M. E. "What is Set-Based Design?," 2008.
10. Sobek II, D. K., Ward, A. C., and Liker, J. K. "Toyota's Principles of Set-Based Concurrent Engineering," *Sloan Management Review*, 1999.
11. "Engineering Design Handbook. Helicopter Engineering, Part One. Preliminary Design. AMCP 706-201." Army Materiel Command, 30 August 1974.

12. Handschuh, R. F., and Zakrajsek, J. J. "Current Research Activities in Drive System Technology in Support of the NASA Rotorcraft Program." NASA/TM—2006-214052, ARL—TR—3707, January 2006.
13. Chae, H. G. "A Possibilistic Approach to Rotorcraft Design through a Multi-Objective Evolutionary Algorithm," *PhD Thesis, School of Aerospace Engineering*. Georgia Institute of Technology, Atlanta, GA, December 2006.
14. Rogers, J. L. "Reducing Design Cycle Time and Cost through Process Resequencing," *11th International Conference on Engineering Design*. Tampere Finland, August 1997.
15. Chen, W., and Lewis, K. "Robust Design Approach for Achieving Flexibility in Multidisciplinary Design " *AIAA Journal* Vol. 37, No. 8, 1999, pp. 982-989.
16. Price, M., Raghunathan, S., and Curran, R. "An Integrated Systems Engineering Approach to Aircraft Design," *Progress in Aerospace Sciences*. 2006.
17. Simonds, R. M. "A Generalized Graphical Method of Minimum Gross Weight Estimation," *The National Conference of the Society of Aeronautical Weight Engineering, INC.*, San Diego, California, May 1956.
18. Joy, D. P., and Simonds, R. M. "Transport Helicopter Design Analysis Methods," *Configuration Selection*. Hiller Helicopters, 1955.
19. Schrage, D. P. "Technology for Rotorcraft Affordability Through Integrated Product/Process Development (IPPD)," *American Helicopter Society 55th Annual Forum*. Montreal, Canada, May 1999.
20. Ashok, S. V., Robledo, A., and Schrage, D. P. "A Systems Engineering Modeling and Simulation Approach for Rotorcraft Drive System Optimization," *American Helicopter Society 67th Annual Forum*. Virginia Beach, VA, May 2011.
21. Sirirojvisuth, A. "Development of Hybrid Lifecycle Cost Estimating Tool (HLCET) for Manufacturing Influenced Design Tradeoff," *PhD Thesis, School of Aerospace Engineering*. Georgia Institute of Technology, Atlanta, GA, August 2012.
22. Mavris, D., Baker, A. P., and Schrage, D. P. "IPPD Through Robust Design Simulation for an Affordable Short Haul Civil Tiltrotor," *American Helicopter Society 53rd Annual Forum*. Virginia Beach, VA, May 1997.

23. Gunduz, M. E. "Software Integration for Automated Stability Analysis and Design Optimization of a Bearingless Rotor Blade," *Phd Thesis, School of Aerospace Engineering*. Georgia Institute of Technology, Atlanta, GA, May 2010.
24. Schrage, D. P. "GIT-KKU IPPD Rotorcraft Preliminary Design Methodology." 2009.
25. Nickol. "Conceptual design shop: Presentation to Conceptual Aircraft Design Working Group (CADWG21)," 2004.
26. "Detailed Weight Statement - H60," *SES-700209*. Sikorsky Aircraft, 1989.
27. "Systems Engineering Handbook," *INCOSE-TP-2003-016-02*. 2004.
28. Simpson, T. W., Toropov, V., Balabanov, V., and Viana, F. A. C. "Design and Analysis of Computer Experiments in Multidisciplinary Design Optimization," *12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference AIAA 2008-5802*, Victoria, British Columbia, Canada, September 2008.
29. Mavris, D. N., Tai, J., and Schrage, D. P. "A Multidisciplinary Design Optimization Approach to Sizing Stopped Rotor Configurations Utilizing Reaction Drive and Circulation Control," *American Institute of Aeronautics and Astronautics*. 1992.
30. Khalid, A. S. "Development and Implementation of Rotorcraft Preliminary Design Methodology using Multidisciplinary Design Optimization," *PhD Thesis, School of Aerospace Engineering*. Georgia Institute of Technology, Atlanta, GA, December 2006.
31. Landor, S. E. "Issues in Multiagent Design Systems," *IEEE*, 1997.
32. Ashok, S. V., Sirirojvisuth, A., and Schrage, D. P. "Systems Engineering Application for Rotorcraft Systems," *American Helicopter Society 68th Annual Forum*. Fort Worth, TX, May 2012.
33. Carty, A. "An Approach to multidisciplinary Design Analysis and Optimization for Rapid Conceptual Design," *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*. AIAA 2002-5438, Atlanta, Georgia, September 2002.
34. Pratt, M. J. "Parametric STEP for Concurrent Engineering," *Advances in Concurrent Engineering*, 2002.

