Dynamic Tailoring and Tuning for Space-Based Precision Optical Structures

by

Rebecca Ann Masterson

Submitted to the Department of Mechanical Engineering
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2005

Author .....
Department of Mechanical Engineering
September 30, 2004

Certified by.....
Warren P. Seering
Professor of Mechanical Engineering
Thesis Committee Chairman

Certified by.....
David W. Miller
Associate Professor of Aeronautics and Astronautics
Thesis Supervisor

Accepted by .....
Lallit Anand
Chairman, Department Committee on Graduate Students

Dynamic Tailoring and Tuning for Space-Based Precision Optical Structures

by

Rebecca Ann Masterson

Submitted to the Department of Mechanical Engineering
on September 30, 2004, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

Next-generation space telescopes in NASA's Origins missions require use of advanced imaging techniques to achieve high optical performance with limited launch mass. Structurally-connected Michelson interferometers meet these demands, but pose specific challenges in the areas of system dynamics and controls, uncertainty management and testing. The telescope optics must meet stringent positional tolerances in the presence of environmental and on-board disturbances, resulting in heavy demands on structural dynamics and control. In addition, fully integrated system tests are cost-prohibitive due to the size and flexibility of the system coupled with the severe differences between the on-orbit and ground testing environments. As a result, the success of these missions relies heavily on the accuracy of the structural and control models used to predict system performance.

In this thesis, dynamic tailoring and tuning are applied to the design of precision optical space structures to meet aggressive performance requirements in the presence of parametric model uncertainty. Tailoring refers to changes made to the system during the design, and tuning refers to adjustments on the physical hardware. Design optimizations aimed at improving both performance and robustness are considered for application to this problem. It is shown that when uncertainty is high and performance requirements are aggressive, existing robust design techniques do not always guarantee mission success. Therefore, dynamic tuning is considered to take advantage of the accuracy of hardware performance data to guide system adjustments to meet requirements. A range of hardware tuning techniques for practical implementation are presented, and a hybrid model updating and tuning methodology using isoperformance analysis is developed.

It is shown that dynamic tuning can enhance the performance of a system designed under high levels of uncertainty. Therefore, robust design is extended to include tuning elements that allow for uncertainty compensation after the structure is built. The new methodology, Robust Performance Tailoring for Tuning creates a design that is both robust to uncertainty and has significant tuning authority to allow for hardware adjustments. The design methodology is particularly well-suited for high-

performance, high-risk missions and improves existing levels of mission confidence in the absence of a fully integrated system test prior to launch. In the early stages of the mission the design is tailored for performance, robustness and tuning authority. The incorporation of carefully chosen tuning elements guarantees that, given an accurate uncertainty model, the physical structure is tunable so that system performance can be brought within requirements. It is shown that tailoring for tuning further extends the level of parametric uncertainty that can be tolerated at a given performance requirement beyond that of sequential tailoring and tuning, and is the only design methodology considered that is consistently successful for all simulated hardware realizations.

Thesis Committee Chairman: Warren P. Seering
Title: Professor of Mechanical Engineering

Thesis Supervisor: David W. Miller
Title: Associate Professor of Aeronautics and Astronautics

Acknowledgments

This work was supported by a Michelson Graduate Fellowship from the Jet Propulsion Laboratory and was performed at the MIT Space Systems Laboratory (SSL) under the direction of Professor David Miller.

I would like to thank all of the members of the MIT Space System Laboratory for the friendship, support and knowledge that they have given to me during my six years as a graduate student. Marilyn Good, Sharon Leah Brown, and Paul Bauer provided their time and assistance in many areas. The members of the DOCS research group: Prof. Oli deWeck, Scott Uebelhart, Deborah Howell, Soon-Jo Chung, Dave Lobosco and Ryan Lim engaged in insightful discussions, made suggestions and did numerous little favors for me along the way. Scott and Dr. Ray Sedwick donated their computers so I could run long optimization problems. Dr. Carl Blaurock provided technical support for the DOCS toolbox and was always available for long discussions on physical parameter gradients. My officmates, Dr. Alice Liu, Simon Nolet and Alvar Saenz-Otero provided a pleasant work environment for me every day. They put up with my mood swings and engaged in useful discussions about tailoring and tuning, MATLAB, graphics and anything else that may have popped into my head in the course of the day. Alice was also my roommate and deserves special recognition for making sure I was fed and being a huge source of emotional support during some very stressful times.

I could not have finished this work without the support and encouragement of my friends and family. Mark Hilstad, Steve Sell, and Karen Marias were always there for the lunches at Chicago Pizza, Wendy's and Rebecca's that kept me sane. Karen, thanks for all countless sanity breaks and puppet shows. I'm so glad that we have been able to travel this little PhD journey together. I owe a large debt of gratitude to Arthur Richards for his outstanding technical advice and personal support. Arthur spent a lot of time talking with me about the big picture and served as a sounding board for all of my ideas along the way. He pointed me to useful references, helped get me up to speed on optimization techniques and read the entire thesis, providing me with valuable feedback. In addition, I could always count on him for the daily crossword puzzle and peppermint patties. I also had the support and encouragement of my parents, Robert and Donna Masterson, my brother, Eric Masterson and my fiance, Gordon Wong. Gordon's support went above and beyond all expectations. He radically changed his lifestyle by moving onto MIT campus from Back Bay while I was finishing this thesis. He has been very patient with me through the entire thesis process and has never wavered in his support. Thank you Gordon, for helping me succeed and keeping me grounded.

I had the pleasure of working with a wonderful PhD committee. Dr. Allen Bronowicki from Northrup Grumman Space Technologies was a great source of information and has taught me much through his example. Prof George Barbastathis taught me optics so I could better understand the systems I was working on and provided discussions and support throughout the process. Prof. Warren Seering was my committee chair and served as my liaison to the mechanical engineering department. Warren

has supported me and my research since I was undergraduate, and I am very lucky to have wandered into his office seven years ago looking for a senior thesis topic. My research advisor, Prof. David Miller, has been a great source of technical guidance and support throughout my entire graduate career. He made my graduate experience very positive and has been an inspiration to me. I cannot thank him enough.

This thesis is dedicated to my parents, Robert and Donna Masterson and to my brother, Eric Masterson.

Contents

1 Introduction **19**
- 1.1 Space-Based Interferometry . . . 20
  - 1.1.1 Astrometry and Imaging . . . 20
  - 1.1.2 Planet Detection . . . 22
  - 1.1.3 Technical Challenges . . . 23
  - 1.1.4 Missions . . . 24
- 1.2 Problem Statement . . . 25
  - 1.2.1 Background . . . 26
  - 1.2.2 Approach . . . 29
  - 1.2.3 Research Objectives . . . 30
- 1.3 Previous Work . . . 31
- 1.4 Thesis Roadmap . . . 37

2 Performance Tailoring **41**
- 2.1 PT Formulation . . . 42
- 2.2 SCI Development Model . . . 42
- 2.3 PT Implementation . . . 48
  - 2.3.1 Performance Metric . . . 48
  - 2.3.2 Performance Gradients . . . 50
  - 2.3.3 Design Variables . . . 53
  - 2.3.4 Finite Element Gradients . . . 54
  - 2.3.5 Constraints . . . 55
- 2.4 Optimization Algorithms . . . 56
  - 2.4.1 Sequential Quadratic Programming . . . 57
  - 2.4.2 Simulated Annealing . . . 59
- 2.5 Performance Tailoring Results . . . 62
  - 2.5.1 Comparison of Optimization Algorithms . . . 63
  - 2.5.2 Performance Tailored Design . . . 66
- 2.6 Summary . . . 71

3 Robust Performance Tailoring **73**
- 3.1 Uncertainty . . . 74
  - 3.1.1 Uncertainty Models . . . 75
  - 3.1.2 Uncertainty Analysis . . . 75
  - 3.1.3 Example: SCI Development Model . . . 78

3.2 RPT Formulation . . . 84
3.2.1 Anti-optimization . . . 85
3.2.2 Multiple Model . . . 87
3.2.3 Statistical Robustness Metric . . . 88
3.3 RPT Designs . . . 90
3.3.1 Algorithm comparisons . . . 90
3.3.2 Objective function comparisons . . . 95
3.3.3 Comparison to PT design . . . 98
3.4 Limitations: Design Regimes . . . 104
3.5 Summary . . . 107

4 Dynamic Tuning **109**
4.1 Tuning Formulation . . . 110
4.1.1 Example . . . 112
4.1.2 Design Regimes . . . 122
4.2 Tuning in Practical Application . . . 124
4.2.1 Hardware-only Tuning . . . 125
4.2.2 Model-only Tuning . . . 129
4.2.3 Example . . . 132
4.3 Isoperformance Updating for Tuning . . . 136
4.3.1 Model Updating . . . 138
4.3.2 Isoperformance . . . 139
4.3.3 Tuning Algorithm . . . 140
4.3.4 Examples . . . 143
4.3.5 Comparison of Tuning Methods . . . 148
4.4 Summary . . . 150

5 Robust Performance Tailoring with Tuning **153**
5.1 RPTT Formulation . . . 154
5.2 SCI Development Model . . . 158
5.2.1 Optimization Algorithms . . . 159
5.2.2 Comparison to PT and RPT Designs . . . 160
5.2.3 Design Regimes . . . 167
5.3 RPTT Simulations . . . 169
5.3.1 Tuning Authority . . . 169
5.3.2 Hardware Simulations . . . 171
5.4 Summary . . . 179

6 Focus Application: Structurally Connected TPF **183**
6.1 Model Description . . . 183
6.1.1 Reaction Wheel Assembly Disturbances . . . 185
6.1.2 Vibration Isolation . . . 187
6.1.3 Plant . . . 188
6.1.4 Attitude Control System . . . 196
6.1.5 Integrated Model Nominal Performance . . . 196

6.2 Design Parameters . . . 199
6.2.1 Tailoring . . . 199
6.2.2 Tuning . . . 201
6.2.3 Uncertainty . . . 201
6.3 Optimization Implementation . . . 202
6.4 Results . . . 205
6.5 Summary . . . 210

7 Conclusions and Recommendations **213**
7.1 Thesis Summary . . . 213
7.2 Contributions . . . 217
7.3 Future Work . . . 218

A Gradient-Based Optimization **221**
A.1 Steepest Descent . . . 222
A.1.1 Stepsize Selection . . . 223
A.2 Newton's Method . . . 224
A.3 Conjugate Gradient . . . 225

List of Figures

1-1 Timeline of Origins missions. 20
1-2 Artists' concepts of SIM and TPF. 25
1-3 Current model validation and performance assessment approach. . . . 27

2-1 Schematic of SCI development model. 43
2-2 Frequency response functions of SCI 47
2-3 T_z FRF of SCI: with and without ACS 50
2-4 Simulated annealing algorithm. 60
2-5 Development model generation algorithm. 63
2-6 Search results for PT with simulated annealing. 65
2-7 Modal energy: nominal and PT designs 68
2-8 Critical modes shapes for nominal and PT designs. 70

3-1 Uncertainty propagation on PT design. 80
3-2 PT model with uncertainty: FRFs and PSDs. 81
3-3 % Total energy by mode: PT with uncertainty. 83
3-4 First bending mode: PT design with uncertainty. 84
3-5 Performance vs. weighting: VSR RPT optimization. 95
3-6 Nominal and worst case performance for PT and RPT optimizations. 96
3-7 Modal energy: PT and RPT designs with nominal uncertainty. 99
3-8 Critical mode shapes: PT and RPT designs, $\vec{p} = \vec{p}_0$. 100
3-9 Modal energy: PT and RPT designs with worst-case uncertainty. . . 102
3-10 Critical mode shapes: PT and RPT designs, $\vec{p} = \vec{p}_{wc}$. 103
3-11 RMS OPD for PT and RPT designs vs Δ. 105
3-12 Performance requirement vs uncertainty: PT and RPT designs. . . . 106

4-1 PT worst-case tuning results. 115
4-2 RPT AO worst-case tuning results 117
4-3 Modal energy: tuned worst-case PT design. 118
4-4 Mode shape comparison: PT untuned and tuned. 119
4-5 Modal energy: tuned worst-case RPT AO design. 121
4-6 Mode shape comparison: RPT AO untuned and tuned. 122
4-7 Requirement vs uncertainty for PT and RPT designs with tuning. . . 123
4-8 Barrier method implementation. 128
4-9 Model tuning results. 135
4-10 Hardware tuning results. 137

4-11 Isoperformance tuning algorithm. 141
4-12 Isoperformance tuning results on SCI model: $p_1 = E_1$, $p_2 = E_2$. 144
4-13 Isoperformance tuning results on SCI model: $p_1 = E_1$, $p_2 = \rho_1$. 147

5-1 Nominal, worst-case and tuned performance for all designs. 162
5-2 Modal energy: RPTT design. 164
5-3 Modal characteristics of first bending mode in all designs. 166
5-4 Performance requirement vs uncertainty: all designs. 168
5-5 RPTT design performance vs. weighting. 170
5-6 Hardware simulation algorithm. 172
5-7 Monte Carlo uncertainty space, $\Delta = 10\%$. 173
5-8 PT, RPT and RPTT Simulation results, $\Delta = 10\%$, $\sigma_{z_{req}} = 220\mu$m . . 174
5-9 Monte Carlo uncertainty space, $\Delta = 21.5\%$. 176
5-10 PT, RPT and RPTT Simulation results, $\Delta = 21.5\%$, $\sigma_{z_{req}} = 330\mu$m. . 178
5-11 RPTT design performance vs. weighting. 180

6-1 Schematic of TPF SCI Model. 184
6-2 SCI TPF plant finite element model. 185
6-3 RWA disturbance model. 188
6-4 Transfer functions of Isolator models. 189
6-5 TPF SCI truss bay geometry. 191
6-6 Structural model schematic showing optical paths. 195
6-7 Output PSD and cumulative variance from nominal TPF SCI design. 197
6-8 Mode Shapes of nominal SCI TPF design. 198
6-9 Schematic of TPF SCI tailoring parameters. 200
6-10 Implementation flow for generating integrated TPF model. 204
6-11 SCI TPF PT design 207
6-12 SCI TPF RPT Design 207
6-13 SCI TPF RPTT Design 208
6-14 Nominal, worst-case and tuned performances for all TPF SCI designs. 208

List of Tables

1.1 Effect of simulation results on mission. 28
1.2 Effect of simulation results on mission with tuning. 30

2.1 Mass breakdown of SCI development model. 44
2.2 Natural frequencies and mode shapes of nominal SCI model 45
2.3 Tailoring parameters for SCI sample problem. 53
2.4 PT optimization results, $J_0 = 471$ μm. 63
2.5 SA algorithm parameters. 64
2.6 PT optimization: MC SQP optimizations. 66

3.1 Uncertainty parameters for SCI development model. 78
3.2 Uncertainty propagation results: PT design. 80
3.3 Algorithm performance: anti-optimization. 91
3.4 Algorithm performance: multiple model, $\beta_i = 1/n_{pv}$. 92
3.5 Algorithm performance: statistical robustness, $\alpha = 0.5$. 93

4.1 Tuning parameters for SCI development model. 113
4.2 Tuning performance summary for PT and RPT designs. 113
4.3 Hardware Model Data. 133
4.4 Uncertainty parameters for SCI development model. 145
4.5 Hardware Model Data: Example 2. 146
4.6 Tuning results on fifty hardware simulations. 149

5.1 Algorithm performance: RPTT, $\alpha = 0.0$, $\Delta = 0.1$. 160
5.2 Performance and design parameters for optimal designs. 161
5.3 Performance and parameters for tuned worst-case realizations. 161

6.1 RWA disturbance model parameters. 187
6.2 TPF SCI model mass breakdown. 190
6.3 Truss properties. 190
6.4 Truss element and grid numbering. 191
6.5 Bus model elements and properties. 192
6.6 Primary mirror properties. 193
6.7 Primary mirror mount properties. 193
6.8 TPF SCI instrument elements. 194
6.9 Critical modes of nominal TPF SCI design. 197
6.10 TPF SCI model tailoring parameters. 199

6.11 TPF SCI model tuning parameters. 201
6.12 TPF SCI model uncertainty parameters. 202
6.13 Design parameters for nominal and optimal TPF SCI designs. 206
6.14 Performance predictions for all TPF SCI designs. 209
6.15 Tuning parameters for tuned worst-case TPF SCI realizations. 209

Nomenclature

Abbreviations

ACS	attitude control system
AO	anti-optimization
BSD	barrier steepest descent
DOCS	Disturbance Optics Controls Structures
CSI	control-structures interaction
CTE	coefficient of thermal expansion
ESO	European Southern Observatory
FEM	finite element model
FRF	frequency response function
GSE	governing sensitivity equation
HST	Hubble Space Telescope
JPL	Jet Propulsion Laboratory
JWST	James Webb Space Telescope
LOS	line of sight
LQR	lineaer quadratic regulator
IR	infrared
MC	Monte Carlo
MM	multiple model
MPC	Model Predictive Control
MUF	model uncertainty factor
NGST	Next Generation Space Telescope
OPD	optical pathlength difference
PM	primary mirror
PSD	power spectral density
PT	Performance Tailoring
QP	quadratic programming
RMS	root mean square
RPT	Roubst Performance Tailoring
RPTT	Robust Performance Tailoring for Tuning
RWA	reaction wheel assembly

SA	simulated annealing
SCI	structurally-connected interferometer
SIM	Space Interferometry Mission
SIT	system intergration test
SQP	sequential quadratic programming
SR	statistical robustness
TPF	Terrestrial Planet Finder
TT	Tailoring for Tuning
ULE	ultra-low expansion
WC	worst-case

Symbols

A_i	cross-sectional area
A, B, C	state-space matrices
$\hat{B}(\cdot)$	barrier function
$B_{\hat{x}w}$	disturbance input mapping matrix
B_I	interferometric baseline
C_i	amplitude coefficient
$C_{z\hat{x}}$	output mapping matrix
E_i	Young's Moudulus
F_x, F_y, T_z	force and torque disturbances
$G(\cdot)$	transfer function
H	Hessian
I	identity matrix, area moment of inertia
K	finite element stiffness matrix
J	optimzation cost
$\mathcal{L}$	Langrangian
L	length
M	finite element mass matrix
M_{coll_i}	collector mass
M_{comb}	combiner mass
$\bar{M}$	total mass limit
P	uncertainty parameter space
P_{iso}	isoperformance set
T	temperature
R	distance to star
S_{zz}	output PSD
S_{ww}	disturbance PSD
X, Y, Z	coordinate system axes
Y	tuning parameter space
d_i	cross-sectional diameter

d_k	optimization search direction
f	frequency
$f(\cdot)$	performance function
$\vec{g}(\cdot)$, $\vec{h}(\cdot)$	constraints
h	height
h_i	harmonic number
k_c, k_a, k_r, k_T	simulated annealing paramters
m_i	design mass
$\vec{p}$	uncertainty parameters
q	modal degrees of freedom
s	Laplace variable
w	white noise, width
$\hat{x}$	physical degrees of freedom
$\vec{x}$	tailoring parameters
$\vec{y}$	tuning paramters
z	output metric, dummy optimization cost
Δ	uncertainty level
Ω, ω_j	natural frequeny
Φ, ϕ_j	mode shapes
Σ_z	output covariance matrix
Σ_q	state covariance matrix
Z, ξ, ζ	damping ratio
α_k	optimization step size
α, β	optimization weights
λ	Lagrange multiplier
μ_k	barrier sequence
ν	Poisson's ratio
ρ	material density
σ^2	variance (mean square if zero mean)
σ	standard deviation (RMS if zero mean)

Subscripts and Superscripts

$(\cdot)^*$	optimal solution
$(\cdot)^H$	Hermitian (complex-conjugate tanspose)
$(\cdot)_{HW}$	hardware value
$(\cdot)^T$	transpose
$(\cdot)_{WC}$	worst-case value
$(\cdot)_0$	nominal value
$(\cdot)_{req}$	requirement
$(\cdot)_t$	tuned value

Chapter 1

Introduction

Next-generation precision telescopes, such as space-based interferometers, push the boundaries of existing design methodologies. In order to achieve science goals, nanometer level pointing stability and precise wavefront control are necessary. These requirements place a large burden on the structural dynamics and control systems. In addition, fully-integrated system tests on interferometers are cost-prohibitive, placing a heavy dependence on models and simulation for pre-launch performance assessment. In effect, inherently uncertain models and approximations are relied upon to provide precise performance predictions. Traditional robust design techniques are not adequate to guarantee success under such conditions.

To address this problem, a two-stage design methodology called Robust Performance Tailoring for Tuning (RPTT) is developed. It is an extension to robust design that includes, in the cost function, the concept of tunability . Uncertainty compensation is shared between robust design and hardware tuning. RPTT is employed during the design stage when uncertainty is high but the design parameter space is large. Hardware tuning is used after the components are built, when the performance is known, but only limited adjustments are possible. The methodology is developed in the context of structurally-connected space-based interferometers such as NASA's Space Interferometry Mission (SIM) and Terrestrial Planet Finder (TPF), but is also applicable to other precision optical systems such as the James Webb Space Telescope (JWST) and coronographs.

1.1 Space-Based Interferometry

NASA's Origins program is an on-going effort aimed at exploring both the nature of the universe and its inhabitants. The specific goals of the program are to answer questions about the origin of our universe and to search for other Earth-like planets in nearby solar systems. The program consists of a family of missions that span over twenty years (Figure 1-1). The earliest telescopes, including the Hubble Space Telescope (HST), are in operation and provide valuable information to scientists and astronomers at the time of this writing. Upcoming missions include SIM, JWST, and TPF (interferometer and coronograph). All three of these missions have ambitious science goals that push the boundaries of engineering across many disciplines, including optics, structures and control.

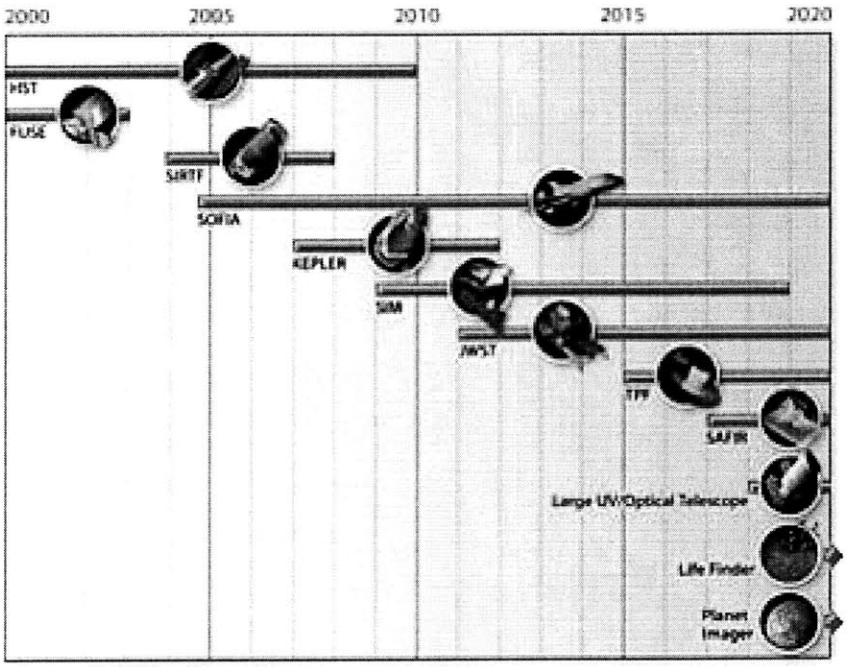


Figure 1-1: Timeline of Origins missions [3].

1.1.1 Astrometry and Imaging

In order to achieve the first of Origins' goals, mapping the universe and exploring its origins, a telescope with very high angular resolution is necessary. Angular resolution

is an optical metric that describes the accuracy of an optical system. It is defined as the minimum resolvable angular separation between two objects [54]. Consider two objects located far from a telescope, but very close to each other. A low-angular resolution telescope is unable to resolve the two objects, and they appear to an observer as a single blurred object. A telescope with high angular resolution, however, presents an image of two distinct light sources.

One challenge faced in the design of these telescopes is the limitations imposed by a trade-off between system mass and optical performance. In the case of a monolithic telescope, such as HST, the angular resolution of the system scales proportionally with the diameter of the primary mirror. However, as the primary mirror becomes larger it also becomes more massive and more expensive to manufacture and launch. Therefore mass and cost budgets limit the angular resolution that can be achieved with a monolithic system.

One alternative to the monolithic telescope design that has generated much interest over the past few decades is astronomical interferometers [105, 8]. These instruments provide high resolution imaging and astrometry at a significant savings of mass and volume compared to a monolithic design. Interferometers function by combining the light gathered by two or more smaller apertures separated by a large baseline. The angular resolution of an interferometer increases proportionally to the length of this baseline. Therefore, it is not necessary to manufacture and launch very large mirrors. Instead, it is only necessary to place collecting mirrors sufficiently far away from each other to achieve the desired performance.

A number of ground-based stellar interferometers are currently in operation. These include the Keck Observatory [29] on Mauna Kea in Hawaii, the Sydney University Stellar Interferometer [33, 34] (SUSI) in Australia, and the Navy Prototype Optical Interferometer [9] (NPOI) in Flagstaff, Arizona. These systems have already provided valuable astronomical data [58, 59, 30, 38, 77, 32] such as stellar images, astrometric measurements of stellar positions and stellar angular diameters. However, the optical performance of these ground interferometers is limited by atmospheric distortions. The next logical step is to place these systems in space where they can operate

unfettered by the Earth's atmosphere [8].

1.1.2 Planet Detection

The second main goal of the Origins program is to search for life in other solar systems. This aim requires the detection and study of extra-solar, Earth-like planets. Currently, planet detection is done by ground-based telescopes that measure the motion of a star due to an orbiting planet. This technique has been successful in locating a number of Jupiter-sized planets, but is not sensitive enough to find smaller planets with mass comparable to that of the Earth. Therefore, the concept of a single instrument that detects Earth-like planets through their radiated heat and then performs a spectroscopic analysis to find signatures of carbon dioxide, water and ozone has been proposed [6].

There are two main challenges associated with this type of detection scheme. First, it is necessary to distinguish the radiated heat from that of the star. If the observing instrument is located on Earth, the heat from the planet is entirely swamped by the radiation from the Earth's atmosphere and the ambient temperature of the optical elements. However, even if the instrument is placed in space, removing these sources of noise, the contrast ratio of the star to the planet is on the order of 10^7. It is necessary, then, to find a way to separate the star radiation from that of the planet. A second challenge arises due to the small angular separation between the planet and the star. As discussed above, a high-angular resolution instrument is necessary to resolve the two sources.

A nulling interferometer operating in the infrared (IR) has been proposed as the solution to the planet detection problem. The concept was first introduced by Bracewell [20]. He and his colleagues suggested using an interferometer to destructively interfere star radiation captured by two apertures separated by a baseline [21]. The destructive interference results in the cancellation of the stellar flux over a broad waveband. If the instrument is designed correctly, the planet emission, which is slightly off-axis from the star, constructively interferes and is reinforced allowing the detection of the planet despite the stronger stellar radiation. Designs for nulling IR

interferometers have been proposed by Leger et al. [71], Mennesson and Mariotti [86] and Angel and Woolf [6].

1.1.3 Technical Challenges

While interferometers do provide solutions to the astrometry and planet detection problems, they also pose a significant set of engineering challenges. In order to produce astrometric measurements, images or to achieve nulling, the interferometers must be able to acquire fringes. Interferometric fringes are the dark and light zones that result from the interference of two light sources (see [54] for a full discussion on interferometry). The acquisition of fringes is possible given that the distances the light travels through the two sides of the interferometer are equal to within a fraction of a wavelength. This optical metric is known as the optical path difference (OPD).

Recall that the angular resolution of the system scales with the baseline. Large baseline can be achieved by placing the collecting and combining optics on a deployable or articulating truss structure or by flying them on individual spacecraft. The first configuration is known as a structurally-connected interferometer (SCI), while the latter is called a formation flown interferometer or a separated spacecraft interferometer [72]. A high-resolution interferometer operating in the visible regime requires that the light paths be equal to within a few nanometers over a baseline of ten or one hundred meters. In the case of the SCI architecture, flexibility in the supporting structure and the presence of on-board disturbances place a large demand on the structural dynamics and control systems to achieve the required stability. The formation flown problem is also difficult, but its challenge lies in the knowledge of position and the control of the spacecraft formation. The work included in this thesis is focused on the SCI architecture.

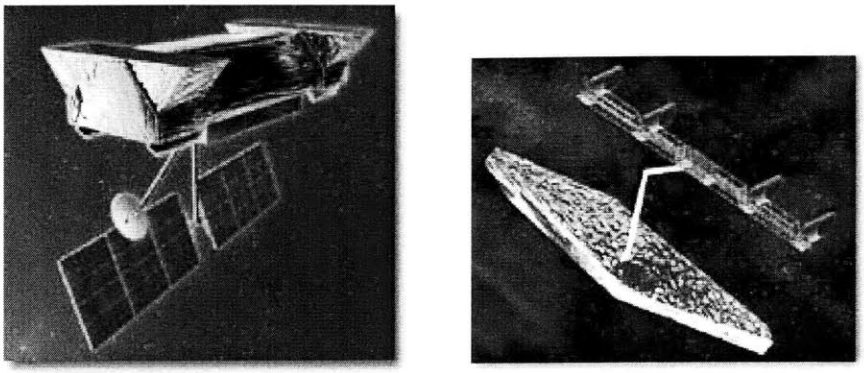
A second area of difficulty associated with space-based interferometers is system integration and test. A full system integration test (SIT) previous to launch is not practical due to the large contrast between the ground testing and operations environments. The system is designed for operation in a vacuous, zero-gravity environment, but a ground test is influenced by gravity and atmospheric distortions. In the case of

the SCI, both the size and flexibility of the structure cause difficulties when creating a testing environment on Earth that is consistent with the operations environment in space. More importantly, a pseudo-star is necessary to inject collimated light across the entire baseline of the interferometer simulating the observation of a star many light years away. A pseudo-star of this type is currently used with the Microarecond Metrology Testbed which is a technology demonstrator for the Space Interferometry Mission [69]. In this case, only one pseudo-star is necessary to test the single baseline interferometer. However, to complete the objectives of the Origins missions a multiple baseline interferometer is necessary. Therefore, one pseudo-star per baseline is required to test all baselines simultaneously. In addition, each pseudo-star must be more sensitive than the interferometer instrument that it is designed to test. Such a SIT is theoretically possible, but is cost-prohibitive.

1.1.4 Missions

The Space Interferometry Mission (SIM) is the first interferometer in the suite of Origins missions and is scheduled for launch in 2009 . The primary goal of SIM is planet detection and galaxy mapping through precision astrometric measurements. SIM is also a technology demonstrator for space-based interferometry and hopes to pave the way for future Origins missions. The current SIM design, shown in Figure 1-2(a) [1], is a structurally-connected Michelson interferometer with four individual parallel baselines. Each baseline is approximately 10 meters long and consists of 35 cm diameter aperture telescopes that collect star light, compress it and direct it through the optical train to the beam combiner. Two of the interferometers are guide interferometers and are pointed directly at guide stars to provide precise inertial reference data. The other two interferometers are science instruments and are used for observing science targets [69].

The Terrestrial Planet Finder (TPF) is a second generation Origins mission scheduled for launch between 2012-2015. The science goals of the mission include detecting Earth-like planets in a habitable zone around nearby stars, searching for atmospheric signatures of life through spectroscopy and performing high-resolution imaging of



(a) (b)

Figure 1-2: Artists' concepts of Origins missions: (a) SIM [1] and (b) TPF [2].

astrophysical targets [2, 70]. The original baseline design for TPF is an infrared separated spacecraft interferometer [13]. However, as the result of a series of industry studies at the time of this writing there are three competing architectures for TPF: an infrared (IR) structurally-connected interferometer, an infrared formation-flown interferometer and a visible light coronograph. Figure 1-2(b) shows an artist's concept of the SCI design proposed by Lockheed Martin. For the purpose of the work in this thesis only the SCI architecture is considered.

1.2 Problem Statement

One problem inherent in complex system design arises due to a trade-off that occurs over the development phase between design flexibility and the accuracy of performance predictions. Early in the mission development, models of the proposed design are created to assess system performance. At this stage, design alterations come at a modest cost, but the models used to predict performance are uncertain, resulting in low-confidence predictions. Designing the system to meet requirements across this large uncertainty space may not be possible. In the later phases of the program, flight components and limited integrated test data become available dramatically increasing the accuracy of the performance predictions. However, an unfortunate consequence

may be that it becomes *certain* that the system will fail. Design changes at this stage are quite expensive now that flight hardware is built.

Space-based, structurally-connected interferometers are particularly affected by this trade since they are classified as both high-performance and high-risk systems. The precision optical performance required for astrometry, nulling and imaging coupled with the size and flexibility of the instrument place heavy demand on the structural dynamics and control systems, while the high cost of a fully-integrated system test limits the ability to guarantee desired on-orbit performance prior to launch. As a result, *it is necessary to design the system very precisely, yet rely heavily on models and simulations, which are approximations, to predict performance.*

1.2.1 Background

One approach to the design of these systems is shown in Figure 1-3. The figure is broken up into three different regions. In Region I, testbeds are used to validate modeling techniques and generate model uncertainty factors (MUFs) [18]. Testbed models are developed and performance predictions from these models are compared to data from the testbeds. The models are refined until all of the major features visible in the testbed data are captured in the model. Then MUFs are chosen to approximate any remaining differences between the model and the data that are difficult to quantify. Model predictions that have been adjusted by MUFs should be conservative when compared to the testbed data.

In Region II, the component models are used to predict performance and drive system design. The component developers deliver models of their respective designs. The MUFs are applied to the component models and they are integrated to evaluate system performance. The component designs and associated models are iterated upon until the predicted system performance meets requirements. Once the designs are validated in this way, the developers build and deliver the flight system components. Upon delivery, the components are tested and compared with the conservative component models before acceptance. If the test data lies within the model predictions the models are considered validated, and the components are accepted.

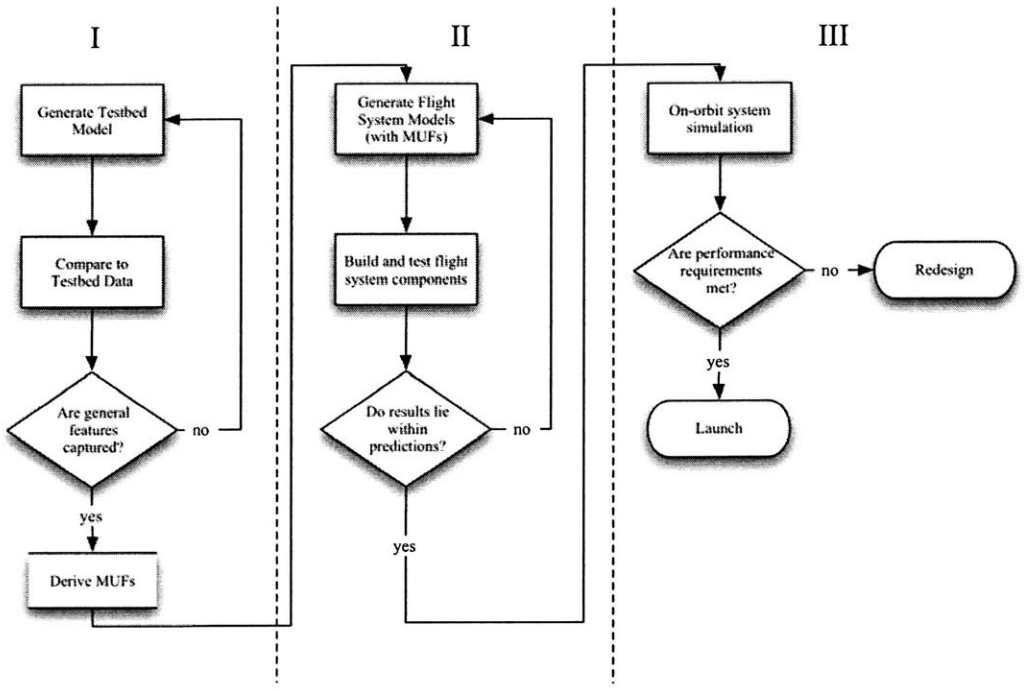


Figure 1-3: Current model validation and performance assessment approach.

In Region III, the test data from the component hardware is combined with an on-orbit simulation to predict system performance in the operational environment. The predictions are compared to the limited system validation test data that is available as well as to the requirements. Component interface uncertainty becomes relevant in this step since a blend of models and data are used in the simulation. If the simulation prediction meets requirements and the validation tests match predictions, the system is launched. If the simulation does not meet requirements, launch is delayed to allow for redesign and adjustments.

Four mission scenarios that could arise based on the process described above are listed in Table 1.1. In the first scenario, the simulation predictions meet performance, the system is launched and the on-orbit performance matches the predictions resulting in a successful mission. In the second scenario, the simulation predicts adequate performance, but the predictions are incorrect and on-orbit performance is not adequate leading to mission failure. In the third scenario, the predictions are incorrect again, but this time the simulation predicts poor performance while the on-orbit behavior would have been adequate, and the result is an unnecessary delay in launch. Finally,

Table 1.1: Effect of simulation results on mission.

#	Simulation Prediction	Action	On-Orbit Performance	Result
1	good	launch	good	success
2	good	launch	bad	failure
3	bad	no launch	good	delay
4	bad	no launch	bad	delay

in the fourth scenario, the simulation correctly predicts poor performance and the resulting launch delay and redesign is the appropriate action.

It is interesting to note that only the first scenario results in a successful mission. All other scenarios lead to failure or a delay in launch. The fourth scenario is appropriate given the conditions and may eventually lead to success as long as the simulation predictions continue to be correct after adjustments and redesign. In the second and third scenarios the simulations predict the system operation incorrectly. As a result, scenario two is a complete failure, while scenario three calls for unnecessary redesign that may lead to scenario two upon eventual launch. *The simulation prediction is therefore a single-point failure in this approach.*

One way to increase the chances of mission success is to ensure that the simulation predictions are always correct. Accomplishing this goal requires a large investment in modeling and analysis. Uncertainty modeling, structural optimization and robust design are all known techniques that are frequently employed to increase the accuracy of performance predictions. Uncertainty modeling involves identifying the sources of model uncertainty, quantifying them and producing bounds on performance predictions. Uncertainty models are combined with structural optimization in the field of robust design to find a design that meets the desired performance requirements and is insensitive to model uncertainties. However, it is well known in the field of robust control that robustness is achieved at the expense of nominal performance, and so, as a result, these tools alone may not be adequate for a system that must meet aggressive performance requirements under a high level of uncertainty. In this thesis, the problem of how to extend robust design techniques to ensure that stringent perfor-

mance requirements are met on-orbit when high-uncertainty models and simulations are depended upon for launch decisions is addressed.

1.2.2 Approach

The approach taken in this thesis is a formal synthesis of robust structural design and hardware tuning. The burden of uncertainty management is shared by both simulation and hardware. Instead of mitigating uncertainty through robust design or excessive modeling and component prototyping, it is managed by optimizing for a mix of performance, robustness and tunability. Such a scheme addresses the problem of performance prediction accuracy vs. design flexibility by choosing parameters that can be tuned during component testing, system integration, or on-orbit, to affect the system performance thereby increasing hardware design flexibility. If the system is designed such that these parameters have enough authority to guarantee that performance requirements can be met regardless of where the actual system lies in the uncertainty space, then the resulting system is significantly more likely to be successful.

Consider the effect of such a design methodology on the scenarios in Table 1.1. The simulations are augmented with a formal pre-launch and on-orbit tuning process so that the predictions only need to meet performance within specified error bounds. If the system is guaranteed to be "tunable" within these bounds then the scenarios are as shown in Table 1.2. The simulation prediction metric has changed from "good" or "bad" to "meet requirement" or "within bound." This distinction is made because if the prediction does not meet requirements, but is within the tunable range then launch is still the correct action. Note that *all* four situations lead to launch and mission success. The dependence on simulation is reduced by allowing for on-orbit adjustments and the requirements for launch are relaxed to allow for model uncertainty resulting in a greater probability of mission success.

When discussing the design methodology throughout the thesis the following terminology is used. The term *Performance Tailoring (PT)* describes the process of structural design for performance only, i.e. structural optimization without consid-

Table 1.2: Effect of simulation results on mission with tuning.

#	Simulation Prediction	Action	On-Orbit Performance	Action	Result
1	meet req	launch	good	none	success
2	meet req	launch	bad	tune	success
3	within bound	launch	good	none	success
4	within bound	launch	bad	tune	success

eration of model uncertainty. *Robust Performance Tailoring (RPT)* refers to robust structural optimization in which robustness to a specified uncertainty model is included in the design objective. *Robust Performance Tailoring for Tuning (RPTT)* is an extension to RPT in which a design is optimized for performance, robustness and tunability. In this approach the tuning or adjustment on hardware is anticipated and explicitly planned for during the design optimization.

1.2.3 Research Objectives

The main objective of this thesis is to develop a design methodology that is appropriate for high-performance and high-risk systems such as space-based interferometers. Available robust design tools are applied to this problem and evaluated for effectiveness. An extension to robust design that includes a formal tuning methodology is developed. The specific research goals, and thesis chapters in which they are addressed, are as follows:

- Apply structural optimization and the robust design framework and tools to space-based interferometer design. Identify control and noise parameters relevant to the design of precision structures for optical space systems [Chapters 2 and 3].

- Evaluate the performance of robust design techniques on high-performance and high-uncertainty systems [Chapter 3].

- Formalize an efficient tuning methodology that can be applied to flight hardware either during component testing or on-orbit operation to bring the system

performance within requirements [Chapter 4].

- Extend robust design techniques to include tailoring for tuning. Define the concept of tunability and incorporate the idea into robust optimization resulting in a formal relationship between structural tailoring and tuning [Chapter 5].

- Demonstrate through simulation that RPTT guarantees that the desired performance can be achieved given an accurate model of parametric uncertainty [Chapter 5].

- Apply the RPTT design methodology to an integrated model of a precision optical space structure. Choose appropriate tailoring and tuning parameters for the focus application [Chapter 6].

1.3 Previous Work

The research presented in this thesis draws from previous work in systems engineering, structural dynamics, integrated modeling, optimization and robust design. In this section a review of the relevant work in these fields is presented.

Structural optimization is a well-developed field with a rich history. The advent of the digital computer made optimizing structural elements in the design process practical. Numerous examples can be found in the literature of sizing, shape and topology optimization [64]. Sizing optimization is typically applied to a truss structure and uses parameters such as member cross-sectional area and plate thickness as design variables. In shape optimization the topology is fixed, but the structural boundaries are allowed to change [56]. In topology optimization much less is known about the structure and the search is for optimal material distributions to yield a preliminary structural configuration [114, 73].

A large body of work exists on the application of sizing optimization to truss problems to minimize mass subject to static constraints such as maximum stress. Many different optimization techniques have been applied to variations of this problem. The reader is referred to Kirsch [66] for an overview of gradient methods and examples. It

has been found that structural design problems tend to have many variables, include nonlinearities in the cost functions and/or constraints, and may be non-convex. As a result, non-gradient or heuristic, search techniques have become popular in recent years. Lagaros et al. apply genetic algorithms to the minimum weight sizing problem and compare the results to those obtained with gradient methods [67]. Hasancebi et al. use simulated annealing to determine the optimal size, shape and layout for minimum weight truss structures subject to stress, stability and displacement constraints [51]. Manoharan and Shammuganathan provide a comparison of four search techniques, Tabu search, simulated annealing, genetic algorithms and branch-and-bound, applied to the truss sizing problem [79].

Structural optimization has also received a large amount of attention from the vibration suppression community. Yamakawa formulated the problem of minimum root mean square (RMS) tip displacement for a cantilevered beam and truss frame structures with deterministic loading [113] . Chen, Bruno and Salama demonstrated increased finite-time energy dissipation through combinatorial optimization of passive and active damping locations in a simulated annealing framework [26]. Langley uses a Quasi-Newton optimization algorithm to minimize kinetic energy and maximum strain in a near-periodic beam system by varying bay lengths and loss factors [68]. Keane, Nair and their colleagues at South Hampton University have done extensive work in vibration minimization through unusual truss geometries. They use genetic and evolutionary algorithms to design trusses that exploit the intrinsic vibration filtering capabilities of non-periodic structures to achieve passive isolation [63, 95, 96]. As a continuation of this work, Moshrefi-Torbati and Keane have published the first experimental validation of topology optimization for passive isolation [91]. The authors built the optimized structure and demonstrated significant vibration suppression over the traditional design.

As the science requirements for telescope missions became move aggressive the structural vibration and control communities began to examine the idea of considering structure and control design in parallel as an alternative to traditional design methods. Historically, designs are driven by mass and static requirements. The struc-

tural dynamics are analyzed after an architecture is chosen and detailed design has begun. The dynamics are then characterized and active control is implemented to compensate for undesirable behavior. The high-performance required by space-based interferometers highlighted the need for a new approach. It was believed that executing the system design sequentially would not guarantee the high levels of stability necessary for mission success. The need for high control bandwidth to acheive high performance coupled with the detrimental effects from the interaction between this control and flexible dynamics led to the field of control-structures interaction.

There are many contributions in the field of combined control/structures design. For example, von Flotow shows that a truss structure can be optimized, or tailored, to generate a design that is more easily controlled [111]. Other authors include control in the optimization by adding control energy to the cost functional [87] or optimizing over closed-loop, instead of open-loop, performance [90, 14]. Uchida and Onoda extend the problem by considering the optimal design of passive damping for performance improvement on space structures. They optimize structural parameters and linear quadratic regulator (LQR) gains to minimize both mass and control effort [109]. In a more recent publication, Anthony and Elliott compare combined structure and control optimization strategies to individual optimizations [7]. Genetic algorithms are used to perform combinatorial optimization of joint and actuator locations to reduce the average vibrational energy in a given frequency band. It is shown that the combined optimization solution achieves greater attenuation then the sequential solutions.

Crawley, Masters and Hyde formalize the CSI ideas by developing a methodology to unify structural dynamics and controls analysis to provide end-to-end design evaluations for high performance structures. They stress the importance of considering both structure and control during conceptual design [31]. These ideas of conceptual design are applied to a model of a stellar interferometer precision structure [81] and validated on a scaled experiment [82]. The experimental validation joins that of reference [91] as one of the few published in this area.

Application of these methodologies to precision telescopes led to the inclusion of optical performance in the models. Bronowicki optimized the truss member sizes of a

precision telescope structure for mass and optical performance metrics such as line of sight and wavefront error [22]. Similar efforts have been made by JPL using structural and optical modeling tools developed in-house. Milman, Salama, and Wette optimized truss member areas and control gains to generate a Pareto optimal set of designs that minimize optical performance, control effort and system mass [89]. Combinatorial optimization of passive damper locations on an interferometer truss to minimize RMS OPD is presented by Joshi, Milman and Melody [62].

As a result of this multidisciplinary approach to design and performance assessment the field of integrated modeling has been developed. An integrated model is a system design tool that includes the effects of cross-disciplinary sub-systems. The MIT Space Systems Laboratory, the Jet Propulsion Laboratory, NASA Goddard Space Flight Center and Ball Aerospace have demonstrated the role of integrated models in performance analysis and design of precision space telescopes. Mosier et al. use an integrated model of the Next-Generation Space Telescope (NGST) to trade line of sight (LOS) pointing error against reaction wheel speed and isolation corner frequency [92]. Miller and de Weck present the DOCS (disturbance-structures-optics-controls) integrated modeling environment and apply it to models of NEXUS and NGST [35, 88]. Manil and Leiber apply Ball Aerospace integrated modeling tools to a ground telescope proposed by the European Southern Observatory (ESO) [78]. The integrated telescope model includes structural and optical dynamics, active mirror control and environmental effects. Other examples of Ball implementation include application to a TPF coronograph model [74, 75]. At JPL the integrated modeling efforts are focused on SIM [49] and include attempts to validate the methodology on an interferometer testbed [85, 84].

The combination of structural optimization and integrated modeling may lead to high-performance designs, but if the models that are used in the optimization are not accurate the system that is ultimately built is not guaranteed to perform as predicted. Historically models have been used to help engineers understand the physical behavior of their system. Models are built of a piece of hardware and then predictions are compared to test data and model updating techniques are employed

to bring the two into agreement. In this sense, the models are simply used to verify reality. However, recent trends in complex structures require models to serve as predictors of future behavior. Since models, by definition, are only approximations to reality, there is considerable risk and uncertainty involved in the prediction process. The field of uncertainty and stochastic modeling is growing rapidly. Researchers are working to understand the sources of model uncertainty, develop models of it and propagate the effects of this uncertainty to provide bounds, or statistics, on the performance predictions.

There are many sources of possible model uncertainty, including global modeling, or model structure errors, parametric errors, discretization errors and environmental discrepancies [23, 27, 12, 93]. However, of these, parametric uncertainty is treated most often in the literature as it is the easiest to model and the most difficult to reduce due to lack of sufficient experimental data. There are several forms of uncertainty models with probabilistic models being the most popular. Simonian has compiled a database of measured damping data from twenty-three satellites [106]. He applies statistical models built from this data to an electro-optic jitter problem to obtain the probability density function of the performance, optical pointing error. Hasselman has also been a contributer in this area. He uses a generic modeling database derived from prior analysis and testing to generate uncertainty models for modal mass, damping and stiffness parameters [53, 52]. He compares the use of linear covariance, interval prediction and Monte Carlo propagation of the uncertainties through the analysis to obtain statistical bounds on frequency response functions.

A short-coming of statistical models is that often the data used to build the uncertainty model is insufficient. Furthermore, the data that does exist originates from diverse structural systems making the application to a particular system, such as an interferometer, suspect. In this sense, the uncertainty model itself is uncertain. Ben-Haim and Elisakoff present an alternative to probabilistic models in their monograph on convex uncertainty modeling [15, 41]. The authors suggest that the performance predictions obtained through statistical analysis are quite sensitive to the distribution chosen for the uncertainty model. Therefore, convex, or bounded, models present an

attractive, and more conservative alternative, when there is insufficient data to build an accurate statistical model of parametric uncertainty. When these models are used in uncertainty propagation, bounds on the response are obtained in lieu of statistical output distributions.

These uncertainty models and propagation techniques are ultimately combined with structural optimization to produce designs that can meet performance requirements despite the model uncertainty. This practice is known as robust design and has been popular among control experts for some time [110, 10, 47, 48]. Robust design optimization is a relatively new field in structural design, but has generated a lot of interest due to the rising complexity of systems [40]. Anderson considers the problem of robust actuator and damping placement for structural control using a model with known errors [5]. Park, Hwang and Lee use Taguchi methods to post-process gradient-based structural optimization to find discrete sizing variables [99]. The Taguchi method [107, 102] is a quality-control technique originally developed for circuit design that has recently found application in structural optimization. In a later paper, Park et al. apply the Taguchi method to unconstrained structural optimization to find robust optimal designs of three and ten-bar trusses subjected to applied loads [100]. The areas of the truss members are optimized with conventional methods first to minimize mass and then with the Taguchi method to minimize the sensitivity of the displacement of a given node to variations in the member areas. Constrained robust structural optimization problems are considered as well. Sandgren and Cameron suggest a two-stage hybrid approach that combines the use of genetic algorithms for topology optimization with Monte Carlo uncertainty propagation to determine the statistics of the objective function and/or constraints [103]. The method is computationally expensive, but is successfully applied to a ten-bar truss problem and a higher-fidelity automobile inner panel using probabilistic models of uncertainty. Elishakoff, Haftka and Fang use convex uncertainty models to perform "anti-optimization" or min-max style robust design [42]. They demonstrate their technique on a ten-bar truss problem subjected to uncertain loading as well as stress and displacement constraints.

Although structural optimization and integrated modeling do much to enable the design of complex structures like space-based interferometers there is a lack of application of robust design to this problem in the literature. Structural optimization has been applied to the opto-mechanical jitter problem presented by interferometers, but robust design techniques have not. Instead, the robust design literature is generally concerned with truss-sizing problems for mass minimization or static displacement reduction and not with dynamic cost functions such as optical path jitter. If space-based interferometers are to be designed and launched based on simulation predictions alone, model uncertainty must be considered during the design process. However, in order to achieve robustness, a system sacrifices nominal performance. Therefore, robust design techniques reach their limitation when performance requirements are aggressive and uncertainty is high. In this thesis both the application of robust optimization to SCI design and a technique that can be applied to problems that lie beyond the limits of robust design are presented.

1.4 Thesis Roadmap

A design methodology that extends robust design techniques for application to high-performance, high-risk systems such as space-based interferometers is developed. In this chapter, background information on space-based interferometry is presented and motivates the need for an extension of current design techniques. A review of relevant literature is included.

In Chapter 2, Performance Tailoring, or the optimization of a structure to minimize performance variance alone, is discussed. A simple model of a structurally-connected interferometer, referred to throughout as the development model, is introduced and discussed in detail. Included in the model are the structure, optical performance, and disturbance sources. The performance metric is RMS OPD, and methods for obtaining its gradient with respect to the design parameters are discussed. In addition, an overview of two popular optimization methods, sequential quadratic programming and simulated annealing, is given. These algorithms are used

to solve the performance tailoring problem on the development model. The results are compared to assess efficiency and performance of the optimization techniques, and the optimal design is analyzed.

In Chapter 3, parametric uncertainties in the model are considered, and it is shown that the performance tailored design is very sensitive to changes in the uncertain parameters. Therefore, the problem of space-based interferometer design is posed in a robust design framework. A selection of robust design approaches including multiple model, anti-optimization and statistical robustness are compared and contrasted on the SCI development model. The resulting RPT design is compared to the PT design at both the nominal and worst-case uncertainty values. Specific design regimes are identified by considering the worst-case performance of the PT and RPT designs as the level of uncertainty increases. It is shown that, although robust design significantly increases the amount of uncertainty that can be tolerated, a limit is reached for systems that are highly uncertain yet require a level of performance at which these techniques are no longer adequate.

In Chapter 4, the concept of dynamic tuning is introduced as a way to extend the performance of the PT and RPT designs. Tuning is defined as changes made to the hardware once it is built to bring the system within performance requirements. Possible tuning parameters and appropriate selection criteria are discussed. A spectrum of tuning methodologies are presented ranging from model updating to experimental hardware optimization. Methods are compared through application to the development model, and a hybrid methodology using isoperformance for model updating is developed. Tuning techniques are applied to hardware simulations of the development model to improve performance under parametric uncertainty.

The focus of Chapter 5 is the development of the RPTT design methodology. It is a formal synthesis of the tailoring and tuning techniques detailed in the previous chapters in which the design is tailored to plan for future tuing adjustments on hardware. Tailoring for tuning is demonstrated on the development model, and a comparison of PT, RPT and RPTT techniques is presented over a range of uncertainty levels to illustrate the benefits of formally combining tailoring and tuning. It is shown that

the level of tolerable uncertainty is extended past that of robust design techniques alone.

In Chapter Six, the PT, RPT and RPTT methodologies are applied to a high-fidelity integrated model of a structurally-connected interferometer architecture for TPF. Appropriate tuning, tailoring and uncertainty parameters are identified, and the design methodologies are demonstrated. Trends similar to those exhibited by the development model are observed. Limitations of the optimizations algorithms are identified and recommendations for increasing computational efficiency are made.

Finally, in Chapter 7 the thesis contributions are highlighted and recommendations for future work are enumerated.

Chapter 2

Performance Tailoring

A common approach to designing high-performance systems is to use optimization techniques to find a design that meets performance requirements [113, 26, 56, 109, 62, 51]. One popular problem statement is the design of a minimum mass structure given constraints on load carrying capability or first natural frequency. In this example, the objective is a static quantity; the mass of the structure does not change over time. In contrast, the structural control problem posed by space-based interferometers has driven the need for both structural and control optimization with a dynamic objective such as output variance or frequency response. In these problems the objective function changes with time due to dynamic loading on the structure. Langley et al. consider the minimization of kinetic energy and maximum strain energy by variation of truss bay length and bay loss factor [68]. Moshrefi-Torbati, Keane et al. reduce the frequency-averaged response of a truss by varying the truss topology [91]. The use of optical performance metrics as cost functions is addressed by both Bronowicki [22] and Milman et al [89].

In this thesis, the term Performance Tailoring (PT) is applied to structural optimizations in which the goal is to tailor the structure to meet a dynamic performance requirement. This chapter provides a formal definition of performance tailoring and gives an illustrative example using a simple structural model that is representative of a structurally-connected interferometer. The model details are presented and the PT problem is formulated specifically for the problem of minimizing the RMS of an

optical metric. Disturbance analysis methods for evaluating performance RMS and sensitivity analyses that provide performance gradients are discussed. Both gradient based and stochastic search techniques are applied to the sample problem to obtain a performance tailored design.

2.1 PT Formulation

The objective of performance tailoring is to design a system using a set of design variables, $\vec{x}$, such that a desired performance metric, $f(\vec{x})$, is below some required value, f_{req}. One approach to this problem is to formulate an optimization that minimizes the performance metric:

$$\begin{aligned} & \min_{\vec{x}} f(\vec{x}) \\ s.t. \quad & \vec{g}(\vec{x}) \leq \vec{0} \end{aligned} \tag{2.1}$$

where $f(\vec{x})$ is the performance, or cost, function, $\vec{x}$ is a vector of design variables that affect the performance, and $\vec{g}(\vec{x})$ are constraints on the design variables. Examples of performance metrics include first mode natural frequency and the variance of the displacement of a particular structural element. For example, consider the design of a truss that must achieve a specified level of stability at a given point, such as the tip, when subjected to a dynamic disturbance environment. A PT optimization would minimize the variance of the tip motion within a given frequency range by varying the cross-sectional areas of the truss bars. Due to practical considerations, the cross-sectional areas may be constrained by a lower bound set by manufacturing and local buckling considerations and an upper bound on the mass of the structure.

2.2 SCI Development Model

The PT formulation is applied to the problem of high-performance systems through a simple sample structure that is representative of a structurally-connected interfer-

ometer (SCI). In the following section the development model is presented in detail. This model is used throughout the thesis to develop and demonstrate the design methodology and provide a comparison among problem formulations and optimization techniques.

The model is two-dimensional and consists of three optical elements, two collectors and a combiner, modeled by lumped masses rigidly connected to a truss structure as shown in Figure 2-1. The truss model is broken into four segments, each with its

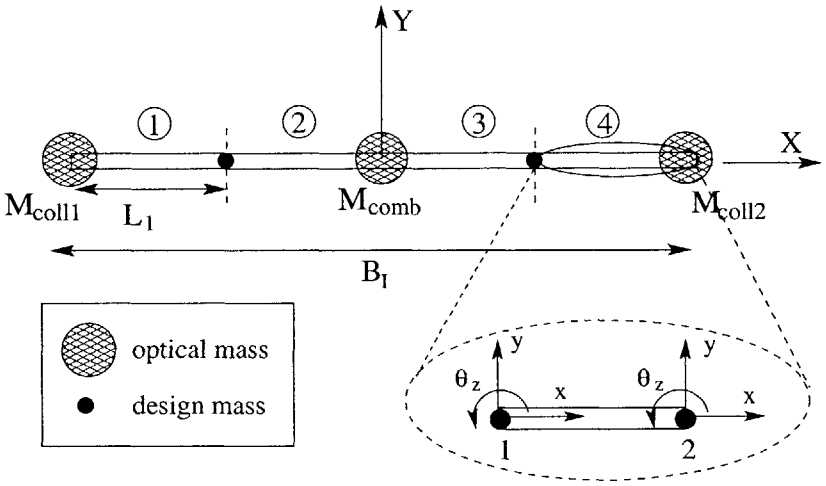


Figure 2-1: Schematic of SCI development model.

own material and geometric properties, to represent an articulated truss. Each truss segment is modeled with one Bernoulli-Euler beam finite element. The beam elements have two nodes each with three degrees of freedom: x-translation, y-translation, and z-rotation (See Fig. 2-1), for a total of four elements, five nodes and fifteen degrees

of freedom in the model. The stiffness and mass matrices for one beam element are:

$$K_e = \begin{bmatrix} \frac{EA}{L} & 0 & 0 & -\frac{EA}{L} & 0 & 0 \\ 0 & \frac{12EI}{L^3} & \frac{6EI}{L^2} & 0 & -\frac{12EI}{L^3} & \frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^2} & \frac{4EI}{L} & 0 & -\frac{6EI}{L^2} & \frac{2EI}{L} \\ -\frac{EA}{L} & 0 & 0 & \frac{EA}{L} & 0 & 0 \\ 0 & -\frac{12EI}{L^3} & -\frac{6EI}{L^2} & 0 & \frac{12EI}{L^3} & -\frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^2} & \frac{2EI}{L} & 0 & -\frac{6EI}{L^2} & \frac{4EI}{L} \end{bmatrix} \tag{2.2}$$

$$M_e = \frac{\rho AL}{420} \begin{bmatrix} 175 & 0 & 0 & 35 & 0 & 0 \\ 0 & 156 & 22L & 0 & 54 & -13L \\ 0 & 22L & 4L^2 & 0 & -13L & -3L^2 \\ 35 & 0 & 0 & 175 & 0 & 0 \\ 0 & 54 & 13L & 0 & 156 & -22L \\ 0 & 13L & 3L^2 & 0 & -22L & 4L^2 \end{bmatrix} \tag{2.3}$$

where E and ρ are the Young's Modulus and material density, A and I are the cross-sectional area and inertia of the beam, and L is the element length. The mass matrix is of the same form as that used by the finite element program NASTRAN. The element matrices are assembled into global mass and stiffness matrices, K and M. The optical elements are simply lumped masses without inertial or optical properties. They are rigidly fixed to the beam elements and are incorporated by adding the appropriate mass to the x and y degrees of freedom of the corresponding node in the global mass matrix. The mass breakdown of the model is given in Table 2.1.

Table 2.1: Mass breakdown of SCI development model.

Component	Mass [kg]
Truss	791.7
Combiner	200
Collector $(-x)$	200
Collector $(+x)$	200
Total	1391.7

The equations of motion of the undamped system are written as:

$$M\ddot{\hat{x}} + K\hat{x} = B_{\hat{x}w}F \tag{2.4}$$

where $B_{\hat{x}w}$ is a mapping matrix between the disturbance forces, F and the physical degrees of freedom, $\hat{x}$. The disturbances enter at the combiner node, where the spacecraft bus and reaction wheels are located and include force and torque in all three directions, F_x, F_y, T_z. Therefore, $B_{\hat{x}w}$ is a sparse matrix with fifteen rows and three columns. The axial force, F_x, and the torque, T_z, disturbances are modeled as unit-intensity white noise, while the y-force, F_y, is white noise with an intensity of $0.01\ \frac{N^2}{Hz}$. The y-force intensity is a fraction of the torque intensity to ensure that the system response from the two disturbances are of similar scale.

Equation 2.4 is transformed to modal coordinates through the eigenvalue problem:

$$\left(-\Omega^2 M + K\right)\Phi = 0 \tag{2.5}$$

where Ω is a diagonal matrix of natural frequencies and Φ contains the associated mode shapes. The natural frequencies and mode shape descriptions for select modes are listed in Table 2.2. The system dynamics are written in a modal state-space

Table 2.2: Natural frequencies and mode shapes of nominal SCI model, unconstrained and with model of Attitude Control System (ACS) included.

	unconstrained		with ACS Model	
Description	Mode #	Freq (Hz)	Mode #	Freq (Hz)
Rigid X-translation	1	0	N/A	N/A
Rigid Y-translation	2	0	N/A	N/A
Rigid Z-rotation	3	0	N/A	N/A
ACS mode, θ_z rotation	N/A	N/A	1	0.082
1st bending mode, symmetric	4	0.197	2	0.197
2nd bending mode, asymmetric	5	0.708	3	0.813
3rd bending mode, symmetric	6	1.294	4	1.294
4th bending mode, asymmetric	7	2.848	5	3.030
1st axial mode, asymmetric	12	47.81	10	47.81
2nd axial mode, symmetric	13	83.83	21	83.83

representation as follows:

$$\begin{Bmatrix} \dot{q} \\ \ddot{q} \end{Bmatrix} = \overbrace{\begin{bmatrix} 0 & I \\ -\Omega^2 & -2Z\Omega \end{bmatrix}}^{A} \begin{Bmatrix} q \\ \dot{q} \end{Bmatrix} + \overbrace{\begin{bmatrix} 0 \\ \Phi^T B_{\hat{x}w} \end{bmatrix}}^{B} w \tag{2.6}$$

$$z = \underbrace{\begin{bmatrix} C_{z\hat{x}}\Phi & 0 \end{bmatrix}}_{C} \begin{Bmatrix} q \\ \dot{q} \end{Bmatrix}$$

where q are modal coordinates, w is white noise, z is the output, and $C_{z\hat{x}}$ is a mapping matrix from physical states to the output. Damping is introduced into the system through the matrix Z, a diagonal matrix of modal damping ratios. In the development model, modal damping is set to 0.001 for all modes.

In interferometer design, the optical performance determines the success of the instrument. There are many optical metrics that are of interest such as the angle of the wave-front at the combiner, beam shear and optical path-length difference. Since the SCI development model is a low-fidelity model without true optics, the output, z, is a linearized geometric approximation of the optical path difference (OPD) between the two arms of the interferometer and is based only on the translations of the mirror nodes. If the star is located at a distance R from the interferometer in the Y-axis direction (Figure 2-1), then the linearized optical path lengths from the star to the combiner through the two interferometer arms are:

$$OP_1 = R - y_1 + x_c - x_1 + \frac{B_I}{2} \tag{2.7}$$

$$OP_2 = R - y_2 - x_c - x_2 + x_3 + \frac{B_I}{2} \tag{2.8}$$

where B_I is the interferometric baseline, x_1 and y_1 are the negative-x collector translations, x_2 and y_2 are the positive-x collector translations, and x_c is the combiner x-translation. The linearized OPD is the difference of Equations 2.7 and 2.8:

$$z = -x_1 - y_1 + 2x_c - x_2 + y_2 \tag{2.9}$$

The frequency response functions from disturbance inputs to the performance are computed from the state-space matrices as follows:

$$G_{zw}(s) = C(sI - A)^{-1} B \tag{2.10}$$

where $s = j\omega$ and, G_{zw} is a matrix of transfer functions. The resulting transfer functions from the SCI model are plotted in Figure 2-2. The transfer function from F_x and T_z to OPD are drawn in solid and dash-dotted lines, respectively. The transfer function from F_y to OPD does not appear on the plot because the force in this direction does not couple to the output due to the symmetry in the model. Forcing the structure at the center node in the y-direction only excites the symmetric bending modes. It is seen from Equation 2.9 that symmetric y motion of the two collector nodes results in zero OPD. The dominant transfer function is that from torque at the

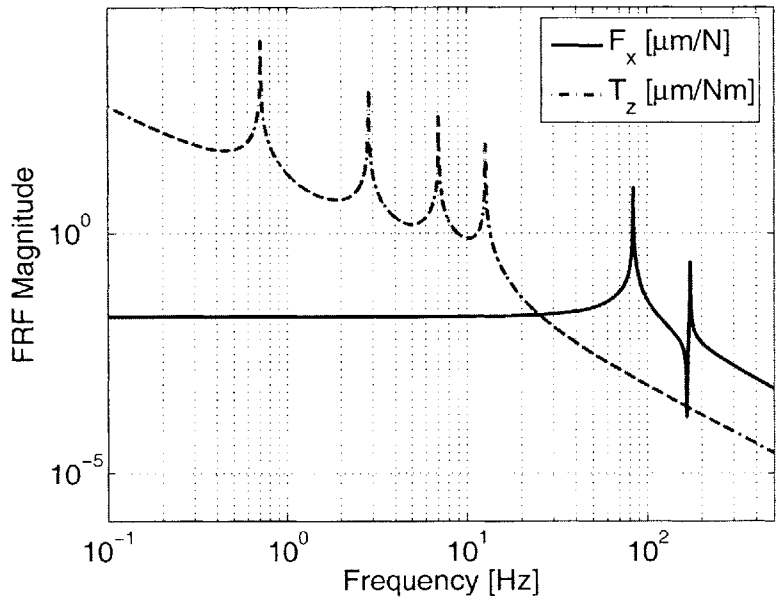


Figure 2-2: Frequency response functions (FRF) of SCI: disturbance input to performance.

center to OPD. The first flexible mode is the first asymmetric bending mode at 0.708 Hz, followed by the higher order asymmetric bending modes. In contrast, the F_x transfer function has no low-frequency modes because the beam elements are much stiffer axially than in bending. The first observable axial mode is at 80.2 Hz.

The transfer function from torque to OPD is nonzero at zero frequency because the model is unconstrained and there is a rigid body rotation about the center of the array. This motion results in equal, but opposite displacement at the ends of the structure, where the collecting optics are located. It is clear from Equation 2.9 that such a motion results in infinite OPD contribution. The transfer function from F_x to OPD does not show any contribution from axial rigid body motion. The translational rigid body modes are unobservable in the output since OPD is only affected by relative changes in the positions of the collectors and combiners.

2.3 PT Implementation

In the next section the PT optimization formulation is applied to the development model. First, the performance metric is defined, then design variables for tailoring the structure are selected.

2.3.1 Performance Metric

The output of interest, z, is the OPD between the two arms of the interferometer which changes as a function of time with the disturbance. The output variance, σ_z^2, is a common measure of system performance that reduces the output time history to a scalar quantity. If the output signal is zero mean, then the variance is also the mean square and its square root is called the root mean square [112] (RMS). Output RMS is often used in disturbance, or jitter, analysis to predict the broadband performance of a system in the presence of a dynamic disturbance environment. Performance variance is calculated in the frequency domain from the power spectral density (PSD), of the output signal, S_{zz}:

$$S_{zz} = G_{zw} S_{ww} G_{zw}^H \tag{2.11}$$

where G_{zw} is the system transfer function from disturbance to output, S_{ww} is the PSD of the input signal and $()^H$ is the matrix Hermitian. Given that $z(t)$ is zero-mean,

the output covariance matrix, Σ_z, is found by integrating S_{zz} over frequency:

$$Sigma_z = \int_{-\infty}^{\infty} S_{zz}(\omega)\, d\omega \tag{2.12}$$

In general, Σ_z, is a matrix, and the diagonal elements are the performance variances, $\sigma_{z_i}^2$, corresponding to the system outputs, z_i.

Alternatively, if the disturbance input is white noise then the output variance is obtained more directly using the Lyapunov equation to obtain the modal state covariance matrix, Σ_q:

$$A\Sigma_q + \Sigma_q A^T + BB^T = 0 \tag{2.13}$$

The performance variance for the i^{th} output is obtained by pre and post-multiplying the state covariance matrix by the appropriate rows of the state-space output matrix C:

$$\sigma_{z_i}^2 = C_i \Sigma_q C_i^T \tag{2.14}$$

The disturbance input for the SCI model is white noise, therefore Equation 2.14 is used in the implementation to compute performance variance. However, a Lyapunov analysis can only be conducted on a stable system. Recall from Figure 2-2 and Table 2.2 that this model contains three rigid body modes with zero frequency. These modes must be removed or stabilized in order to perform the necessary jitter analysis.

The translational rigid body modes are not observable in the output and, therefore, can be removed from the model without consequences. The rotational rigid body mode, on the other hand, does result in positive OPD. In practice, such motion is controlled with an attitude control system (ACS). The ACS is modeled with a rotational spring with stiffness, $k_{ACS} = 10,000\ \frac{Nm}{rad}$ at the rotational node of the combiner. The value of k_{ACS} is chosen to produce an ACS mode that has a frequency well below the flexible modes. The effect of the ACS model on the transfer functions from T_z to OPD is shown in Figure 2-3. Notice that the transfer function with ACS does not have a negative slope at zero frequency. The addition of the ACS model does affect the first few asymmetric bending modes, but only slightly. The new frequency

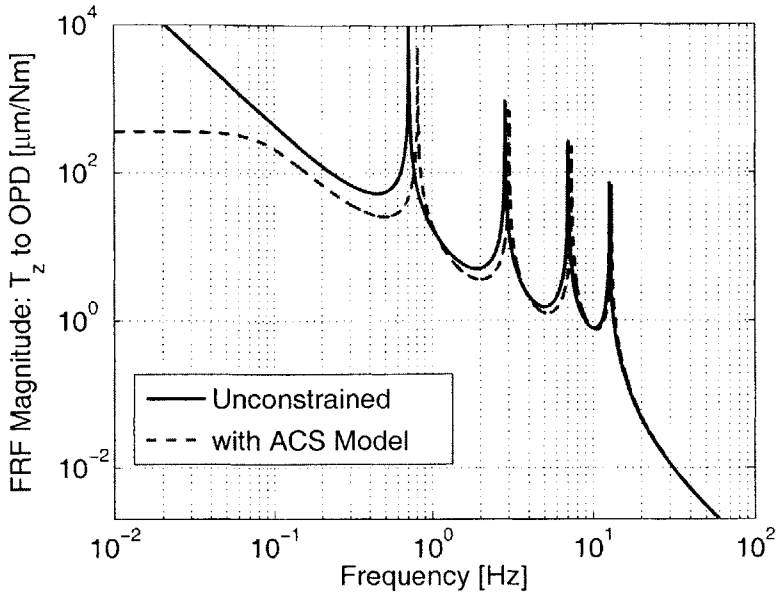


Figure 2-3: Frequency response function of SCI: disturbance input T_z to performance, '-.': without ACS model, '–': with ACS.

values are listed in Table 2.2. This method of modeling the ACS is simplistic, but will suffice for a model of this fidelity. The F_x and F_y transfer functions are not shown in Figure 2-3 since they are not affected by the rigid body modes.

2.3.2 Performance Gradients

A large number of optimization techniques, such as steepest descent, conjugate gradient and Newtons method (see Appendix A), require the gradient of the performance with respect to the design variable, x. In the following discussion the gradients of the output variance are derived. For a more thorough presentation of these equations the reader is referred to the thesis by H. Gutierrez [50].

To begin, notice that the performance metric is defined by Equation 2.14 with the constraint that Σ_q satisfies Equation 2.13. Therefore, the Lagrangian of the variance is written by augmenting the expression for the variance with a symmetric Lagrange multiplier matrix, Λ_i:

$$\left(\sigma_{z_i}^2\right)_L = C_i \Sigma_q C_i^T + \Lambda_i \left(A\Sigma_q + \Sigma_q A^T + BB^T\right) \tag{2.15}$$

where the subscript indicates the i^{th} performance metric. In order for the derivative of Equation 2.15 to equal the derivative of the variance, σ_z^2, its derivatives with respect to both Σ_q and Λ must be zero. As a result, the Lagrange multiplier is computed by solving the following equation:

$$A^T\Lambda_i + \Lambda_i A + C^T C = 0 \tag{2.16}$$

Equation 2.16 is similar in form to that of Equation 2.13 and is in fact simply another Lyapunov equation in A^T and C instead of A and B.

Then, taking the derivative of Equation 2.15 with respect to a design variable, x and using well-known matrix identities gives:

$$\frac{\partial \sigma_z}{\partial x} = tr\left\{\Sigma_q \frac{\partial\left(C^T C\right)}{\partial x}\right\} + tr\left\{\Lambda_i\left(\frac{\partial A}{\partial x}\Sigma_q + \Sigma_q\frac{\partial A^T}{\partial x} + \frac{\partial\left(BB^T\right)}{\partial x}\right)\right\} \tag{2.17}$$

Equation 2.17 is referred to as the Governing Sensitivity Equation (GSE) and provides an analytical method of obtaining cost function gradients for design optimizations that use output variance as a performance metric. However, in order to use this equation it is necessary to calculate the gradients of the state-space matrices. Recall that these matrices are based on the modal representation of the structure and therefore, in general, are not explicit functions of the design variables. The design variables are related to these matrices through the modal quantities, ω and Φ as follows:

$$\frac{\partial A}{\partial x} = \sum_{j=1}^{m}\frac{\partial A}{\partial \omega_j}\frac{\partial \omega_j}{\partial x} \tag{2.18}$$

$$\frac{\partial B}{\partial x} = \sum_{j=1}^{m}\frac{\partial B}{\partial \phi_j}\frac{\partial \phi_j}{\partial x} \tag{2.19}$$

$$\frac{\partial C}{\partial x} = \sum_{j=1}^{m}\frac{\partial C}{\partial \phi_j}\frac{\partial \phi_j}{\partial x} \tag{2.20}$$

where the summations are performed over the modes included in the model.

Since the natural frequencies and mode shapes of the model are obtained through

the eigenvalue equation, the derivatives of the frequencies are obtained by differentiating Equation 2.5:

$$\frac{\partial \omega_j}{\partial x} = \frac{1}{2\omega_j} \phi_j^T \left(-\omega_j^2 \frac{\partial M}{\partial x} + \frac{\partial K}{\partial x} \right) \phi_j \tag{2.21}$$

Obtaining the derivatives of the eigenvectors, or mode shapes, is slightly more involved. Using Nelson's method [98], it is assumed that the j^{th} eigenvector derivative is written as a linear combination of the j^{th} eigenvector and a linearly independent vector, ψ_j:

$$\frac{\partial \phi_j}{\partial x} = \psi_j + a_j \phi_j \tag{2.22}$$

The scalar a_j is obtained through derivation of the mass normalization equation:

$$a_j = -\phi_j^T \left(M\psi_j + \frac{1}{2} \frac{\partial M}{\partial x} \phi_j \right) \tag{2.23}$$

The vector ψ_j is found by solving the following matrix equation:

$$b_j = \mathcal{K}_j \psi_j \tag{2.24}$$

where $\mathcal{K}_j$ and b_j are defined as:

$$\mathcal{K}_j = -\omega_j^2 M + K \tag{2.25}$$

$$b_j = \frac{\partial \omega_j^2}{\partial x} M\phi_j - \left(-\omega_j^2 \frac{\partial M}{\partial x} + \frac{\partial K}{\partial x} \right) \phi_j \tag{2.26}$$

The matrix $\mathcal{K}_j$ is singular with rank of $n-1$ if the eigenvalues are distinct. This issue is addressed by arbitrarily removing one element from b_j as well as the corresponding row and columns from $\mathcal{K}_j$. The corresponding element in ψ_j is set to zero. Equations 2.21 and 2.22 provide expressions for the derivatives of the modal quantities with respect to the design variables as a function of the mass and stiffness matrix derivatives, $\frac{\partial M}{\partial x}$ and $\frac{\partial K}{\partial x}$. These final quantities will differ depending on how the design variables enter the stiffness and mass matrices. These gradients are derived in the following section for the specific design variables chosen for the SCI model PT optimization.

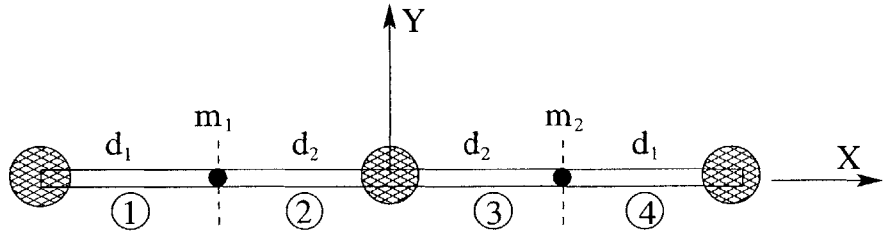
2.3.3 Design Variables

The choice of design parameters for performance tailoring varies from application to application. In general, it is desirable to choose variables that have a significant effect on the performance metric and can be controlled to high accuracy during manufacturing. For example, geometric properties, such as the cross-sectional diameter of the truss elements affect the performance metric and can be specified to a relatively high precision in practice. In contrast, the material properties of the truss elements also affect the performance metric, but are not good choices for tailoring parameters because it is difficult and costly to manufacture a material with a very specific Young's Modulus.

The design variables used in the SCI development model are the cross-sectional diameters of the truss segments and lumped masses placed at pre-determined locations along the truss. The truss diameters are constrained to be symmetric about the center of the array so that there are only four tailoring parameters as listed in Table 2.3 and shown in the accompanying figure.

Table 2.3: Tailoring parameters for SCI sample problem.

Name	Description	x_0	$\frac{\partial \bar{\sigma}_z}{\partial x}$
d_1	cross-sectional diameter of truss 1 & 4	0.10 [m]	-11.81
d_2	cross-sectional diameter of truss 2 & 3	0.10 [m]	-31.15
m_1	lumped mass on -x truss	0 [kg]	9.31
m_2	lumped mass on +x truss	0 [kg]	9.31



The third column of the table gives the nominal value for the parameters, x_0, and the fourth column lists the normalized sensitivity, $\frac{d\bar{\sigma}_z}{dx}$ of the performance metric with respect to the tailoring parameter. The normalized sensitivity is defined as the

percent change in the performance due to a percent change in the parameter:

$$\frac{d\bar{\sigma}_z}{dp} = \frac{x_0}{\sigma_{z_0}} \frac{\partial \sigma_z}{\partial p} \tag{2.27}$$

where σ_{z_0} is the RMS performance of the nominal design and $\frac{\partial \sigma_z}{\partial x}$ is the derivative of the RMS performance with respect to the parameter, x as defined in Equation 2.17. In the case of the design masses, the nominal values are zero, so the average of the structural mass in the consistent mass matrix in the x and y degrees of freedom at the appropriate node is used in the sensitivity calculation. Note from the table that increasing the cross-sectional area of the truss segments in the nominal design decreases the variance of the OPD thereby improving the performance of the system.

2.3.4 Finite Element Gradients

In order to compute the performance gradients it is necessary to obtain the gradients of the stiffness and mass matrices with respect to the design variables. The stiffness and mass matrices depend on the cross-sectional diameter of the truss segments through the area and inertia properties of the beam elements. Therefore, the chain rule is employed to find the derivatives of the global stiffness and mass matrices with respect to d_i:

$$\frac{\partial K}{\partial d_i} = \frac{\partial K}{\partial I_i} \frac{\partial I_i}{\partial d_i} + \frac{\partial K}{\partial A_i} \frac{\partial A_i}{\partial d_i} \tag{2.28}$$

$$\frac{\partial M}{\partial d_i} = \frac{\partial M}{\partial A_i} \frac{\partial A_i}{\partial d_i} \tag{2.29}$$

It is assumed that the bar elements have a circular cross-section, so that the inertia and area of the elements in the i^{th} truss segment are defined as:

$$I_i = \frac{\pi}{64} d_i^4 \tag{2.30}$$

$$A_i = \frac{\pi}{4} d_i^2 \tag{2.31}$$

and then, by inspection, the inertia and area derivatives required for Equations 2.28 and 2.29 are:

$$\frac{\partial I_i}{\partial d_i} = \frac{\pi}{16} d_i^3 \tag{2.32}$$

$$\frac{\partial A_i}{\partial d_i} = \frac{\pi}{2} d_i \tag{2.33}$$

The stiffness and mass matrix derivatives with respect to d_i follow directly from the matrices for one beam element (Equations 2.2 and 2.3). The individual element derivatives are assembled into the global mass and stiffness matrices as is appropriate for the truss segment under consideration.

The lumped masses enter the model as concentrated mass at the x and y translation degrees of freedom on the appropriate grids in the mass matrix. The derivatives of the stiffness and mass matrices with respect to these masses are therefore:

$$\frac{\partial K}{\partial m_i} = \mathbf{0}, \qquad \frac{\partial M}{\partial m_i} = \begin{bmatrix} \mathbf{0} & & \cdots & & \mathbf{0} \\ \vdots & \ddots & & & \vdots \\ & & 1\ 0\ 0 & & \\ \vdots & & 0\ 1\ 0 & & \vdots \\ & & 0\ 0\ 0 & & \\ \vdots & & & \ddots & \vdots \\ \mathbf{0} & & \cdots & & \mathbf{0} \end{bmatrix} \tag{2.34}$$

2.3.5 Constraints

The cross-sectional diameters are constrained to be greater than a lower bound of 30 mm. This constraint prevents the optimization algorithm from choosing a design in which the collectors are perfectly isolated from the disturbance by removing the truss altogether. A lower bound of zero is also enforced on the lumped masses as mass may only be added to the truss in this manner. An additional constraint is placed on the total mass of the tailored design, $\bar{M}$, to represent the mass constraints imposed by launch vehicle specifications:

$$0.03 - d_i \leq 0 \ \forall i = 1,2 \tag{2.35}$$

$$-m_i \leq 0 \ \forall i = 1,2 \tag{2.36}$$

$$\frac{\pi}{2} L\rho \sum_{i=1}^{2} \left(d_i^2 + m_i\right) - 0.90\bar{M} \leq 0 \tag{2.37}$$

where $\bar{M}$ is the total system mass allocated to the design. Only 90% of the mass budget is allocated at design in order to reserve 10% as margin. Note that the mass constraint is nonlinear in the cross-sectional diameters.

The equations presented throughout this section describe the PT formulation for the SCI development model and are summarized in Equation 2.38. The objective is to minimize the RMS of the performance metric over all frequencies given a white noise disturbance input. Four design variables, a symmetric distribution of the cross-sectional areas of the four truss segments and two lumped masses, are considered. The variables are constrained by a mixed set of linear and nonlinear inequalities that ensure the optimized design meets practical criteria. In the following section the optimization algorithm used to produce performance tailored designs are discussed and results from the SCI development model are presented.

$$\begin{aligned} \vec{x}^T &= \begin{bmatrix} d_1 & d_2 & m_1 & m_2 \end{bmatrix} \\ \vec{x}^* &= \arg\min_{x \in X} \sigma_z(\vec{x}) \\ s.t. \quad & 0.03 - d_i \leq 0 \ \forall i = 1,2 \\ & -m_i \leq 0 \ \forall i = 1,2 \\ & \frac{\pi}{2} L\rho \sum_{i=1}^{2} \left(d_i^2 + m_i\right) - 0.90\bar{M} \leq 0 \end{aligned} \tag{2.38}$$

2.4 Optimization Algorithms

The field of optimization is quite large and there are many algorithms available to solve problems such as the PT formulation given in Equation 2.38. In general, the

algorithms begin at an initial guess for the design variables and then try some number of configurations until an optimal design is achieved or some termination conditions are met. The approaches differ in the methods used to move from one iteration to the next. Gradient-based methods use gradient information to guide the search for points that satisfy the necessary conditions for optimality. However, if the solution space is not convex there is no guarantee that a global optimum is found. Heuristic methods do not require gradients and use randomized algorithms to search for good, but not necessarily optimal, solutions. These methods are most useful when the solution space is non-convex and/or highly constrained as the search is able to jump out of local minima and across islands of feasibility. A more detailed discussion of the gradient-based algorithms and a simple example is given in Appendix A. Both a gradient-based method, sequential quadratic programming (SQP), and a heuristic method, simulated annealing (SA), are applied to the PT SCI problem.

2.4.1 Sequential Quadratic Programming

Sequential quadratic programming (SQP) is a quasi-Newton method developed to solve constrained optimization problems such as that of Equation 2.1. Constraints are handled by augmenting the objective function to produce the Lagrangian function:

$$\mathcal{L}(\vec{x}, \lambda) = f(\vec{x}) + \sum_{i=1}^{m} \lambda_i g_i(\vec{x}) \tag{2.39}$$

The general goal of SQP is to find the stationary point of this Lagrangian using Newton's method. Therefore, the algorithm is also referred to as the Lagrange-Newton method. For a detailed discussion of SQP the reader is referred to Fletcher [44].

SQP is chosen as the gradient-based optimization algorithm for the SCI development model because it can handle constrained problems with nonlinear objectives and constraints, and is already implemented in the MATLAB optimization toolbox [108]. Also, analytical gradients of the cost function are available, as described above, enabling a computationally efficient search.

In the MATLAB SQP implementation there are two levels of iterations. At each

major iteration an approximation of the Hessian of the Lagrangian function is made using the quasi-Newton updating method of Broyden, Fletcher, Goldfarb and Shanno (BFGS). The Hessian approximation is then used along with first-order Taylor series approximations of the nonlinear constraints to generate a quadratic programming (QP) subproblem:

$$\begin{aligned} &\min_{d\in\mathcal{R}^n} \frac{1}{2} d^T H_k d + \nabla f\left(x_k\right)^T d \\ &\nabla g_i\left(x_k\right)^T d + g_i\left(x_k\right) = 0 \qquad i = 1, \ldots m_e \\ &\nabla g_i\left(x_k\right)^T d + g_i\left(x_k\right) \leq 0 \qquad i = m_e + 1, \ldots m \end{aligned} \tag{2.40}$$

where H is the Hessian approximation, g_i are the constraint equations and m and m_e are the total number of constraints and the number of equality constraints, respectively. The solution, d_k, is obtained through an active set quadratic programming strategy and is used to form a new major iterate:

$$x_{k+1} = x_k + \alpha_k d_k \tag{2.41}$$

The step length, α_k is found through a line search that requires sufficient decrease in a particular merit function.

If a problem is well-behaved and properly scaled, then gradient-based algorithms, such as SQP, are likely to find a global optimum as long as the objective function and constraints are convex. However, if the problem is non-convex, i.e. there are multiple solutions in the space that are locally optimal, then the algorithm may converge to a local minima, instead of the global one. In fact, the solution that is obtained depends on the initial guess chosen by the user. Non-convexity poses a difficult problem since there is no known way to prove definitively that a global optimum has been found instead of simply a local one. Therefore, there is an entire body of heuristic methods that can be employed to search for a global optimum. One simple heuristic is to randomly chose some number of initial guesses and run SQP, or some other gradient-based algorithm, from each of these starting points. Each resulting solution is then

at least locally optimal, and the solution with the lowest cost can be chosen as the "global" optimum. Such a method is applied to the PT SCI problem and is referred to as Monte Carlo (MC) SQP.

2.4.2 Simulated Annealing

Simulated annealing (SA) is another example of a heuristic search method. Originally developed for application to the field of combinatorial optimization, it uses principles from statistical mechanics to conduct a random search of the solution space. In the original paper on the subject, Kirkpatrick et al. [65] draw an analogy between the cooling of a liquid material, such as metal, and a stochastic search optimization. If a metal is cooled very quickly then the material will solidify into a sub-optimal configuration. However, if the cooling is done slowly through careful annealing the material solidifies into a state of minimum energy. In SA, this minimum energy state is analogous to the minimum of the objective function.

The authors outline four key ingredients necessary to implement a simulated annealing algorithm [65]. The first is a concise description of the configuration of the system, such as a mathematical model that depends directly on the design variables. The second requirement is a method of randomly generating new rearrangements of the design variables. Recall that in the gradient-based methods a new iteration is chosen based on the search direction found through the cost function gradients. In simulated annealing, it is up to the user to set up a method of randomly generating design variable perturbations. The third ingredient is a quantitative objective function containing the trade-offs that have to be made. In the SCI problem this function is the combination of the cost and constraints given in Equation 2.38. Finally, an annealing schedule of the temperatures and length of times for which the system is to be evolved is necessary. In the optimization context the temperature has no physical meaning but defines as a set of rules that guides the random search.

An SA algorithm based on elements from the Kirkpatrick reference [65] and the thesis by Jilla [61] is implemented in MATLAB for use in this thesis. The algorithm is presented in Figure 2-4 and described here in detail. The process begins with an

Data: initial iterate, x_0, initial temperature, T_0, change bounds on x, Δx, algorithm parameters, k_a, k_r, k_c, n_{dof}
Result: optimal design variables and cost: x^*, J^*
begin
  Evaluate initial cost: $J_0 = f(x_0)$
  Initialize: $x = x_0$, $J = J_0$, $k = 1$, $T = T_0$, $A = R = 0$
  while *not frozen and number of function evaluations* $<$ *max* **do**
    Randomly choose design variables to change based on n_{dof}
    Generate random change with bounds Δx [Equations 2.42 and 2.43]
    if x_k *is feasible* **then**
      Evaluate new cost, $J_k = f(x_k)$
      Calculate change in cost, $\Delta J = J_k - J_{k-1}$
      if $\Delta J < 0$ **then**
        accept design, A^{++}, k^{++}, store: J_k, x_k
      else
        Calculate acceptance probability, p_A [Equation 2.44]
        if *rand(0,1)*$< p_A$ **then**
          accept design, A^{++}, k^{++}, store: J_k, x_k
        else
          reject design, R^{++}
        end
      end
      if $R > k_r$ *or* $A > k_A$ **then**
        Increment temperature [Equation 2.45]
        Reset counters, $A = R = 0$
      end
    end
  end
  Return lowest-cost design: $J^* = \min J$, $x^* = \arg\min J(x)$
end

Figure 2-4: Simulated annealing algorithm.

initial design variable state, $x = x_0$, an initial temperature, $T = T_0$ and a set of optimization parameters. The initial cost is calculated from x_0 and stored. In the SCI development model, this step is equivalent to determining the RMS OPD for the initial design. The counters, A and R, are used to track the number of designs accepted and rejected at each temperature and are initially set to zero.