35. Lee, K. *Principles of CAD/CAM/CAE Systems.*: Addison Wesley Longman, Inc., 1999.
36. Zeid, I. *CAD/CAM Theory and Practice*: McGraw-Hill, Inc., 1991.
37. Chang, K. H., Silva, J., and Bryant, I. "Concurrent Engineering and Manufacturing for Mechanical Systems," *Concurrent Engineering Research and Applications*, 1999.
38. Robinson, T. T., Armstrong, C. G., and Chua, H. S. "Determining the Parametric Effectiveness of a CAD Model," *Engineering with Computers*, 2011.
39. Ashok, S. V., Sirirojvisuth, A., and Schrage, D. P. "Drive System Design Integration for Rotorcraft through Integrated Product-Process Development (IPPD)," *American Helicopter Society International, 2nd Technical Specialists' Meeting on Systems Engineering*. Ithaca, NY, September 2011.
40. Silva, J., and Chang, K.-H. "Design Parameterization for Concurrent Design and Manufacturing of Mechanical Systems," *Concurrent Engineering: Research and Applications*, November 2001.
41. Stark, J. "Engineering Information Management Systems: Beyond CAD/CAM to Concurrent Engineering Support " *Automation in Manufacturing*. 1992.
42. "CATIA, Dassault Systemes. Paris, FRANCE. Version 5 Release 15." 2005.
43. Xu, X. *Integrating Advanced Computer - Aided Design, Manufacturing, and Numeric Control: Principles and Implementation*. Hershey, NY: IGI Global, 2009.
44. Inoue, M., Nahm, Y. E., Okawa, S., and Ishikawa, H. "Design Support System by Combination of 3D-CAD and CAE with Preference Set-based Design Method," *Concurrent Engineering*, 2010.
45. Chironis, N. *Gear Design and Application*. New York: McGraw-Hill, 1967.
46. Dooner, D., and Seireg, A. *The Kinematic Geometry of Gearing*. New York: Wiley, 1995.
47. Bellocchio, A. T. "Drive System Design Methodology for a Single Main Rotor Heavy Lift Helicopter," *Master of Science Thesis, Aerospace Engineering*. Georgia Institute of Technology, Atlanta, GA, December 2005.

48. "AGMA 911 A94 - Design Guidelines for Aerospace Gearing," *AGMA INFORMATION SHEET*. American Gear Manufacturers Association, 1994.
49. "AGMA 908 B89 - Gear Factors for Determining the Pitting Resistance and Bending Strength of Spur, Helical and Herringbone Gear Teeth," *AGMA Information Sheet*. American Gear Manufacturers Association, 1989.
50. "ANSI/AGMA 2001 - C95 - Fundamental Rating Factors and Calculation Methods for Involute Spur and Helical Gear Teeth," *AGMA Standard*. American National Standard, American Gear Manufacturers Association, 1995.
51. "ANSI/AGMA 2005-C96 - Design Manual for Bevel Gears," *American National Standard*. American Gear Manufacturers Association, 2003.
52. "ANSI/AGMA 2003-B97 - Rating The Pitting Resistance and Bending Strength of Generated Straight Bevel, Zerol Bevel and Spiral Bevel Gear Teeth," *American National Standard*. American Gear Manufacturers Association, 2003.
53. Mugnier, M. "AHS Student Design Competition - Odyssey Final Report." Georgia Institute of Technology, Atlanta, Georgia, 2011.
54. Hameer, S. "A Comparative Study and Application of Continuously Variable Transmission to a Single Main Rotor Heavy Lift Helicopter," *PhD Thesis, School of Aerospace Engineering*. Georgia Institute of Technology, Atlanta, GA, December 2009.
55. Willis, R. J. "New Equations and Charts to Pick Off Lightest-Weight Gears." 1963.
56. Stepniewski, W. Z., and Shinn, R. A. "A Comparative Study of Soviet vs. Western Helicopters: Part 2-Evaluation of Weight, Maintainability and Design Aspects of Major Components," *NASA TR 82-A-10* Ames Research Center: AVRADCOM Research and Technology Laboratories, 1983.
57. Saribay, Z. B., Wei, F.-S., and Sahay, C. "Optimization of an Intermeshing Rotor Transmission System Design," *AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics & Materials Conference*. Austin, TX, 2005.
58. Chong, T. H., and Lee, J. S. "Genetic Algorithm Based Design for Gear Trains," *International Journal of the Korean Society of Precision Engineering*, June 2000.