To begin the search, new values for the design variables, x_k, in the neighborhood of the current design, x_{k-1} are generated:

$$x_k^i = x_{k-1}^i + dx_k^i \tag{2.42}$$

where the superscript indicates the i_{th} element of x. A study in [61] shows that to conduct an efficient search of the solution space it is desirable to change only two or three design variables at a time. The optimization parameter, n_{dof}, indicates how many degrees of freedom to change per iteration. If $n_{dof} = 2$, two design variables, chosen randomly from the vector $\vec{x}$, are changed and the others remain static. The magnitude of the change, dx_k^i, is chosen from a random uniform distribution bounded by Δx:

$$-\Delta x^i < dx_k^i < \Delta x^i \tag{2.43}$$

The new design is checked for feasibility by considering the constraint equations. If the design is not feasible it is discarded and a new dx is chosen. This process continues until a feasible new design is obtained.

The new design variables, x_k, are then used to calculate a new cost, J_k and the cost differential, $\Delta J = J_k - J_{k-1}$. If the change in cost is negative, i.e. the new design is better than the current, it is accepted. The new cost becomes the current cost, $J = J_i$ and the counter A is incremented. If ΔJ is positive it is accepted with probability:

$$p_A(\Delta J) = e^{\left\{\frac{-\Delta J}{k_T T}\right\}} \tag{2.44}$$

where T is the current temperature and k_T is a scaling constant. Rejected designs are not recorded, but the counter R is incremented. The constants, k_A and k_R are

defined by the user and indicate the number of designs that must be accepted or rejected, respectively, at a given temperature. When either $A = k_A$ or $R = k_R$ the temperature is reduced according to a user-defined cooling constant, k_C:

$$T_k = T_{k-1}\,(1 - k_C) \tag{2.45}$$

and the A and R counters are reset to zero. The search continues in this manner until the termination conditions are met. The termination conditions are chosen by the user based on the specific problem statement. In the PT SCI search the algorithm terminates when k_R designs are rejected at three temperatures in a row.

Equation 2.44 is called the Boltzman probability, and is the critical feature of the SA algorithm. At high temperatures the probability of accepting a design that increases the cost is high. By randomly accepting less favorable designs the algorithm allows itself to climb out of areas of local minimum and more thoroughly search the space. As the temperature decreases the acceptance probability, p_A, becomes lower, and the search eventually converges in the neighborhood of a favorable design. The minimum cost and corresponding design variables are identified from the history of accepted design costs. There is no guarantee that the resulting design is an optimum in the mathematical sense, in fact it most likely is not. However, if the cooling schedule is set appropriately for the given problem the resulting design is a good design and in many cases is better than a locally optimal design resulting from a gradient search.

2.5 Performance Tailoring Results

The results of the PT optimizations run on the SCI development model are presented in the following section. Three optimization algorithms, SQP, MC SQP and SA are compared for performance and efficiency. The SQP algorithm is part of the MATLAB optimization toolbox, and the SA algorithm is implemented as outlined in Figure 2-4.

At each optimization or search iteration a new SCI state-space model is built

and the RMS OPD is computed using Equations 2.13 and 2.14. The SCI model is built in MATLAB as indicated in the algorithm outlined in Figure 2-5. The input to the model generation code is a vector of tailoring design variables, x. These design variables are used to build the appropriate finite element model that includes the truss elements, optics, design masses and ACS model. The eigenvalue problem is solved, resulting in the natural frequencies and mode shapes for this particular design. The rigid body modes are removed and a state-space model, $SYS\,(x)$ is assembled.

Data: tailoring parameters, $\vec{x}$
Result: SCI system model, $SYS(\vec{x})$
begin
  Build finite element model including: truss, optics, design masses and ACS $\rightarrow K(\vec{x}), M(\vec{x})$
  Solve eigenvalue problem $\rightarrow \Omega(K,M), \Phi(K,M)$
  Remove rigid body modes
  Build state-space model $\rightarrow A(\Omega, Z), B(\Phi), C(\Phi)$
  Build data structure and return model
end

Figure 2-5: Development model generation algorithm.

2.5.1 Comparison of Optimization Algorithms

The model performance is evaluated in the nominal configuration to provide a baseline for comparison with the optimized designs. Building the model with the nominal tailoring parameters listed in Table 2.3 and running the disturbance analysis results in a nominal RMS OPD of 471 μm. The cost, optimal tailoring parameters, computation time and number of function evaluations for each optimized design are listed in Table 2.4.

Table 2.4: PT optimization results, $J_0 = 471$ μm.

		Tailoring Parameters					
Algorithm	J [μm]	d_1[m]	d_2[m]	m_1[kg]	m_2[kg]	time [min]	n_{fun}
SA	101.57	0.03	0.03	0.3614	2.520	8.74	1619
SQP	100.53	0.03	0.03	0.0	1.934	1.28	29
MC SQP	100.53	0.03	0.03	1.941	0.0	18.1	447

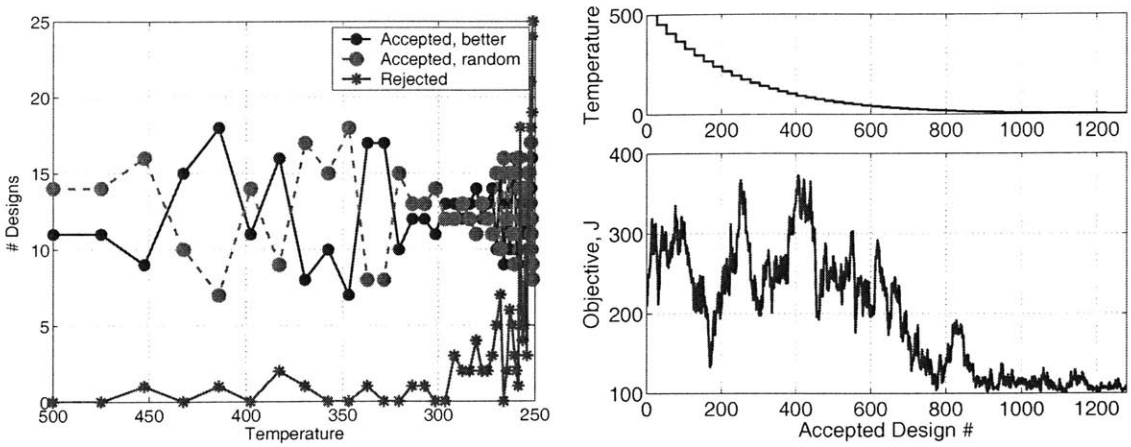
Simulated annealing results in a very good tailored design that improves the performance by 78.4% when compared to the nominal design. The algorithm freezes on a design in just under nine minutes and requires 1619 function evaluations. The cooling schedule and configuration change parameters used for this SA search are listed in Table 2.5. The algorithm performance is illustrated by the plots in Figure 2-6. The

Table 2.5: SA algorithm parameters.

Parameter	Description	Value
T_0	initial temperature	500
k_A	acceptance quota	25
k_R	rejection quota	25
k_C	cooling rate	0.1
k_T	scaling	1.0
n_{dof}	number of degrees of freedom	2
Δd_i	perturbation bounds [m]	± 0.005
Δm_i	perturbation bounds [kg]	± 1

first plot, Figure 2-6(a), shows the number of designs accepted due to a reduction in cost (solid line with circles), accepted randomly based on p_A (dashed line with circles), and rejected (solid line with stars), at each temperature. Notice that initially very few designs are rejected and the accepted designs are split between those that improve the cost and those that are accepted randomly. As the temperature decreases the number of rejected designs steadily increases until the maximum quota of 25 rejected designs is reached at three consecutive temperatures and the design is frozen. The temperature and objective function histories are shown in the upper and lower plots of Figure 2-6(b), respectively. The performance starts at around 300μm and fluctuates quite a bit at high temperatures. The performance even becomes worse than the starting point due to the high value of p_A at high temperatures. As the process cools the performance becomes steadily better until freezing in the neighborhood of the tailored design.

The SA-optimized design is used as the initial guess for the SQP design, and both the SQP and MC SQP algorithms find designs with a cost that is slightly lower than that of the SA design. With an RMS OPD of only 100.53 μm, the SQP-tailored



(a) (b)

Figure 2-6: Search Results for PT with Simulated Annealing: (a) Temperature and objective history for accepted designs (b) Accepted and rejected design statistics vs. temperature.

designs improve the performance by 78.6% over the nominal design, but are only 0.2% better than SA. When started at the SA design, the SQP algorithm converges very quickly and results in the same performance as the MC SQP algorithm. In fact, the combination of SA and SQP takes nearly half the time (9 minutes) as MC SQP (18.1 min) to find the optimal design. It is interesting to note that, although the SQP and MC SQP designs have the same performance, the mass distributions are nearly mirror images of each other. This result makes sense, since the design masses affect the performance symmetrically as evidenced by the sensitivities listed in Table 2.3, but does indicate that the PT solution space is non-convex.

To further explore the convexity of the design space, the resulting solutions and performances from the individual MC SQP optimizations are listed in Table 2.6. Note that over the ten trials the optimization converges at one of five different solutions. One solution has zero design mass and a performance of 102.88μm. The other four solutions have pretty much the same performance, 100.52μm, but show slight variation in the size of m_1. Also note that the tailored cross-sectional diameters are the same across all ten solutions; it is only the mass values that change. This result indicates that the solution space is convex with respect to the cross-sectional parameters, but is

non-convex with respect to the lumped masses. This theory is supported by the fact that both the single SQP run and the MC SQP run resulted in the same values for the d_i's; with different mass distributions. Referring back to the normalized parameter sensitivities in Table 2.3 it is apparent that changing the cross-sectional diameters has a greater impact on the cost than changing the lumped mass values. Therefore, the cost value is dominated by the d_i solution, which seems to be a global optimum and is found by all three algorithms.

Table 2.6: PT optimization: MC SQP optimizations.

	J	Tailoring, x^*			
#	[μm]	d_1[m]	d_2[m]	m_1[kg]	m_2[kg]
1	100.53	0.03	0.03	1.9410	0.0
2	102.88	0.03	0.03	0.0	0.0
3	102.88	0.03	0.03	0.0	0.0
4	100.53	0.03	0.03	1.9344	0.0
5	102.88	0.03	0.03	0.0	0.0
6	102.88	0.03	0.03	0.0	0.0
7	102.88	0.03	0.03	0.0	0.0
8	102.88	0.03	0.03	0.0	0.0
9	100.53	0.03	0.03	1.9311	0.0
10	100.53	0.03	0.03	1.9347	0.0

All three optimization algorithms result in a much improved PT-tailored design. The combination of SA and SQP converges on what appears to be a globally optimal design in around ten minutes, while running SQP optimizations at random initial guesses took almost twice as long (18 minutes) and 447 function evaluations. Since the solution space is non-convex due, in part, to the symmetry in the design masses, the SA algorithm provides a good starting point for the SQP optimization and improves the likelihood of locating the best design quickly.

2.5.2 Performance Tailored Design

It is clear from comparing the cost values, RMS OPD, that the PT design is a large improvement over the nominal design in terms of performance. However, since the cost metric is RMS, an average energy over all frequencies, comparing these numbers

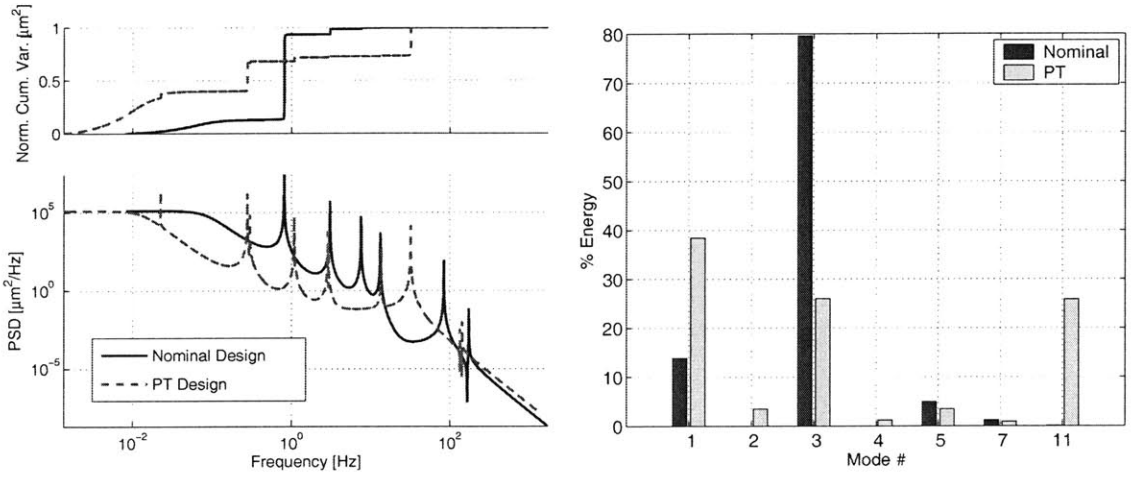
does not provide information on *why* the PT design performs so much better. In this section, the nominal and SQP PT designs are compared more closely.

First, consider the output PSDs of each design obtained from the transfer functions using Equation 2.11. The PSDs are presented in the lower plot of Figure 2-7(a), and show a significant difference in the behavior of the two systems. The backbones of the systems are quite different and all of the modes in the PT design are shifted lower in frequency. The upper plot in the figure is the normalized cumulative variance, $\bar{\sigma}^2_{z,c}$. This quantity is the running total of energy in the system and is obtained by integrating the PSD over discrete frequency bins [50]. The cumulative variance presented here is normalized by the total system energy so that the two systems can be compared on a single plot:

$$\bar{\sigma}^2_{z,c}(f_0) = \frac{2}{\sigma^2_z} \int_{f_{min}}^{+f_0} [S_{zz}(f)]\, df \tag{2.46}$$

where $f_0 \in [f_{min} \ldots f_{max}]$ and $\bar{\sigma}^2_{z,c} \simeq 1.0$. The normalized cumulative variance plot is especially interesting because it indicates how the energy in the system accumulates as a function of frequency. It is possible to pick out the modes that are most critical to the performance by visually correlating the largest steps in the cumulative variance curve to the peaks in the output PSDs directly below it. The cumulative variance curve of the PT design shows that the energy is distributed somewhat evenly over a few critical modes instead of concentrated in a single mode, as in the nominal design.

The critical modes and the distribution of energy among them for the two designs are shown graphically in a bar chart in Figure 2-7(b) and listed in full in Table 2-7(c). A critical mode is defined as one that accounts for at least 1% of the total output energy. The nominal and PT design data is presented side-by-side in the table with the modal frequencies in the first column, followed by the percent contribution of the mode to the total energy, and then the output variance and RMS attributed to the mode. The bar chart shows that the third mode, or second bending mode, is responsible for most of the energy in the nominal design ($\approx 80\%$), while the PT design



(a) (b)

Mode #	Not tuned f_n [Hz]	energy %	σ_z^2 [μm^2]	σ_z [μm]	Tuned f_n [Hz]	energy %	σ_z^2 [μm^2]	σ_z [μm]
1	0.082	13.87	24898	58.78	0.013	38.46	3886	38.66
2	0.197	0.0	0	0.0	0.023	3.48	352	3.50
3	0.813	79.65	142940	337.42	0.281	26.01	2628	26.15
4	1.294	0.0	0	0.0	0.302	1.26	127	1.26
5	3.03	4.97	8927	21.07	1.08	3.47	351	3.49
7	7.41	1.27	2275	5.37	2.85	0.93	94	0.94
11	83.8	0.02	41	0.10	31.7	25.85	2612	25.98
	Total:	97.78	179052	422.74		99.46	10050	99.98

(c)

Figure 2-7: Modal energy breakdown for nominal and PT designs (a) output PSDs (b) percent energy comparison: nominal (dark), PT (light) (c) results table.

energy is distributed almost evenly among the first (ACS), third and eleventh (second axial) modes. The ACS mode and second axial mode contribute a small percentage to the output in the nominal design, but account for 38% and 26% of the energy, respectively, in the PT design. Comparing the natural frequencies, it is found that the PT design frequencies are at least a factor of two lower than those of the nominal design. The lower frequencies make sense considering the fact that the cross-sectional areas of the beam have decreased greatly from the nominal system (0.10 m) to the PT design (0.03 m). The cross-section diameters map directly to area and inertia (Equations 2.31 and 2.30) which in turn map to the stiffness matrix (Equation 2.2). Lowering the cross-sectional diameters lowers the global stiffness matrix of the system and, in turn, the natural frequencies.

The final step in understanding the differences between the two systems is to examine the mode shapes of the critical modes, plotted in Figure 2-8. The ACS mode changes between the two systems (Figure 2-8(a)) partly due to the change in inertia, which in turn effects the modal frequency. The shape of this mode (Figure 2-8(a)) is similar in the nominal and PT design but its frequency is much lower in the tailored case (PT). There is a large difference in percent energy contribution from this mode between the two systems (from 14% to 38.5%), but the absolute RMS values accumulated in the motion are of similar magnitude (59 and 39μm).

The major improvement in performance is due to the tailoring of the second bending mode (Figure 2-8(b)). Notice that in the nominal design the two outer nodal points, or points of zero deflection, are located slightly away from the endpoints of the array towards the center. However, in the PT design, these nodal points are right at the ends of the array, where the collectors are located. Since the collector motions feature prominently in the OPD equation (Equation 2.9), tailoring the mode shape such that the nodal points are located at the collectors significantly reduces the OPD output in this mode, and, as a result, the total RMS OPD.

The second and fourth modes (Figures 2-8(c) and 2-8(d)) are account for zero energy in the nominal case, but a finite (although small) amount of energy in the PT design. In the nominal design the system is perfectly symmetric so these modes are

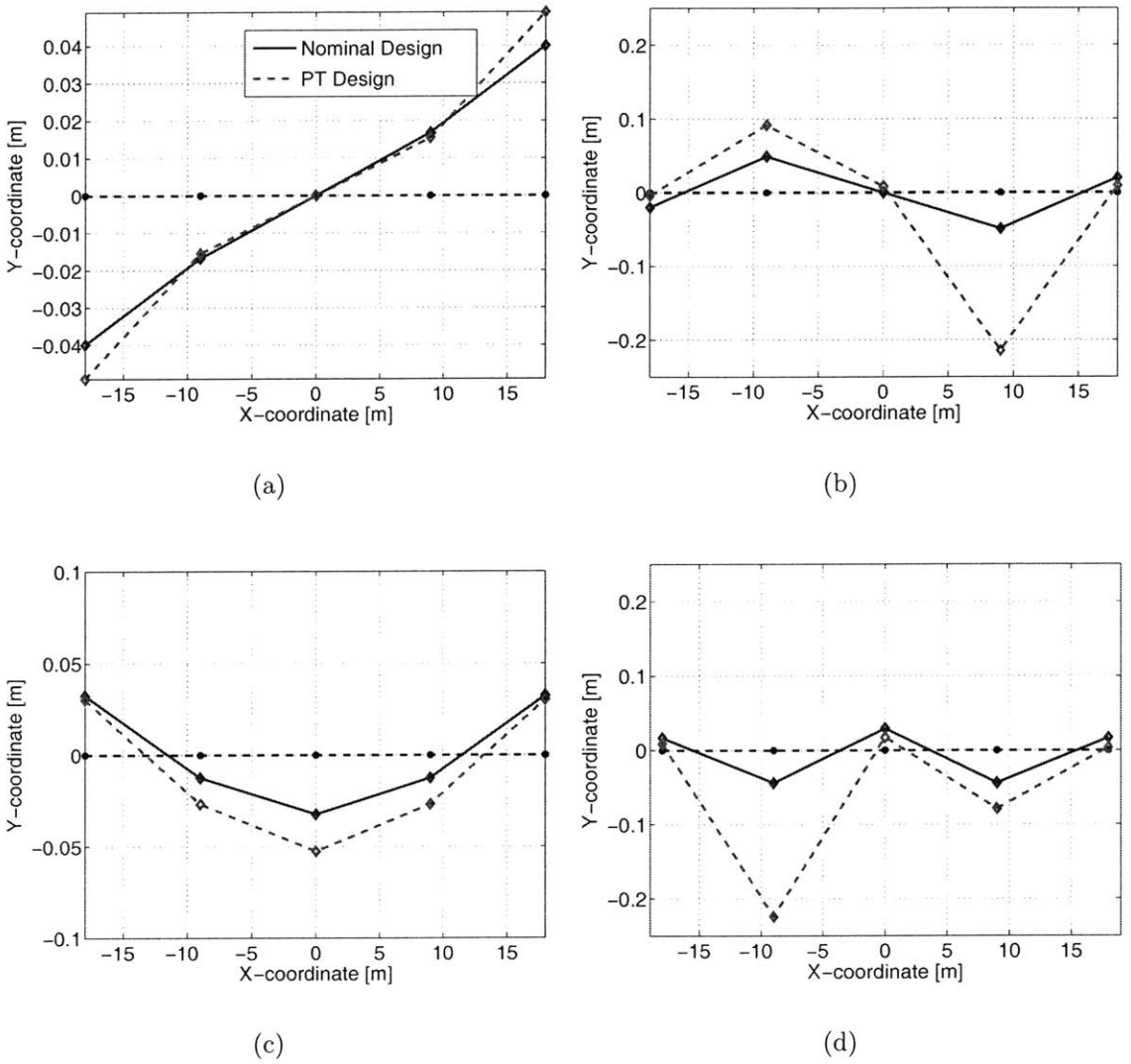


Figure 2-8: Comparison of critical mode shapes for nominal (–) and PT (- -) designs: (a) Mode #1: ACS (b) Mode #3: second bending (c) Mode #2: first bending (d) Mode #4: third bending.

also symmetric as seen in the figure. It is obvious from inspection of the OPD equation (Equation 2.9) that symmetric y-translation of the collectors does not result in OPD. It is only relative motion that is important. The PT design, however, is slightly asymmetric due to the small mass added to the negative-x arm of the interferometer. The mode shapes of the PT design show definite asymmetry resulting in a small relative displacement between the end points and a slight energy accumulation in these modes. The asymmetry is slight however, and does not have a large affect on the total OPD.

The final mode of interest is the second observable axial mode, Mode #11, listed in the table. The mode shape is not pictured here because the motion is only axial and difficult to discern on a linear plot. In both designs the positive x-motion of the collectors together with the negative x-displacement of the combiner node increase the OPD (Equation 2.9). The main difference between the two systems is that in the nominal design this mode is much higher in frequency and therefore contributes very little to the overall OPD, while the axial stiffness in the PT case is decreased significantly so that this mode plays a major role in the accumulation of output energy. The increase is only relevant to the distribution of energy; the total PT energy is still much lower than that of the nominal design.

In summary, the optimized design achieves better performance by choosing very small truss elements that result in lower natural frequencies overall and move the nodal points of the first two asymmetric bending modes to the ends of the array, where the collector optics are located. The large mass of the collectors on a very flexible truss effectively pin the collectors in place, so that the truss isolates the optics from the disturbances entering at the center.

2.6 Summary

In this chapter Performance Tailoring (PT) is introduced and formalized. A simple model of structurally-connected interferometer is presented in detail and used to step through the process of applying the PT formalization to a structural model. The

problem of minimizing a dynamic cost, RMS performance, by varying geometric properties of the structure, cross-sectional diameters and mass distribution, is addressed. The design variables are introduced and practical constraints as well as analytical calculation of cost gradients are discussed.

Two well-known optimization algorithms, sequential quadratic programming (SQP) and simulated annealing (SA), are applied to the problem. The results are compared for performance and efficiency and it is found that this particular problem can be solved with a combination of heuristic and gradient methods with reasonable computational effort. The PT design obtained through a heuristic SQP approach shows dramatic improvement in performance over the nominal design. The nominal and PT designs are examined in detail to understand why the tailored design produces such a large improvement in performance RMS.

Chapter 3

Robust Performance Tailoring

"There is nothing so wrong with the analysis as believing the answer!... Uncertainties appear everywhere in the model.. When using a mathematical model, careful attention must be given to the uncertainties in the model." - Richard Feynman [43]

Performance tailoring optimization results in a design that is tailored to achieve a high level of performance given the design variables and constraints. However, in the early design stage, the performance assessment is based only on the model predictions and not on real data. Therefore, there is a question regarding how well the model can predict the actual behavior of the system. In the following chapter this issue of prediction accuracy is explored in detail. First, model uncertainty is defined and methods for quantifying it and assessing the effects of uncertainty on the performance predictions are discussed. Then, parametric uncertainty is identified in the SCI development model and its effect on the performance predictions of the PT design is explored. This study motivates the need for robust design techniques and three existing Robust Performance Tailoring (RPT) cost functions are presented. Results of RPT optimization on the SCI development model are presented and analyzed. The chapter concludes with a discussion of the limitations of robust design on high-performance systems.

3.1 Uncertainty

Historically, structural models are used along with test data to help engineers better understand the physical behavior of the hardware. Model predictions are compared to test data, and the data is used to provide deeper understanding of the underlying physics and validate the model for use in trade analyses. However, as space systems become more complex and more difficult to test on the ground, test data is harder to obtain and it is necessary to rely solely on models and simulations to provide predictions of future system behavior without the benefit of validation data. This task is a much more demanding one and requires a high level of confidence in the models. As a result, much attention is focused in the area of prediction accuracy and model uncertainty.

The sources of inaccuracy in model predictions are grouped into three main categories: parametric errors, discretization errors and model structure errors [27, 93, 12]. Parametric errors refer to inaccuracies in the values of model parameters. For example, a finite element model may consist of beam elements that have certain material properties, such as Young's Modulus. Although the value of Young's Modulus is published for a wide range of materials it is likely that components made from the same material have slightly different Young's Modulus values. A value of Young's modulus in a finite element model that is slightly different from the Young's modulus of the physical component is an example of a parametric error. Discretization errors exist because finite element models are composed of discrete elements while physical parts are continuous. These errors can be reduced by using a high-fidelity mesh, but not eliminated. Finally, model structure errors are global modeling omissions or mistakes. This category includes any physical system behavior that is not captured in the model. Examples of these types of errors are unmodelled nonlinearities and improper choice of element types.

Parametric errors are most often considered in stochastic analysis because they are the easiest to model and the hardest to reduce. Discretization error can be reduced by refining the finite element model mesh, and model structure errors are reduced

through careful and experienced modeling [57]. Parametric errors, on the other hand, are nearly impossible to eliminate entirely since the model is built and used to make predictions well before components are available for testing. Even when test data is available, measurements are affected by noise in the sensor and data acquisition making it nearly impossible to obtain accurate data for model updating. Therefore, there is always some level of uncertainty inherent in the parameter values.

3.1.1 Uncertainty Models

The modeling of parametric uncertainties is currently a popular field of research as engineers move away from purely deterministic models and analysis to stochastic analogues. There are three accepted ways of modeling parametric uncertainty: probabilistic models, fuzzy logic, and convex, i.e. bounded, models [41]. In probabilistic modeling, a random distribution, usually normal, is assigned to a parameter and the uncertainty model is defined by choosing the mean parameter value and a standard deviation. These models are propagated through the analysis to provide statistical information about the performance such as probability of mission success [19, 52]. One drawback to probabilistic modeling is that there are generally insufficient data to accurately determine the statistical properties. In effect, the uncertainty model is itself uncertain. Convex, or bounded, uncertainty models address this concern by taking a more conservative approach. The uncertain parameter is assumed to be distributed between some bounds, and the worst-case performance prediction is considered. The uncertainty model used in this thesis is a type of convex model known as "envelope bounds." For a complete treatment of convex uncertainty models see the monograph by Elishakoff and Ben-Haim [15].

3.1.2 Uncertainty Analysis

Once a model of the parametric uncertainty is chosen, the effects of the uncertainty on the model predictions are assessed through an uncertainty analysis. Quite a few techniques for propagating uncertainty through a structural model are found in the

literature, and four methods compatible with bounded parametric uncertainty models are described below. The goal of the analysis is to provide a model, or mapping, of the performance as a function of the uncertainty parameters. This model is referred to in this work as the uncertainty space and can be used to provide a metric of the performance prediction accuracy. Such a metric is used to assess the accuracy of the performance prediction or in combination with optimization to produce a design that is robust to uncertainty.

Exhaustive Search

When the uncertainty is modeled as a bounded set, the uncertainty space may be evaluated by discretizing the uncertainty parameters with a fine grid and performing a full-factorial search of the space. The problem with this method is that it requires a large number of function evaluations and is therefore computationally expensive. For example, if there are four uncertainty parameters and a grid of 100 points is chosen for each parameter, the full-factorial uncertainty space consists of $100^4 = 1x10^8$ possible combinations.

Taguchi Arrays

One way to reduce the full-factorial space is to carefully choose a few "noise levels" and take advantage of orthogonal arrays as in the Taguchi method. The Taguchi method is a quality engineering technique developed by Genichi Taguchi for application to electronic circuit design [107], but has since been applied to problems of robust structural design [100]. In the Taguchi framework the parameters in a process are classified as either "control" or "noise" factors. Control factors are parameters that can be specified freely by the designer and are analogous to the tailoring parameters in our formulation. Noise factors cannot be controlled by the designer, but still affect the quality, or performance of the system, like the uncertainty parameters. An objective function that captures a measure of design quality is then evaluated using matrix arrays of the control and noise factors at specified levels. If a full-factorial experiment is performed then the method is no different from the exhaustive search

described above. However, one can reduce the full factorial matrix by choosing orthogonal sets so that for any two sets of factor levels all combinations occur an equal number of times. One caveat of this technique is that for the results of an orthogonal experiment array to be accurate the factor effects must be independent of each other. This assumption is generally only true to first order. For more information on the Taguchi method and its application to robust design the reader is referred to the book by Phadke [102].

Monte Carlo Propagation

A third approach, that lies somewhere between the computational expense of an exhaustive search and the rigor of the Taguchi method, is Monte Carlo uncertainty propagation. Instead of searching through a grid of all possible combinations, a number of uncertainty values are chosen randomly from the parameter distributions. The idea is that the performance space can be approximated relatively well by sampling at a large number of random points. Monte Carlo propagation is a useful way to check the performance of other analysis methods and provides a baseline result that is obtained with less computational effort than an exhaustive search.

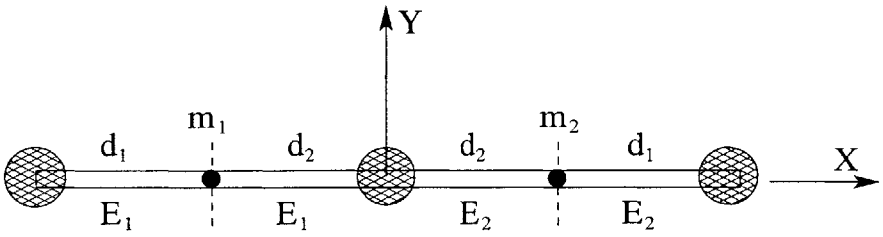
Vertex Method

While all three of these methods are valid ways to define and map out the uncertainty space, they all require significant computational effort and do not necessarily guarantee that the entire space is captured. Each method requires the discretization of a continuous uncertainty space. As a result, the performance map is discrete and there may be significant uncertainty values that are not explored. However, if the uncertainty space is convex about a given design point, then it is possible to severely reduce the computational effort involved by searching only the vertices of the space. This method is particularly useful when the uncertainty metric of interest is the worst-case performance.

In this thesis only parametric errors modeled with bounded uncertainty models are considered. Uncertainty analysis is performed with either a vertex search or Monte

Table 3.1: Uncertainty parameters for SCI development model.

Name	Description	p_0
E_1	Young's Modulus of truss 1 & 2	72 GPa
E_2	Young's Modulus of truss 3 & 4	72 GPa



Carlo propagation, depending on the application. The Monte Carlo propagation serves as a baseline comparison to verify the assumption of convexity. In the following section these techniques are demonstrated through an uncertainty analysis of the PT SCI design.

3.1.3 Example: SCI Development Model

In general, material properties represent a significant source of parametric uncertainty. Properties such as Young's modulus can be difficult to measure or predict accurately and often vary from sample to sample of the same material. Therefore, in the SCI sample problem the Young's modulus of the four truss segments, taken in pairs, are chosen as the uncertainty parameters, as listed in Table 3.1. The parameter, E_1 is the Young's modulus of the truss segments with negative-x coordinates, and E_2 corresponds to the truss segments with positive-x coordinates, as shown in the accompanying figure. The nominal values of 72 GPa are based on the material properties of aluminum [17]. The Young's Modulus of the truss segments are considered in pairs to reduce the computation required for the development model. In application, any number of uncertainty parameters can be considered, but the computation time required for uncertainty analyses and robust optimizations increases significantly with the number of parameters.

The addition of uncertainty parameters to the model results in a performance met-

ric, $\sigma_z(\vec{x}, \vec{p})$, that is dependent on *both* the tailoring, $\vec{x}$ and uncertainty parameters, $\vec{p}$. Like the tailoring parameters, the uncertainty parameters affect the performance through the finite element matrices. In this particular case, only the stiffness matrix is affected, since Young's modulus does not appear in the mass matrix at all (Equations 2.2 and 2.3). The uncertainty model is bounded and uniformly distributed about the nominal parameter value over a ranged defined by Δ_i:

$$\left(1 - \frac{\Delta_i}{100}\right) p_{i_0} \leq p_i \leq \left(1 + \frac{\Delta_i}{100}\right) p_{i_0} \tag{3.1}$$

where p_i is one of the uncertainty parameters in Table 3.1. Δ_i is a percent of the nominal parameter value and is referred to throughout as the uncertainty level.

The uncertainty model is propagated through the PT design with both a vertex search method and Monte Carlo analysis. In the vertex search, only the vertices, or corners, of the uncertainty space are considered. The number of performance evaluations necessary for this propagation method, n_{pv}, grows exponentially with the number of uncertainty parameters, n_p:

$$n_{pv} = 2^{n_p} \tag{3.2}$$

As discussed previously, the worst-case performance is at a vertex if the uncertainty space is convex. To check this assumption, a Monte Carlo analysis is run in which values for the uncertainty parameters are chosen randomly from their distributions. The performance is then evaluated at each uncertainty combination. If the convexity assumption holds, all of the performance values from the Monte Carlo should be at or below the worst-case value from the vertex search.

The results of the uncertainty analyses on the PT design are plotted in Figure 3-1. The uncertainty level for all parameters is 10%, and the Monte Carlo analysis is run with 500 random uncertainty values. The Monte Carlo results are shown in a histogram plotted against RMS performance. It is clear that even a small amount of uncertainty in the Young's Modulus values results in a large spread on the performance prediction. The dotted line to the right of the plot indicates the worst-case

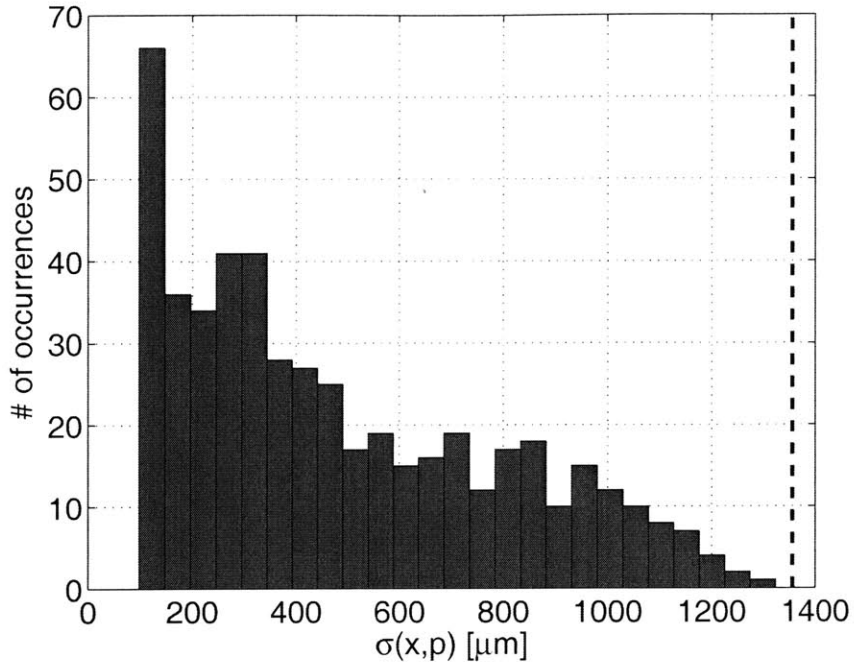


Figure 3-1: Uncertainty propagation results on PT optimized design: histogram - MC results, (- -) worst-case vertex search, $\Delta = 10\%$.

Table 3.2: Uncertainty propagation results: PT design.

	RMS	Young's Modulus [GPa]	
	[μm]	E_1	E_2
nominal	100.53	72.0	72.0
WC vertex	1355.5	64.8	79.2
WC Monte Carlo	1315.9	64.85	78.73

performance value from the vertex search. The fact that all of the Monte Carlo performance predictions are below this value is a good indication that the convexity assumption is valid in this case. The values of nominal and worst case RMS, along with the uncertainty parameter values, are listed in Table 3.2. Note that the worst-case performance occurs at an asymmetric configuration when E_1 and E_2 are at the lower and upper limits, respectively.

The frequency response functions (FRF) from disturbance forces and torques to the output are presented in Figures 3-2(a) through 3-2(c). The transfer functions for the PT configuration at nominal uncertainty values are depicted with a solid line, while those for the worst-case uncertainty configuration obtained through the vertex

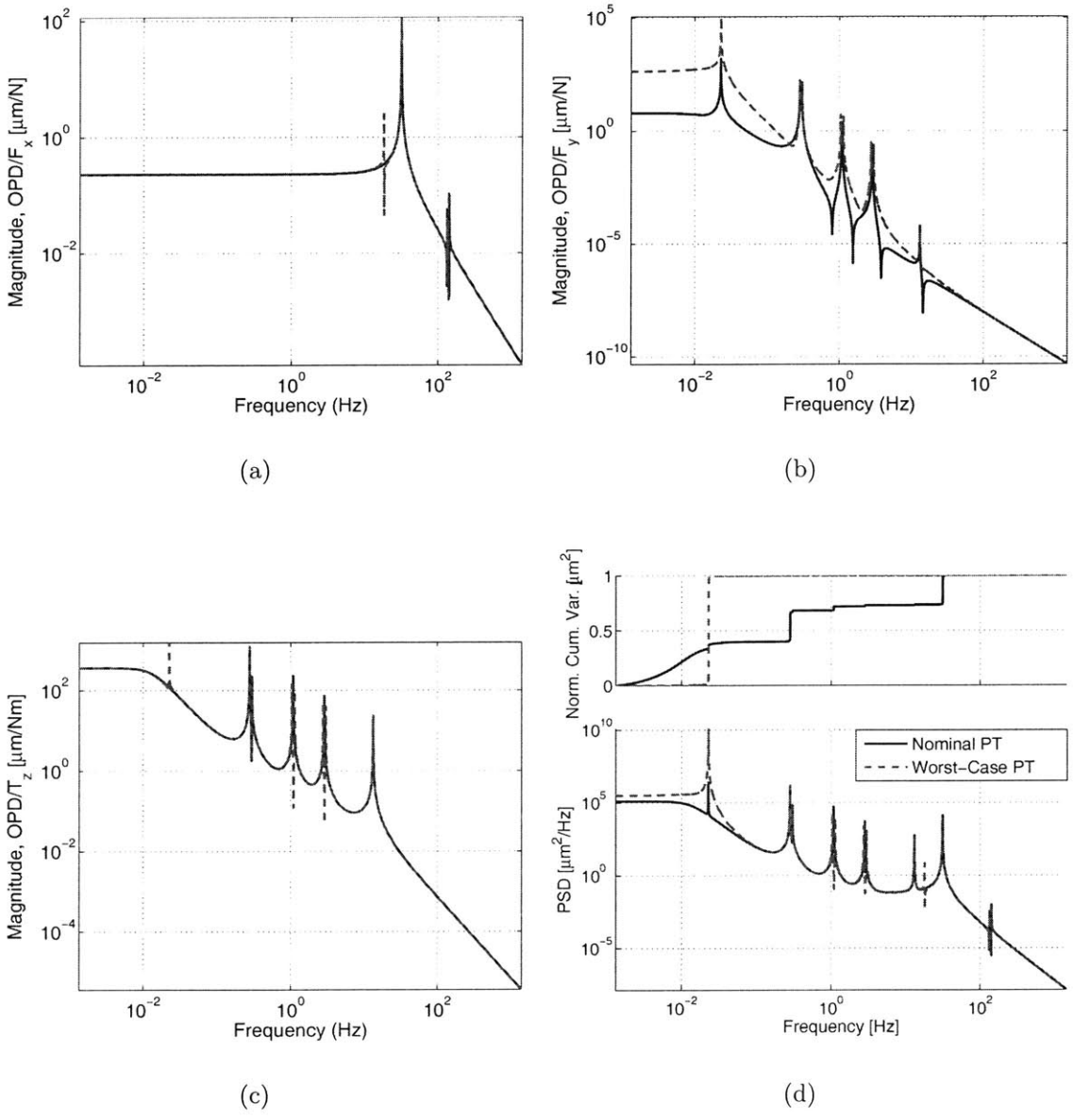


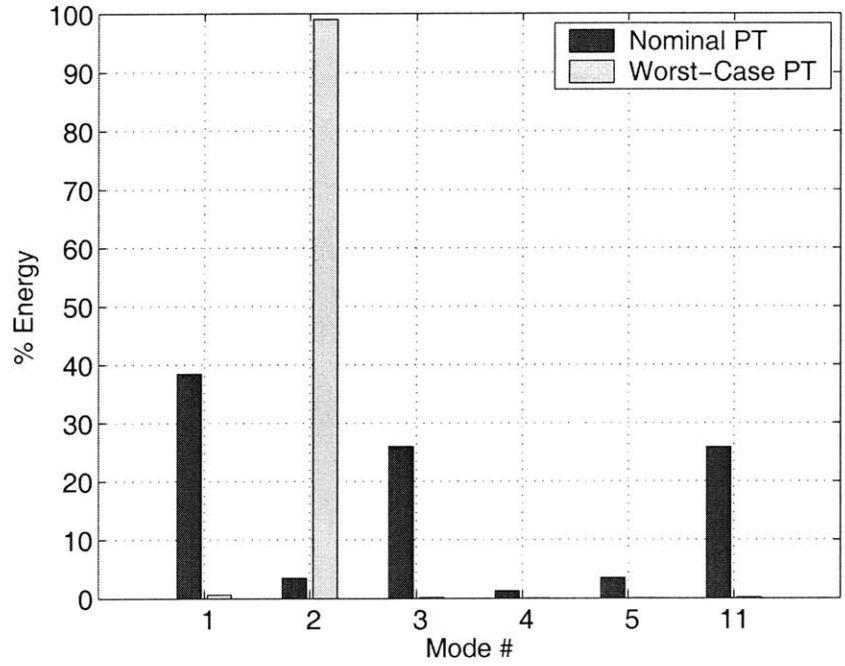
Figure 3-2: Comparison of nominal PT model (–) and PT model at worst-case uncertainty vertices(- -): (a) frequency response function, F_x to σ_z (b) frequency response function, F_y to σ_z (c) frequency response function, T_z to σ_z and (d) cumulative variance and output PSD.

method are shown with dashed lines. The transfer functions from axial force and torque to the output do not show much change between the two models. However, there is a significant increase in FRF magnitude between the nominal and worst-case designs for the y-direction force input. Recall, that in the PT configuration, the design is nearly symmetric and that the y-force disturbance has little effect on the output. When the system asymmetry is increased the symmetric modes become asymmetric and are observable in the OPD.

The normalized cumulative variance and output PSD plots are shown in Figure 3-2(d). It is clear from this plot that nearly all of the RMS in the worst-case PT model is attributed to one mode. This mode is the first bending mode at 0.023 Hz, and does not contribute significantly to the output in the nominal PT design, as evidenced by the modal energy distributions plotted and tabulated in Figure 3-3. The modal energy breakdown for the nominal uncertainty PT system is repeated here for comparison. In the nominal PT model the first bending mode contributes only 3.50μm to the total RMS performance. However, at the worst-case uncertainty vertex this mode accounts for 99.1% (1343μm) of the output energy.

The reason for the dramatic degradation in performance is found by comparing the shape of the first bending modes of the two systems (Figure 3-4). Note that the nominal PT mode shape (solid line) is nearly symmetric about the center so that there is no discernible relative y-motion between the collectors and no resulting OPD contribution. However, at the worst-case uncertainty vertex the system is highly asymmetric, and this bending mode is not symmetric about the center of the array (dotted line). In that case, the collector nodes do not move the same amount and OPD results. The addition of this mode to the output is significant since it is excited by both the force in the y-direction and torque about the z-axis.

The uncertainty propagation analysis of the PT optimized design shows that this design is only optimal if the model represents the physical system exactly. Small changes in the model parameters have drastic effects on the performance. This result is common in optimization problems as the design is often pushed to sensitive areas of the solution space in order to achieve maximum performance. In order to increase



(a)

	Nominal Uncertainty				Worst-case Uncertainty			
Mode #	f_n [Hz]	energy %	σ_z^2 [μm^2]	σ_z [μm]	f_n [Hz]	energy %	σ_z^2 [μm^2]	σ_z [μm]
1	0.013	38.46	3886	38.66	0.013	0.61	11144	8.22
2	0.023	3.48	352	3.50	0.023	99.09	1820600	1343.10
3	0.281	26.01	2628	26.15	0.281	0.14	2641	1.95
4	0.302	1.26	127	1.26	0.300	0.01	119	0.09
5	1.08	3.47	351	3.49	1.04	0.01	140	0.10
11	31.7	25.85	2612	25.98	31.7	0.14	2554	1.88
	Total:	98.53	9956	99.04		100.00	1837198	1355.34

(b)

Figure 3-3: % Total energy by mode ($\Delta = 0.10$): PT with nominal uncertainty (dark), PT with worst-case uncertainty (light).

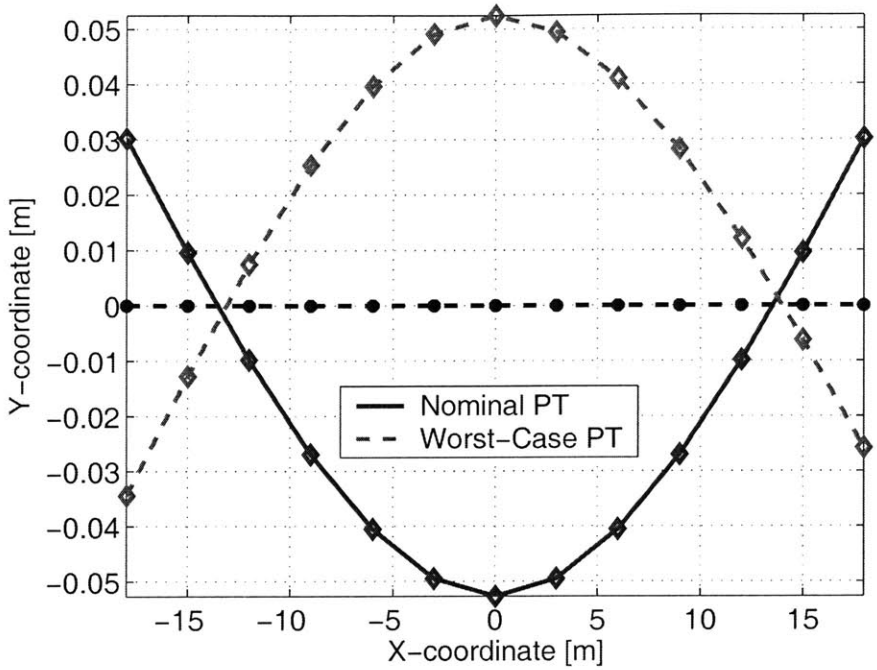


Figure 3-4: Mode shape of first bending mode for the nominal (–) and worst-case (- -) uncertainty PT designs.

the probability of success for a mission it is desirable to design a system such that it both meets performance requirements and is insensitive to model uncertainty. In the following section common robust design techniques aimed at decreasing the sensitivity of the design to the uncertainty parameters are discussed.

3.2 RPT Formulation

Robust performance tailoring accounts for the effects of uncertainty on the performance predictions by minimizing a robust objective function, J_{RPT}, in lieu of the nominal performance:

$$\begin{aligned} &\min_{\vec{x}} J_{RPT}(\vec{x}) \\ s.t. \quad &\vec{g}(\vec{x}) \leq 0 \end{aligned} \tag{3.3}$$

Unlike the PT optimization formulation, the RPT problem specifically accounts for the fact that there is uncertainty in the model parameters by adding a robustness

metric to the cost function. Note, however, that the uncertainty parameters do not appear in the constraint equations. The RPT optimizations considered in this thesis do not include constraints that are a function of the uncertainty parameters. For example, the uncertainty parameters chosen for the sample problem affect only the stiffness matrix and not the constraint on total system mass. In the most general form of Equation 3.3, uncertainty does affect the constraints and can be incorporated by requiring that the tailoring parameters satisfy the constraint equations for the uncertainty values that are worst-case in relation to the constraints. Examples of such problems are found in [42] and [103].

There are many possible formulations for J_{RPT} since there are numerous ways to account for uncertainty and penalize design sensitivity. In the following sections three known techniques are presented and discussed in detail.

3.2.1 Anti-optimization

Optimization with anti-optimization (AO) is a design approach for structural optimization with bounded uncertainty formulated by Elishakoff, Haftka and Fang in [42]. The authors discuss both of the formulations described below and demonstrate the design method on a standard ten-bar truss optimization problem using sequential linear programming. The truss is subjected to uncertain loads, and the cross-sectional areas of the truss members are optimized for minimum mass subject to stress and minimum gage constraints.

Anti-optimization can be defined as a min-max optimization problem that includes the process of identifying the critical uncertainty parameter values.

$$\begin{aligned} & \min_{\vec{x}} \overbrace{\max_{\vec{p} \in P} f(\vec{x}, \vec{p})}^{J_{AO}} \\ s.t \quad & \vec{g}(\vec{x}) \leq 0 \end{aligned} \tag{3.4}$$

At each iteration of tailoring parameters, $\vec{x}$, an uncertainty analysis is run to determine the values of uncertainty parameters, $\vec{p}$, that result in the worst-case per-

formance. This worst-case performance is then the objective function for the outer tailoring optimization. In effect, anti-optimization is analogous to performance tailoring for worst-case, instead of nominal, performance.

In order to use a gradient-based optimization algorithm to solve the anti-optimization problem efficiently analytical gradients of the objective are required. These gradients are difficult to obtain given the form of Equation 3.4. As the tailoring parameters change, the worst-case uncertainty vector may move from one vertex to another causing a discontinuity in the gradient. If the objective and constraints are linear, then the problem can be solved with a linear programming algorithm and the discontinuities do not cause a problem. However, if a quadratic approximation algorithm, such as SQP, is applied to a problem with nonlinear objectives and/or constraints, the discontinuity causes the optimization to misbehave and search inefficiently.

The problem can be formulated in a manner that is better suited for SQP by minimizing a dummy variable, z, and moving the performance at the uncertainty vertices to the constraints:

$$\begin{aligned} & \min_{\vec{x},z} z \\ s.t. \quad & \vec{g}(\vec{x}) \leq 0 \\ & h_i\left(z, \vec{x}, \vec{p}_i\right)) \leq 0 \quad \forall\, i = 1 \ldots n_{pv} \end{aligned} \tag{3.5}$$

where the augmented constraints, $h_i\left(z, \vec{x}, \vec{p}_i\right)$, are defined as follows:

$$h_i\left(z, \vec{x}, \vec{p}_i\right) = -z + f\left(\vec{x}, \vec{p}_i\right) \tag{3.6}$$

By inspection, the gradients of the objective with respect to the tailoring, $\vec{x}$ and dummy, z, variables are zero and one, respectively. The performance gradients are included through the augmented constraint gradient, instead of in the objective func-

tion:

$$\frac{\partial h_i\left(z,\vec{x},\vec{p}_i\right)}{\partial \vec{x}} = \frac{\partial f\left(\vec{x},\vec{p}_i\right)}{\partial \vec{x}} \tag{3.7}$$

$$\frac{\partial h_i\left(z,\vec{x},\vec{p}_i\right)}{\partial z} = -1 \tag{3.8}$$

In this alternate formulation (Equation 3.5) the optimization is a minimization with nonlinear constraints. Although the performance at each of the vertices is still required at each iteration, it is no longer necessary to determine the worst-case vertex. The problem is set up such that the optimal cost must be at one of the constraint boundaries, and as a result the variable z is the worst-case performance.

This robust design method is particularly well-suited for convex parametric uncertainty models. In their monograph [15], Ben-Haim and Elishakoff define convex models and discuss their application to problems in applied mechanics. The authors show that for most practical problems the uncertainty space is convex and therefore only the vertices of the space need be considered in robust design applications. This result is fortuitous as it allows a large reduction in the uncertainty set and guarantees robustness to all other uncertainty values within the bounds.

3.2.2 Multiple Model

Multiple model is a robust design technique borrowed from the field of robust control. It is applied to control system design in order to obtain a controller that is stable for a range of parameter values [10, 47]. In order to achieve this goal the weighted average of the $\mathcal{H}_2$ norms of a discrete set of plants is minimized. The resulting solution is guaranteed to stabilize each of the plants in the set.

The multiple model principle is readily applied to the robust performance tailoring problem since the output RMS value calculated with the Lyapunov expression is also an $\mathcal{H}_2$ norm. Instead of minimizing the nominal performance, as in the PT case, a weighted sum of the performances of a set of models within the uncertainty space is

minimized:

$$\min_{\vec{x}} \overbrace{\sum_{i=1}^{n} \beta_i f\left(\vec{x}, \vec{p_i}\right)}^{J_{MM}} \qquad (3.9)$$
$$s.t. \quad \vec{g}(\vec{x}) \leq 0$$

where β_i is a weighting factor on each of the model realizations in the set. The gradients of the cost function are simply the weighted sum of the performance gradients at each uncertainty realization:

$$\frac{\partial J_{MM}}{\partial x} = \sum_{i=1}^{n} \beta_i \frac{\partial f\left(\vec{x}, \vec{p_i}\right)}{\partial x} \qquad (3.10)$$

This method is similar to the anti-optimization technique in that at each tailoring iteration the model must be evaluated at a set of uncertainty values. However, it is less conservative since the average performance over the uncertainty space is minimized instead of the worst-case performance. Any of these uncertainty analysis methods are appropriate for defining the uncertainty space for use with this cost function. The vertex method is the most computationally efficient, as it requires a small number of function evaluations at each iteration.

3.2.3 Statistical Robustness Metric

A third approach is to augment the performance tailoring cost with a statistical robustness metric such as the standard deviation of the performance. An example of such a technique is found in a recent article by Sandgren and Cameron [103]. The authors propose a two-stage optimization for application to structural optimization problems in which sensitivity to parameter variations is penalized by including the standard deviation of the constraint in the formulation. Similar to Elishakoff et al. [42], a mass minimization problem with stress and displacement constraints is considered. The inner loop of the process requires a Monte Carlo simulation for each design point in which the design parameters are fixed and the uncertain parameter

values are chosen from their statistical distributions. The standard deviation(s) of the constraint(s) and/or objective, are then found from the results of the Monte Carlo analysis.

The formulation of this statistical approach for RPT is as follows:

$$\begin{aligned} & \min_{\vec{x}} \overbrace{\left\{\alpha f\left(\vec{x}, \vec{p_0}\right) + (1-\alpha)\,\sigma_f\left(\vec{x}, \vec{p}\right)\right\}}^{J_{SR}} \\ & s.t. \quad \vec{g}(\vec{x}) \leq 0 \end{aligned} \tag{3.11}$$

where $\vec{p_0}$ denotes the nominal values of the uncertain parameters, α is a relative weighting and σ_f is the standard deviation of the performance:

$$\sigma_f^2 = \frac{1}{N}\sum_{i=1}^{N}\left(f\left(\vec{x}, \vec{p_i}\right) - \mu_f\right)^2 \tag{3.12}$$

where N is the number of uncertainty samples chosen to populate the output distribution. The mean, μ_f, is simply the weighted average of the performance:

$$\mu_f = \frac{1}{N}\sum_{i=1}^{N} f\left(\vec{x}, \vec{p_i}\right) \tag{3.13}$$

As with the other two methods discussed thus far, an uncertainty analysis is necessary at each iteration of the tailoring parameters. If Monte Carlo analysis is used then the performance at all values in the uncertainty sample space is computed and Equations 3.13 and 3.12 are used to calculate the performance mean and standard deviation, respectively.

The gradient of the objective is obtained by differentiating J_{SR} directly and substituting the derivatives of Equations 3.12 and 3.13 appropriately:

$$\begin{aligned} \frac{\partial J_{SR}}{\partial x} &= \alpha\frac{\partial f\left(\vec{x}, \vec{p_0}\right)}{\partial x} \\ &+ (1-\alpha)\frac{1}{N\sigma_f}\sum_{i=1}^{N}\left(f\left(\vec{x}, \vec{p_i}\right) - \mu_f\right)\left(\frac{\partial f\left(\vec{x}, \vec{p_i}\right)}{\partial x} - \frac{1}{N}\sum_{j=i}^{N}\frac{\partial f\left(\vec{x}, \vec{p_j}\right)}{\partial x}\right) \end{aligned} \tag{3.14}$$

This formulation differs from that of Sandgren and Cameron in that the standard deviation of the objective is considered instead of that of the constraints. Also, the weighting factor α is included to allow a trade between nominal performance and robustness. When $\alpha = 1.0$ the cost function in Equation 3.11 reduces to the PT objective (Equation 2.1), and when $\alpha = 0$ nominal performance is completely sacrificed in favor of robustness, as in AO.

3.3 RPT Designs

The RPT optimizations are run with each of the cost functions described previously on the SCI development model with $\Delta = 0.10$ on both of the uncertainty parameters. The design variables and constraints are equivalent to those in Equation 2.38. Both the SQP and SA algorithms are used, and the resulting designs and algorithm performance for each cost function are presented in the following section.

3.3.1 Algorithm comparisons

Anti-optimization

RPT through anti-optimization is applied to the two-stage optimization problem of Equation 3.4 with SA and then with SQP. For comparison, the alternate form (Equation 3.5) is run with SQP starting from the SA design as well as a series of different initial guesses. The optimization performance data and resulting robust designs are listed in Table 3.3. The "iter" column lists the number of iterations required for convergence, and the column labelled "feval" gives the total number of function evaluations. In the SA case, the number of iterations is equivalent to the number of temperatures necessary to freeze the design, and in MC SQP it represents the total number of iterations and function evaluations required by all of the individual SQP runs.

The table shows that the SQP algorithms all find the same optimal design. Rows two and three of the table list the results of running SQP on the two different for-

Table 3.3: Algorithm performance: anti-optimization.

Alg.	Form	J^* [μm]	iter #	fevals #	time [min]	x^* [m] d_1	d_2	x^* [kg] m_1	m_2
SA	Eq 3.4	307.17	59	1800	11.52	0.0481	0.0580	0.1234	0.0164
SQP	Eq 3.4	306.86	25	86	1.48	0.0486	0.0581	0.0	0.0
SQP	Eq 3.5	306.86	17	35	2.00	0.0486	0.0581	0.0	0.0
MC SQP	Eq 3.5	306.86	698	1431	53.52	0.0486	0.0581	0.0	0.0

mulations using the SA design as an initial guess. In both cases the same design is found, but the simple minimization problem (Equation 3.5) is more efficient. The min-max problem (Equation 3.4) takes 2.0 minutes and requires 25 iterations to converge, while the simple optimization converges in only 1.5 minutes and 17 iterations. The MC SQP algorithm finds the same optimal design as the SA-SQP combination indicating that this design is likely to be a global optimum. The combination of SA and SQP proves to be a much quicker way to heuristically search the space than MC SQP, requiring a combined 13.52 minutes to find the optimal design in contrast to 53.32 minutes needed for the ten MC SQP runs.

The SA design is slightly sub-optimal, but is close to the SQP designs and provides a good starting point for the gradient search. The min-max formulation is used for SA since gradients are not required, and the algorithm may have trouble finding feasible iterates due to the additional constraints in the alternate formulation (Equation 3.5). The SA algorithm requires far more function evaluations then the SQP optimizers because the search is not guided by gradient information. Although the SA optimization does not perform quite as well as SQP, the worst-case performance of the resulting design, J^*, is significantly lower than that of the worst-case PT design indicating that a more robust design has indeed been found. Note that in the SQP-optimized design the design masses are all zero, while those in the SA design are small, but non-zero. Due to the random nature of the SA search, it is highly unlikely that the resulting design is an optimal solution, especially if that solution is along a constraint boundary, as is the case with the design masses.

Multiple Model

The multiple model RPT optimization is run on the SCI sample problem using both SA, SQP and MC SQP algorithms. The optimization performance metrics and resulting designs are listed in Table 3.4. The optimal costs listed in the table are much

Table 3.4: Algorithm performance: multiple model, $\beta_i = 1/n_{pv}$.

	J^*	iter	fevals	time	x^* [m]		x^* [kg]	
Alg.	[μm]	#	#	[min]	d_1	d_2	m_1	m_2
SA	273.51	64	2059	12.79	0.0440	0.0515	0.8464	19.5131
SQP	271.78	20	48	3.83	0.0432	0.0529	0.0	22.613
MC SQP	271.78	238	545	41.06	0.0432	0.0529	0.0	22.613

lower than those of the AO designs (Table 3.3), since the multiple model objective is the weighted sum of the performance values at each of the uncertainty vertices, and not the worst-case performance. In the results presented here the weighting β_i is the same at each vertex and equals $1/n_{pv}$, so that the cost is the average of the performance values at the uncertainty vertices.

The SQP and SA designs are nearly equivalent as the SA cost is only 0.6% higher than SQP. The individual runs in the MC SQP algorithm result in a few different optimal solutions with slight variations in the diameter of the inner array segments and the design mass values. All designs, however, are at the maximum mass constraint, and the best design is found with either MC SQP or the combination of SA and SQP. As seen previously with AO, using SA in conjunction with SQP finds the global optimum much more quickly (16.62 minutes) than performing ten randomly started MC SQP searches (41.06 minutes). The optimal cross-sectional diameters are similar to those in the anti-optimization designs, in that the diameters of the inner truss segments are larger than those of the outer segments. However, the MM design has 22 kg of design mass on the positive-x arm of the interferometer while the AO design has no lumped mass at all.

Statistical Robustness

The statistical robustness RPT optimization is run with the SA, SQP, and MC SQP algorithms and two measures of standard deviation. The algorithm performance metrics and resulting designs for are listed in Table 3.5. For the cases with vertex uncertainty propagation the standard deviation is calculated by solving Equation 3.12 using only the designs at the uncertainty vertices. This method is a conservative measure of the standard deviation since only the extremes of the distribution are considered. In the cases labelled "MC," a Monte Carlo uncertainty analysis with 500 samples is conducted to obtain an output distribution and then the standard deviation of that distribution is computed from Equation 3.12 at each iteration. This method is closer to a true standard deviation measure.

Table 3.5: Algorithm performance: statistical robustness, $\alpha = 0.5$.

	Unc.	J^*	iter	fevals	time	x^* [m]		x^* [kg]	
Alg.	Prop.	[μm]	#	#	[min]	d_1	d_2	m_1	m_2
SA	vertex	137.27	72	2223	15.54	0.0415	0.0543	0.0426	0.0477
SQP	vertex	137.22	14	30	1.55	0.0423	0.0540	0.0	0.0
MC SQP	vertex	137.22	238	895	25.19	0.0423	0.0540	0.0	0.0
SA	MC	114.75	68	2043	75.53	0.0344	0.0597	0.0683	0.0197
SQP	MC	114.63	16	38	75.83	0.0360	0.0485	0.0	0.0

Recall from Equation 3.11 that the statistical robustness objective function includes both the nominal performance and its standard deviation over the uncertainty space. In the results presented here the relative weighting on the nominal performance is $\alpha = 0.5$, so that it is as important as the robustness metric. The first three rows in the table compare the results of the SA, SQP and MC SQP algorithms with the vertex standard deviation measure. The results are similar to those seen in the anti-optimization and multiple model studies in that the performance of all three algorithms is similar with the SA-SQP combination obtaining the optimal design in less time than MC SQP.

The Monte Carlo standard deviation results for both SQP and SA have optimal costs that are significantly lower than the vertex method counter-parts. This differ-

ence in cost is due to the fact that the standard deviation metric is less conservative in this implementation. Note that both optimization algorithms take a very long time to converge when Monte Carlo standard deviations are calculated. This jump in computational effort is due to the fact that to get a good estimate of standard deviation it is necessary to run a large number of uncertainty combinations. In the case of the SQP algorithm, it is important that the Monte Carlo uncertainty distribution is chosen before running the optimization so that the uncertainty space are consistent from one iteration to the next. Otherwise, the gradients given in Equation 3.14 do not track the design changes.

The differences between the SQP and SA designs are similar to the other cases considered thus far in that SA finds a sub-optimal design, but provides a starting point for SQP that is very close to the optimal design. Although the Monte Carlo method provides a more accurate measure of the standard deviation, the results indicate that the increase in accuracy is not worth the additional computational effort required. The combination of SA and SQP optimization with the vertex method converges in 17 minutes compared to 151 minutes required for the Monte Carlo metric. Although the vertex method may not be an accurate measure of standard deviation for the uniform distribution, it does provide a conservative measure of robustness for bounded uncertainty models.

In order to assess the effect of the weighting parameter, α, the statistical robustness algorithm is run over a range of relative weightings from 0.0 to 1.0. The nominal and worst-case performance for the resulting designs are plotted in Figure 3-5 along with the standard deviation measure obtained from the vertex method. The nominal performance is depicted with circles and the standard deviation with stars. As α increases, the weight on the standard deviation decreases and that on nominal performance increases as evidenced by the figure. The nominal performance decreases as α increases while the standard deviation shows the opposite trend. At $\alpha = 0.0$, the nominal performance is not included in the cost at all and only the standard deviation is minimized. At the other extreme, when $\alpha = 1.0$, the standard deviation is eliminated from the cost function and the problem is equivalent to the PT optimization.

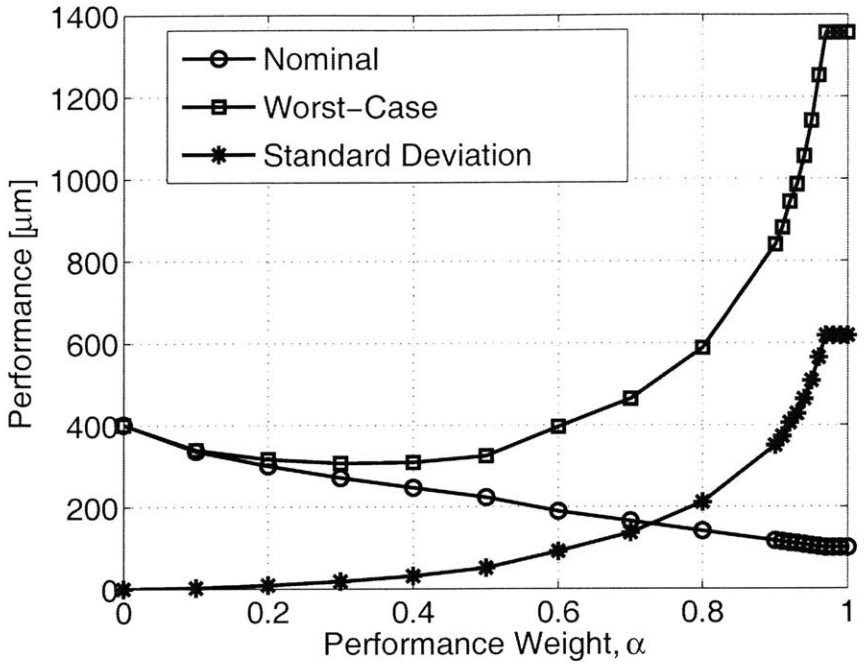


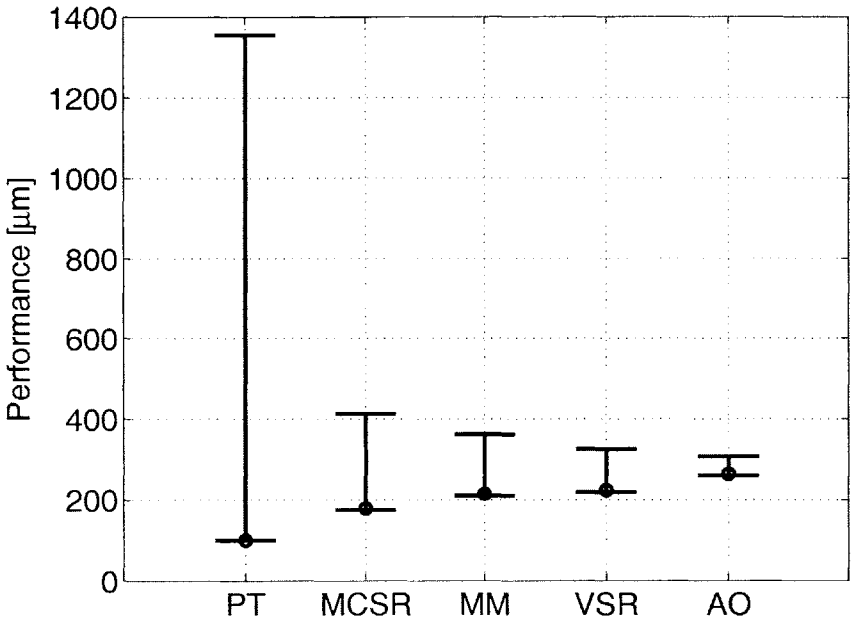
Figure 3-5: Nominal performance (o), worst-case (□) performance and standard deviation (*) for vertex statistical robustness (VSR) RPT designs vs nominal performance weighting, α.

The squares on the plot represent the worst-case performance, or the performance at the worst-case uncertainty vertex. As α increases, and the weight on robustness decreases, the worst-case performance increases nonlinearly. There is significant jump in the worst-case performance at $\alpha = 0.8$, and as the weighting approaches $\alpha = 1.0$ the curves plateau to the performance of the PT design.

3.3.2 Objective function comparisons

In the previous section the different implementations of optimization with the three RPT cost functions are compared for computational efficiency and performance. The combination of SA and SQP algorithms consistently achieves a lower cost value in less time than MC SQP. In this section the SQP RPT designs are compared against each other and the PT design for robustness.

The nominal and worst-case performance values for the PT and RPT designs are plotted in Figure 3-6(a), and the values are listed in the accompanying table



(a)

	σ_0	σ_{WC} [μm]		X-S diameters [m]		mass [kg]		
	[μm]	Vertex	MC	d_1	d_2	m_1	m_2	m_{truss}
PT	100.53	1355.5	1240.9	0.03	0.03	0.0	0.1934	72.15
MCSR	178.74	412.75	403.92	0.0360	0.0485	0.0	0.0	144.61
MM	214.75	362.06	352.63	0.0432	0.0529	0.0	22.61	184.89
VSR	223.45	325.09	312.40	0.0423	0.0540	0.0	0.0	185.97
AO	263.87	306.86	300.54	0.0486	0.0581	0.0	0.0	227.04

(b)

Figure 3-6: Nominal and worst case performance for PT and RPT optimizations.

(Figure 3-6(b)). The nominal performance for each design is depicted in the plot with circles and the worst-case performance, obtained through a vertex analysis, is denoted with an error-bar to indicate that the true performance could lie anywhere in this range. The designs are ordered, from highest to lowest, by worst-case performance. The error bar of the PT design is by far the largest and is many times higher than those of the RPT designs. The RPT designs all show a significant improvement in worst-case performance, but the AO design is the clear winner in terms of robustness as the error-bar is the shortest and the worst-case performance the lowest of all the designs. The worst-case value for performance of the AO design is only 306.86μ m, while vertex statistical robustness (VSR) and multiple model (MM) designs have worst-case value performances of 325.09μm and 362.06μm, respectively. The Monte Carlo statistical robustness (MCSR) design is the least robust of the RPT designs, with a worst-case performance of 412.6μm.

The plot andq table show that the nominal performance is inversely related to the robustness; as the designs become more robust, the nominal performance degrades. The nominal performance of the PT design is significantly better than that of the RPT designs, and the AO design has the worst nominal performance of the group. It is interesting to note, however, that the improvement in worst-case performance is much more significant that the degradation in nominal performance. The nominal performance of the PT design is roughly two times better than that of the AO design, but the worst-case performance of the AO design is nearly 4.5 times better that that of the PT design. Although there is a trade between nominal performance and robustness, large improvements in robustness can be achieved with moderate sacrifices in nominal performance.

The worst-case performance values shown in the plot are obtained through a vertex uncertainty analysis. These results are listed in Table 3-6(b) along with the worst-case values resulting from a Monte Carlo uncertainty analysis with 500 samples chosen from the distribution. Note that the worst-case performance results from the MC propagation are consistently below those from the vertex method confirming the assumption of a convex uncertainty space. In addition, the design parameters

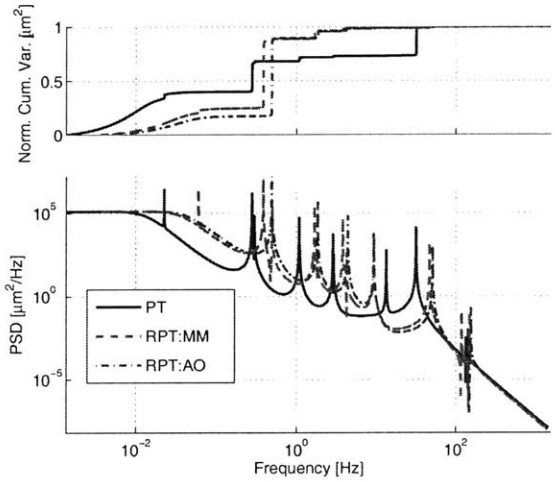
and truss mass for the designs are given in the table. Note that all of the RPT designs achieve robustness by increasing the cross-section diameter of the inner truss segments. It seems that the cross-sectional diameters have a greater effect on the system robustness due to the fact that in the most robust designs all of the mass is allocated to the truss members and none to the design masses. These issues are explored further in the following section by comparing the energy distributions of the PT and RPT designs.

3.3.3 Comparison to PT design

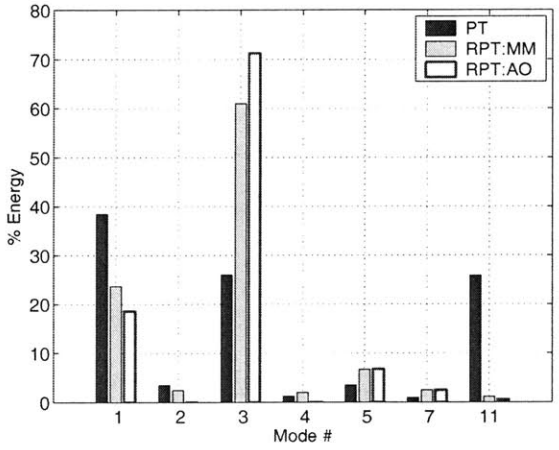
The results presented thus far show that the RPT design is dramatically less sensitive to uncertainty than the PT design. However, it is not yet clear how the increase in robustness is achieved physically. To answer this question the model energy distributions of the PT and RPT (AO and MM) designs are compared in the nominal and worst-case uncertainty configurations.

The output PSDs and normalized cumulative variance plots are shown in Figure 3-7(a) for the PT (solid line), multiple model RPT (dashed line) and anti-optimization RPT (dash-dotted line) designs with nominal uncertainty parameter values. The PSDs of the RPT designs are similar to each other, but are quite different from that of the PT design. The robust designs are stiffer with increased frequencies in all modes, and their energy is concentrated in one mode instead of being spread out over a few critical modes. A bar chart of the percent energy in the critical modes for the PT, RPT MM and RPT AO designs is shown in Figure 3-7(b), and the frequencies and energy breakdown of these modes for the two RPT designs are listed in Table 3-7(c). It is clear form the bar chart that the energy in the robust designs is concentrated largely in the second bending mode (mode #3). Referring back to Table 3-3(b) it is seen that the PT design only accumulates 26.15μm of RMS OPD in this mode compared to 131 and 188μm in the RPT MM and AO designs, respectively.

A plot of this mode shape for the three designs, Figure 3-8, readily illustrates the reason for this difference. In the PT design the nodal points of the mode coincide



(a)



(b)

	RPT: Multiple Model				RPT: Anti-optimization			
Mode #	f_n (Hz)	energy %	σ_z^2 (μm^2)	σ_z (μm)	f_n (Hz)	energy %	σ_z^2 (μm^2)	σ_z (μm)
1	0.036	23.67	10917	50.83	0.043	18.58	12936	49.02
2	0.061	2.41	1114	5.19	0.073	0.0	0	0.0
3	0.388	61.04	28152	131.09	0.494	71.23	49595	187.95
4	0.502	2.02	930	4.33	0.594	0.0	0	0.0
5	1.677	6.70	3091	14.39	1.842	6.74	4695	17.79
7	3.787	2.49	1148	5.34	4.375	2.47	1722	6.52
11	46.69	1.11	512	2.39	51.36	0.53	367	1.39
	Total:	99.44	45864	213.56		99.55	69315	262.67

(c)

Figure 3-7: Modal energy breakdown for PT and RPT designs with nominal uncertainty: (a) output PSDs (b) % energy comparison: PT (dark), RPT MM (light) RPT AO (white) (c) results table.

almost exactly with the collector locations. Therefore, when this mode is excited there is very little OPD accumulated. However, in the robust designs the nodal points are moved inwards towards the center of the array and away from the collectors. As a result, the collectors move in opposite directions when this mode is excited leading to larger OPD.

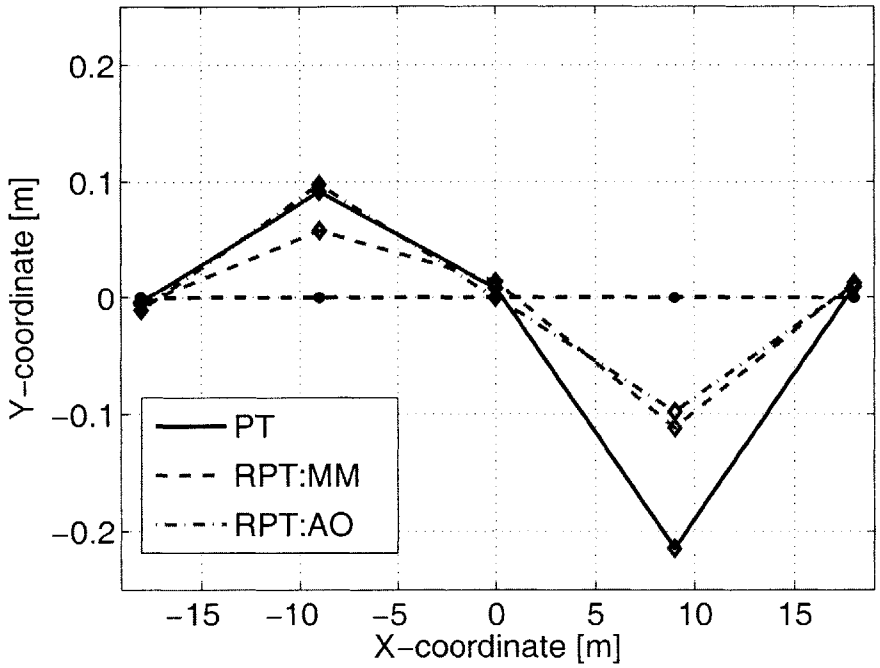


Figure 3-8: Critical modes shape comparisons of PT (–), MM RPT (- -) and AO RPT (-.) designs with nominal uncertainty parameters: second bending mode.

In order to understand why the robust designs are so much less sensitive to uncertainty than the PT design it is necessary to compare the designs at the worst-case uncertainty parameter values. The output PSDs and cumulative variance plots for this case are presented in Figure 3-9(a). The PT design is depicted with a solid line, and the RPT MM and AO design results are plotted in dashed and dash-dotted lines, respectively. At the worst-case uncertainty vertex a situation directly opposite to that at the nominal uncertainty values is observed. All of the energy is contained in the first observable bending mode, while the output energy in both of the RPT designs is distributed over two critical modes. The exact modal energy numbers for the critical modes are listed in Figure 3-9(c) along with a bar chart of percent total

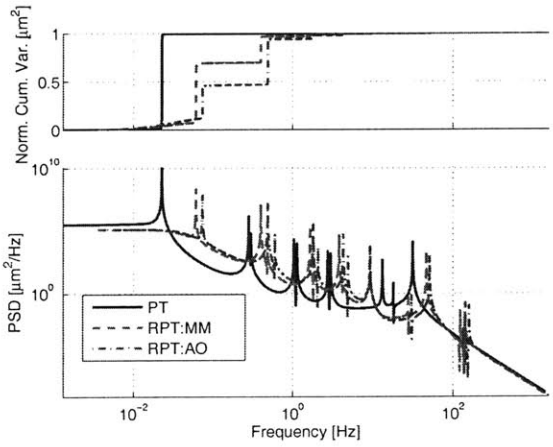
energy by mode for easy comparison.

Comparing the values in this table to those for the worst-case PT system in Figure 3-3(b) it is observed that the first bending mode (Mode 2), which accounts for 99.1% of the total PT energy, is responsible for only 61.2% (MM) and 33.4% (AO) of the RPT performance energy. In contrast, there is very little energy in Mode 3 in the PT design, but this mode is the second most critical in the RPT designs. To understand the physical mechanism behind the energy redistribution consider the contributions of a given mode, k, to the frequency response function:

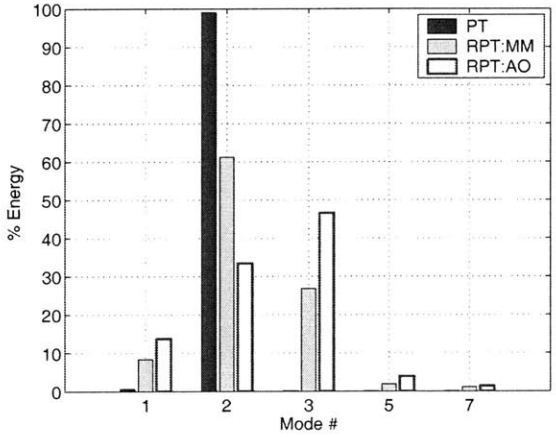
$$G^k_{z_i w_j}(\omega) = \frac{C_{z_i \hat{x}} \phi_k \phi_k^T B_{\hat{x} w_j}}{\omega^2 - 2\xi_k \omega_k + \omega_k^2} \tag{3.15}$$

The quantities $C_{z_i \hat{x}}$ and $B_{\hat{x} w_j}$ are vectors mapping the physical degrees of freedom to the i^{th} output and j^{th} disturbance, respectively as defined in Chapter 2. Since the modal damping, ξ_k and general model structure are constant across designs, the energy in a given mode can only be reduced by changing the mode shape, ϕ_k, in such a way that less energy is transferred to the output or increasing the modal frequency. The mode shapes for Modes 2 and 3 are plotted for all three designs in Figure 3-10. The PT design mode shape is shown with a solid line, while the RPT MM and AO are drawn in dashed and dash-dotted lines. The plot of Mode 2 in Figure 3-10(a), shows that this mode is asymmetric in all three designs. Therefore, it seems that the robustness is not achieved through a change in the mode shape. However, comparing the frequencies of this mode in the PT and RPT designs it is obvious from the tables and the PSD plot (Figure 3-9(a)) that this mode has a significantly lower frequency in the PT design than in the RPT designs. Since the disturbance input is white noise all modes are excited, but those with lower frequencies contribute more to the output energy.

The energy distribution in the third mode shape also changes and is much more significant in the RPT designs. The physical mechanism behind this shift is the same as in the nominal uncertainty case as shown in Figure 3-10(b). Note that the mode shape of the PT design, although asymmetric, has nodal points at the collectors,



(a)

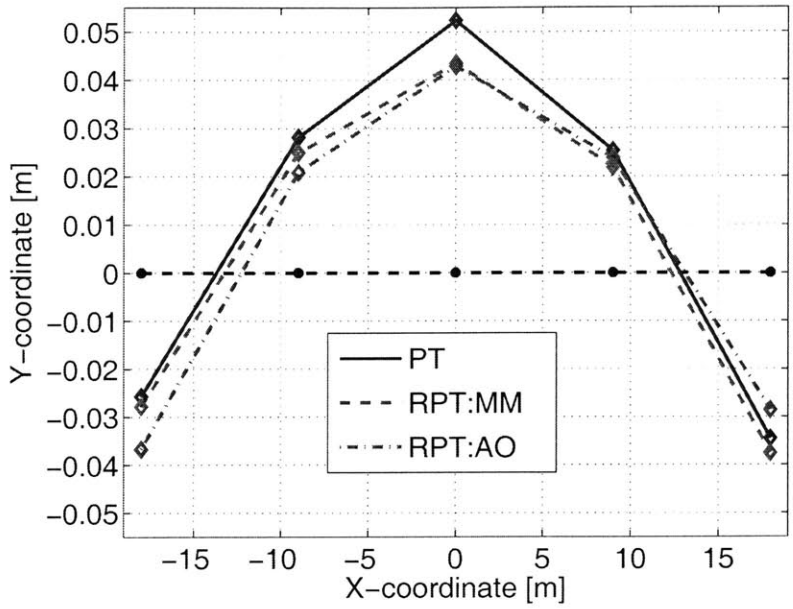


(b)

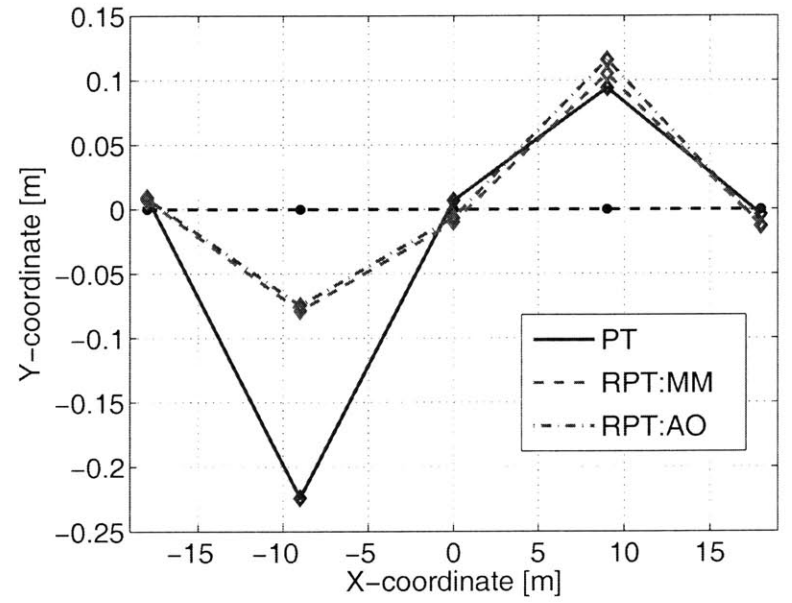
	RPT: Multiple Model				RPT: Anti-optimization			
Mode #	f_n (Hz)	energy %	σ_z^2 (μm^2)	σ_z (μm)	f_n (Hz)	energy %	σ_z^2 (μm^2)	σ_z (μm)
1	0.036	8.34	10933	30.20	0.043	13.70	12902	42.04
2	0.061	61.21	80242	221.63	0.073	33.43	31479	102.58
3	0.399	26.86	35207	97.24	0.488	46.63	43906	143.08
5	1.655	1.87	2449	6.76	1.808	3.85	3625	11.81
7	3.819	1.02	1341	3.70	4.30	1.40	1315	4.29
	Total:	99.30	130172	359.53		99.01	93227	303.80

(c)

Figure 3-9: Modal energy breakdown for PT and RPT designs with worst-case uncertainty: (a) output PSDs (b) % energy comparison: PT (dark), RPT MM (light) RPT AO (white) (c) results table.



(a)



(b)

Figure 3-10: Critical modes shape comparisons of PT (–), MM RPT (- -) and AO RPT (-.) designs with worst-case uncertainty parameters: (a) first bending mode (b) second bending mode.

while the nodal points of the RPT designs are moved in towards the center of the array. This shift in the mode shape accounts for the greater accumulation of energy in this mode in the RPT design when compared to the PT design. However, the price of the increased energy in mode #3 is justifiable since reducing the effect of mode #2 has such a dramatic effect on the worst case performance.

Investigation of the energy distribution among modes and the frequencies and mode shapes of the PT and RPT designs allows a physical understanding of the trades that are made to achieve robustness to uncertainty. In the PT design there is no balance between nominal performance and robustness; instead the resulting system performs very well nominally, but is highly sensitive to uncertainty. The RPT design strikes a balance by making a small sacrifice in nominal performance for a dramatic increase in robustness. The nominal performance of the RPT designs is worse than that of the PT design since the asymmetric bending mode shapes are no longer tailored such that the nodal points are at the ends of the truss where the collectors are located. However, by tailoring the truss such that the mass is concentrated on the inner truss segments, the frequency of the second mode, and consequently its output energy, is significantly reduced at the worst-case uncertainty vertices.

3.4 Limitations: Design Regimes

In order to obtain a complete comparison of PT and RPT methods, the tailoring optimizations and uncertainty propagations are run over a range of uncertainty values from 0.01% to 25%. Since the RPT AO design is the most robust design in terms of worst-case performance, it alone is considered in the study. At each uncertainty level, the optimal design is found and the nominal performance is calculated. Then the uncertainty vertex resulting in the worst-case performance, σ_{WC}, is identified and applied to the design to produce a worst-case performance prediction.

The results of the simulations are plotted in Fig. 3-11. The PT and RPT AO performance predictions are represented with solid and dash-dotted lines, respectively. The nominal performance is denoted by circles and the worst case performance by

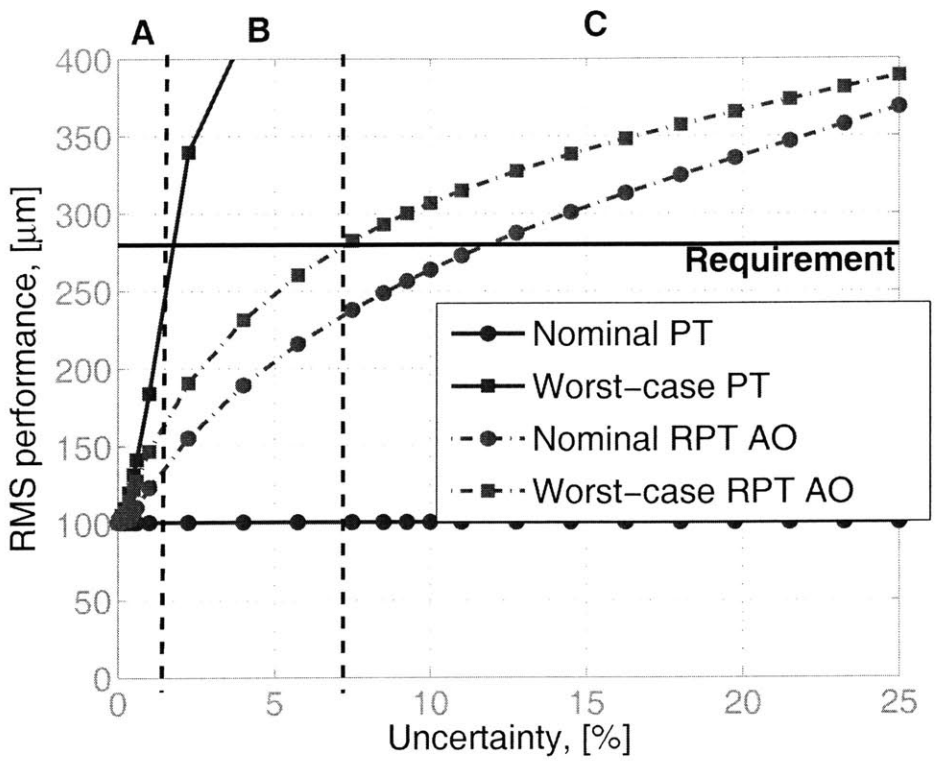


Figure 3-11: RMS OPD for PT (–) RPT MM (- -) and RPT AO (.-) designs vs. % Uncertainty: nominal performance (o), worst-case performance (□), requirement (black –).

squares. Since the PT formulation does not include uncertainty, the nominal PT performance is independent of uncertainty level and remains at a value of 100.53μm across the entire range. However, as the uncertainty increases, the worst case performance of the PT design increases dramatically. In fact, the worst case for values for uncertainty over 3% are too large to show on this plot. The RPT AO formulation does include uncertainty in the cost function and therefore produces different designs at each value of Δ. The worst-case RPT performance predictions at all uncertainty levels are much lower than those of the corresponding PT design, while the nominal RPT performance is larger than the PT nominal and increases with the uncertainty level. This behavior is the classic trade between nominal performance and robustness explored in the previous section. As the uncertainty increases the nominal performance of the RPT design gets slowly worse, but the design is relatively insensitive to uncertainty so the worst-case performance stays close to the nominal prediction.