59. Padmanabhan, S., Ganesan, S., Chandrashekar, M., and Raman, V. S. "Gear Pair Design Optimization by Genetic Algorithm and FEA," *IEEE*, 2010.
60. Yokota, T., Taguchi, T., and Gen, M. "A Solution Method for Optimal Weight Design Problem of the Gear Using Genetic Algorithms," *Computers and Industrial Engineering* Vol. 35, 1998, pp. 523-526.
61. Savsani, V., Rao, R. V., and Vakharia, D. P. "Optimal Weight Design of a Gear Train using Particle Swarm Optimization and Simulated Annealing Algorithms," *Mechanism and Machine Theory* Vol. 45, No. 3, 2010, pp. 531-541.
62. Ashok, S. V., Wade, B., and Schrage, D. P. "Variable Speed Transmission using Planetary Gear System for High Speed Rotorcraft Application," *American Helicopter Society 66th Annual Forum*. Phoenix, AZ, May 2010.
63. "AGMA 901 - A92 - A Rational Procedure for the Preliminary Design of Minimum Volume Gears," *AGMA INFORMATION SHEET*. American Gear Manufacturers Association, 1992.
64. Dudley, D. W. *Handbook of Practical Gear Design*: Technomic Publication, 1994.
65. Townsend, D. P. *Dudley's Gear Handbook - The Design, Manufacture, and Application of Gears*. Second Edition: Tata McGraw Hill, 2011.
66. Norton, R. L. *Machine Design - An Integrated Approach*: Pearson Prentice Hall, Third Edition, 2006.
67. Brown, F. W., Davidson, S. R., Hanes, D. B., Weires, D. J., and Kapelevich, A. "Analysis and Testing of Asymmetric Involute Tooth Form and Optimized Fillet Form for Potential Application in Helicopter Main Drives," *American Gear Manufacturers Association*. 2010.
68. White, G. "Design study of a split-torque helicopter transmission," *Proceedings of the Institution of Mechanical Engineers, Journal of Aerospace Engineering* Vol. 212, Number 2, November 1998.
69. Lewicki, D. G., Heath, G. F., Filler, R. R., Slaughter, S. C., and Fetty, J. "RDS-21 Face Gear Surface Durability Tests." NASA/TM—2007-214970, ARL-TR-4089.

70. Litvin, F. L., Egelja, A., Tan, J., Chen, D. Y.-D., and Heath, G. "Handbook on Face Gear Drives With a Spur Involute Pinion." NASA / CR 2000-209909, ARL-CR-447, March 2000.
71. Lewicki, D. G. "Advanced Rotorcraft Transmission Technology." Propulsion Directorate, NASA Lewis Research Center. , Cleveland OH.
72. Kapelevich, A. L., and Shekhtman, Y. V. "Direct Gear Design: Bending Stress Minimization," *Gear Technology*. 2003.
73. Math, V. B., and Chand, S. "An Approach to the Determination of Spur Gear Tooth Root Fillet," *Journal of Mechanical Design* Vol. 126, 2004.
74. Arikan, M. A. S. "Direct Calculation of AGMA Geometry Factor J by making use of Polynomial Equations," *Mechanics Research Communications* Vol. 29, 2002.
75. Glover, F., and Greenberg, H. "New Approaches for Heuristic Search: a Bilateral Linkage with Artificial Intelligence," *European Journal of Operational Research*, 1989.
76. Cheng, F. Y., and Li, D. "Genetic Algorithm Development for Multiobjective Optimization of Structures," *AIAA Journal* Vol. 36, No. 6, June 1998.
77. Gen, M., and Cheng, R. "A Survey of Penalty Techniques in Genetic Algorithms," *Proceedings of IEEE International Conference on Evolutionary Computation*. May 1996.
78. Yeniay, O. "Penalty Function Methods for Constrained Optimization with Generic Algorithms," *Mathematical and Computational Applications*, 2005.
79. Srinivas, M., and Patnaik, L. M. "Adaptive Probabilities of Crossover and Mutation in Genetic Algorithms," *IEEE Transactions on Systems, Man and Cybernetics*, April 1994.
80. Grefenstette, J. J. "Optimization of Control Parameters for Genetic Parameters for Genetic Algorithms," *IEEE Transactions on Systems, Man and Cybernetics* 1986.
81. Bingul, Z. "Adaptive Genetic Algorithms Applied to Dynamic Multiobjective Problems," *Applied Soft Computing*, 2007.
82. DeJong, K. A. "An Analysis of the Behavior of a Class of Genetic Adaptive Systems," *PhD Thesis*. University of Michigan, 1975.

83. Goldberg, D. E., and Richardson, J. "Genetic Algorithms with Sharing for Multimodal Function Optimization," *Second International Conference on Genetic Algorithms*. 1987.
84. Cantu-Paz, E. "A Survey of Parallel Genetic Algorithms."
85. Belkadi, K., Gourgand, M., and Benyettou, M. "Parallel Genetic Algorithms with Migration for the Hybrid Flow Shop Scheduling Problem."
86. Motro, A., and Smets, P. *Imperfect Information: Imprecision-Uncertainty, Uncertainty Management in Information Systems, from Needs to Solutions*. Massachusetts, USA: Kluwer Academic Publishers, 1997.
87. Antonsson, E. K., and Otto, K. N. "Imprecision in Engineering Design," *Transaction of the ASME Journal of Mechanical Design*, 1995.
88. Kara, S., Kayis, B., and Kaebnick, H. "Modeling Concurrent Engineering Projects Under Uncertainty," *Concurrent Engineering: Research and Applications*, 1999.
89. Saleh, J. H., Hastings, D. E., and Newman, D. J. "Flexibility in System Design and Implications for Aerospace Systems," *Acta Astronautica* Vol. 53, 2002, pp. 927-944.
90. Raouf, A., and Ben-Daya, M. "Flexible Manufacturing Systems: Recent Developments," *Elsevier*. Amsterdam, 1995.
91. Suarez, F., Cusumano, M., and Fine, C. "Flexibility and Performance: A Literature Critique and Strategic Framework," *Sloan School WP, Massachusetts Institute of Technology*. Cambridge, MA, 1991.
92. Taylor, T. "Evaluating and Selecting Manufacturing Flexibility," *S.M. Thesis, Department of Mechanical Engineering*. Massachusetts Institute of Technology, Cambridge, MA, 1991.
93. Piore, M. "Corporate Reform in American Manufacturing and the Challenge to Economic Reform," *Massachusetts Institute of Technology*. Cambridge, MA, 1989.
94. Oleson, J. "Pathways to Agility," *Wiley*. New York, 1998.

95. Fricke, E., Schultz, A., Wenzel, S., and Negele, H. "Design for Changeability of Integrated Systems within a Hyper-Competitive Environment," *INCOSE Conference*. Denver CO, 2000.
96. Thurston, D. L. "A Formal Method for Subjective Design Evaluation with Multiple Attributes," *Research in Engineering Design* Vol. 3, 1991, pp. 105-122.
97. Wallace, D. R., Jakiela, M., and Flowers, W. C. "Design Search under Probabilistic Specifications using Genetic Algorithms," *Computer Aided Design* Vol. 28, No. 5, 1996, pp. 404 - 406.
98. Sandgren, S. U. M. E. "Multiobjective Optimization Dealing with Uncertainty," *Advances in Design Automation* Vol. 19, No. 2, 1989, pp. 241-248.
99. Ward, A. C., Liker, J. K., Cristiano, J. J., and Sobek, D. K. "The Second Toyota Paradox: How Delaying Decision Can Make Better Cars Faster," *Sloan Management Review*, 1995.
100. Lewis, W. C. K. "Robust Design Approach for Achieving Flexibility in Multidisciplinary Design " *AIAA Journal* Vol. 37, No. 8, 1999, pp. 982-989.
101. DeLaurentis, D. A., and Mavris, D. N. "Uncertainty Modeling and Management in Multidisciplinary Analysis and Synthesis," *AIAA*. 2000.
102. Mavris, D. N., and Bandte, O. "A Probabilistic Approach to Multivariate Constrained Robust Design Simulation." 1997.
103. Trigeorgis, L. "Real Options: Managerial Flexibility and Strategy in Resource Allocation," *Massachusetts Institute of Technology Press*. Cambridge, MA, 1996.
104. Stoker, K., Chauduri, A., and Kim, N. H. "Safety of Spur Gear Design Under Non-Ideal Conditions with Uncertainty," *ASME International Design Engineering Technical Conferences and Computer and Information in Engineering Conference*. Montreal, Canada, August 2010.
105. Coy, J. J., Townsend, D. P., and Zaretsky, E. V. "Gearing," *NASA Reference Publication 1152* Vol. AVSCOM Tehnical Report 84-C-15, 1985.
106. Vijayarangan, S., and Ganesan, N. "A Study of Dynamic Stresses in a Spur Gear Under a Moving Line Load and Impact Load Conditions by a Three-dimensional Finite Element Method," *Journal of Sound and Vibration* Vol. 162, 1993, pp. 185-189.