Distinct design regimes appear when the PT and RPT results are compared to the requirement. The horizontal black line in Fig. 3-11 represents a performance requirement of 280μm, and the design regimes are indicated by dashed vertical lines. In regime A ($\Delta = 0.01\%$ to $\Delta \approx 2\%$), the uncertainty is very low and can be ignored since the worst-case prediction from the PT design is below the requirement. As the uncertainty increases, the worst-case PT performance moves beyond the requirement and the design moves into regime B. In this regime, ($\Delta \approx 2\%$ to $\Delta \approx 7\%$), it is necessary to use RPT optimizations to find a design that meets the requirement at the worst-case uncertainty vertices. However, RPT also has its limits, and once the uncertainty becomes larger than $\Delta \approx 7\%$ even the robust designs fail to meet the requirement in the worst-case realizations. In this regime, labelled C, neither the PT nor the RPT designs are adequate. At these high uncertainty levels, RPT is unable to produce a design that is both insensitive to uncertainty and meets the aggressive performance requirement.

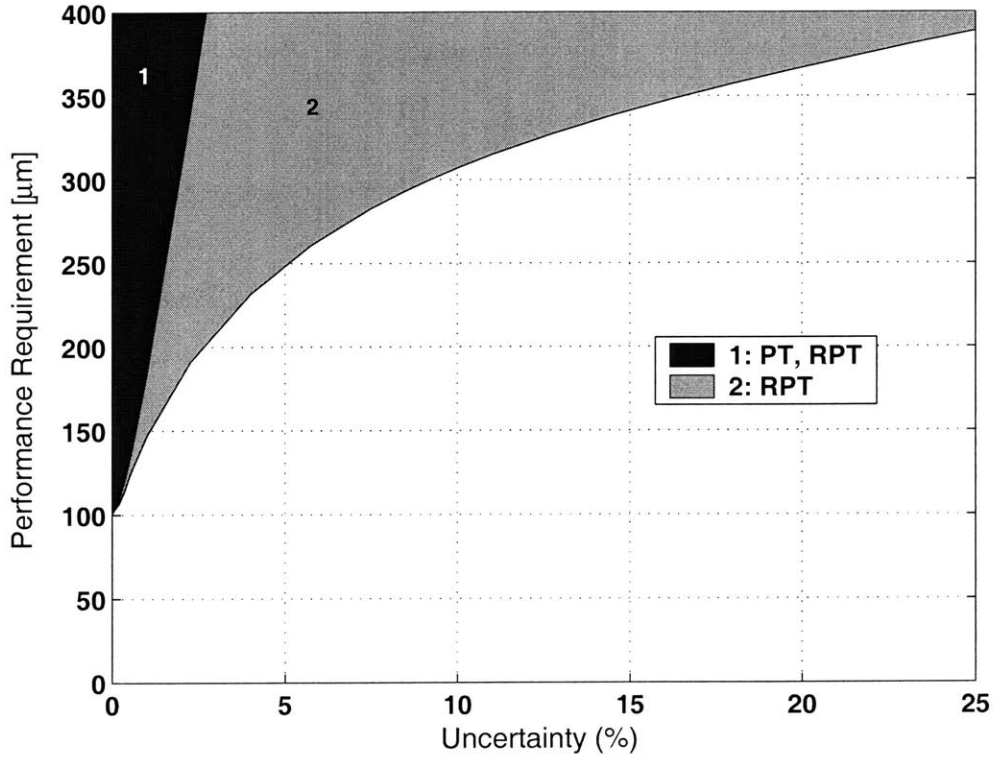


Figure 3-12: Performance requirement vs uncertainty: PT and RPT designs.

The requirement chosen here is somewhat arbitrary, but it is important to note that as the requirement changes so does the level of uncertainty that can be tolerated by each design. To illustrate this point consider the contours shown in Figure 3-12. The performance requirement is shown on the y-axis, and the maximum level of uncertainty that can be tolerated is plotted along the x-axis. The dark patch in the upper left sector of the plot, Region 1, represents the design regime that can be accommodated with PT design methods. As the value of the performance variance increases, indicating more relaxed requirements, a higher level of parametric uncertainty can be tolerated. For example, if $\sigma_{req} = 250\mu$m, then just over 2.3% uncertainty variation can be tolerated with PT design techniques. However, if the performance requirement is tightened to $200\mu m$ then only $\approx 1\%$ variation in the uncertainty parameters can be tolerated.

The RPT design, Region 2 (light patch), covers a greater uncertainty range than the PT design, significantly opening the design space. At a performance requirement of 350μm the RPT design can accommodate over 16% variation in the uncertainty parameters. This value is a significant improvement over the PT design. However, as the performance requirement becomes more aggressive, the RPT design, like the PT design, can tolerate much less uncertainty. At a requirement of $200\mu m$, the RPT design can tolerate $\approx 2.5\%$ variation in the uncertainty parameters, only slightly better than the PT design at the same level. Although the RPT design methods do result in designs that are significantly more robust than simple PT, they are not adequate for high-performance and high-uncertainty designs, as evidenced by the large unfilled area in the lower right corner of the plot.

3.5 Summary

Robust performance tailoring is a design methodology that is used to produce designs that are robust to parametric uncertainty. In this chapter the effects of uncertainty on the PT design are explored motivating the need for robust design techniques. Three different robust cost functions are reviewed and applied to the SCI development

problem. The optimizations are run with both SQP and SA search techniques to produce robust optimal designs. The performance of the optimization techniques are compared, and it is found that the SA design is a good initial guess for SQP, enabling quick convergence to the optimal design for this particular problem. It is also found that the anti-optimization cost function is the most conservative of the three chosen and produces the design with the lowest worst-case performance variance.

The RPT AO and MM designs are compared to the PT design, and it is shown that the RPT designs achieve greater robustness by tailoring the truss such that the inner segments are larger than the outer segments resulting in an increase in the natural frequency of the first bending mode. The PT and RPT designs are compared over a range of uncertainty levels and the concept of design regimes is introduced. PT designs are only adequate in the cases of relaxed performance requirements and low uncertainty levels. RPT designs cover much more of the design space and can accommodate greater levels of uncertainty than the PT designs at equivalent performance requirements. However, even the RPT methodology is not adequate in the high-performance, high-uncertainty regime.

Chapter 4

Dynamic Tuning

Robust Performance Tailoring (RPT) optimization results in a design that is tailored to be robust to parametric uncertainty across a large range of values. Robustness is achieved by sacrificing nominal performance, so that the nominal, and consequently worst-case, performance predictions increase with the level of uncertainty. Therefore, although the RPT design is insensitive to uncertainty, it may not meet agressive performance requirements. The trade between robustness and nominal performance places high-performance and high-uncertainty systems outside of the RPT design regime.

In the following chapter, dynamic tuning is explored as a method of extending the capabilities of PT and RPT design, by exploiting the additional information available from hardware testing. First, a formal definition of dynamic tuning is provided, and the optimization problem is formulated. Tuning parameters are identified in the SCI development model and SQP and SA optimization techniques are employed to tune the worst-case uncertainty realizations of the PT and RPT designs. The tuned designs are considered in the context of the design regimes introduced in the previous chapter, and it is shown that tuning increases the level of uncertainty that can be tolerated at a given performance requirement. Then, a spectrum of tuning methods for practical application, in which the value of the uncertainty parameters are unknown, ranging from pure hardware tuning to model-based techniques are discussed. A hybrid method that uses isoperformance techniques to facilitate model

updating for tuning is developed and demonstrated on the SCI development model. The tuning methods are compared to one another for performance and testing cost over a large sample of hardware simulations, and it is shown that only isoperformance tuning consistently requires a small number of hardware tests and is successful across the sample space.

4.1 Tuning Formulation

In this thesis, dynamic tuning is defined as adjustments made to the hardware once it is built to affect the performance and bring it within requirements. Consider a situation in which models predict that the system meets performance requirements across most of the uncertainty space, but not at the extremes. If only one such system is built, there is a possibility that the physical system may lie outside the area in which the RPT design meets requirements. However, if there are tuning adjustments that can be made to the hardware at this stage to affect the performance, it may be possible to improve the system performance to within the desired range. Due to model inaccuracies and manufacturing discrepancies, situations such as this arise frequently in practice. Engineers often make ad hoc adjustments to hardware to improve performance or bring a component or entire system within specifications. However, this type of tuning is not formalized, and it is difficult to find references on these practices.

Dynamic tuning is a hardware-based procedure, and therefore, the tuning parameters must be chosen carefully. As in the case of performance tailoring, the parameters must have some effect on the performance metric. The range of this effect, or the difference between the tuned and untuned performance, is referred to as the tuning authority. In addition, the parameters must be easy to adjust on the hardware, significantly limiting the possible design variables compared to those available for tailoring. For example, although it has been shown previously that the cross-sectional diameters of the truss members are good tailoring parameters for the SCI development model, they are not well-suited for tuning as there is not an easy way to change these values

on a physical truss. Since tailoring takes place on models it is easy to try a range of truss diameters and optimize the design in this manner. However, tuning with these parameters requires physically cutting into the existing truss and replacing members. This procedure is expensive and may have undesired global effects on the performance resulting in mission delays.

One example of dynamic tuning on hardware is found in a recent paper by Glaese and Bales [46]. The authors study the effects of a range of structural adjustments to a gossamer structure. In the paper, they call the process dynamic tailoring because the goal is to affect the dynamic performance of the membrane. However, in the context of this thesis, their efforts are classified as dynamic tuning since the authors make adjustments to the hardware itself. A 80-inch major diameter, 8-inch minor diameter, pre-formed Kapton torus is the structure used to demonstrate the dynamic tuning. The authors choose four possible tuning parameters: inert masses added at random locations on the structure to minimize the disturbance forces by increasing the effective impedance, tuned mass dampers to reduce narrow-band disturbance effects, piezo-electro actuators and sensors that enhance broadband structural damping, and shunts added to the piezo-electric elements to actively control the flow of energy in the structure. All of these techniques are good examples of tuning parameters since they can be added to the structure and adjusted with minimal disruption to the hardware itself.

The formulation of the tuning optimization problem is similar to performance tailoring with the major distinction being that the system performance is now a function of the tailoring, $\vec{x}$, uncertainty, $\vec{p}$, *and* tuning parameters, $\vec{y}$, but the tuning parameters are the only design variables:

$$\begin{aligned} &\min_{\vec{y}} f\left(\tilde{x}, \vec{y}, \tilde{p}\right) \\ s.t. \quad &\vec{g}\left(\tilde{x}, \vec{y}\right) \leq 0 \end{aligned} \tag{4.1}$$

where $\tilde{x}$ and $\tilde{p}$ are the actual values of the tailoring and uncertainty parameters realized in the hardware and $g\left(\tilde{x}, \vec{y}\right)$ are constraints on the design variables. The

tailoring parameters are fixed by the design, and the uncertainty parameters by the actual hardware realization. At the tuning stage the hardware exists and performance data is available so it is no longer necessary to rely on performance predictions from the model. Furthermore, the hardware represents one instantiation of the uncertainty model which, if identified through the data, no longer requires the system to be robust across the entire uncertainty space. The parametric model uncertainty is effectively removed from the performance data and the uncertainty parameters become deterministic.

4.1.1 Example

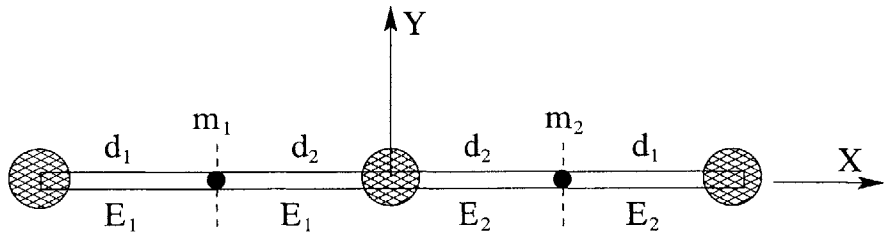
The SCI development model is used to demonstrate the tuning optimization. The design masses originally introduced in Chapter 2 as tailoring parameters are now used as tuning parameters. This choice of tuning element is similar to the inert masses used by Glaese and Bales in [46]. The tuning parameters are listed in Table 4.1 along with a figure of the model reproduced here from Chapter 3 for the reader's convenience. The tuning formulation for this specific problem is as follows:

$$\begin{aligned} & \min_{\vec{y}} \sigma_{OPD}\left(\tilde{x}, \vec{y}, \tilde{p}\right) \\ s.t. \quad & -y_i \leq 0 \\ & \underbrace{\frac{\pi}{4}\sum_{i=1}^{4} \tilde{x}_i^2 L_i \rho_i}_{m_{truss}} + \sum_{i=1}^{2} y_i \leq \bar{M} \end{aligned} \tag{4.2}$$

where $\bar{M}$ is the total mass of the structure, without the optics. The first part of the mass constraint equation, m_{truss}, is static as it is only dependant on the tailoring parameters, the cross-sectional diameters of the beam elements. It is included in the equation here because it changes from one tailored design to another. Recall from Chapter 2 that a mass margin of 10% is held back when the design is tailored. This margin, plus any mass not used by the tailoring optimization, is available for tuning.

Table 4.1: Tuning parameters for SCI development model.

Name	Description	y_0
m_1	-x tuning mass	0 kg
m_2	+x tuning mass	0 kg



To demonstrate the technique the worst-case uncertainty realizations ($\Delta = 10\%$) of the PT and RPT AO designs are tuned using the SA and SQP algorithms. The nominal, worst-case and tuned performances are listed in Table 4.2. The worst case PT performance is significantly higher than nominal, and tuning results in a large improvement. Tuning also improves the RPT worst-case designs, but the results are not as dramatic.

Table 4.2: Tuning performance summary for PT and RPT designs.

	Performance [μm]		
Design	nominal	worst-case	tuned
PT	100.53	1355.5	303.17
RPT AO	263.87	306.86	273.32

Algorithm Performance

In order to evaluate the performance of the different optimization algorithms, tuning is performed with SA-SQP, MC-SQP and an exhaustive search (ES) of the solution space. The results for the PT and RPT designs are given in Figures 4-1 and 4-2, respectively. Each figure includes a table of the algorithm performance data as well as a two-dimensional surface plot of the tuning solution space. In SA-SQP, a feasible initial guess is chosen randomly for SA, and the resulting design is used as the initial guess for the SQP algorithm. In MC-SQP ten feasible initial guesses are randomly

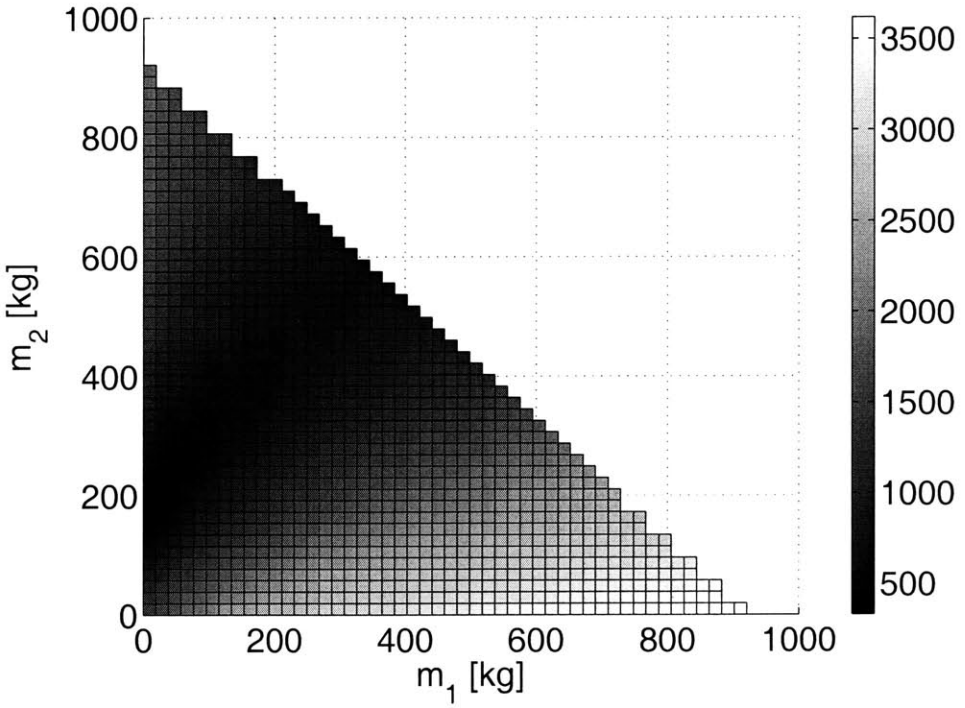
generated and SQP is run from each one. The minimum-cost design is then identified from the results. The cost and design variables listed for this algorithm in the tables are the best of the ten runs, and the number of iterations and function evaluations are the sum of the values from each individual run. The exhaustive search is conducted by discretizing the tuning masses into fifty distinct values within the allowable range and performing a full-factorial search of the space, neglecting any mass combinations that violate the constraints.

First, consider the tuning results for the worst-case PT design (Figure 4-1). The SA-SQP combination finds the optimal tuning solution in the shortest amount of time. The combined time required to tune the PT design using first SA and then SQP is 54.21 seconds, while MC SQP with ten initial guesses takes over almost four times as long (210.51 seconds). The MC-SQP and SA-SQP results are equivalent indicating that a global optimum is found. The SA design providesa a good starting point for the SQP optimization in that the tuned performance is very close to the optimial design and the majority of the mass is placed at m_2. The SA-SQP combination is a good way to handle convexity issues when the space is not well known. The exhaustive search results further support this conclusion, as does the surface plot of the solution space, Figure 4-1(b). The light and dark regions in the plot indicate high and low performance variance, respectively. The darkest area is along the constraint boundary of zero m_1, near $m_2 = 200$ kg.

The RPT AO results, Figure 4-2, show similar trends to the PT results. Both the SA-SQP and MC-SQP algorithms find an optimal tuning solution, but SA-SQP does so much more quickly (37 seconds, instead of 264 seconds). SA finds a sub-optimal design due to the randomness of the search, but provides a good starting point for SQP. The exhaustive search design is nearly equivalent to the SA-SQP and MC-SQP designs indicate that a global optimum is found. The ES design is slightly sub-optimal due to coarse discretization of the tuning parameter values. The accompanying plot of the tuning space, Figure 4-2(b), shows that the tuning solution is a global minimum. A comparison of the performance scale of the RPT space to that of the PT space indicates that the RPT design is less tunable. In Figure 4-1(b) the performance

Algorithm	J^* [μm]	# iter	# fvals	time [sec]	y^* [kg] m_1	m_2
SA	303.55	64	2445	44.15	174.01	0.15
SQP	303.17	3	14	10.06	175.05	0.00
MC SQP	303.17	98	73	210.51	175.07	0.00
ES	303.46	N/A	1275	23.02	172.66	0.00.

(a)



(b)

Figure 4-1: PT worst-case tuning results: (a) table of algorithm performance results (b) solution space, 2D surface view.

ranges from 300μ m to 3650μ, while in the RPT performances range from 273μm to only 665μm. The shorter range suggests that the tuning parameters have less of an effect on the performance of the RPT design.

It is clear from these reults that tuning the worst-case uncertainty realization has a significant effect on the performance. In the following sections, a physical understanding of how the tuning masses improve the performance is obtained by investigation of the output PSDs and energy distribution among modes.

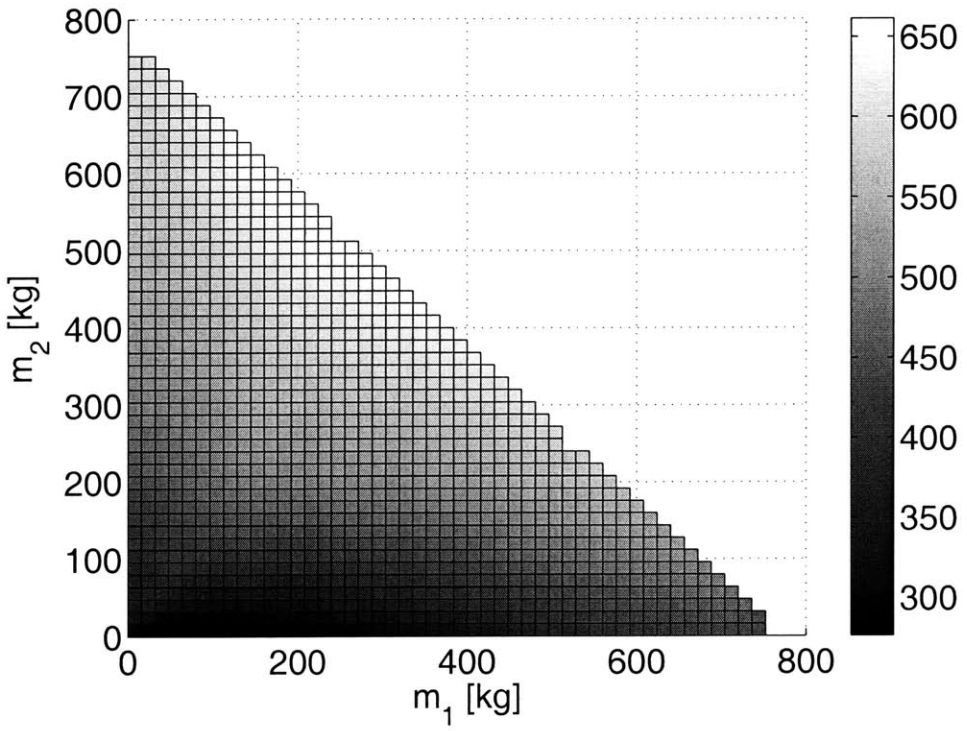
Physical Interpretation: PT Tuning

To begin, consider the worst-case PT design in both the untuned and tuned configurations. As seen from Table 4.2 and Figure 4-1, a 77% reduction in performance variance is achieved by adding 175 kg of tuning mass to the positive-x arm of the interferometer. The plot of the tuning space (Figure 4-1(b)) shows that this particular tuning configuration is a global optimum. The RMS increases as mass is added to the negative-x arm or if more/less mass is added to the positive-x arm. To understand why this tuning configuration has such an impact on the performance consider the energy information presented in Figure 4-3.

The first plot (Figure 4-3(a)) is the normalized cumulative variance and output PSDs for both the untuned (solid line) and tuned (dashed line) worst-case PT systems. The largest differences in energy distribution between the two systems occur at low frequencies. The second mode, which accounts for almost all of the RMS in the untuned case, has only a small effect in the tuned system, and the third mode is at a lower frequency in the tuned system. The bar chart, (Figure 4-3(b)), shows the percent of total energy in each of the critical modes. It is clear that the addition of tuning mass shifts the energy from the second mode to the third mode. The accompanying table lists the natural frequencies and energy of each of the critical modes. The absolute energy values show that not only has the energy been redistributed among the modes, but the total energy in the modes is greatly reduced. For example the second mode has 1325μm of RMS in the untuned PT design, but only 14.96μm in the tuned configuration. The third mode accounts for more energy in the tuned

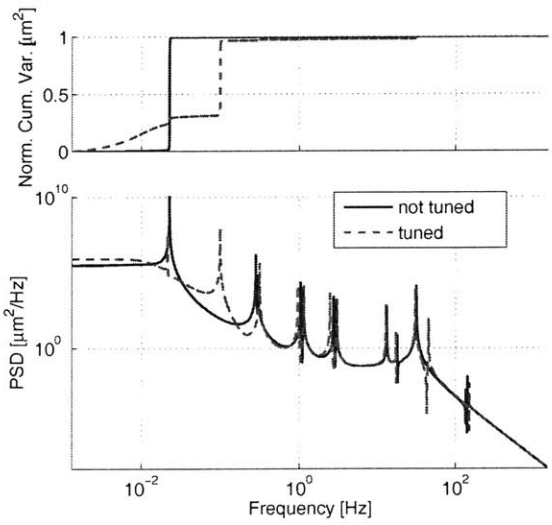
Algorithm	J^* [μm]	# iter	# fvals	time [sec]	y^* [kg] m_1	m_2
SA	298.28	68	1980	18.97	6.77	0.02
SQP	273.32	6	18	18.36	81.05	0.0
MC SQP	273.32	84	259	264.3	80.99	0.0
ES	273.32	N/A	1271	12.42	80.05	0.0

(a)

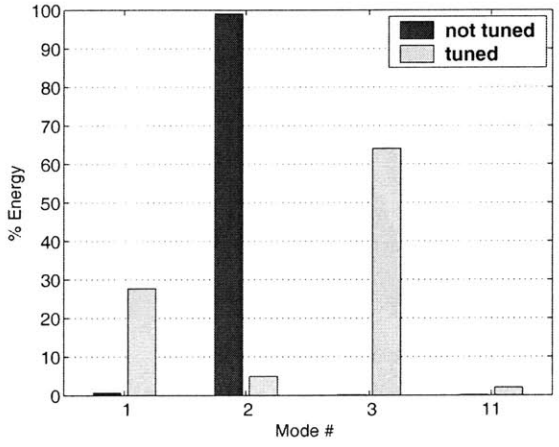


(b)

Figure 4-2: RPT AO worst-case tuning results: (a) table of algorithm performance results (b) solution space, 2D surface view.



(a)



(b)

	Not tuned				Tuned			
Mode #	f_n (Hz)	energy %	σ_z^2 (μm^2)	σ_z (μm)	f_n (Hz)	energy %	σ_z^2 (μm^2)	σ_z (μm)
1	0.013	0.61	11144	8.22	0.013	27.68	25446	83.93
2	0.023	99.09	1820600	1343.10	0.021	4.93	4537	14.96
3	0.281	0.14	2641	1.95	0.097	64.03	58852	194.12
11	31.77	0.14	2554	1.88	30.95	2.05	1884	6.21
	Total:	99.98	1836939	1355.15		98.69	90719	298.22

(c)

Figure 4-3: Modal energy breakdown for worst-case PT design (a) output PSDs (b) % energy comparison: not tuned (blue), tuned (green) (c) results table.

configuration than the untuned, but the increase is only from 1.33μm to 194.13μm so that the total energy is much lower when the design is tuned.

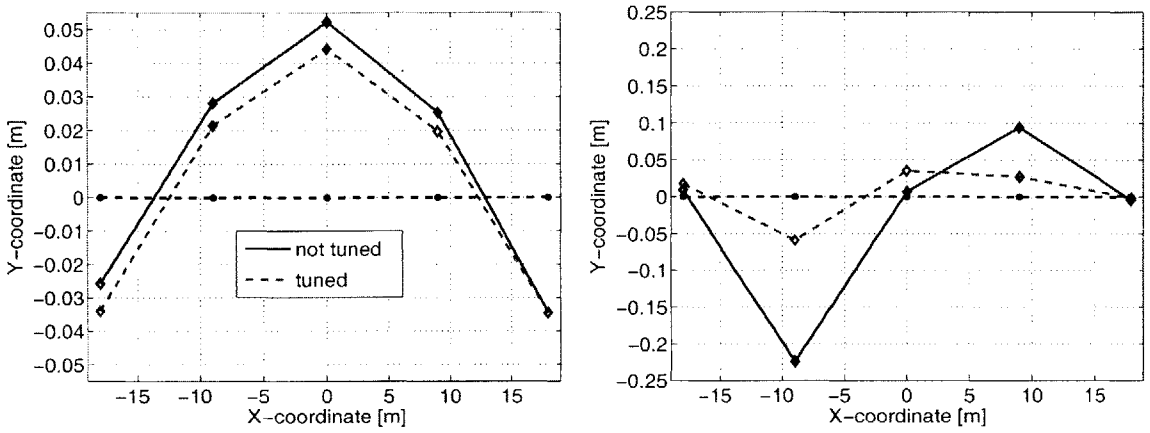


Figure 4-4: Mode shape comparisons, worst-case PT untuned (blue solid) and tuned (green dashed): (a) Mode #2, first bending (b) Mode #3, second bending.

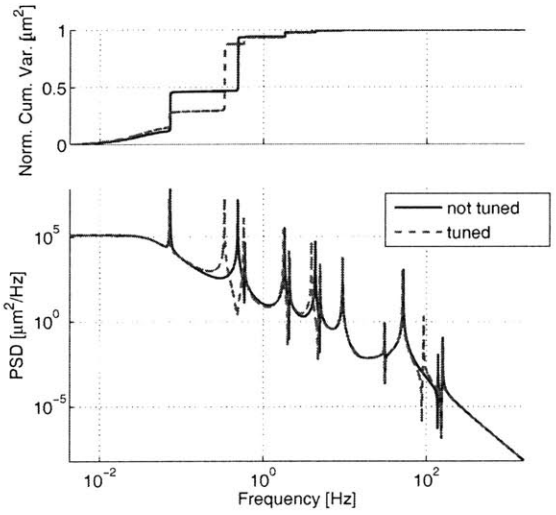
The second and third modes of the untuned and tuned PT designs are plotted in Figure 4-4. Recall from the last chapter that the worst-case PT design has a high RMS due to an asymmetric uncertainty distribution that results in an asymmetric first bending mode shown with the solid line in Figure 4-4(a). When the design is tuned, mass is added asymmetrically (only to the positive-x arm) to counter the asymmetric uncertainty distribution. As a result, the second mode in the tuned system (dashed line) is nearly symmetric, and little OPD is accumulated by this motion. The price that is paid for this improvement is observable in the third mode, Figure 4-4(b). Note that the outer nodal points of the untuned mode (solid line) are located very close to the ends of the structure where the collectors are located resulting in little OPD accumulation. However, when the design is tuned (dashed line), these nodal points move towards the center of the array, and the collector nodes experience greater relative displacement. The increase in OPD in the third mode energy (192.8μm) is more than offset by the decrease in that of the second mode (1310.84μm). The tuning algorithm properly identifies the second mode as the critical one and focuses all the

tuning effort there.

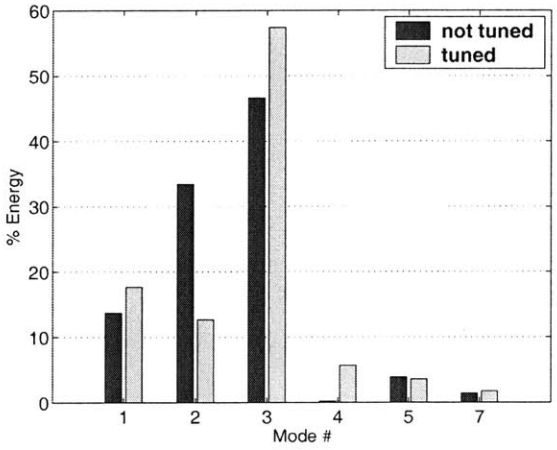
Physical Interpretation: RPT Tuning

The results for the RPT AO design exhibits trends similar, although less dramatic, to those of the PT design. Recall from Table 4.2 that tuning the worst-case AO design reduces the performance variance by 11%. The energy results for this design is presented in Figure 4-5. The cumulative variance and output PSD plots, Figure 4-5(a), show a decrease in the energy in Mode 2 and the frequency of Mode 3. The bar chart of percent energy, Figure 4-5(b), supports the PSD plots indicating that energy distribution has been shifted from the second mode to the third mode and a few higher modes. It is interesting to compare this bar chart to that of the PT design (Figure 4-3(b)). Note that the second mode contains a smaller percentage of the energy in the RPT design, and additional high-frequency modes are critical. Recall from Chapter 3 that the robust optimization tailors the system to reduce the effect of the second mode by increasing its frequency. The resulting design is insensitive to the uncertainty parameters, but *is also insensitive to the tuning parameters.* However it is still most advantageous, given the tuning parameters chosen, to place the mass such that the energy in mode two is reduced further. The table included in the figure provides additional results that support these observations. Note that the energy in Mode 2 is reduced from 102.58 to 34.66μm, but there is an increase in the accumulated RMS in Mode 3.

The second and third mode shapes for the RPT AO design are plotted in Figure 4-6. The effect of tuning on the second mode (Figure 4-6(a)) is more difficult to observe in this case. Recall from chapter 3 that the AO design is tailored for robustness and as a result has more effect on the second mode in the worst-case uncertainty realization. Therefore, since the tuning parameters are limited to masses in only two locations, there is less tuning authority for this system. The second mode is made slightly more symmetric through the tuning masses, but not significantly so. The third mode (Figure 4-6(b)) exhibits characteristics similar to that of the tuned PT design in that the nodal crossings are shifted slightly inwards resulting in a small



(a)

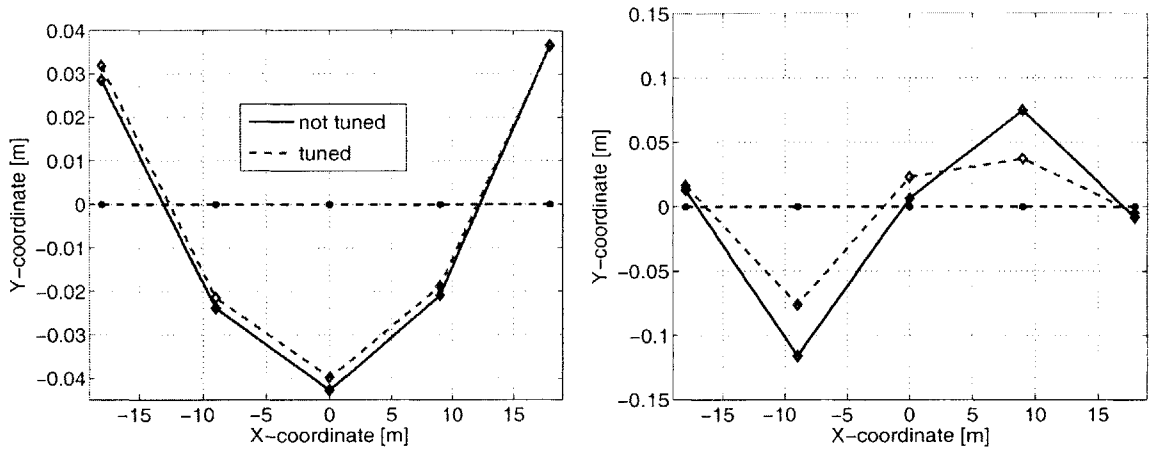


(b)

Mode #	Not tuned f_n (Hz)	energy %	σ_z^2 (μm^2)	σ_z (μm)	Tuned f_n (Hz)	energy %	σ_z^2 (μm^2)	σ_z (μm)
1	0.043	13.70	12902	42.04	0.043	17.64	13179	48.22
2	0.073	33.43	31478	102.58	0.072	12.68	9474	34.66
3	0.487	46.63	43906	143.08	0.336	57.43	42907	156.98
4	0.599	0.19	175	0.57	0.578	5.64	4217	15.43
5	1.808	3.85	3625	11.81	1.754	3.57	2670	9.77
7	4.300	1.34	1315	4.29	3.864	1.69	1265	4.62
	Total:	99.14	93401	304.37		98.65	73712	269.68

(c)

Figure 4-5: Modal energy breakdown for worst-case RPT AO design (a) output PSDs (b) % energy comparison: not tuned (blue), tuned (green) (c) results table.



(a) (b)

Figure 4-6: Mode shape comparisons, worst-case RPT AO untuned (blue solid) and tuned (green dashed): (a) Mode #2, first bending (b) Mode #3, second bending.

increase in OPD in this mode.

4.1.2 Design Regimes

In order to assess the impact of tuning on the design space it is applied to the worst-case PT and RPT AO models across a range of uncertainty values. The results are shown in Figure 4-7, a further evolution of the design regimes plot introduced at the end of Chapter 3. The y-axis represents the performance requirement of the system and the x-axis is the level of uncertainty in the parameters. It is assumed that the uncertainty levels range $\pm\Delta$ about the nominal parameter value and are the same for both uncertainty parameters. The design regimes are the numbered areas, and the design methods that are successful in each regime are listed in the legend.

The addition of hardware tuning to the design process changes the design regimes significantly from those observed with PT and RPT alone (Figure 3-12). There are now six separate regimes instead of only two due to intersecting regions, where more than one technique is applicable. Consider, for example, a performance requirement of 200μm. PT is adequate for this level of performance if the uncertainty is under

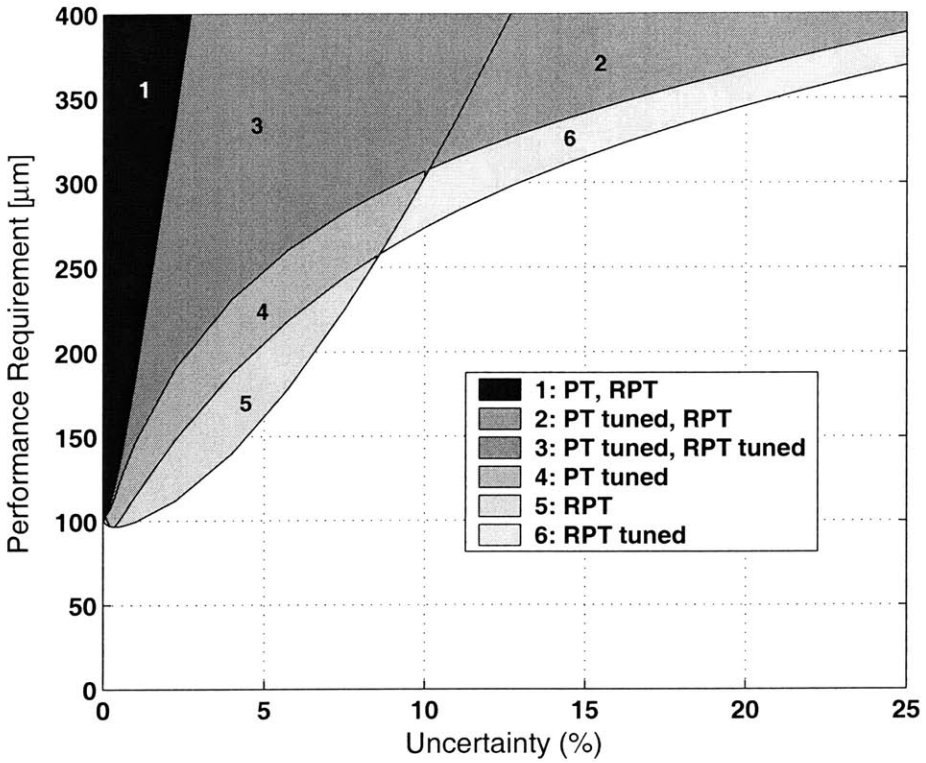


Figure 4-7: Requirement vs uncertainty for PT and RPT designs with tuning: design regimes are numbered and labelled on plot.

2%. Tuning the PT design increases the tolerated uncertainty level to just about 7%. It is interesting to note that the RPT AO method is only applicable up to 3% uncertainty at this perfomrance, and that this range is only increased to 5% by the addition of tuning. Therefore there is a regime, Region 4 in the figure, in which tuning the PT design is the only successful method. This result indicates that for this problem tailoring the system to be robust actually reduces the tuning authority available for later adjustments on the hardware. At the more stringent performance requirements it is better to performance tailor the design and then compensate for the uncertainty with tuning. This approach is somewhat worrisome because the success of the mission relies heavily on the ability to tune the hardware since the predicted worst case of the PT design is many times that of the nominal performance even at the low uncertainty levels.

As the requirement is relaxed the RPT and tuned RPT designs have a great effect on the design space. At a requirement of 280μm, PT is only adequate up to

2% uncertainty, but the tuned RPT design can tolerate up to 11%. Tuning the PT design also does better at this requirement, allowing just under 9.2% variation in the uncertainty parameters. At this relaxed requirement. there is no longer a regime where only tuned PT works. The RPT designs can both compensate for uncertainty and provide a fair amount of tuning authority. Relaxing the requirement further, above 350μm for example, greatly increases the benefits of tuning the RPT design. At these requirement levels tuning the PT design can accommodate 11% variation in uncertainty parameters, but tuning the RPT design extends the range out to 20%.

It is clear from the comparison of Figures 4-7 and 3-12 that tuning the hardware has a significant impact on the design space. At a given requirement, a larger variation in the uncertainty parameters can be tolerated if the hardware is tuned to improve the system performance. The addition of a tuning step to the overall design process allows a greater level of risk for systems with aggressive performance goals. In the next section, practical methods for tuning hardware, when the realized values of the uncertainty parameters are *not* exlicitly known, are explored.

4.2 Tuning in Practical Application

The results presented in the previous section demonstrate that hardware tuning is effective in improving system performance and extends the limits of RPT and PT design. The studies are conducted on models in which the uncertain parameters are known, since systems under worst-case uncertainty are considered. In practical application, however, the engineer does not have the advantage of full knowledge of the uncertain parameters. Even though the hardware is built and these parameters are no longer random, their values can rarely be measured directly. For example, in the case of Young's Modulus, it is very difficult, if not impossible, to measure the exact value of this parameter even with access to the hardware. Despite this difficulty there is information available that can indirectly lead to the identification of the uncertainty parameters and successful tuning of the hardware. In this section, the practical problem of tuning the hardware when only performance data is available

is considered.

4.2.1 Hardware-only Tuning

One approach to the problem is to perform a tuning optimization and replace performance predictions from the model with hardware data. In effect, a real-time optimization is conducted using the hardware and test data. At each iteration of a gradient-based optimization algorithm the cost function, $f(x_k)$, is evaluated, the gradient of the objective is calculated and a new search direction and step-size are obtained. The use of a model in this process allows for quick computation and analytical gradient calculations.

Replacing the model simulation with actual test data is both computationally expensive and labor intensive as each function call requires an actual hardware test. In addition, analytical gradients are no longer available and finite-difference approximations must be used instead. There are two methods to compute finite-difference gradients, the forward-difference and central-difference equations:

$$\frac{\partial f(x)}{\partial x_i} = \frac{f(x + \Delta x e_i) - f(x)}{\Delta x} \tag{4.3}$$

$$\frac{\partial f(x)}{\partial x_i} = \frac{f(x + \Delta x e_i) - f(x - \Delta x e_i)}{2\Delta x} \tag{4.4}$$

where i denotes an element of x, e_i is a unit vector with a 1 in the i^{th} location, and Δx is a small change in the design parameter. The central difference equation (Equation 4.4) is more accurate, but requires an additional function evaluation at each step. These approximations are both sensitive to the size of Δx, and large parameter changes may be outside the range of linear approximation. The need to use finite-difference gradient approximations adds to the time and cost burden of real-time tuning optimizations.

A second consideration of hardware optimizations is that it is not always possible to evaluate iterates that are beyond the constraint boundaries. In the example of mass tuning considered in this chapter there are two constraints: a total mass constraint

and a lower bound constraint at zero. When performing a hardware-based empirical optimization it may be okay to break the total mass constraint, but it is not physically possible to add negative tuning mass. Therefore, the lower bound is a hard constraint that cannot be broken during the optimization. For this reason, it is desirable to use an optimization method that guarantees that the search path contains only feasible iterates.

Barrier Method

Barrier methods, or interior point methods, have recently gained popularity for solving optimization with inequality constraints. The constrained problem is converted into a sequence of unconstrained problems that involve a high cost for approaching the boundaries of the feasible region. As a result, if the initial iterate is feasible, all subsequent ones are feasible as well. The following brief overview is taken from the book by Bertsekas [16].

The interior of the set, defined by the inequality constraints, $g_j(\vec{y})$, is:

$$S = \{\vec{y} \in Y | g_j(\vec{y}) < 0, \quad j = 1, \ldots, r\} \tag{4.5}$$

It is assumed that S is non-empty and that any feasible point not in S can be approached arbitrarily closely by a vector from S. The optimization objective is augmented with a barrier function that is defined in the interior set, S, and goes to infinity at the boundary of S:

$$B(y) = -\sum_{j=1}^{r} \ln\{-g_j(\vec{y})\} \tag{4.6}$$

Equation 4.6 is a logarithmic barrier function. Alternate forms, such as inverse barrier functions are also used. The barrier function is modified by a parameter sequence, $\{\mu_k\}$:

$$0 < \mu_{k+1} < \mu_k \quad k = 0, 1, \ldots, \quad \mu_k \to 0 \tag{4.7}$$

and added to the objective function:

$$y_k = \arg\min_{y \in S} \{f(y) + \mu_k B(y)\} \quad k = 0, 1, \ldots \tag{4.8}$$

The subscript k indicates the iteration number, and the weight of the barrier function relative to the objective decreases at each iteration allowing the search to approach the constraint boundary. The search direction can be determined through any standard gradient search procedure such as steepest descent, Newton's method or conjugate gradient (See Appendix A). However, the step-size must be properly selected to ensure that all iterates lie within the feasible region, S. An interior point method is attractive for the problem of hardware tuning since all iterates are feasible.

The barrier method is implemented in MATLAB as outlined in Figure 4-8. The initial iterate, x_0, termination condition tolerances and barrier parameters, ϵ and μ_0, are inputs to the algorithm. The iteration loop begins by initializing the algorithm as shown. Then the constraint equations, their derivatives and the barrier function (Equation 4.6) are calculated analytically at the current iterate. The tuning parameters are set to the current iterate on the hardware and a test is run. This data is saved as $f(y_k)$, and the objective cost at this iterate is evaluated. Next, the gradients of the objective are needed to generate a new search direction:

$$\begin{aligned} \frac{\partial J(y_k)}{\partial y} &= \frac{\partial f(y_k)}{\partial y} + \mu_k \frac{\partial B(y_k)}{\partial y} \\ \nabla J(y_k) &= \nabla f(y_k) + \mu_k \nabla B(y_k) \end{aligned} \tag{4.9}$$

The performance gradients, $\nabla f(y_k)$, are determined with Equation 4.3, requiring an additional hardware setup and test, while the barrier gradients are calculated analytically:

$$\nabla B(y_k) = -\sum_{j=1}^{r} \left\{ -\frac{1}{g(y_k)} \frac{\partial g(y_k)}{\partial y} \right\} \tag{4.10}$$

The objective gradients are then used to find the new search direction. The MATLAB implementation developed for this thesis allows either a steepest descent or conjugate gradient directions. Once the search direction is obtained the step-size is calculated

Data: initial iterate, y_0, termination tolerances, barrier parameters, ϵ, μ_0
Result: optimal design variables
begin
  Initialize: $y = y_0$, $k = 1$, $\mu = \mu_0$s;
  while *termination conditions not met* **do**
    evaluate $g(y)$, $\frac{\partial g(y)}{\partial y}$;
    evaluate $B(y)$ (Equation 4.6);
    evaluate $f(y)$ [Hardware test];
    evaluate $J = f(y) + \mu B(y)$;
    calculate finite difference gradients [Hardware test];
    calculate $\nabla B(y)$ and $\nabla J(y)$;
    calculate descent direction;
    if *steepest descent* **then**
      $d = -\nabla J(y)$ (Equation A.3);
    else if *conjugate gradient* **then**
      $d = -\nabla J(y) + \beta d_{k-1}$ (Equation A.15);
    end
    calculate step-size;
    if *decreasing* **then**
      $\alpha_k = \frac{\alpha_0}{\sqrt{k}}$ (Equation A.10);
    else if *line minimization* **then**
      $\alpha_k = \arg\min_{\alpha \in [0,s]} f(y_k + \alpha d_k)$ (Equation A.11);
    end
    evaluate new iterate, $y_{k+1} = y_k + \alpha_k d_k$;
    increment barrier sequence, $\mu_{k+1} = \mu_0 \epsilon^k$;
    increment iterate, $k = k + 1$;
  end
end

Figure 4-8: Barrier method implementation.

using either a decreasing step-size or a line search algorithm. The calculation of these search directions and step-sizes is discussed in detail in Appendix A. Special care is taken in the step-size selection such that the next iterate remains within the feasible set. Finally, the new iterate is calculated and the barrier sequence and iteration numbers are incremented.

Three termination conditions are implemented for the hardware tuning application. Since the goal of the exercise is simply to bring the hardware within the required performance it is not actually necessary to reach an optimum tuned performance. Therefore at each iteration the algorithm checks to see if the hardware performance is better than the requirement, if so the tuning optimization terminates. If the tuned performance does not get below the requirements the optimization terminates when either the change in objective function or the change in the design variables fall below a specified tolerance. These conditions only exist to prevent an infinite loop if the system cannot be tuned below the requirement. In reality, tuning on such a system has failed.

Simulated Annealing

Many stochastic search methods are easily modified such that only feasible iterates are considered. Simulated annealing is a particularly attractive search technique for a hardware optimization problem due to the simplicity of the algorithm. The search algorithm and MATLAB implementation are discussed in detail in Chapter 2 (see Figure 2-4). The algorithm is modified slightly for application to hardware tuning by adding $f(y_k) < f_{req}$ as a termination condition. In effect, simulated annealing for hardware tuning is simply a random search of tuning configurations in the hope of finding one that works.

4.2.2 Model-only Tuning

A major drawback of the hardware optimization methods described above is that they are labor and time intensive and therefore, costly. Each function evaluation

required by the optimization or stochastic search algorithm involves a new test setup and data acquisition. In contrast, optimizations performed on the model are much cheaper requiring only computational effort. In this section some model-only tuning schemes are considered.

Nominal Model Tuning

A very simple approach to model-only tuning is to tune the nominal model for improved performance and apply this tuning configuration to the hardware. If the shape of the performance space with respect to the tuning parameters is similar for the nominal model and the hardware, then a tuning configuration that improves the nominal model performance may also improve the hardware performance. If the hardware performance is only slightly above the requirement or the degree of uncertainty is low then this improvement may be enough to bring the system within specification. The optimization formulation is very similar to that of the tuning formulation in Equation 4.2 with $\vec{p}_0$ replacing the actual uncertainty values, $\tilde{p}$:

$$\begin{aligned} &\min_{\vec{y}} f\left(\tilde{x}, \vec{y}, \vec{p}_0\right) \\ s.t. \quad &\vec{g}\left(\tilde{x}, \vec{y}\right) \leq 0 \end{aligned} \tag{4.11}$$

This technique is referred to as nominal model tuning throughout the rest of this chapter.

Robust Tuning with Anti-optimization

A second model-only tuning method is to apply the robust performance tailoring ideas to the tuning problem. The anti-optimization method is chosen as the cost function to obtain a conservatively robust design. The formulation is the nearly the same as

for tailoring, but tuning parameters are the design variables:

$$\begin{aligned} & \min_{\vec{y},z} z \\ s.t. \quad & \vec{g}(\tilde{x},\vec{y}) \leq 0 \\ & h_i\left(z,\tilde{x},\vec{y},\vec{p}_i\right)) \leq 0 \quad i = 1 \ldots n_{pv} \end{aligned} \tag{4.12}$$

where the augmented constraints, $h_i\left(z,\tilde{x},\vec{y},\vec{p}_i\right)$, are defined as follows:

$$h_i\left(z,\tilde{x},\vec{y},\vec{p}_i\right) = -z + f\left(\tilde{x},\vec{y},\vec{p}_i\right) \tag{4.13}$$

The goal of the AO tuning optimization is to find a set of tuning parameters that reduce the worst-case model performance over the uncertainty vertices. The resulting tuning parameters are then used to tune the hardware.

Worst-Case Model Tuning

Another approach that is similar in nature to nominal model tuning and less conservative than AO tuning is worst-case model tuning. Instead of tuning the nominal model to find a tuning configuration, or trying to tune over the entire uncertainty space, only the model at the worst-case uncertainty vertex is tuned. The worst-case combination of uncertainty parameters is found by searching over the uncertainty vertices of the untuned model. If the design has been tailored with RPT methods then this information may already be available. The optimization formulation is similar to that of nominal model tuning with the worst-case uncertainty values, $\vec{p}_{WC}$, replacing the nominal values:

$$\begin{aligned} & \min_{\vec{y}} f\left(\tilde{x},\vec{y},\vec{p}_{WC}\right) \\ s.t. \quad & \vec{g}\left(\tilde{x},\vec{y}\right) \leq 0 \end{aligned} \tag{4.14}$$

Optimized Model Tuning

None of the model tuning methods presented thus far use any information from the hardware to guide the tuning process. Tuning is only necessary if the hardware does not meet performance requirements, so an initial test must be conducted to evaluate the hardware. Therefore, there is at least one data point, the performance of the untuned hardware, f_{HW}, available for the model tuning process. One way to incorporate this data is to formulate an optimization to find uncertainty parameters that result in a performance prediction from the model equal to that of the hardware:

$$\begin{aligned} \hat{p} \quad &= \quad \arg\min_{\vec{p} \in P} \left(f\left(\tilde{x}, \vec{y}_0, \vec{p}\right) - f_{HW} \right)^2 \\ &s.t. \quad \vec{p} - (1+\Delta)\vec{p}_0 \leq 0 \\ &\qquad (1-\Delta)\,\vec{p}_0 - \vec{p} \leq 0 \end{aligned} \tag{4.15}$$

The constraint equations ensure that the optimal uncertainty parameters are within the bounds of the uncertainty model. The resulting uncertainty values, $\hat{p}$, are used in the model tuning optimization to obtain a hardware tuning configuration:

$$\begin{aligned} &\min_{\vec{y}} f\left(\tilde{x}, \vec{y}, \hat{p}\right) \\ s.t. \quad &\vec{g}\left(\tilde{x}, \vec{y}\right) \leq 0 \end{aligned} \tag{4.16}$$

This technique is referred to as optimized model tuning throughout the rest of this chapter.

4.2.3 Example

The hardware and model tuning methods are applied to the SCI development model to assess their performance. In this example, a hardware simulation is used in lieu of actual hardware. It is assumed that uncertainty exists only in the uncertain parameters and is within the bounds of the uncertainty model. All other model parameters and the general form of the model are assumed exact. A hardware simulation is ob-

tained by randomly choosing parameters from the uncertainty distribution. In order for a system to be a candidate for tuning the hardware performance must be worse than the requirement. For this example, the RPT AO design at $\Delta = 10\%$ uncertainty is used to generate a hardware model, and the requirement is set at 280μm. The model parameters and performance data for the resulting hardware model is listed in Table 4.3. Recall from Chapter 3 that in RPT design all of the mass is given

Table 4.3: Hardware Model Data.

Tailoring, x	0.0486	0.0581	[m]
Uncertainty, p	76.6	65.4	[GPa]
Tuning, y	0.0	60.01	[kg]
HW Performance σ_{HW}	290.83		[μm]
Tuned Performance σ_{HWt}	262.75		[μm]

to the cross-sectional diameters and the design masses are not used at all. However, 10% of the allowable mass is reserved during the design stage as margin and is now available for tuning. The uncertainty parameters listed in the table are chosen randomly from the uncertainty model. The performances, σ_{HW} and σ_{HWt}, are the system performance variances before and after tuning, respectively. The tuned performance is obtained by directly tuning the hardware with the known uncertainty values as in the previous section. This result serves as a baseline for the other tuning schemes and confirms that this system can be tuned below the requirement.

Model Tuning

To begin, the four model tuning schemes, nominal model tuning (NomMod), AO model tuning (AOMod), worst-case model tuning (WCMod), and optimal model tuning (OptMod) are applied to the hardware simulation. We start with model-only tuning in the example since it is the easiest to implement and requires no additional hardware testing. In a sense, these tuning methods are "free" and, at the very least, provide starting points for the hardware tuning schemes. The results of these schemes are presented in both tabular and graphical form in Figure 4-9. The table (Figure 4-9(a)) includes the uncertainty parameters used in the model, the tuning

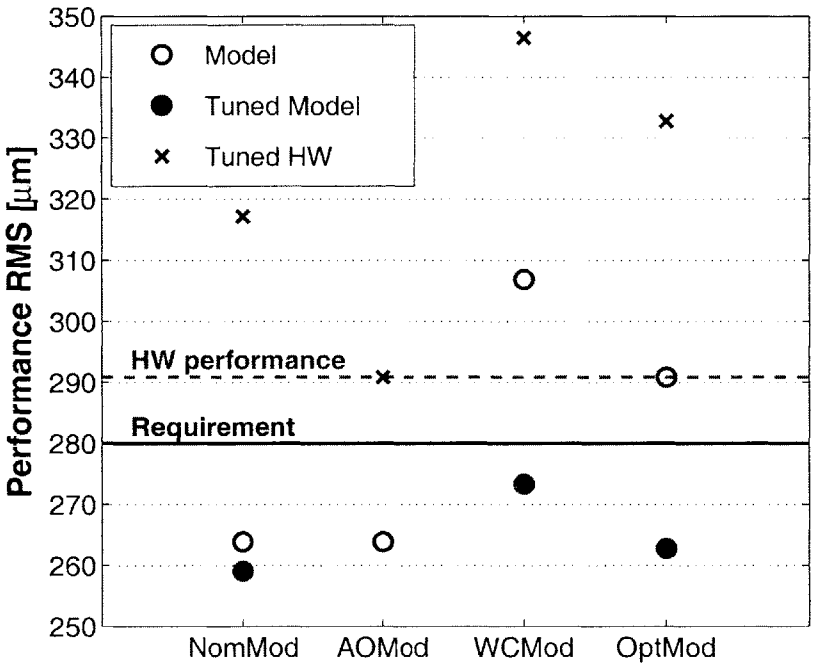
parameters obtained by tuning the model and the performance variance of the nominal and tuned models as well as that of the tuned hardware. These results are plotted graphically in Figure 4-9(b) with open circles, filled circles and black x's representing the nominal model, tuned model and tuned hardware performance RMS, respectively. The tuned hardware preformance is obtained by applying the tuning parameters resulting from a given tuning algorithm to the hardware simulation. The requirement is indicated by the solid line, and the untuned hardware performance by the dashed line.

It is clear from both the table and the figure that none of the model-only tuning schemes result in a tuning configuration that successfully tunes this hardware simulation since none of the x's lie below the requirement line. This result is not surprising for the nominal, AO and worst-case model methods since there no hardware data are used at all in those techniques. Both the nominal and worst-case model tuning provide tuning configurations that improve the model performance, but make the hardware performance worse. The AO tuning fails to find a tuning configuration that improves all of the uncertainty vertices. The untuned configuration is actually the most robust since the uncertainty space is so large. Therefore the nominal and tuned model performance (open and filled circles) are directly on top of each other and the hardware performance (black x) stays at its nominal value.

It is a bit more surprising that optimal model tuning also fails to improve hardware performance. The optimal model has uncertainty parameters chosen specifically to result in a model prediction equal to the hardware performance. This goal is met as indicated in Figure 4-9(b) by the fact that the open circle is directly on the dashed line for the optimized model case. However, despite the fact that the performances are the same, the tuned configuration obtained from the optimized model does not also tune the hardware. In fact, in the figure, the x associated with this method is well above the HW performance line indicating that tuning has made the performance worse. This result suggests that the uncertainty parameters found by the optimization are a non-unique solution and there are multiple uncertainty configurations that result in the same performance. This problem of uniqueness is well-known in the field of

	p [GPa]		y^* [kg]		Performance [μm]		
	E_1	E_2	m_1	m_2	Model	Tuned Model	Tuned HW
NomMod	72	72	28.61	0.0	263.87	259.07	317.19
AOMod	72	72	0.0	0.0	263.87	263.87	290.83
WCMod	64.8	79.2	81.04	0.0	306.86	273.32	346.46
OptMod	65.5	76.7	59.92	0.0	290.83	262.81	332.79

(a)



(b)

Figure 4-9: Results of model-only tuning algorithms (a) table of results (b) nominal model (open circle), tuned model (filled circle) and tuned hardware (black x) performance, $\sigma_{HW_0} = 290.83\mu$m (dashed line) and $\sigma_{req} = 280\mu$m (solid line).

model updating [60, 55, 12] and is addressed in a future section.

Hardware Tuning

The hardware simulation is tuned in real time with the barrier steepest descent algorithm (BSD) and simulated annealing (SA). The tuning configuration from the model-only method that resulted in the best performance (in this case, nominal model tuning) is used as the starting point. The resulting tuning parameters, tuned performance and required number of hardware tests are listed in the table in Figure 4-10. Although each method resulted in a different tuning mass configuration, both succeeded in bringing performance below the requirement. However, in order to achieve this goal a large number of tests, more than 2000 in the case of SA, are required.

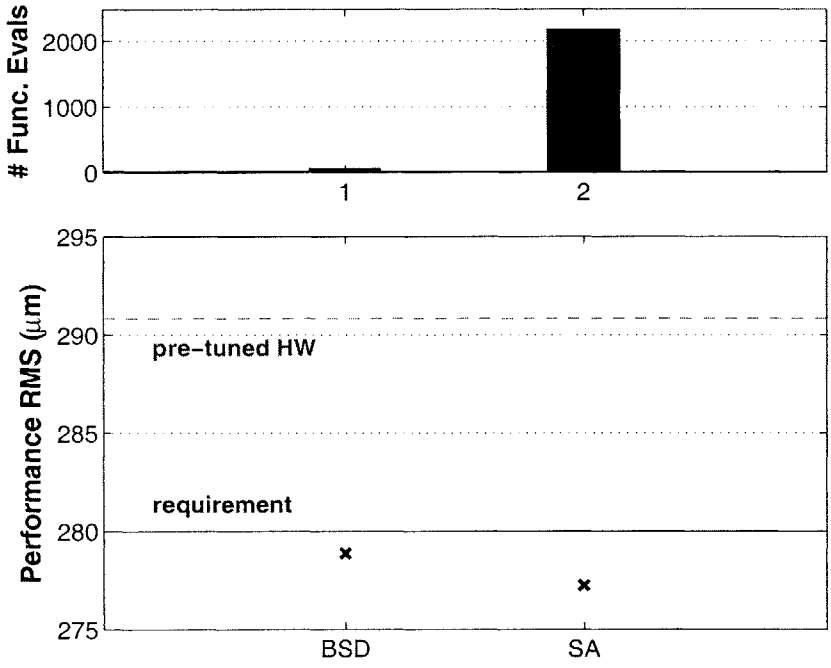
The results are shown graphically in the lower figure, Figure 4-10(b). The bar chart shows the number of function evaluations required for each method. In this example, the barrier method found a working configuration much more quickly than simulated annealing. Simulated annealing takes a long time since it is a random search and is not directed by gradient information. The lower subplot shows the performance results obtained by each algorithm. The solid line is the requirement, and the dotted lines indicates the nominal hardware performances. It is clear that both algorithms succeed in meeting the requirement.

4.3 Isoperformance Updating for Tuning

In the previous section tuning methods ranging from those that use only the model to others that reject the model and tune directly on hardware are explored. It is shown that a trade exists between cost and reliability. The model-only methods are low-cost since only one hardware test is necessary. The tuning optimization is done using only the model and the resulting tuning parameters are then applied to the hardware. However, since these methods rely only on a model that is uncertain and consequently not exact they do not always succeed in tuning the hardware within performance. In fact, for the hardware simulation considered, the tuned hardware performance is

	y^*	[kg]	Performance	#
	m_1	m_2	[μm]	Tests
BSD	0.0	10.40	278.87	61
SA	0.09	12.42	277.24	2176

(a)



(b)

Figure 4-10: Results of Hardware tuning algorithms: (a) table of results (b) number of tests required and tuned performances, $\sigma_{HW_0} = 290.83\mu$m (dashed line), $\sigma_{req} = 280\mu$m (solid line).

actually worse. In contrast, the hardware-only methods disregard the model entirely and perform a real-time tuning optimization directly on the hardware. If theres is enough tuning authority to meet the requirement, these methods are successful, but often require a large number of hardware tests. Such tests are generally expensive as they require a large investment of time and labor.

4.3.1 Model Updating

One way to balance this trade between reliability and cost is to use a hybrid method that takes advantage of available performance data from the hardware to guide the model tuning. One approach is to use the hardware data to update the model such that its predictions match the hardware performance. Then the updated model can be used to run tuning optimizations for application to the hardware. The concept of model updating from data is not a new one and many different techniques can be found in the literature [93]. One problem that is common to all methods is non-uniqueness of the updating solutions. The updated model is generally quite sensitive to the parameters chosen to do the updating [12] and although the updated model may match the test data for one configuration, there is no guarantee that the performance predictions will track actual performance across multiple configurations. If incorrect model updating solutions are used for the tuning optimization then the optimal tuning parameters will not tune the hardware as predicted. In fact, this situation has already been observed in the optimized model tuning technique described previously. The model uncertainty parameters are optimized such that the model prediction matches the hardware. However, the tuning parameters that result from tuning the updated model do not always successfully tune the hardware. The method fails because there are multiple solutions to the updating optimization so although the updated model may predict the same performance these predictions do not track the hardware as other parameters are changed.

One approach to solve the problem of solution uniqueness is to include configuration tracking in the updating procedure. In a recent work, Howell develops a general updating methodology along with a metric for evaluating how well configura-

tion changes in the updated solution track analogous changes in the real system [57]. Cha and Pillis use the idea of configuration changes to perform model updating by adding known masses to the physical structure to obtain a new set of measurements. These new measurements are used in conjunction with previous data to correct the mass and stiffness matrices of the model [25].

These ideas are readily applied to the tuning problem posed in this chapter with a few modifications. The updating method of Cha and Pillis is developed to improve the frequency and mode shape predictions of the model through manipulation of the mass and stiffness matrices. In the tuning application it is only necessary to find tuning parameters that successfully tune the model; it is not required that the model be accurately updated. Also, in this application the data used for updating is a single performance metric and not a set of modes and frequencies, and the uncertainty model is parametric and assumed to be accurate. Therefore it is desirable to update the uncertainty parameters and not operate directly on the mass and stiffness matrices. However, the idea of tracking the hardware and model through configuration changes is key in developing a tuning algorithm that is both successful and low cost.

The hybrid method, referred to as isoperformance tuning, exploits a design tool called isoperformance to reduce the uncertainty space by considering the changes in the performance data as the hardware is tuned. It is an iterative, cooperative process in which tuning optimizations on the model guide the hardware tests, and the resulting data further refines and guides the model tuning optimizations.

4.3.2 Isoperformance

Isoperformance is a methodology developed by deWeck for multi-objective design and analysis of complex, high-performance systems [36]. Instead of fixing the design costs or resources *a priori* and optimizing for system performance within those constraints, the isoperformance methodology constrains the performance and searches for a family of designs that achieve this performance. The idea is that the "optimal" family of designs can then be evaluated with respect to other considerations such as cost and risk. In the thesis [36], deWeck develops and compares three methods of finding

an isoperformance set for both bivariate (two design parameters) and multi-variable design problems.

In this thesis the isoperformance methodology is applied to the hardware tuning process. Instead of using isoperformance to obtain a design, the methodology aids in finding a set of uncertainty parameters that, when used in the model, predict the actual hardware performance. In order to locate the isoperformance set for a system with two uncertainty parameters and one performance objective, deWeck's gradient-based contour following algorithm is employed. This algorithm traces out an isoperformance contour using performance gradient information. First, a gradient-based optimization is used to find an initial point on the contour. In fact, the uncertainty values obtained through optimized model tuning could be used for this purpose. A neighboring point on the contour is then found by taking a step in a direction tangential to the contour. The tangent direction is obtained through a singular value decomposition of the performance gradient with respect to the uncertain parameters. The contour following algorithm takes advantage of the fact that there are only two uncertain parameters, but is extensible to the n-dimensional case. The extended algorithm is called tangential front following, and detailed descriptions of it and contour following are found in [36].

4.3.3 Tuning Algorithm

If the dominant source of uncertainty is parametric and the uncertainty model is well-known and bounded, then isoperformance can be applied to the model updating problem to reduce the uncertainty set. The isoperformance tuning algorithm is given in Figure 4-11 for reference.

To begin, there is one data point from the hardware available, the performance of the untuned hardware, denoted σ_0. Gradient-based contour following is applied over the uncertainty bounds to obtain the initial isoperformance set, P_{iso_0}:

$$P_{iso_0} = \{\vec{p} \in P | \sigma(\tilde{x}, \vec{y}_0, \vec{p}) = \sigma_0\} \tag{4.17}$$

Data: initial iterate, p_0, performance requirement, σ_{req}, uncertainty model, tuning constraints, initial HW performance, σ_0
Result: optimal design variables
begin
  Obtain initial isoperformance set, P_{iso_1}
  Initialize: $k = 1$, $\sigma_{HW} = \sigma_0$ **while** $\sigma_{HW} \leq \sigma_{req}$ **do**
    for *each iso-segment in* P_{iso_k} **do**
      Find uncertainty bounds, B
      Run AO tuning on model over uncertainty space
      Obtain tuning configuration $\vec{y}_k$
      if $|\vec{y}_k - \vec{y}_{k-1}|_2$ *is small* **then**
        Discard $\vec{y}_k$
        Choose a new tuning configuration, $\vec{y}_k$, randomly
      end
    end
    for *each* $\vec{y}_k$ **do**
      Set tuning parameters on hardware
      Test hardware, obtain performance data, σ_{HW}
      if $\sigma_{HW} \leq \sigma_{req}$ **then**
        Stop! tuning is successful
      end
    end
    Search along P_{iso_k} for intersections with new HW configuration performance: $P_{iso_k} = J_{iso}(\tilde{x}, \vec{y}_k, \vec{p}) \cap P_{iso_{k-1}}$
  end
end

Figure 4-11: Isoperformance tuning algorithm.

This contour is a subset of the uncertainty space, P, and consists of all the uncertainty values that, when applied to the model, yield a performance prediction equal to the nominal hardware performance. Since the parameters in $\vec{p}$ are the only source of model uncertainty the real hardware parameters must lie somewhere along this contour. Therefore, the bounds on the uncertainty space can be reduced by finding the minimum and maximum values of each of the parameters in P_{iso_0}. In some cases, the isoperformance line (iso-line) is a single open or closed contour, but it is also possible for there to be multiple isoperformance contours in a given space. Each individual contour is referred to as an isoperformance segment (iso-segment). This situation is handled in the algorithm (Figure 4-11) by looping over the iso-segments and creating subsets of the uncertainty space for each one.

Robust tuning with anti-optimization (Equation 4.12) is run on the model over the reduced uncertainty space(s), and the resulting tuning parameters are applied to the hardware to obtain new performance data, σ_{HW_k}. The subscript k indicates the iteration number. If the hardware performance, J_k, is within requirement then the tuning is successful and the algorithm ends. Otherwise, the uncertainty space is still too large to tune robustly, and the new performance data is used to reduce the space further by searching for intersections in the uncertainty parameter space between the original isoperformance line and that defined by the newly acquired data. Although the tuning parameters, and consequently, the hardware performance, have changed, the values of the uncertainty parameters have not. Therefore, it is guaranteed that the iso-lines intersect somewhere in the uncertainty space. Conceptually, it is convenient to think of two iso-contours crossing, but in the implementation it is not necessary to trace out the second curve. Instead, it is sufficient to evaluate the model prediction at each uncertainty point in P_{iso_0} with the new tuning parameters and to retain those points for which the predicted performance is equal to the new hardware performance within a set tolerance, ϵ:

$$P_{iso_k} = \left\{ \vec{p} \in P_{iso_{k-1}} \,|\, |\sigma\left(\tilde{x}, \vec{y}_k, \vec{p}\right) - \sigma_{HW_k}| \leq \epsilon \right\} \tag{4.18}$$

The new isoperformance set, P_{iso_k} is a subset of P_{iso_0} and allows a further reduction of the uncertainty bounds. This new uncertainty set replaces P_{iso_0} and the algorithm begins another iteration. This process of robust tuning and uncertainty reduction continues until the uncertainty space is small enough that robust tuning is successful.

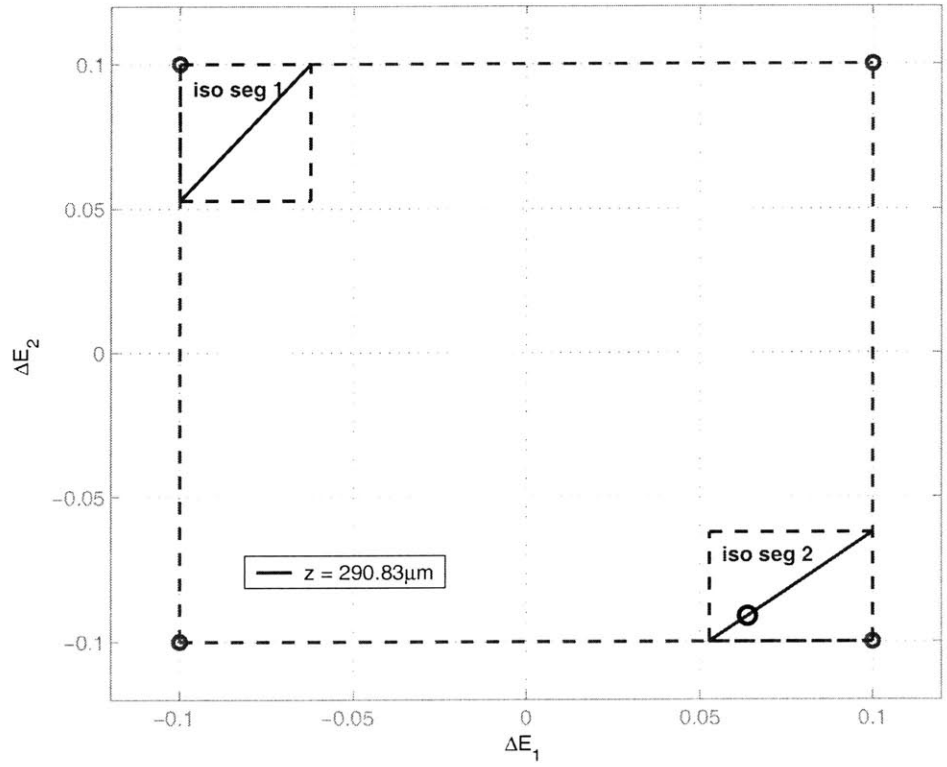
It is possible that the tuning configuration resulting from the robust model tuning does not successfully tune the hardware or produce a new iso-contour that significantly reduces the uncertainty space. For example, the new iso-line may only be slightly different from the previous one, so that all the points in $P_{iso_{k-1}}$ evaluate to the new performance within the given tolerance, ϵ. This situation is handled in the algorithm by evaluating the norm of the difference between the new tuning configuration and the previous one before performing the hardware test. If this value is small indicating that

the new tuning configuration is very close to the previous one, then the optimized tuning configuration is disregarded and a new one is chosen randomly. It is also possible to perform a new optimization in which the goal is to find a set of tuning parameters that result in the greatest reduction of the uncertainty space. It is not necessary to implement the alternate method in this thesis due to the form of the development model. However, such a technique may be useful for more complex design problems and should be considered in future work.

4.3.4 Examples

The isoperformance tuning algorithm is best illustrated through an example and is applied to the problem of tuning the SCI development model considered in the previous section on the same hardware model. The uncertainty space ranges $\pm 10\%$ about the nominal for both parameters, E_1 and E_2, and the required performance is 280μm. The nominal hardware performance is 290.93μm so tuning is necessary on this system. Although the baseline tuning on this system is successful (see Table 4.3) applying AO tuning over the entire uncertainty space does not result in a tuning configuration. In fact, the optimization chooses to not tune the structure at all since any additional mass increases the performance at one of the uncertainty vertices. The isoperformance tuning algorithm is therefore applied to reduce the uncertainty space and bring the hardware performance within requirements.

The results of applying the isoperformance tuning algorithm to this problem are given in both graphical and tabular form in Figure 4-12. The tuning parameters, uncertainty bounds and performance data for each hardware test are listed in the Table 4-12(b). The first test is the hardware in its nominal configuration with no tuning parameters. The uncertainty bounds for this iteration are drawn on the accompanying Figure 4-12(a) in dashed lines. The isoperformance contour for this performance is drawn in a solid lines and consists of two labelled segments. The segments are mirror images of each other due to the model's symmetry. Since there are two distinct iso-segments it is possible to reduce the uncertainty space significantly into two much smaller regions as indicted by the dotted lines around the contours.



(a)

Test	$\vec{y}$ [kg]		Source	Uncertainty Bounds p_1		p_2		σ_{HW}
#	m_1	m_2		Δ_L	Δ_U	Δ_L	Δ_U	[μm]
1	0.0	0.0	nominal HW	-0.1	0.1	-0.1	0.1	290.83
2	81.04	0.0	AO tune: iso-seg 1	-0.1	0.05	-0.62	0.1	346.47
3	0.0	81.05	AO tune: iso-seg 2	0.05	0.1	-0.1	0.62	264.07

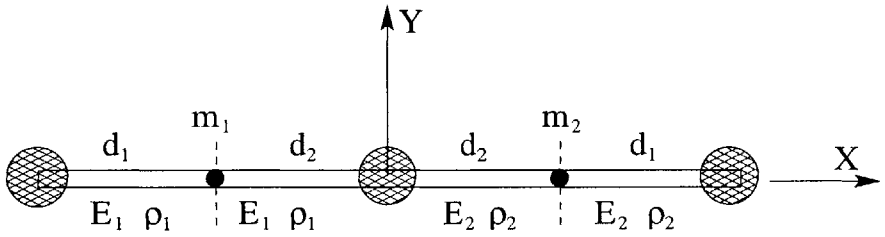
(b)

Figure 4-12: Isoperformance tuning, $\vec{p} = \begin{bmatrix} E_1 & E_2 \end{bmatrix}$: (a) uncertainty space and isoperformance contours, actual hardware parameters are denoted with black circle (b) table of hardware tests, uncertainty bounds and results.

Performing an AO tuning optimization about the uncertainty space of the first iso-segment results in the tuning configuration listed in the table under Test 2. These parameters are set on the hardware and a new performance measure of 346.47μm is obtained. This tuned configuration does not meet requirements, and in fact, is worse than the nominal hardware performance indicating that the hardware parameters are most likely on the second iso-segment. Performing an AO tuning optimization on the other iso-segment results in a tuning configuration that is symmetric to that found in Test 2. This tuning configuration successfully tunes the hardware performance to below the requirement ending the process after only three hardware tests. Comparing these results to the baseline tuning results in Table 4.3 it is interesting to note that the isoperformance tuning methodology results in a similar tuning configuration and leads to a hardware performance that is very close to the baseline tuned performance.

Table 4.4: Uncertainty parameters for SCI development model.

Name	Description	p_0
E_1	Young's Modulus of truss 1 & 2	72 [GPa]
ρ_1	material density of truss 1 & 2	2900 [kg\m^3]



The development model tunes easily with isoperformance tuning due to the symmetry in the model. The iso-contours are distinct mirror images of each other allowing a dramatic reduction of the uncertainty space from only one hardware measurement. In order to demonstrate the latter half of the algorithm, the search for intersecting subsets of the isoperformance space, a slight variation of the development model is considered. In this example the uncertainty parameters are Young's Modulus, E_1, and density, ρ_1, of the beams in truss segments 1 and 2, as listed in Table 4.4 and shown in the accompanying figure. In addition, the uncertainty range is increased to $\pm 50\%$ about the nominal parameter values. Since the uncertainty model has changed

it is necessary to generate a new hardware simulation to test the isoperformance tuning methodology. The hardware simulation is obtained as previously described and tuned using full knowledge of the uncertainty parameters to obtain a baseline tuning configuration for comparison. The hardware model parameters and nominal and tuned performance are listed in Table 4.5. The performance requirement in this case is 295μm.

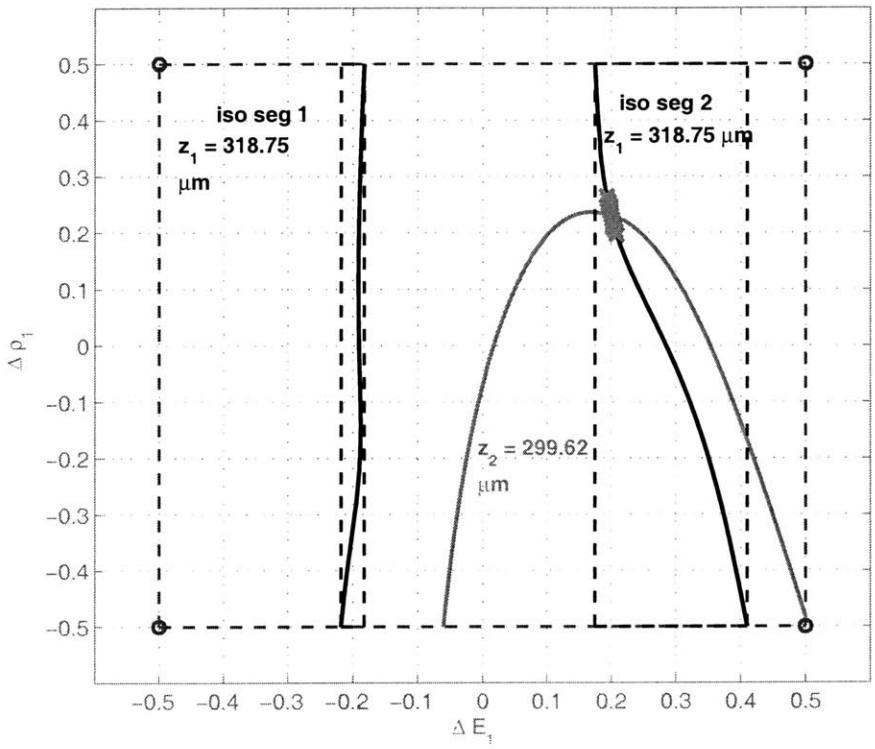
Table 4.5: Hardware Model Data: Example 2.

Tailoring, x	0.0486	[m]	0.0581	[m]
Uncertainty, p	86.45	[GPa]	3440	[kg\m^3]
Tuning, y	0.0	[kg]	60.01	[kg]
HW Performance σ_{HW}		318.75		[μm]
Tuned Performance σ_{HWt}		289.98		[μm]

The results of applying the isoperformance tuning methodology to this problem are given in Figure 4-13. The hardware tests are described in the table (Figure 4-13(a)), and the corresponding isoperformance curves and uncertainty space reduction are shown graphically in Figure 4-13(b). The first hardware test is performed in its nominal configuration with the tuning masses set to zero. At 318.76μm the untuned hardware performance is significantly higher than the requirement. The dashed lines in the plot represent the boundaries of the uncertainty space. The solid lines, labelled "iso seg 1" and "iso seg 2", are the isoperformance contours of the nominal, untuned, hardware performance. Note that there are two distinct contours, but in this case each contour covers a significant portion of the uncertainty space. The reduced uncertainty bounds are shown around the contours in dashed lines and listed in the table. Tracing these iso-contours reduces the bounds on E_1 dramatically, but does not affect the bounds on ρ_1. The tuning configurations for the second and third hardware tests result from an AO tuning optimizations on the model over the blue uncertainty spaces. The second test results in a hardware performance that is worse than nominal, while the third test improves the hardware performance to 299.62μm. These results indicate that the hardware uncertainty parameters are most likely located on the second iso-segment.

Test	$\vec{y}$ [kg]		Source	Uncertainty Bounds p_1		p_2		σ_{HW}
#	m_1	m_2		Δ_L	Δ_U	Δ_L	Δ_U	[μm]
1	0.0	0.0	nominal HW	-0.5	0.5	-0.5	0.5	318.76
2	156.77	0.0	AO tune: iso-seg 1	-0.218	0.183	-0.5	0.05	397.74
3	0.0	134.12	AO tune: iso-seg 2	-0.715	0.41	-0.5	0.05	299.62
4	0.0	69.23	AO tune over P_{iso_1}	-0.193	0.208	0.196	0.265	290.00

(a)



(b)

Figure 4-13: Isoperformance tuning, $\vec{p} = \begin{bmatrix} E_1 & \rho_1 \end{bmatrix}$ (a) uncertainty space and isoperformance contours, actual hardware parameters are denoted with black circle (b) table of hardware tests, uncertainty bounds and results.

Since neither tuning configuration results in successfully tuned hardware the two iso-segments are searched for intersections with either of the hardware performance values obtained in these two tests. The first tuning configuration does not yield any intersecting points, further supporting the conclusion that the hardware uncertainty parameters do not lie on the first iso-segment. The search for intersections with the second tuning configuration performance bears fruit and a new isoperformance set is extracted from the original. The new set, P_{iso_1} is indicated with gray x's in the plot. For visualization purposes, the entire isoperformance line for this performance value is traced with a light gray line. It is clear that there is an intersection between the first and second iso-lines precisely at the value of the hardware uncertainty parameters (marked by a black circle on the plot and listed in Table 4.5). The two contours intersect in only one location yielding greatly reduced uncertainty boundaries. The fourth, and final, tuning configuration is found by performing the AO tuning optimization over this smaller uncertainty space (bounded by dashed gray lines) and successfully tunes the hardware performance to a value below 295μm.

4.3.5 Comparison of Tuning Methods

The tuning results presented thus far are all for one particular hardware simulation. In order to fairly assess the performance of the methods and compare them, fifty hardware models are generated and tuned with the eight techniques: nominal model, AO tuning, optimized model, worst-case model, BSD hardware tuning, SA hardware tuning and isoperformance tuning. The model-only methods are tried first and then the best performing tuning configuration is used as a starting point for the hardware tuning process. The methods are compared based on reliability, how often the hardware model tunes successfully, and efficiency, how many hardware tests are required. The results of the tuning simulations are presented in Table 4.6. The tuning methods are sorted from top to bottom, first by the number of successful tuning trials, and then by the average number of hardware tests required. Therefore, the method that performs best in both categories, isoperformance tuning, is listed first.

The model-only methods are grouped at the bottom of the table because they are

Table 4.6: Tuning results on fifty hardware simulations: performance and number of required tests.

	# of Sims		# Hardware tests	
Method	Success	Failure	Maximum	Average
Iso Tuning	50	0	4	2.5
Hardware: BSD	50	0	231	46.2
Hardware: SA	50	0	2618	790
Worst-Case Model	29	21	2	2
Nominal Model	29	21	2	2
Optimized Model	24	26	2	2
AO Tuning	0	50	2	2

successful on only 50% of the trials. The only exception is AO tuning, which does not successfully tune any of the simulations. It is worthwhile to note that, although the model-only methods failed in the example discussed previously, there are some hardware realizations for which these methods are adequate. Since each of these methods require only two hardware tests, the initial test to assess nominal hardware performance and a second one to try the tuning configuration, there is little cost in starting the tuning process with these methods as much time and effort can be avoided if they succeed. If the model-only method fails, the tuning configuration is a good starting point for a hardware tuning scheme. Optimized model tuning and isoperformance tuning are particularly compatible in this way since the uncertainty parameters used for the model tuning also serve as an initial point on the isoperformance contour. It is also interesting to note that the nominal model and worst-case model tuning methods result in the exact same numbers of successes. It turns out that these methods succeed and fail on the same hardware realizations indicating that these methods are interchangeable for this particular problem.

The hardware tuning methods and isoperformance tuning, on the other hand, are consistently successful across the entire sample space. There is quite a difference in the number of hardware tests that are required by the three methods, however. Real-time simulated annealing is by far the least efficient requiring an average of 790 hardware tests across the sample space. This result is not surprising since the method is a random search of the tuning space. It is equivalent to simply trying random tuning

configurations until one works. The fact that such a large number of tests are required for such a simple problem (only two tuning and two uncertainty parameters) indicates that this approach is fairly ill-suited for this application. The barrier steepest-descent real-time optimization performs much better than SA, requiring an average of 46 hardware tests to tune successfully. The maximum value of tests is still pretty high at 231, however, and is expected to increase with the number of tuning and uncertainty parameters in the problem.

The isoperformance tuning algorithm, a hybrid of model and hardware tuning techniques, stands out as the clear winner. Like the hardware-only methods it successfully tunes each and every one of the hardware realizations, but does so with only a few hardware tests. The maximum number of tests required by the isoperformance tuning method is 4 and the average across the sample space is 2.5. These statistics are a factor of ten less than BSD and a hundred times less than SA. It is true that the iso-lines in this problem are particularly well-suited for this tuning methodology as discussed previously, but the additional example considered shows that even if the isoperformance lines are less favorable the method performs very well. As the complexity of the model and tuning problem increases, it is expected that the model-only methods will fail more often and that the hardware tuning methods will require even more tests to find a successful tuning configuration. Therefore, although the model-only methods are attractive for their simplicity and low-cost, and the hardware methods are attractive for their success rate, neither method is really a practical solution as there is always the chance of failure or prohibitive expense. In contrast, the isoperformance tuning method is able to consistently provide successful tuning solutions with only a small number of hardware tests.

4.4 Summary

In this chapter, the concept of dynamic tuning is defined as the process of adjusting hardware to bring the system performance within requirements. The tuning process is formalized as an optimization, and guidelines for choosing appropriate tuning param-

eters are discussed. The optimization is run on the worst-case uncertainty realizations of the SCI development model PT and RPT designs to demonstrate the performance improvement that results from tuning the hardware. Three different optimization algorithms are explored, and a physical interpretation of how tuning masses reduce the performance variance in these models is provided. The design regime plot introduced in the previous chapter is updated to include tuned PT and RPT designs, and it is shown that adding a tuning step to the design process extends the uncertainty tolerance of the tailoring techniques.

The problem of tuning in practical application is also considered. In order to tune the hardware successfully additional information about the values of the uncertainty parameters is required. Eight different tuning schemes are presented and demonstrated on a hardware simulation of the SCI development model. The tuning methods range from model-only methods that use little or no hardware data, to hardware-only methods that disregard the uncertain model altogether. The methods are evaluated based on reliability and efficiency. A hybrid tuning method that exploits a technique from multi-objective optimization, isoperformance, is shown to be both consistently successful and low-cost in that a small number of hardware tests are required to tune successfully.

Chapter 5

Robust Performance Tailoring with Tuning

Dynamically tuning hardware that does not meet specification may be successful in bringing the performance within requirements. This added step takes advantage of the additional knowledge provided by hardware data to meet aggressive requirements that can not be met through modeling and robust design alone. However, since tuning is performed after the hardware is built, options are limited to only changes that do not disrupt the integrity of the system, and therefore, it is not always be possible to tune the hardware to meet performance. If this is the case, then the engineer gains the knowledge that the system does not meet requirements, but is left without easy options to pursue.

A better approach is to explicitly plan for dynamic tuning from the beginning of the design process. The design is tailored for both robustness to uncertainty and tunability. Tuning is added to the design optimization as a complement to robust tailoring and shares in the uncertainty management. This approach is named Robust Performance Tailoring for Tuning (RPTT), and is the major contribution of this thesis.

In the following chapter the RPTT methodology is developed as an optimization problem in various forms. It is applied to the SCI development model using SA and SQP optimization algorithms, and the results are compared for algorithm performance

and efficiency. The resulting RPTT design is compared to the PT and RPT designs presented in the previous chapters. The modal energy distributions are explored to gain physical insight into the tailored and tuned designs. The RPTT methodology is run over a range of uncertainty bounds, and the design regimes plot presented in the previous chapters is updated to include the RPTT design. It is shown that RPTT further extends the uncertainty level that can be tolerated for a given performance requirement. Finally, random hardware simulations are generated and the entire tailoring-tuning process is applied using the PT, RPT and RPTT designs. It is shown that only the RPTT designs can consistently meet aggressive performance requirements over a large range of uncertainty bounds.