107. Rao, C. R. M., and Muthuveerappan, G. "Finite Element Modeling and Stress Analysis of Helical Gear Teeth," *Computers and Structures* Vol. 49, No. 6, 1993, pp. 1095-1106.
108. Rameshkumar, M., Venkatesan, G., and Sivakumar, P. "Finite Element Analysis of High Contact Ratio Gear," *American Gear Manufacturers Association, Technical Resources*. 2010.
109. Stoker, K. C. "A Finite Element Approach to Spur Gear Response and Wear Under Non-Ideal Loading," *Master of Science Thesis*. University of Florida, 2009.
110. Hassan, A. R. "Contact Stress Analysis of Spur Gear Teeth Pair," *World Academy of Science, Engineering and Technology* Vol 58, 2009.
111. Li, Z., Freborg, A., and Hansen, B. D. "Modeling Study of Carburization and Quenching Effects on Residual Stresses and Fatigue of Gears," *American Helicopter Society 67th Annual Forum*. Virginia Beach, VA, May 2011.
112. Kirov, V. "Comparing AGMA and FEA Calculations," *gearsolutions.com*. 2011.
113. Shigley, J. E., Mischke, C. R., and Budynas, R. G. *Mechanical Engineering Design*. New York, NY: McGraw Hill, 2004.
114. Gopinath, K., and Mayuram, M. "Gear Failure Lecture." Indian Institute of Technology - Madras.
115. Archard, J. F. "Contact and Rubbing of Flat Surfaces," *Journal of Applied Physics* Vol. 24, No. 8, August 1953.
116. Juvinall, R. C., and Marshek, K. M. *Fundamentals of Machine Component Design*: John Wiley and Sons, 1983.
117. Vanderplaats, G. N. *Numerical Optimization Techniques for Engineering Design*, Fourth Edition, 2005.
118. Deb, K. *Multi-Objective Optimization Using Evolutionary Algorithms*. New York, 2002.
119. Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. "A Fast and Elitist Multi-Objective Genetic Algorithm: NSGA-II," *Evolutionary Computation, IEEE Transactions on* Vol. 6, 2002.

120. Gen, M., and Cheng, R. *Genetic Algorithms & Engineering Optimization*. Canada, 2000.
121. Weile, D. S., and Michielssen, E. "Genetic Algorithm Optimization Applied to Electromagnetics: A Review," *IEEE Transactions on Antennas and Propagation* Vol. 45, No. 3, 1997.
122. Blickle, T., and Thiele, L. "A Mathematical Analysis of Tournament Selection," *Genetic Algorithms: Proceedings of the 6th International Conference*. San Francisco, CA, 1995.
123. Ghasemi, M. R., Hinton, E., and Wood, R. D. "Optimization of Trusses using Genetic Algorithms for Discrete and Continuous Variables," *Engineering Computations*, 1997.
124. WenBin, C., YiJun, L., Li, W., and Xiao-Ling, L. "A Study of the Multi-Objective Evolutionary Algorithm Based on Elitist Strategy," *Asia-Pacific Conference on Information Processing*. 2009.
125. Deb, K., Pratap, A., and Moitra, S. "Mechanical Component Design for Multiple Objectives using Elitist Non-Dominated Sorting GA." Indian Institute of Technology Kanpur, Kanpur, India.
126. Back, T., Hoffmeister, F., and Schwel, H. P. "A Survey of Evolution Strategies," *Proceedings of the Fourth International Conference on Genetic Algorithms*, 1991.
127. Homaifar, A., Lai, S. H. Y., and Qi, X. "Constrained Optimization via Genetic Algorithms," *Simulation*, 1994.
128. Morales, A. K., and Quezada, C. C. "A Universal Eclectic Genetic Algorithm for Constrained Optimization," *Proceedings 6th European Congress on Intelligent Techniques and Soft Computing*, 1998.
129. Joines, J., and Houck, C. "On the use of Non-Stationary Penalty Functions to Solve Nonlinear Constrained Optimization Problems with GAs," *Proceedings of First IEEE International Conference on Evolutionary Computation*, 1994.
130. Kazarlis, S., and Petridis, V. "Varying Fitness Functions in Genetic Algorithms: Studying the Rate of Increase in the Dynamic Penalty Terms," *Proceedings of the Fifth International Conference on Parallel Problem Solving from Nature*. 1998.