5.1 RPTT Formulation

Robust Performance Tailoring for Tuning extends Robust Performance Tailoring by designing the system to be tunable across the entire uncertainty space. Recall that the tailoring process chooses design variables, $\vec{x}$, to optimize the predicted performance of the final system. In RPT, $\vec{x}$ is chosen such that the performance predictions are insensitive to the uncertainty parameters, $\vec{p}$, given fixed, nominal values for the tuning parameters, $\vec{y}$. The key feature of RPTT is that the prediction accounts for the fact that the tuning parameters can be adjusted after the hardware is built and the uncertainty is fixed (and known). The goal of the optimization is to choose $\vec{x}$ to find a design that is *tunable* across all possible hardware realizations. In the implementation, the tuning parameters change depending on the uncertainty parameters. This consideration of future tuning adds extra degrees of freedom to the problem in the form of additional $\vec{y}$ realizations, and, as a result, the optimization is less constrained and may have better solutions. Tailoring for tuning idea is similar to an optimization problem presented by Scokaert and Mayne for application to receding-horizon model predictive control (MPC)[104]. The MPC formulation accounts for the fact that future control decisions are made with more information than is currently available. Scokaert's min-max problem introduces the notion of feedback into the

MPC optimization by allowing a different control sequence for each realization of the disturbance.

The RPTT cost function is a weighted sum of a tailoring for tuning objective, J_{TT}, and a robust performance tailoring objective, J_{RPT}:

$$J_{RPTT} = (1-\alpha)J_{TT} + \alpha J_{RPT} \tag{5.1}$$

The weighting parameter, α, allows adjustment of the relative weight between tuning authority and robustness. If $\alpha = 0$, the design is tailored for maximum tuning authority, and if $\alpha = 1$, J_{RPTT} reduces to an RPT optimization. Optimizing for only J_{TT} could lead to a design that is tunable, but is highly sensitive to uncertainty and therefore relies heavily on hardware tuning for mission success. Since hardware tuning requires additional time and resources, the ideal structure is one that will most likely meet performance, but can be tuned to meet requirements in the event that the hardware realization falls short. Therefore, it is preferable to find a design that is both robust to uncertainty and tunable, so that the uncertainty compensation is shared between robust design and hardware tuning.

One way to formulate the two objectives is with nested optimizations:

$$\min_{\vec{x}} \left\{ (1-\alpha) \overbrace{\left[\max_{\vec{p}\in P} \min_{\vec{y}\in Y} f(\vec{x},\vec{y},\vec{p}) \right]}^{J_{TT}} + \alpha \overbrace{\max_{\vec{p}\in P} f(\vec{x},\vec{y}_0,\vec{p})}^{J_{RPT}} \right\} \tag{5.2}$$

$$s.t.\ \ g(\vec{x}) \leq \vec{0}$$

The tailoring for tuning cost is a max-min optimization in which the tuning optimization, Equation 4.2, is performed at each of the uncertainty vertices. In effect, J_{TT}, is the worst-case tuned performance over the uncertainty space. The anti-optimization cost, Equation 3.4, is used as the robust performance tailoring objective. Note that the difference between J_{TT} and J_{RPT} is that, in the robust objective the tuning parameters are fixed at their nominal value, $\vec{y}_0$ and no tuning optimization is performed. The outer tailoring optimization is performed only over the tailoring parameters, $\vec{x}$,

which are constrained by $\vec{g}$. The uncertainty bounds and tuning parameter constraints are handled in the inner optimizations.

Although Equation 5.2 is conceptually the simplest formulation of the RPTT problem, it is rather cumbersome to implement. At each major iteration of tailoring parameters a tuning optimization is performed at each of the uncertainty vertices. Therefore a problem with only two uncertainty parameters requires four tuning optimizations each time the objective function is evaluated. Such a nested optimization is computationally inefficient. In addition, similar to the min-max AO objective (Equation 3.4), it is not clear how to derive analytical gradients for this objective function. The tuned performance is a function of the uncertainty and tuning parameters, but the tailoring parameters are the only design variables in the outer optimization.

One way to improve the computational efficiency is to move the tuning parameters to the outer optimization, eliminating the need for the nested tuning optimization:

$$\min_{\vec{x},\vec{y_i}} \left\{ (1-\alpha) \overbrace{\max_{\vec{p}\in P} f\left(\vec{x},\vec{y_i},\vec{p}\right)}^{J_{TT}} + \alpha \overbrace{\max_{\vec{p}\in P} f\left(\vec{x},\vec{y}_0,\vec{p}\right)}^{J_{RPT}} \right\} \tag{5.3}$$

$$s.t. \;\; \vec{g}\left(\vec{x},\vec{y_i}\right) \leq 0$$

$$\forall i = 1 \ldots n_{p_v}$$

Equation 5.3 is equivalent to Equation 5.2 in that the tailoring for tuning objective is the maximum tuned performance over the uncertainty space. The nested tuning optimization is removed by expanding the design variables to include *multiple* sets of tuning parameters, $\vec{y_i}$, one set at each uncertainty realization. For example, consider a problem with two tailoring parameters, two tuning parameters and two uncertainty parameters. If the problem is locally-convex so that the vertex method of uncertainty propagation can be used, then there are four uncertainty vertices to consider, and the design variables for the optimization in Equation 5.3 are:

$$x_{dv}^T = \left[\begin{array}{cccccccccc} x_1 & x_2 & y_{11} & y_{12} & y_{21} & y_{22} & y_{31} & y_{32} & y_{41} & y_{42} \end{array} \right] \tag{5.4}$$

where the notation y_{ij} indicates the j^{th} tuning parameter at the i^{th} uncertainty vertex. Keep in mind, that the $\vec{y}_i$ vectors are only the tuning configuration at the vertices of the uncertainty space and are not necessarily the final values of the tuning paramters. The hardware tuning process discussed in Chapter 4 must still be employed if the hardware realization does not meet requirements. However, the assumption is that if $\vec{x}$ is chosen during tailoring such that the system is tunable at the uncertainty vertices then the actual hardware will be tunable across the entire uncertainty space. In this formulation, J_{TT} and J_{RPTT} are almost identical except that the tuning parameters are allowed to change with the uncertainty vertices in J_{TT}.

The formulation in Equation 5.3 improves the computational efficiency of the tailoring optimization since tuning optimizations are no longer required at each evaluation of the objective function. However the optimization is still a combination of min-max formulations, and it is unclear how to obtain the analytical gradients. It is possible to further simplify the RPTT formulation by posing it as a simple minimization problem similar to the minimization form of anti-optimization (Equation 3.5):

$$\begin{aligned} \min_{\vec{x},\vec{y}_i,\vec{z}} & \left\{ (1-\alpha)\overbrace{z_1}^{J_{TT}} + \alpha\overbrace{z_2}^{J_{RPT}} \right\} \\ s.t \;\; & \vec{g}\left(\vec{x},\vec{y}_i\right) \leq \vec{0} \\ & h_{1i}\left(z_1,\vec{x},\vec{y}_i,\vec{p}_i\right) \leq 0 \\ & h_{2i}\left(z_2,\vec{x},\vec{p}_i\right) \leq 0 \\ & \forall i = 1 \ldots n_{p_v} \end{aligned} \tag{5.5}$$

In this formulation the cost function consists only of the weighted sum of two dummy variables, z_1 and z_2. The tailoring for tuning and robustness metrics are included through the augmented constraints:

$$h_{1i}\left(z_1,\vec{x},\vec{y}_i,\vec{p}_i\right) = -z_1 + f\left(\vec{x},\vec{y}_i,\vec{p}_i\right) \tag{5.6}$$

$$h_{2i}\left(z_2,\vec{x},\vec{p}_i\right) = -z_2 + f\left(\vec{x},\vec{y}_0,\vec{p}_i\right) \tag{5.7}$$

The first constraint, Equation 5.6, requires the tailoring for tuning dummy variable, z_1, to be greater than or equal to the maximum tuned performance over the uncertainty vertices. Equation 5.7 is equivalent to Equation 3.6 from the AO formulation, and requires that z_2 be greater than or equal to the maximum un-tuned performance over the uncertainty space. Since the objective is to minimize a weighted sum of z_1 and z_2, the constraints ensure that these variables are effectively the worst-case tuned and worst-case untuned performances, respectively.

In this form, the analytical gradients of the cost function are easily derived by inspection:

$$\frac{\partial J_{RPTT}}{\partial \vec{x}} = \vec{0} \qquad \frac{\partial J_{RPTT}}{\partial \vec{y}_i} = \vec{0} \qquad \frac{\partial J_{RPTT}}{\partial z_1} = 1 - \alpha \qquad \frac{\partial J_{RPTT}}{\partial z_2} = \alpha \tag{5.8}$$

Constraint gradients are also required for a constrained gradient-based optimization. The gradients of the parameter constraints, $\vec{g}$, depend on the particular design problem under consideration. The augmented constraint gradients include the gradients of the performance with respect to the tailoring and tuning parameters:

$$\frac{\partial h_{1i}}{\partial \vec{x}} = \frac{\partial f\left(\vec{x}, \vec{y}_i, \vec{p}_i\right)}{\partial \vec{x}} \qquad \frac{\partial h_{1i}}{\partial \vec{y}_i} = \frac{\partial f\left(\vec{x}, \vec{y}_i, \vec{p}_i\right)}{\partial \vec{y}_i} \qquad \frac{\partial h_{1i}}{\partial z_1} = -1 \qquad \frac{\partial h_{1i}}{\partial z_2} = 0 \tag{5.9}$$

$$\frac{\partial h_{2i}}{\partial \vec{x}} = \frac{\partial f\left(\vec{x}, \vec{y}_0, \vec{p}_i\right)}{\partial \vec{x}} \qquad \frac{\partial h_{2i}}{\partial \vec{y}_i} = \vec{0} \qquad \frac{\partial h_{2i}}{\partial z_1} = 0 \qquad \frac{\partial h_{2i}}{\partial z_2} = -1 \tag{5.10}$$

In the following section the RPTT optimization is applied to the design of a structurally connected interferometer using the SCI development model. A variety of optimization algorithms are used to solve the problem and the performance of the different formulations are compared for efficiency.

5.2 SCI Development Model

Application of the RPTT methodology to the SCI development model requires all of the elements presented in the previous chapters. The tailoring parameters are the two cross-sectional areas (Table 2.3), the tuning parameters are the design masses

(Table 4.1), and the uncertainty parameters are the Young's Moduli of the two truss arms (Table 3.1), so that the design variable vector is:

$$x_{dv}^T = \begin{bmatrix} d_1 & d_2 & m_{11} & m_{12} & m_{21} & m_{22} & m_{31} & m_{32} & m_{41} & m_{42} \end{bmatrix} \tag{5.11}$$

The design masses are no longer considered as tailoring parameters in this formulation since they are accounted for by the tuning component. The performance metric is the RMS of the OPD between the two collectors and is a function of all the parameters. A constraint equation limiting the total mass of the tailored and tuned system is enforced at each uncertainty vertex:

$$g\left(\vec{x}, \vec{y}_i\right) = \begin{cases} \frac{\pi}{2}\rho L \sum_{j=1}^{2} d_j^2 + \sum_{j=1}^{2} m_{1j} - \bar{M} \\ \vdots \qquad\qquad \forall i = 1 \ldots 4 \\ \frac{\pi}{2}\rho L \sum_{j=1}^{2} d_j^2 + \sum_{j=1}^{2} m_{4j} - \bar{M} \end{cases} \tag{5.12}$$

This constraint couples the tailoring and tuning parameters into the mass equation allowing the optimization algorithm to spend mass where it is most beneficial.

The results presented in this section are for the case of bounded uncertainty within $\pm 10\%$ of the nominal uncertainty values. The weighting on robustness, α, in the RPTT formulations is set to zero to produce a design with maximum tuning authority.

5.2.1 Optimization Algorithms

RPTT designs of the SCI development problem are generated by running the optimizations formulated in Equations 5.3 and 5.5 through the simulated annealing and sequential quadratic programming algorithms. The resulting designs are presented in Table 5.1 along with algorithm performance data. Simulated annealing is run first on the formulation in Equation 5.3 to provide an initial guess to the SQP algorithm. SA is not run with the simple minimization formulation (Equation 5.5) as it is difficult to obtain feasible guesses due to the augmented constraints. SQP is also run with ten randomly-chosen initial guesses to assess the convexity of the solution space.

The results show that nearly all of the SQP runs converged on the same design.

Table 5.1: Algorithm performance: RPTT, $\alpha = 0.0$, $\Delta = 0.1$.

		J^*	iter	fevals	time	x^* [m]		y^*_{WC} [kg]	
Alg.	Form	[μm]	#	#	[min]	d_1	d_2	m_1	m_2
SA	Eq 5.3	216.03	201	8304	5.26	0.0451	0.0388	175.63	0.0774
SQP	Eq 5.5	215.94	23	53	1.77	0.0451	0.0388	175.88	0.0
SQP	Eq 5.3	215.97	13	1001	1.62	0.0452	0.0388	175.66	0.028
MC SQP	Eq 5.5	215.94	414	906	31.47	0.0451	0.0388	175.90	0.0
MC SQP	Eq 5.3	215.94	661	5843	51.82	0.0451	0.0388	175.93	0.0

Since the weighting on robustness is set to zero the optimal cost, J^*, is the tuned performance at the worst-case uncertainty vertex. SA provides a starting point that is very close to the optimal design after 201 temperature iterations and 8304 function evaluations. The SQP algorithm performs nicely when started at the SA design and applied to the problem of Equation 5.5. It converges to a design in only 23 iterations and 53 function evaluations. In contrast, the SQP solution obtained from the formulation in Equation 5.3 is slightly sub-optimal. Note that the algorithm ran for only 13 iterations, but 1001 function evaluations. The algorithm fails to properly converge because it becomes stuck in the line search and the maximum number of function evaluations (1000) is reached. One possible reason for this behavior is that the objective function gradients required for the SQP algorithm are not well-posed for this problem. The MC SQP results are equivalent to those obtained by starting from the SA design indicating that a globally optimal design has been found. MC SQP does find the optimal design when applied to Equation 5.3 since five out of ten trials do converge successfully. The time results show that the combination of SA and SQP finds the optimal design much more quickly (7 minutes) than MC SQP (31.5 minutes).

5.2.2 Comparison to PT and RPT Designs

The optimal tailoring parameters for the RPTT design and its performance in the nominal, and worst-case (untuned) uncertainty configurations are listed in Table 5.2 along with the corresponding design variables and performances for the PT and RPT

Table 5.2: Performance and design parameters for optimal designs.

	Tailoring, $\vec{x}$				Performance, [μm]		Total Mass
	d_1 [m]	d_2 [m]	m_1	m_2	σ_0	σ_{WC}	[kg]
PT	0.030	0.030	0.0	1.934	100.53	1355.50	673.18
RPT	0.0486	0.0581	0.0	0.0	263.87	306.86	827.04
RPTT	0.0451	0.0388	N/A	N/A	158.53	572.66	740.05

(AO) designs. The RPTT design is significantly different from both the PT and RPT designs. In the PT design all of the truss segments are of equal diameter and are at the lower bound, while in the RPT design they are tailored so that the mass is concentrated towards the center of the array with the inner truss diameters are about a centimeter larger that those of the outer truss segments. The RPTT design is opposite the RPT as the inner segments have a smaller cross-sectional diameter than the outer segments, so mass is distributed towards the ends of the array. The tuning parameters are used as tailoring parameters in the PT and RPT optimizations so that the same design variables are used to ensure a fair comparison among the methods.

The PT design has the lowest nominal cost by far (100.53μm), but is very sensitive to uncertainty with a worst case performance of 1355μm. The RPT design is much more robust to uncertainty, and has a much lower worst-case performance value (306.86μm), but sacrifices nominal performance (263.87μm) to achieve this robustness. The RPTT design lies somewhere in between the PT and RPT designs. It has a nominal performance of 158.53μm, higher than that of the PT design, but lower than that of the RPT design. It is not quite as robust as the RPT design, but has a much lower worst-case performance (572.66μm) than the PT design.

Table 5.3: Performance and parameters for tuned worst-case realizations.

	Tuning, $\vec{y}$ [kg]		Performance	Total Mass
	m_1	m_2	σ_t [μm]	[kg]
PT	175.06	0.0	303.17	848.25
RPT	81.04	0.0	273.32	908.08
RPTT	0.0	175.88	215.94	915.94

The tuned performance of the designs is obtained by applying the tuning op-

timization (Equation 4.2) to the model at the worst-case uncertainty vertex. The resulting optimal tuning parameters and tuned performances for the three designs are listed in Table 5.3. In all cases, the tuning mass is concentrated in one arm of the interferometer. The RPTT design is tuned on the positive-x arm while the PT and RPT designs are tuned by adding mass to the negative-x arm. This mirror effect is simply due to the choice of worst-case uncertainty vertex. Since the model is perfectly symmetric there are two vertices that result in the same worst-case performance.

The tuned performance values indicate that the RPTT design can achieve the best performance through a combination of tailoring and tuning. Consider, for example, a performance requirement of 240μm, shown with a solid horizontal line in Figure 5-1. Nominal performance is indicated by circles and tuned performance by triangles. The top of the solid error bars is the worst-case untuned performance, and the dashed error bar denotes the worst-case tuned performance. The PT design meets

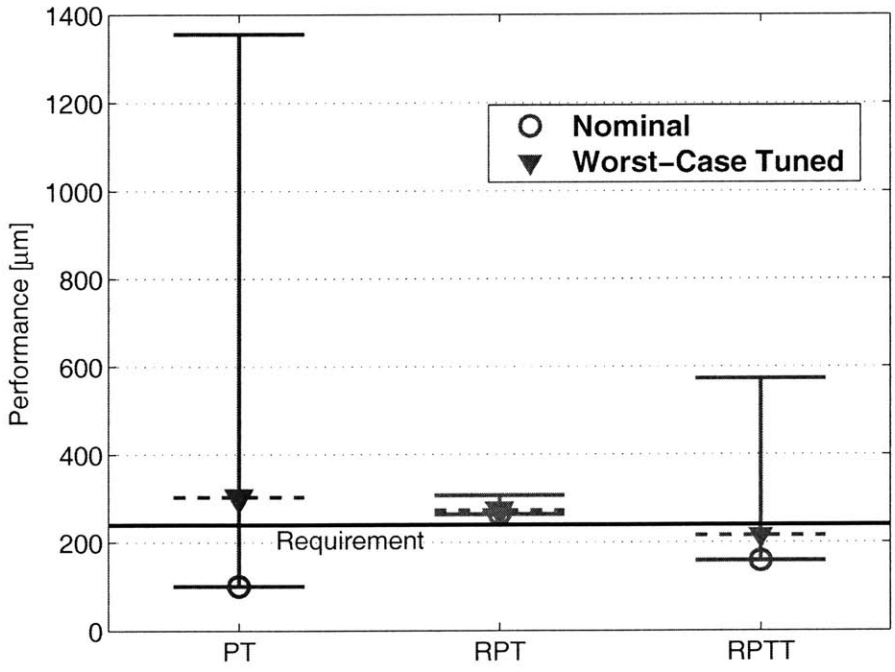


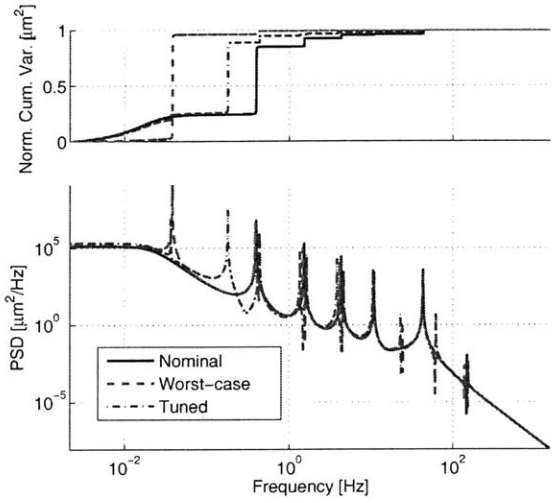
Figure 5-1: Nominal, worst-case and tuned performance for PT, RPT and RPTT designs. Nominal and tuned performances shown with circles and triangles, respectively, and worst-case performance indicated by error bars. The solid black horizontal line indicates a performance requirement of 240 μm.

this requirement nominally with a performance value of 100.58μm, but is far above

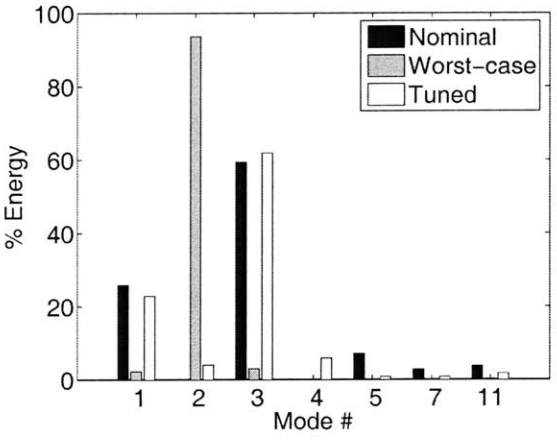
it in the worst-case uncertainty realization at 1355μm. Tuning the PT design at this worst-case vertex improves the performance greatly to 303μm, but does not succeed in bringing the system performance within the requirement. The RPT design is much less sensitive to uncertainty, but sacrifices nominal performance, and consequently tunability, to gain robustness. As a result, it does not meet the requirement in either the nominal (263.87μm) or tuned worst-case (273.32μm) configurations. The RPTT design methodology improves on RPT by tailoring for multiple sets of tuning parameters instead of just one. RPTT sacrifices some robustness for tunability resulting in a worst-case performance of 573μm, higher than that of RPT, but this worst-case hardware realization is tunable to just under 216μm and meets the requirement. In this context, tunability is considered a form of robustness. Although the *design* is somewhat sensitive to uncertainty, the physical hardware is guaranteed to be tunable to below the requirement resulting in a robust *system.*

To understand the physical source of the increased tuning authority consider the energy information provided in Figure 5-2. The output PSDs of the RPTT design in nominal (solid line), worst-case (dotted line) and tuned worst-case (dash-dot line) uncertainty configurations are plotted in Figure 5-2(a). The normalized cumulative variance plot shows that the majority of the energy in the worst-case realization is concentrated in the first bending mode. This result is consistent with the PT and RPT designs. The distribution of energy in the nominal and tuned configurations is similar, but the modal frequencies are much lower in the tuned case due to the additional tuning mass.

The bar chart, Figure 5-2(b), presents the percent of energy accumulated in the critical modes. The nominal uncertainty case is shown by the black bars, the worst-case uncertainty realizations by gray bars and the tuned configurations by white bars. The first three modes are most critical, and the first bending mode contains most of the energy in the worst-case uncertainty situation. The accompanying table (Figure 5-2(c)) lists the modal frequencies, percent energy and absolute RMS of each mode. Note the large increase in energy in Mode #2 in the worst-case realization and the drop in frequency of Mode #3 in the tuned case.



(a)



(b)

Mode #	Nominal f_n [Hz]	Nominal energy %	Nominal σ_z [μm]	Worst-case f_n [Hz]	Worst-case energy %	Worst-case σ_z [μm]	Tuned f_n [Hz]	Tuned energy %	Tuned σ_z [μm]
1	0.022	25.92	41.10	0.022	2.38	13.61	0.022	22.86	49.36
2	0.038	0.00	0.00	0.038	93.63	536.17	0.036	4.10	8.84
3	0.395	59.39	94.16	0.386	3.07	17.58	0.178	61.93	133.73
4	0.431	0.00	0.00	0.440	0.16	0.92	0.434	5.90	12.75
5	1.517	7.20	11.42	1.461	0.23	1.32	1.345	0.89	1.92
7	4.275	2.87	4.55	4.121	0.09	0.52	3.801	0.88	1.90
11	43.04	3.83	6.07	43.14	0.29	1.64	42.74	1.76	3.80

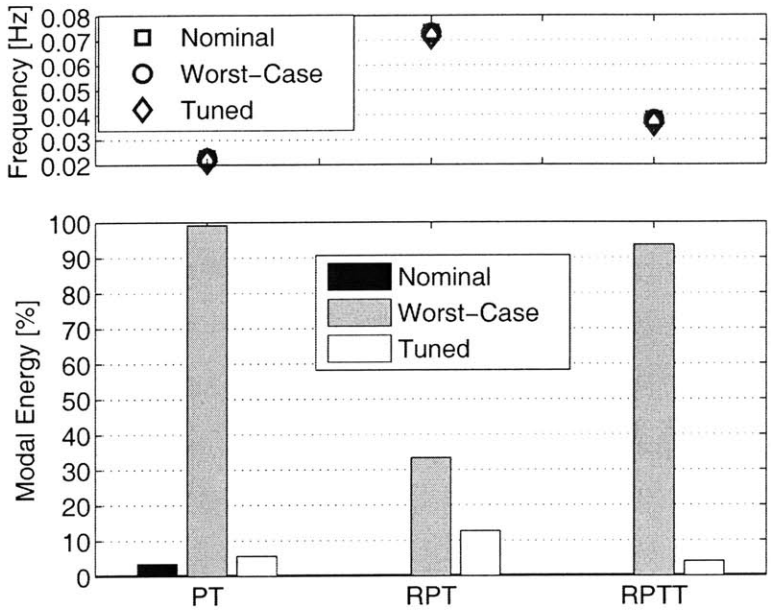
(c)

Figure 5-2: Modal energy breakdown for RPTT design ($\Delta = 0.1$ and $\alpha = 0.0$) for nominal uncertainty, worst-case uncertainty, and tuned configurations (a) output PSDs: nominal (solid blue), worst-case (dashed green), tuned (dash-dot red) (b) % energy comparison (c) results table.

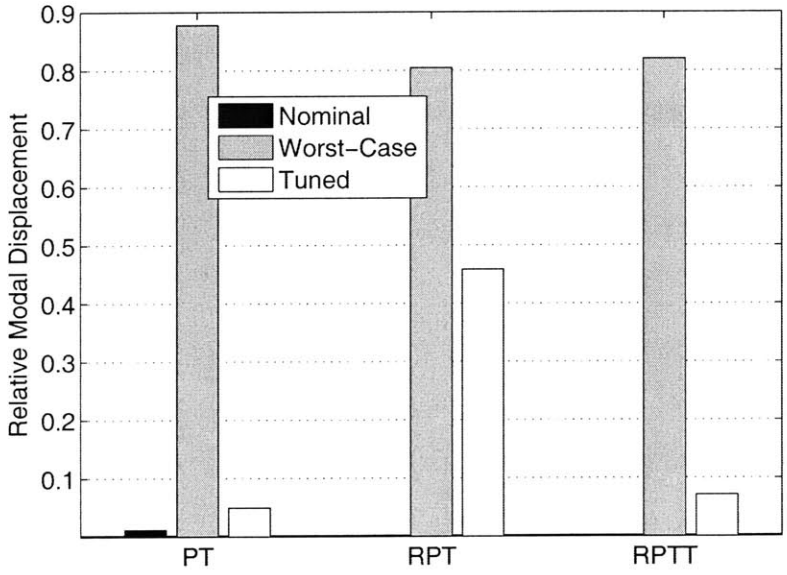
The data in Figure 5-2 indicate that the first bending mode at 0.038 Hz is the focus of the tailoring and tuning mechanisms. Recall from discussions in previous chapters that this mode is critical for all of the designs. A comparison of the modal characteristics of this mode among the PT, RPT and RPTT designs in the nominal, worst-case and tuned configurations is presented in Figure 5-3 to provide insight into the underlying physics.

The natural frequency of the first bending mode in all three designs is shown in the upper subplot of Figure 5-3(a). The frequency of the nominal, worst-case and tuned configurations are marked with circles, squares and diamonds, respectively. The symbols are difficult to distinguish from one another because, for a given design, the frequency of the mode does not change appreciably from one configuration to the next. However, there is quite a difference in the frequency of this mode among the three designs. The natural frequency in the AO design (0.072Hz) is over twice that of the PT design (0.022Hz). Recall from the discussion in Chapter 3 that it is the stiffening of this mode that gives the RPT design its robustness to uncertainty. However, a side effect is that the RPT design is robust to the tuning parameters, and little can be done to improve the performance at a given uncertainty realization. The first bending mode frequency in the RPTT design is between that of the PT and RPT designs, but is closer to the PT design (0.036Hz) so that tuning authority is preserved.

The lower subplot in Figure 5-3(a) is a bar chart of the percent energy in the first bending mode for all three systems in the three configurations. The nominal configuration is shown in black, worst-case in gray and tuned worst-case in white. Keep in mind that the total energy of the designs in the different configurations are different from one another, so that the bars only indicate how the energy is distributed among the modes, and not the absolute energy in each mode. In the worst-case configuration the first bending mode is the dominant mode for the PT and RPTT systems, accounting for over ninety percent of the output energy. In contrast the RPT design, which is less sensitive to this mode due to its higher natural frequency, shows only thirty percent of the output energy in first bending in the worst-case



(a) Modal frequency (upper) and percent energy (lower)



(b) Relative modal displacement

Figure 5-3: Modal characteristics of first bending mode for PT, RPT and RPTT designs in the nominal (blue, circle), worst-case (green, square) and tuned (white, diamond) configurations.

configuration. The tuned configurations are especially interesting as it is clear that the tuning parameters are able to drastically reduce the percent energy of this mode in both the PT and RPTT designs, while the RPT design is not affected as strongly.

To complete the story the relative modal displacement between the collector nodes for each design is shown in Figure 5-3(b). Again, the results for the nominal, worst-case and tuned worst-case configurations are shown in black, gray and white, respectively. This metric is important since the y-translation contribution to the OPD is due to the relative motion of these nodes (Equation 2.9). It is clear from the chart that the addition of the tuning masses to the PT and RPTT designs reduces the relative motion between collectors quite drastically. In effect, the asymmetric addition of tuning mass balances the asymmetry introduced by the uncertainty parameters at the worst-case vertex. The RPT design also shows a reduction, but it is not as significant. The tuning parameters are not able to affect the shape of the first mode as much, because the design is robust to uncertainty but is also robust to tuning.

The RPTT design formulation balances the key elements of the PT and RPT designs to produce a design that is both robust to uncertainty, yet tunable. In the SCI development model all effort is directed to the first bending mode as this is the significant mode in the worst-case uncertainty realization due to asymmetric motion of the collector nodes. The natural frequency of this mode in the RPTT design is higher than that of the PT design so that the mode contributes less to the output energy. However, it is not stiffened as sharply as in the RPT design so that the tuning parameters have sufficient authority to affect the performance.

5.2.3 Design Regimes

The tuned RPTT design is compared to the PT, RPT, tuned PT and tuned RPT designs over a range of uncertainty values. The formulation in Equation 5.5 is run for maximum tuning authority ($\alpha = 0.0$) at various values of uncertainty bound, Δ, ranging from 0.01% to 25%. The uncertainty values vary $\pm\Delta\%$ about nominal and are the same for both uncertainty parameters. The tuned results are obtained by running a tuning optimization on the worst-case uncertainty realization. The results

are shown in Figure 5-4, the final evolution of the design regimes plot introduced at the end of Chapter 3. The y-axis represents the performance requirement of the system and the x-axis is the level of uncertainty in the parameters. The numbered regions indicate areas in which particular design methodologies are successful for all possible uncertainty realizations as listed in the legend. For example, in Region 1, PT, RPT and RPTT all produce designs that can meet the requirement within the uncertainty bounds, while in region 4 only PT tuned and RPTT designs are successful.

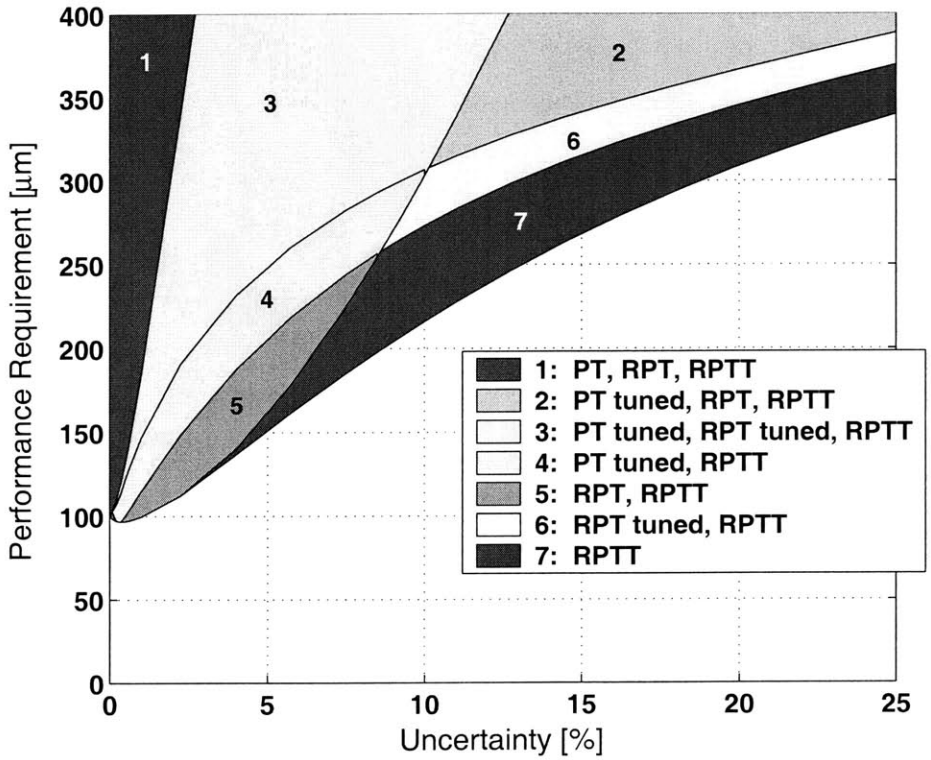


Figure 5-4: Performance requirement vs uncertainty: all designs.

In Region 7, only RPTT can produce a successful design indicating that considering tuning in the tailoring stage of the design further extends the uncertainty that can be tolerated at a given performance requirement. For example, at a requirement of $\sigma_{req} = 200\mu$m the maximum uncertainty that can be tolerated without RPTT is 7% and is achieved through a combination of PT design and hardware tuning. However, if RPTT is applied to the system the uncertainty range is extended to 9% for the

same requirement. The effect becomes more dramatic as the requirement is relaxed slightly. At a requirement of 280μm the tuned RPT design can tolerate up to 11% uncertainty, but the RPTT design allows $\approx 16.3\%$. The entire 25% uncertainty range is covered for requirements of 320μm and higher with the RPTT design. This requirement is nearly 13% lower (more aggressive) than that which can be met by the tuned RPT design at the same uncertainty level.

In addition to extending the reachable design space, RPTT is the only method that is successful in *all* of the design regions. As noted in the previous chapter, Region 4 is interesting because the tuned PT design is adequate, but RPT and RPT tuned are not. This result is concerning because the uncertainty is relatively low in this region and it is reasonable to assume that RPT or tuned RPT formulations are the correct approach here. However, the plot shows that, for this problem, tuning the PT design achieves a more aggressive performance requirement than tuning the RPT design at the same level of uncertainty. For example, at $\Delta = 5.75\%$ the tuned PT design can meet a requirement of 178μm, while the tuned RPT design can only meet a requirement of 218.62μm. In contrast, RPTT is able to meet the same requirement as tuned PT, and can go slightly further to 161.43μm if necessary. In fact, RPTT is the only method that is appropriate in all of the regions shown here.

5.3 RPTT Simulations

In this section the RPTT design space is investigated more thoroughly through a series of simulations. First the trade between tuning authority and robustness is considered by varying the robustness weight, α. Then, the PT, RPT and RPTT designs are compared over a set of random hardware simulations to assess the methodology performance the entire uncertainty space.

5.3.1 Tuning Authority

Recall from Equation 5.1 that the RPTT objective function includes weighted robustness and tuning authority costs allowing a trade of relative importance between the

two. The effects of this weighting, α, on the design performance predictions are explored by finding the optimal RPTT design for a range of α at a constant uncertainty bound, $\Delta = 10\%$. The nominal performance prediction of each design is evaluated along with the performance at the worst-case uncertainty vertex. The worst-case realization is then tuned to obtain the tuned performance.

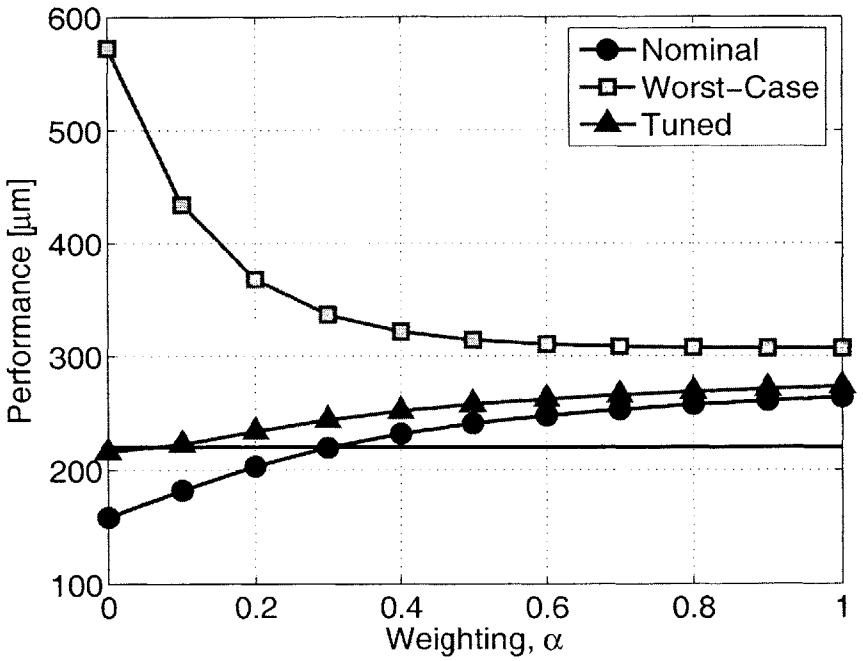


Figure 5-5: RPTT design performance as a function of weighting (α) for $\Delta = 0.10$: (–) requirement.

The results of this study are shown in Figure 5-5. Nominal, worst-case and tuned performance are depicted by circles, squares and triangles, respectively. At low values of α the tuning authority is weighed heavier than robustness, and, as expected, the tuned performance is best in this region. It is interesting to note that the nominal performance is also very good when α is low, but the worst-case performance is very high. As α increases, robustness becomes more important, and the worst-case performance decreases while the nominal and tuned performance increase. It is clear that, for this model, a trade exists between tuning authority and robustness.

Consider a requirement of 220μm, indicated by a solid black line on the figure. The tuned performance meets this requirement only at robustness weight close to zero

indicating that this requirement is aggressive for this system at this uncertainty level. Therefore, to guarantee success through tailoring and tuning the system should be designed for maximum tunability with a robustness weight at or near $\alpha = 0$. However, if this requirement is relaxed to 250μm then less tuning authority is necessary to meet requirements, and a higher weight, up to $\alpha = 0.4$, can be placed on robustness. Placing the maximum weight possible on robustness produces a design with a worst-case performance prediction that is closer to the requirement increasing the chance that the hardware will not need tuning at all.

5.3.2 Hardware Simulations

The results presented thus far indicate that the RPTT is more tunable than both the PT and RPT designs at the worst-case uncertainty vertex. This result is not surprising given the formulations of the design optimizations. PT does not consider uncertainty at all, and RPT is only concerned with being robust to uncertainty at the worst-case vertex and does not take advantage of hardware tuning. The RPTT formulation anticipates the decrease in uncertainty effected by building hardware and incorporates that benefit into the design by allowing different tuning configurations at each uncertainty vertex and minimizing the worst-case tuned performance. As a result, it is guaranteed that RPTT is tunable at the uncertainty vertices, but it is unclear if that assumption holds throughout the rest of the uncertainty space. In this final section, a series of hardware simulations are run to evaluate the performance of the designs across all of the uncertainty space.

The algorithm used to generate the hardware simulations is given in Figure 5-6. The outer loop in the algorithm is over the number of simulations desired, n_{sims}. For each simulation an uncertainty realization, $\vec{p}_{MC}$, is chosen randomly from the uncertainty model. Recall from Equation 3.1 that the uncertainty values in the SCI development model are assumed to be uniformly distributed about the nominal values within bounds of $\pm\Delta\%$. A hardware simulation is generated for each of the three designs, PT, RPT and RPTT, by applying $\vec{p}_{MC}$ to the models and evaluating the performance, σ_{HW}. The hardware performance is then compared to the requirement.

If the hardware meets the requirement then no tuning is necessary and it is considered nominally successful. However, if σ_{HW} is not within the requirement, then the hardware simulation is tuned assuming that the exact uncertainty parameter values, $\vec{p}_{MC}$, are known, a procedure referred to as baseline tuning in Chapter 4. In reality, the tuning would take place through a model updating technique such as isoperformance tuning (see Chapter 4), however for the purposes of this study it is only necessary to determine if it is *possible* to tune the hardware below requirement, so baseline tuning is adequate. The results of the simulations include the nominal hardware performances and the tuned hardware performances for each of the designs.

```
Data: PT, RPT, RPTT designs, uncertainty model, n_sims
Result: nominal HW performance, tuned HW performance
begin
    for i = to n_sims do
        Randomly choose p_MC from uncertainty model
        for PT, RPT and RPTT designs do
            Generate HW simulation by applying p_MC to model
            Evaluate nominal HW performance, σ_HW
            if σ_HW > σ_req then
                Tune HW simulation with knowledge of p_MC → baseline tuning
                Evaluate tuned performance, σ_tune, and store
            end
        end
    end
end
```

Figure 5-6: Hardware simulation algorithm.

The PT, RPT (AO) and RPTT designs generated based on an uncertainty model with $\Delta = 10\%$, $\alpha = 0.0$ and $\sigma_{req} = 220\mu$m are used to generate 200 hardware simulations. A map of the uncertainty space explored is plotted in Figure 5-7. The nominal uncertainty values are marked with the large dot in the middle of the grid and the bounds are denoted by the box around the grid. Each of the realizations used to generate the hardware simulations is marked with a dot. Note that although the 200 simulations aren't quite enough to fill the space, they do an adequate job of sampling it.

The results of the hardware simulation are presented in Figure 5-8. The upper plot,

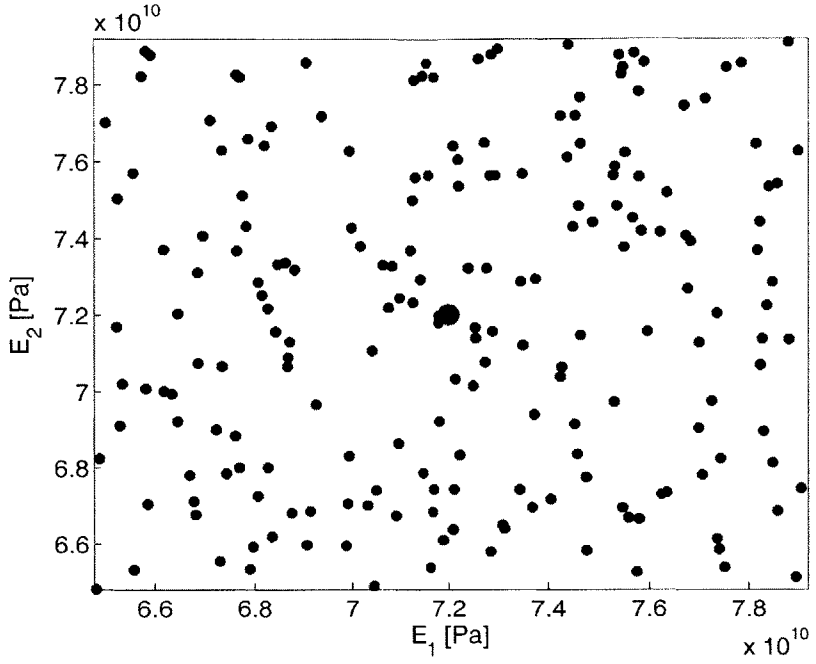
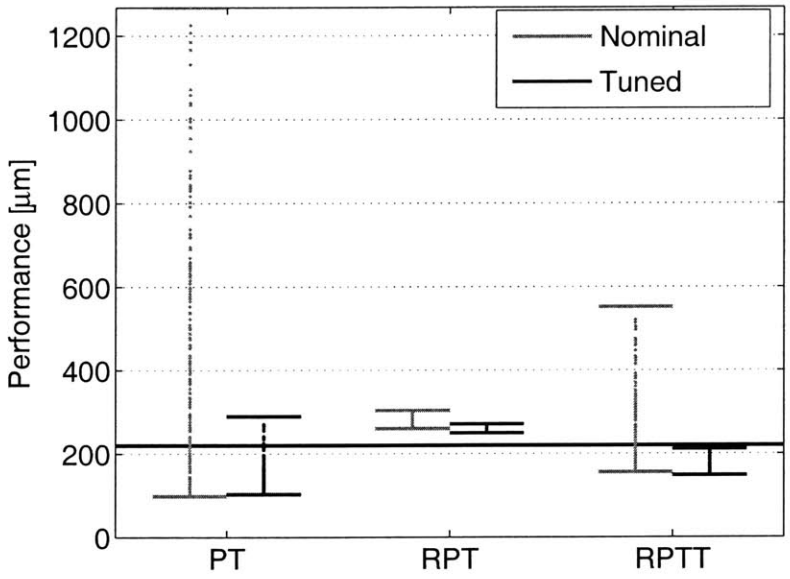
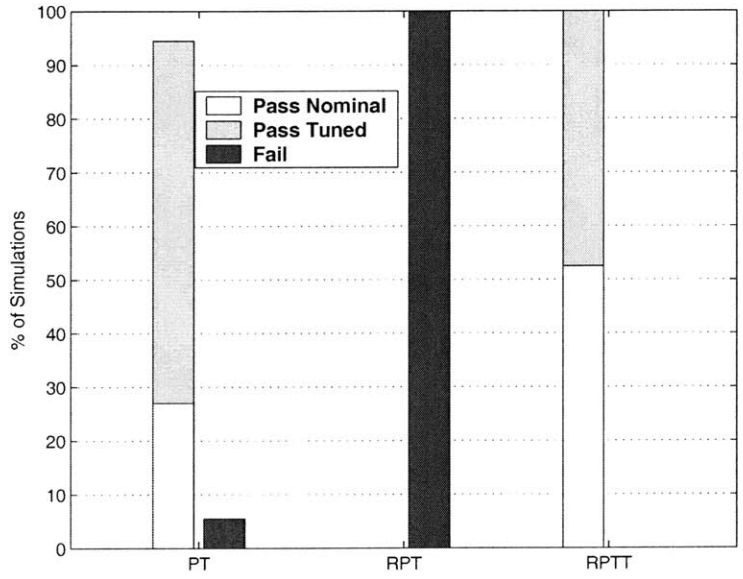


Figure 5-7: Uncertainty points considered in Monte Carlo hardware simulation study, $\Delta = 10\%$.