131. Potter, M. A., and Jong, K. A. D. "A Cooperative Coevolutionary Approach to Function Optimization."
132. Bendsoe, M. P., and Sigmund, O. *Topology Optimization: Theory, Methods and Applications*: Springer Verlag Berlin Heidelberg New York, 2004.
133. Bendsoe, M. P., and Sigmund, O. "Material Interpolation Schemes in Topology Optimization," *Archive of Applied Mechanics*, 1999.
134. Wang, M. Y., Wang, X., and Guo, D. "A Level Set Method for Structural Topology Optimization," *Computer Methods in Applied Mechanics and Engineering* Vol. 192, No. 1-2, 2003, pp. 227-246.
135. "Multi-Disciplinary Design of an Aircraft Landing Gear with Altair HyperWorks." Altair Engineering, October 2008.
136. "OptiStruct Optimization: Analysis, Concept and Optimization," *HyperWorks, Altair Engineering, Inc.*, 2009.
137. Ashok, S. V., and Schrage, D. P. "Optimization Techniques for Gear Train Design," *American Helicopter Society 69th Annual Forum*. Phoenix, AZ, May 2013.
138. Mavris, D. "Design of Experiments for Practical Applications in Modeling, Simulation and Analysis. Introduction to Response Surface Methods," *Advanced Design Methods I*. ASDL, Georgia Institute of Technology, 2008.
139. Balabanov, V. O., and Venter, G. "Response Surface Optimization with Discrete Variables," *45th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics & Materials Conference*. Palm Springs, California, April 2004.
140. Kirby, M. R. "An Overview of Response Surface Methodology," *Advanced Design Methods I* ASDL, Georgia Institute of Technology, 2004.
141. Baker, T. J. "Mesh Generation: Art or science," *Progress in Aerospace Sciences* Vol. 41, 2005, pp. 29 - 63.
142. "ANSYS Meshing User's Guide." ANSYS, INC., November 2012.
143. Miller, E. "Mapped Face Meshing in ANSYS Workbench." [http://www.padtinc.com/blog/post/2011/02/17/Mapped\\_Face\\_Meshing.aspx](http://www.padtinc.com/blog/post/2011/02/17/Mapped_Face_Meshing.aspx), accessed on November 08, 2011.
144. "ANSYS Workbench User's Guide." ANSYS, Inc., November 2010.

145. Ashok, S. V., and Zender, F. "Topology Optimization of Gear Web using OptiStruct " *HyperWorks Technology Conference*. Detroit, MI, May 2012.
146. Davis, J. "Design Methodology for Developing Concept Independent Rotorcraft Analysis and Design Software," *Masters Thesis, Aerospace Engineering*. Georgia Institute of Technology, Atlanta, GA, 2007.
147. Kish, J. G. "Stacked Compound Planetary Gear Train for an Upgraded Powertrain System for a Helicopter." Vol. 5472386, United Technologies Corporation, Hartford, CT, United States, 1995.
148. Kish, J. "Advanced Rotorcraft Transmission (ART) Program Status," *AIAA/SAE/ASME/ASEE, 27th Joint Propulsion Conference*. Sacramento, CA, 1991.
149. Ashok, S. V., Zender, F., Sirirojvisuth, A., and Schrage, D. P. "A Method to Integrate Drive System Design," *American Helicopter Society 68th Annual Forum*. Fort Worth, TX, May 2012.
150. "PC Based Development, Recurring Production, and Operating and Support Cost Model User's Guide ". Bell Helicopter, October 2001.