(Figure 5-8(a)), shows the nominal and tuned performances obtained for each of the designs. The nominal performance values are denoted by gray dots forming a vertical line. The maximum and minimum values are marked with horizontal error bars. The tuned performances are plotted in a similar fashion but in balck. The requirement is indicated by the solid line at 220μm. The nominal PT hardware covers a large range of performance values ranging from 98.54μm to 1266.8μm. Once the poorly performing designs are tuned the range is decreased considerably to a maximum value of only 289.50μm. However, the maximum tuned value does not meet the performance requirement indicating that some number of the PT simulations do not have sufficient tuning authority. In contrast, the range of RPT peformances is very small. In the nominal configuration the performances range from 259.81μm to 303.5μm, and once tuned range from 249μm to 271.21μm. Note that none of the nominal designs meet the requirement, and although tuning improves the performance, it is not sufficient to bring the simulations within the required performance, and not a single tuned RPT design is successful. The RPTT design has a greater range of nominal performance



(a)



(b)

Figure 5-8: PT, RPT and RPTT simulations results, $\Delta = 10\%$, $\sigma_{z_{req}} = 220\mu$m: (a) performance results (b) design success.

than the RPT design, 155.45μm to 551.5μm, but is less sensitive to uncertainty than the PT design. The lower range of the RPTT nominal performance is below the requirement indicating that some of the nominal hardware simulations are successful without tuning. More importantly, the entire tuned range, 148.16μm to 211.23μm, is below the requirement indicating that *all* of the designs are successful once tuned.

The lower subplot, Figure 5-8(b), is a bar chart showing the percent of simulations that are successful, i.e. meets requirements, for each design. Successful designs are broken into two subcategories: those that pass nominally (white bars) and those that pass after tuning (gray bars). The failed designs are indicated by black bars. The PT design is largely successful, with 94.5% of the simulations meeting the requirement. However, the majority of simulations need to be tuned (only 27% pass nominally), and 5.5% of the simulations fail even with tuning indicating that there is no guarantee of success with the PT design. The RPT design fares much worse with a 100% failure rate over the simulations. As discussed in Chapters 3 and 4, the RPT is much less sensitive to uncertainty, but is also insenstive to the tuning parameters resulting in a design with a small range on both nominal performance and tuning. Only the RPTT design is successful for 100% of the simulations. In addition, over half of the RPTT simulations pass nominally, and tuning is only required in 47.5% of the cases. Even though the robust weight, α was set to zero, RPTT achieves a blend of tunability and robustness since the design is tuned at all of the uncertainty vertices. The resulting design is more robust to uncertainty than the PT design and is more likely to meet requirements in the nominal hardware configuration.

The results of the simulations at $\Delta = 10\%$ are interesting because although RPTT is the only design that is successfull 100% of the time, it is surprising to see that the PT design is highly tunable and largely successful despite its high sensitivity to the uncertainty parameters. To further explore this issue 200 simulations are run with a higher uncertainty level, $\Delta = 21.5\%$. The design regimes in Figure 5-4 indicate that none of the designs can accomodate such a high uncertainty level and a requirement of 220μm, so for these simulations the requirement is relaxed to $\sigma_{req} =$ 330μm. In addition, two RPTT designs are generated, one with $\alpha = 0.0$ and the other

at $\alpha = 0.1$ to further explore the trade between robustness and tuning authority. The uncertainty parameters used in this set of simulations are plotted in Figure 5-9. Note that the bounds on the uncertainty parameters are increased compared to those in the corresponding figure from the previous set of simulations, Figure 5-7. The 200 samples do not cover the entire space evenly, but do provide decent coverage across the grid.

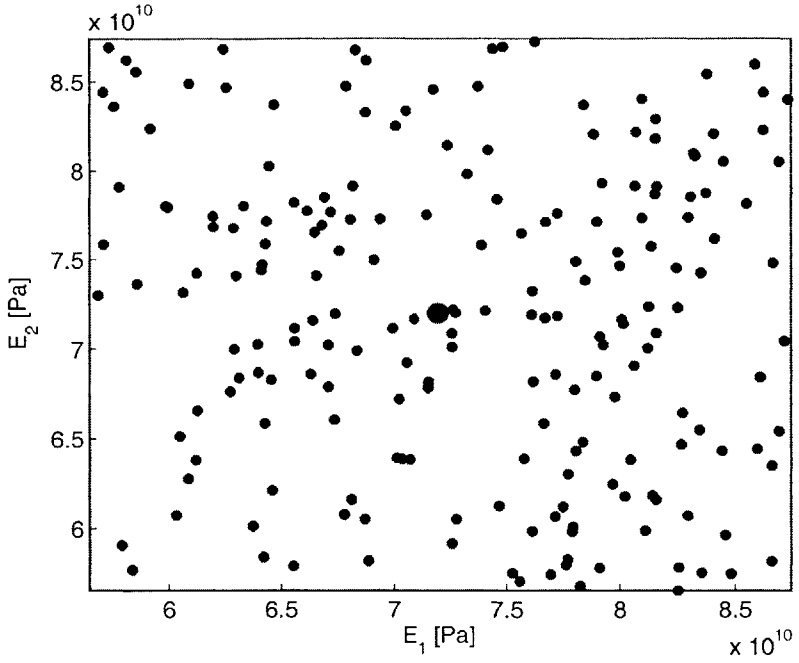


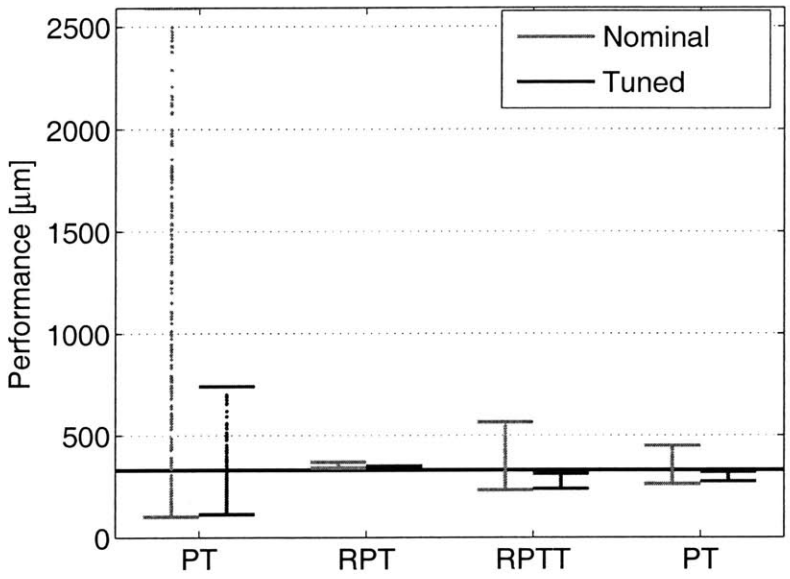
Figure 5-9: Uncertainty points considered in Monte Carlo hardware simulation study, $\Delta = 21.5\%$.

The nominal and tuned perfromances from the hardware simulations are plotted in Figure 5-10 along with a bar chart indicating the success rate of each design. The performance plot (Figure 5-10(a)) is similar in trend to that from the previous set of simulations (Figure 5-8(a)), but there are some key differences. First, the range of the PT nominal performance values is much higher due to the increase in uncertainty bounds. The PT hardware simulations range from 101.87μm all the way to 2591.10μm, over twice the value of the maximum at $\Delta = 10\%$. A similar increase occurs in the tuned range as well (112.98μm to 740.47μm), and the maximum tuned value is over twice that of the requirement. The RPT design simulations

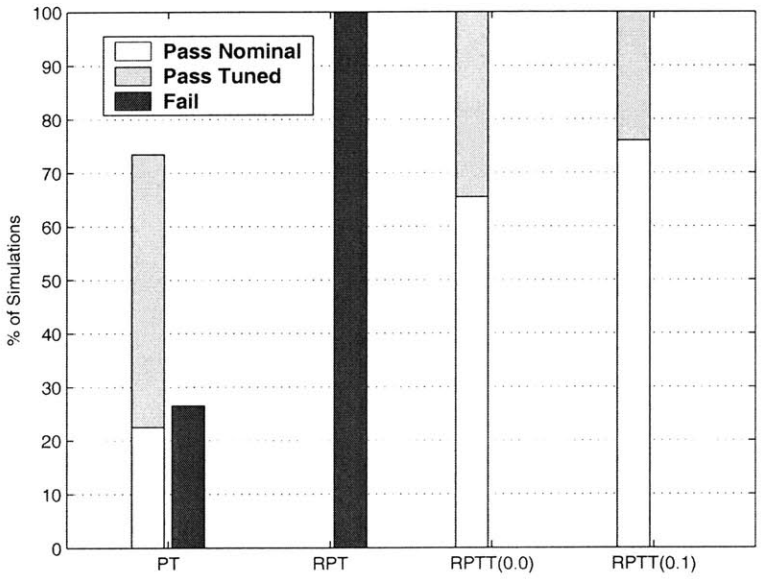
have a very small nominal performance range (341.43μm to 369.1μm) indicating that the design is quite robust to uncertainty. Unfortunately, the robustness comes at the price of nominal performance and none of the hardware simulations meet the requirement. Upon tuning these hardware simulations the performance range shrinks further (331μm to 349.63μm) due to a lack of tuning authority in the robust design, and all of the tuned designs also fail to meet the requirement.

The two RPTT designs are differentiated in the plot by the value of α placed parenthetically in the label. As seen in the previous simulation, both RPTT designs show a blend of the PT and RPT design characteristics. At $\alpha = 0$, the nominal performance range (232.52μm to 565.5μm) is greater than that of the RPT design but much smaller than that of the PT design. This result indicates that even with no weight on the robustness cost the RPTT formulation produces a design that is dramatically less sensitive to uncertainty than the PT optimization. The RPTT design also has greater tuning authority than the RPT design, and the tuned performances range from 237.67μm to 312.45μm. The maximum value of this range is well below the requriement of 330μm, so that all of the tuned RPTT designs are successful. The second RPTT design, acheived through an RPTT optimization with $\alpha = 0.1$, shows increased robustness to uncertainty with a nominal performance ranging from 261.10μm to 448.9μm. The range of tuned performances is also smaller than that in the first RPTT design starting at 274.30μm and continuing to just under the requirement at 320.15μm. All of the design simulations are again tunable to below the requirement, but the RPTT design with $\alpha = 0.1$ is superior to that for $\alpha = 0$ in that it is just tunable enough and therefore more robust to uncertainty. The tuning authority and robustness is balanced such that the design is perfectly tailored to meet the requirement.

The accompanying bar chart, Figure 5-10(b), presents the percent of successful simulations for each design. As seen in the previous hardware simulation (Figure 5-8(b)) only the RPTT designs are successful for all of the simulations, and the RPT design never meets requirements. At the higher uncertainty level just over 70% of the PT simulations succeed, with 51% requiring hardware tuning. Over a quarter



(a)



(b)

Figure 5-10: PT, RPT and RPTT simulations results, $\Delta = 21.5\%$, $\sigma_{z_{req}} = 330\mu$m: (a) performance results (b) designs success.

of these simulations fail to meet requirements all together. In contrast, all of the simulations meet requirements for both of the RPTT designs, and only 34.5% and 24% of the simulations require tuning for the $\alpha = 0$ and $\alpha = 0.1$ designs, respectively. Since there is a cost associated with hardware tuning it is perferable to design the system such that it is likely to meet requirements nominally. In this sense, the RPTT design with $\alpha = 0.1$ performs the best at this requirement and uncertainty level since all simulations meet the requirement and only 24% require hardware tuning.

A plot of the nominal, worst-case and tuned performances of RPTT designs at various values of α at this uncertainty level ($\Delta = 21.5\%$) is shown in Figure 5-11 along with the requirment at 330μm. Note that the tuned worst-case performance is above the requirement for values of α just over 0.1. The results of the hardware simulation indicate that to produce the best RPTT design α should be set to the maximum value at which the tuned worst-case performance meets the requirement. This robustness weighting produces a design that is robust to uncertainty, yet is still tunable to meet requirements. As indicated by the bar chart (Figure 5-10(b)), including maximum allowable robustness in the cost increases the number of uncertainty realizations in which hardware tuning is not necessary.

5.4 Summary

Robust performance tailoring for tuning blends the concepts of robust design and hardware tuning to produce a design with good nominal performance that is both robust to uncertainty and has sufficient authority to meet requirements through hardware tuning if necessary. The RPTT formulation extends the idea of robustness from the design to the physical system by optimizing to facilitate hardware tuning by minimizing the worst-case *tuned* performance over the uncertainty space. During tailoring, the performance prediction accounts for the fact that the tuning parameters can be changed when the hardware is built and the uncertainty is fixed (and known) i.e. the value of $\vec{y}$ is different depending on the value of $\vec{p}$. This knowledge adds extra degrees of freedom to the problem and results in a less constrained optimization.

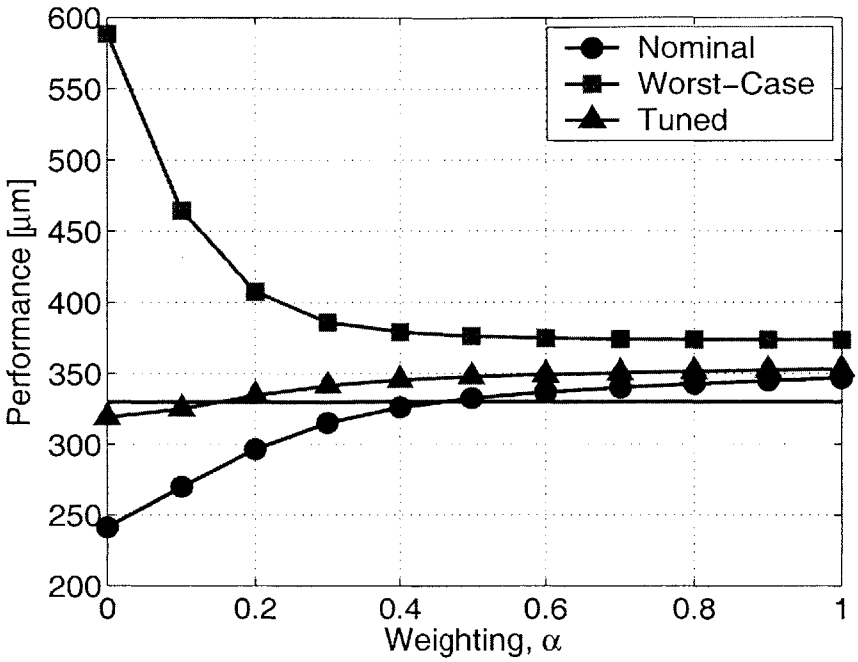


Figure 5-11: RPTT design performance as a function of weighting (α)for $\Delta = 21.5\%$: (–) requirement.

The RPTT formulations are applied to the SCI development model with uncertainty bounds of $\Delta = 10\%$ using SA and SQP optimization algorithms. The results are compared for performance and efficiency. It is found that a blend of SA and SQP in which the SA design is used as an initial guess for SQP performs best. In addition, a simple minimization form of the RPTT cost function in which the tuned and nominal performances at the uncertainty vertices are included in the constraints is found to be the best posed for SQP optimization.

The nominal, worst-case and tuned performances of the optimal RPTT design are compared to those of the PT and RPT designs. It is found that the RPTT design is somewhere between the PT and RPT designs in terms of nominal performance and robustness. It has a better nominal performance than the RPT design, but is more sensitive to uncertainty. However, the tuned worst-case performance of the RPTT design is better than those of either the PT or RPT designs. A design regimes plot shows that the RPTT design further extends the design space for this model allowing more aggressive performance requirements to be met at higher levels of uncertainty

and that it is the only design method that is successful in all of the reachable regimes.

Finally, a set of randomly generated hardware simulations are run for two different levels of uncertainty and performance requirement. Simulations of the RPT designs are never successful because too much focus is placed on insensitivity to uncertainty. A good number of the simulated PT designs are successful, but there are some that fail. In addition, much dependance is placed on tuning as few of the nominal hardware configurations meet the requirement. In contrast, all simulations of the RPTT design succeed and the majority do not require tuning.

Chapter 6

Focus Application: Structurally Connected TPF

In the previous chapters, three tailoring methodologies are explored and it is shown through application to a low-fidelity development model that RPTT, Robust Performance Tailoring for Tuning, produces a design with the most desirable blend of nominal performance, robustness to uncertainty and tuning authority. The development model is representative of a structurally-connected interferometer but has only fifteen degrees of freedom and contains only structural elements. In realistic applications a high-fidelity integrated model is required for design and analysis. Therefore, in this chapter the design methodologies are applied to an integrated TPF model that includes a high-fidelity structural component as well as a realistic disturbance model, vibration isolation and a simple ACS controller. PT, RPT and RPTT designs for a structurally-connected TPF interferometer (TPF SCI) are obtained, and performance trends similar to those in the previous chapters are observed. Implementation issues encountered with such a model are identified and addressed.

6.1 Model Description

The TPF SCI model used for the study is an integrated model with components built in MATLAB and NASTRAN and integrated with the DOCS (Disturbance Optics

Controls Structure) tools [24]. The model is based largely on one created at the MIT Space Systems Laboratory by D. Lobosco [76] with some updates resulting from communication with the Jet Propulsion Laboratory [1]. A schematic of the integrated model is shown in Figure 6-1. The main model components are the plant, attitude

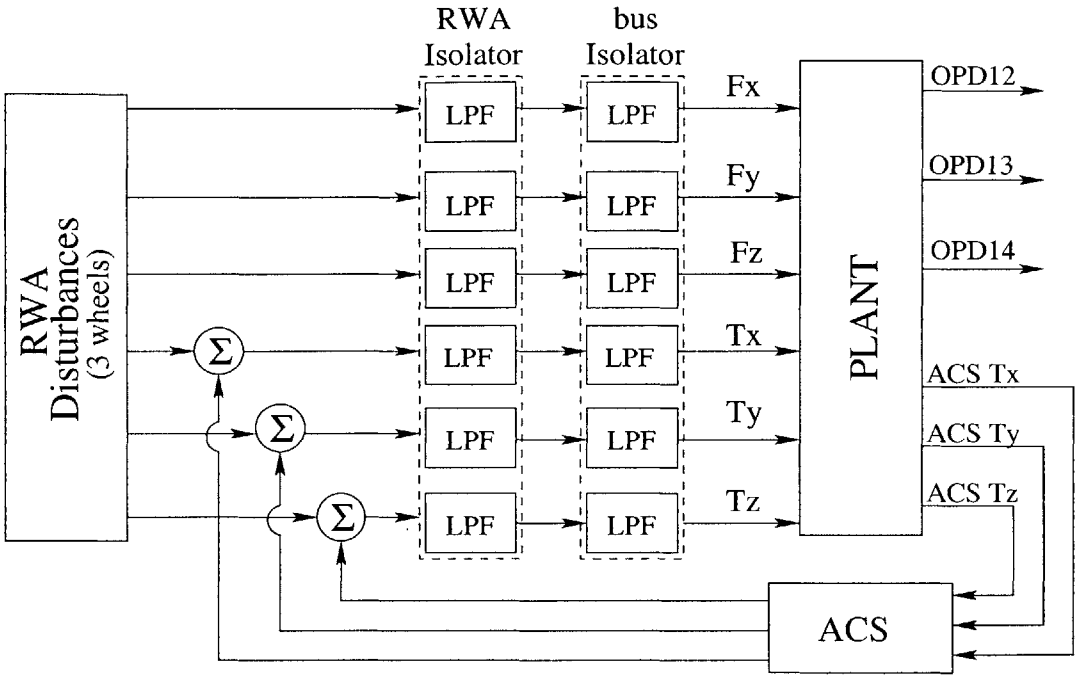


Figure 6-1: Schematic of TPF SCI Model.

control system, disturbance model and vibration isolation stages. The inputs to the integrated model are reaction wheel assembly (RWA) disturbances, shown at the left-hand side of the figure. The disturbances are combined with torque commands from the attitude control system (ACS) and pass through a two-stage vibration isolation system. The first stage represents an isolator between the RWA and the bus, and the second stage models isolation between the bus and the supporting structure. The isolated disturbances then pass through the plant. This component has the highest fidelity and includes the structural finite element model and optical sensitivities. The structural model consists of four evenly-spaced collectors mounted to a truss with a triangular cross-section as shown in Figure 6-2. The plant has six outputs, three

[1] personal communication with Douglas Adams, Member of Technical Staff, Jet Propulsion Laboratory

optical path differences between the collector pairs and three angular torque measurements from the bus that drive the ACS. Since the ACS commands are fed back to the plant torque inputs, the only outputs of the integrated model are the three optical performance metrics. The model components are described in detail in the following sections.

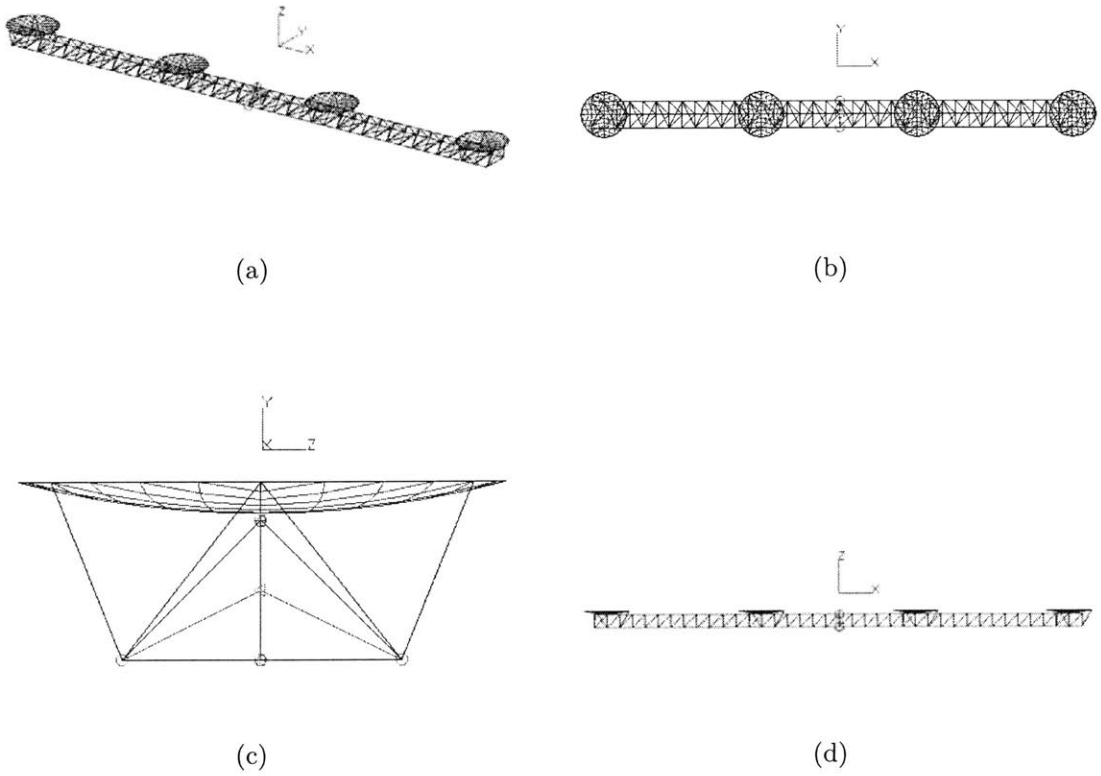


Figure 6-2: Nominal SCI TPF plant finite element model (a) Isometric view (b) top view (XY) (c) side view (ZY) (d) front view (XZ).

6.1.1 Reaction Wheel Assembly Disturbances

The inputs to the integrated model are the disturbances. The are many possible sources of disturbances that may impinge on a spacecraft such as TPF. Some are functions of the environment, such as gravity gradients and solar torque. Other disturbances are from on-board mechanisms such as fast-steering mirrors or thrusters. Vibrations from the reaction wheel assembly (RWA) are anticipated to be the dom-

inant disturbance source and much work has been done regarding the modeling of these vibrations. As a result it is the only disturbance source considered in this model.

The wheel vibrations are assumed to be harmonic in nature with amplitudes proportional to the wheel speed squared:

$$m(t,\Omega) = \sum_{i=1}^{n} C_i f^2 \sin(2\pi h_i \Omega t + \alpha_i) \tag{6.1}$$

where $m(t,\Omega)$ is the disturbance force or torque in Newtons (N) or Newton-meters (Nm), n is the number of harmonics included in the model, C_i is the amplitude of the i^{th} harmonic in N^2/Hz (or $[Nm]^2/Hz$), f is the wheel speed in Hz, h_i is the i^{th} harmonic number and α_i is a random phase (assumed to be uniform over $[0, 2\pi]$). The harmonic numbers and amplitude coefficients for an Ithaco E-type wheel are given in [83]. Note that Equation 6.1 is a function of the wheel speed. Therefore, in order to perform a disturbance analysis with the model in this form a separate analysis is required for each wheel speed.

In the course of operation the wheels traverse through a range of speeds. Therefore, a frequency-domain, stochastic wheel model was developed by J. Melody to estimate the wheel disturbances over a given speed range [97]. The model is developed from Equation 6.1 by assuming that the wheel speed is a uniformly distributed random variable:

$$S_{ww}(\omega) = \begin{cases} \frac{\pi C_i^2}{2(f_2 - f_1)(2\pi h_i)^5}\omega^4 & \text{for } 2\pi h_i f_1 < |\omega| < 2\pi h_i f_2 \\ 0 & \text{otherwise} \end{cases} \tag{6.2}$$

where S_{ww} is the power spectral density (PSD) of the wheel force or torque disturbance, f_1 and f_2 are the lowest and highest expected wheel speeds (Hz), respectively, and ω is the frequency variable.

Equations 6.1 and 6.2 model the disturbances from a single reaction wheel in a coordinate system that is aligned with the body frame of the wheel. However, a reaction wheel assembly is generally composed of multiple wheels, usually between three and six. In order to create a disturbance model of the entire assembly that can

Table 6.1: RWA disturbance model parameters.

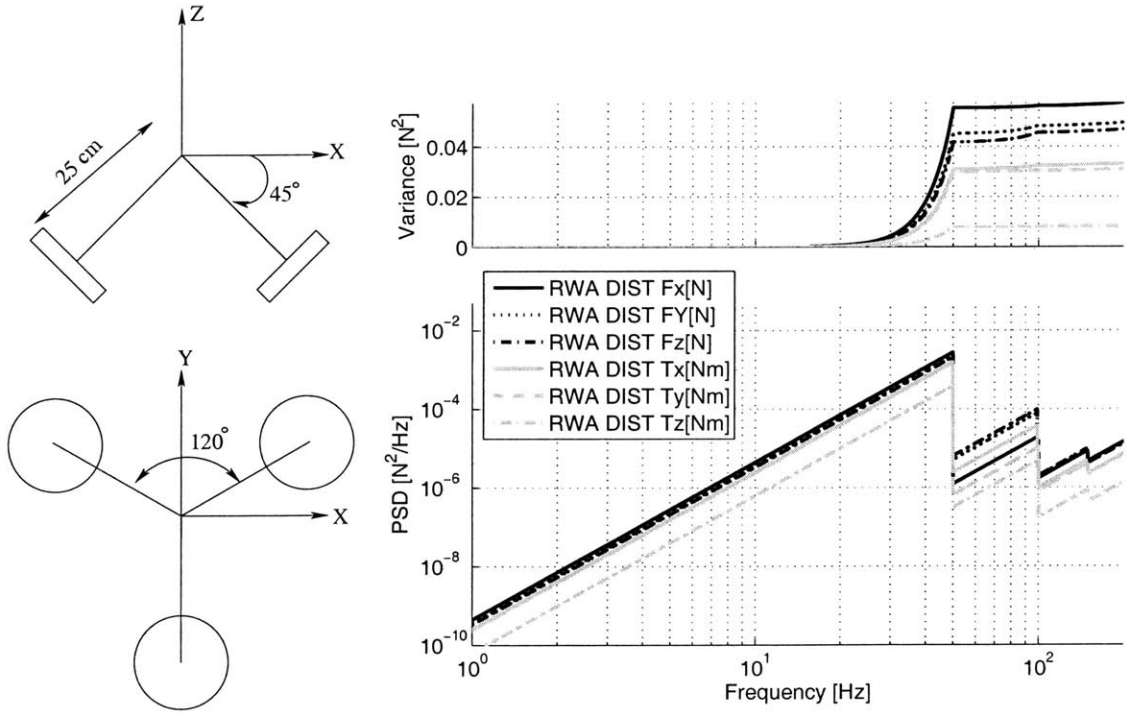
Name	Value	Units	Description
f_1	0	Hz	minimum wheel speed
f_2	50	Hz	maximum wheel speed
n_{rwa}	3	N/A	number of wheels
α	30	degrees	orientation angle
θ	45	degrees	orientation angle
γ	0	degrees	orientation angle
d_{rwa}	0.25	m	distance to spacecraft origin

be used to drive the structural model, the number and configuration of the wheels must be taken into account. A multiple wheel model was developed by H. Gutierrez that combines the disturbance models of n wheels in a given configuration into a single RWA disturbance model [50]. This model is a 6x6 matrix of power and cross-spectral densities of the RWA vibrations transformed to the body-frame of the spacecraft.

The disturbance model implemented in this version of the TPF model is a stochastic, multiple-wheel model created using the DOCS `docs_psd_rwa_dist.m` function. The model parameters used are listed in Table 6.1. The wheel speeds are assumed to range from 0 to 3000 RPM. The angles α, θ, and γ locate the three wheels with respect to the body-frame spacecraft axes. The parameter d_{rwa} is the radial distance from the wheel centers to the spacecraft origin. In the current TPF configuration, the spin-axes of the three wheels are 45° from the TPF Z-axis, and their projections in the X-Y plane are spaced 120° apart as shown in Figure 6-3(a). The modelled disturbance PSDs are shown in Figure 6-3(b).

6.1.2 Vibration Isolation

Due to the precision required by the TPF instrument it is likely that the RWA will be isolated from the truss structure that supports the optics. Simple models of a two-stage isolation system based on the isolator design for SIM are included in the integrated model. The first model represents an isolator located between the RWA and the bus, and the second isolates the supporting structure from the bus. The isolator models are included in the TPF SCI model by placing two second-order systems in



(a) Configuration

(b) Power Spectral Densities

Figure 6-3: RWA disturbance model.

series between the RWA disturbances and the plant as shown in Figure 6-1. The isolator transfer functions are:

$$G_{iso} = \frac{(2\pi f_{iso})^2}{s^2 + 2\zeta_{iso}(2\pi f_{iso})s + (2\pi f_{iso})^2} \tag{6.3}$$

where G_{iso} is the transfer function from the input to the isolated disturbance, f_{iso} is the corner frequency of the isolator (Hz), and ζ_{iso} is the isolator damping ratio. The damping ratio for both isolators is assumed to be 0.1 and the frequencies of the first and second stages are set to 3 Hz and 8 Hz, respectively. The transfer functions for both stages are shown in Figure 6-4.

6.1.3 Plant

The plant is a finite element model of a baseline design of the TPF structurally-connected interferometer concept consisting of four collecting apertures, a supporting

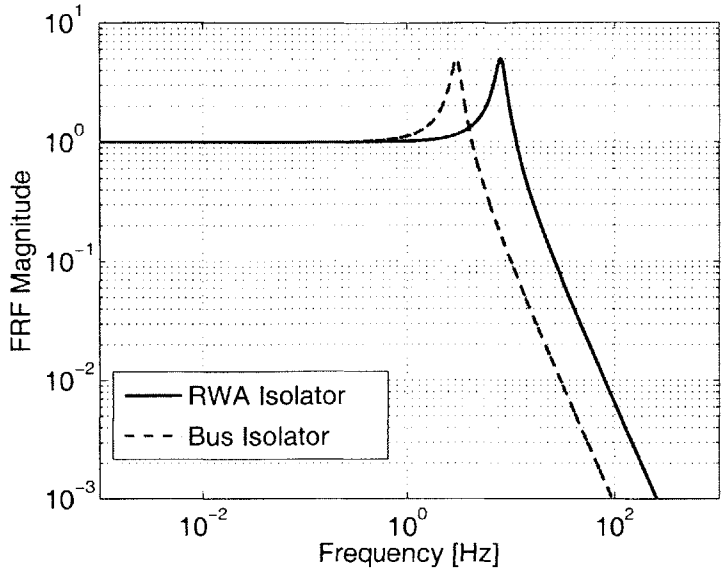


Figure 6-4: Transfer functions of Isolator models: RWA isolator, $\zeta_{iso} = 0.4$ and $f_{iso} = 8$ Hz (solid line), bus isolator, $\zeta_{iso} = 0.1$ and $f_{iso} = 3$ Hz (dashed line).

truss and a combiner. The interferometric baseline, measured as the distance between the centers of the two outermost collectors, is 36 meters. The collectors are evenly spaced along the baseline such that there are 12 meters between any two neighboring collectors. The plant model is built mainly in NASTRAN and is shown in its nominal configuration in Figure 6-2.

It is comprised of NASTRAN beam elements (CBAR), concentrated masses (CONM2), plate elements (CQUAD8, CTRIA6) and rigid elements (RBAR, RBE2). The reader is referred to the NASTRAN documentation for more information on these finite elements [94]. The inputs to the plant are the six isolated RWA disturbances, three forces and three torques. The plant outputs are three OPD metrics from the interferometer pairs and three angular rotations from the ACS sensor. The structural mass breakdown of the plant model is given in Table 6.2. The masses listed are based on current best estimates from JPL, and the total structural mass is 2679 kg.

Truss

The main structural component of the TPF model is the supporting truss. The origin of the spacecraft frame is at the center of the truss so that the truss is broken into

Table 6.2: TPF SCI model mass breakdown.

Component	Mass [kg]
Truss	850.1
Combiner	250.0
RWA (3 Ithaco E)	31.8
Collectors (4)	1547.1
TOTAL	2679.0

two arms, one that extends from the origin in the positive X-direction and one that extends in the negative X-direction. Note from Figure 6-2(a) that the X-axis is aligned with the interferometer baseline and the Z-axis with the observation direction. The Y-axis completes the right-handed coordinate system. The truss is symmetric about the origin so that the negative truss arm is simply the reflection of the positive arm.

The truss is 38 meters in length and is composed of 38 individual truss bays. Each bay consists of nine members: three longerons, three battens forming a triangular frame and three diagonals as shown in Figure 6-5. Each truss member is modeled with a NASTRAN CBAR element, and the material properties are based on those of a low-CTE (coefficient of thermal expansion) MJ55 composite (Table 6.3). The truss design and material are based on a model from the Jet Propulsion Laboratory [2]. The numbering scheme used for the truss elements is given in Table 6.4.

Table 6.3: Truss properties.

Name	Description	Value	Units	Source
ρ	material density	1830	kg/m^3	low CTE MJ55
E	Young's Modulus	111.7	GPa	low CTE MJ55
ν	Poisson's Ratio	0.3	none	low CTE MJ55
r	cross-sectional radius	.01	m	mass and frequency estimates
h	batten frame height	1	m	sized from mass/freq estimates
w	batten frame width	2	m	sized from mass/freq estimates
L	truss bay length	1	m	modeling convenience

[2]personal communication with Douglas Adams, Member of Technical Staff, Jet Propulsion Laboratory

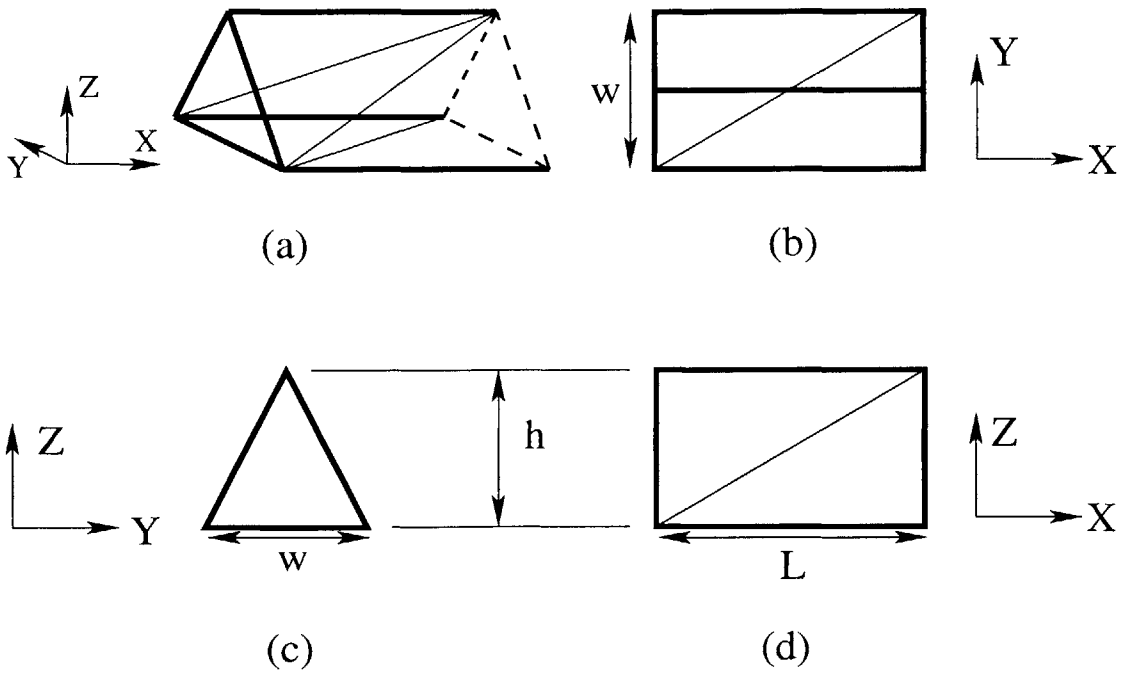


Figure 6-5: Geometry of truss bays with longerons and battens indicated by heavy lines: (a) isometric view (b) top view (c) cross-section, batten frame (d) side view.

Table 6.4: Truss element and grid numbering.

ID	Type	Description
1-117	GRID	truss grid points
1001 - 1435	CBAR	truss members
2001-2019	MAT1	material properties by truss bay
2001 - 2019	PBAR	member properties by truss bay

Bus

The spacecraft bus contains the operations sub-systems such as communication and data handling, attitude control, propulsion and power systems. The bus mass is not included in this model, but its location is modeled with a grid point at the XY origin of the spacecraft frame in the middle of the center truss bay. The bus node is connected to the supporting truss at three grid points through a distributed rigid connection (RBE2), and serves as a connection point for the combining optics and reaction wheel assembly.

The only attitude control system component (ACS) included in the model is a reaction wheel assembly (RWA). The RWA is modeled as a concentrated mass on a

node located on the negative Z surface of the truss bay at the spacecraft XY origin and is rigidly connected to the bus. It is assumed that the assembly consists of three Ithaco E-type wheels. This particular wheel is chosen because it has a mass that is close to the current best estimate (CBE) from JPL and a well-documented disturbance model of this type of wheel is available (see Section 6.1.1). The bus grid points and masses are listed in Table 6.5.

Table 6.5: Bus model elements and properties.

ID	Type	Description
5001	GRID	bus location
5100	RBE2	bus-truss connection
5002	GRID	RWA location
5012	CONM2	RWA mass (31.8 kg)
5102	RBAR	RWA-bus connection

Instrument

The telescope instrument consists of a combiner, located at the origin of the spacecraft coordinate system, and collectors evenly spaced along the truss such that there as 12 meters between neighboring pairs as show in Figure 6-2. In reality the collecting optics consist of a primary mirror, a secondary mirror and relay optics. However, for simplicity, each collector is modeled only by a 3.5 diameter circular primary mirror built from NASTRAN plate elements. The mirror material is Ultra-Low Expansion (ULE) glass, a titanium silicate glass manufactured by Corning with desirable thermal properties[4]. The material properties of the glass are listed in Table 6.6. Further details regarding the geometry and construction of the mirror model are found in [76].

The collectors are mounted to the truss with six beam elements connected to three locations on the mirror and six grids on the bottom surface of the truss. Each mirror mount point has two beam elements connecting it to two truss grids. Low CTE MJ55 is used as the material for these primary mirror mounts. Their cross-sectional radius is set to 5 centimeters to provide a reasonably stiff optical mount. The primary mirror

Table 6.6: Primary mirror properties.

Name	Description	Value	Units	Source
d_{PM}	mirror diameter	3.5	m	JPL
ρ_M	material density	2210	kg/m^3	ULE Glass
E_M	Young's Modulus	67.6	GPa	ULE glass
ν_M	Poisson's Ratio	0.17	none	ULE glass
ρ_a	areal density	40	kg/m^2	JPL estimate

mount properties are listed in Table 6.7 for reference.

Table 6.7: Primary mirror mount properties.

Name	Description	Value	Units	Source
ρ_{PM}	material density	1830	kg/m^3	low CTE MJ55
E_{PM}	Young's Modulus	111.7	GPa	low CTE MJ55
ν_{PM}	Poisson's Ratio	0.3	none	low CTE MJ55
r_{PM}	cross-sectional radius	.05	m	frequency estimates

The combiner consists of the combining optics, supporting structure, a cryo-cooler, and thermal and mechanical controls and is located at the center of the interferometric array. All of these elements are modelled together as a concentrated mass (250 kg) with no inertia. The mass of the combiner is based on the CBE from JPL. The combiner mass is located at the origin of the spacecraft coordinate system and is connected to the bus with a rigid element. All of the finite elements that make up the instrument model are listed in Table 6.8.

Optical Performance

The performance metrics of interest are the optical path differences (OPD) between the reference collector, collector 1, and the other three apertures as shown schematically in Figure 6-6. Similar to the development model the OPD calculation is based only on the perturbations of the center node of the collector primary mirrors since there is no optical train detail in the model.

When the interferometer is in its nominal configuration, i.e. the collectors and combiners are not perturbed from their locations, the path lengths from the star to

Table 6.8: TPF SCI instrument elements.

Element ID	Element Type	Description
5000	GRID	
5010	CONM2	Combiner
5101	RBAR	
3000	PSHELL, MAT1	Collector glass properties
10001-10085	GRID	
13001 - 13060	CQUAD8	Collector 1 (X=-18m) Mirror
14001 - 14024	CTRIA6	
12203 - 12206	CBAR	Collector 1 Mounts
12200	PBAR, MAT1	
20001-20085	GRID	
23001 - 23060	CQUAD8	Collector 2 (X=-6m) Mirror
24001 - 24024	CTRIA6	
22203 - 22206	CBAR	Collector 2 Mounts
22200	PBAR, MAT1	
30001-30085	GRID	
33001 - 33060	CQUAD8	Collector 3 (X=+6m) Mirror
34001 - 34024	CTRIA6	
32203 - 32206	CBAR	Collector 3 Mounts
32200	PBAR, MAT1	
40001-40085	GRID	
43001 - 43060	CQUAD8	Collector 4 (X=+18m) Mirror
44001 - 44024	CTRIA6	
42203 - 42206	CBAR	Collector 4 Mounts
42200	PBAR, MAT1	

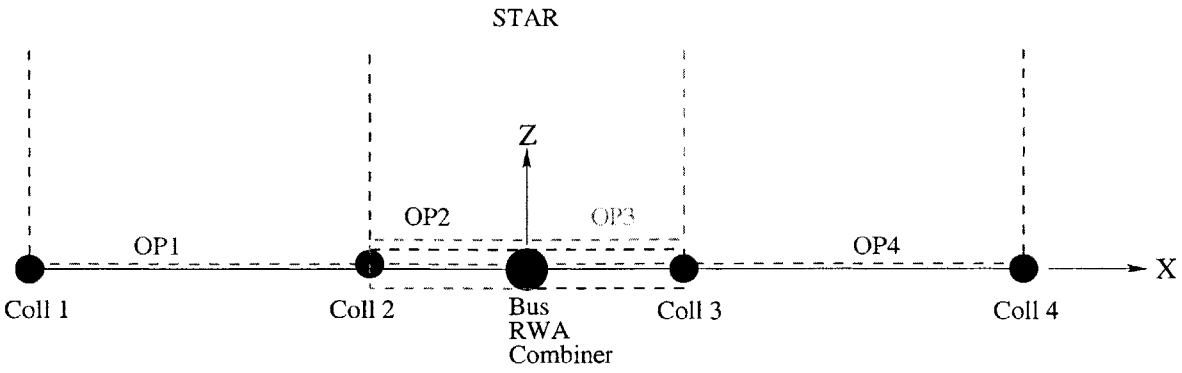


Figure 6-6: Structural model schematic showing optical paths.

the combiner through each collector are all equal to $\frac{B}{2} + R$, where R is the distance from the star to the collector. These paths are shown in in Figure 6-6. However, if the collectors and/or combiner are perturbed from their nominal positions, the path lengths are affected as follows:

$$OP_1 = R - Z_1 + X_c - X_1 + \frac{B}{2} \tag{6.4}$$

$$OP_2 = R - Z_2 - X_c - X_2 + 2x_3 + \frac{B}{2} \tag{6.5}$$

$$OP_3 = R - Z_3 + X_c - 2X_2 + x_3 + \frac{B}{2} \tag{6.6}$$

$$OP_4 = R - Z_4 - X_c + X_4 + \frac{B}{2} \tag{6.7}$$

where Z_i and X_i are the Z and X coordinates of the i^{th} collector, and Z_c and X_c are the Z and X coordinates of the combiner. The OPDs are then found by simply subtracting the relevant path lengths:

$$OPD_{12} = Z_2 - Z_1 + 2X_c - X_1 + X_2 - 2X_3 \tag{6.8}$$

$$OPD_{13} = Z_3 - Z_1 - X_1 + 2X_2 - X_3 \tag{6.9}$$

$$OPD_{14} = Z_4 - Z_1 + 2X_c - X_1 - X_4 \tag{6.10}$$

where OPD_{ij} is the optical path difference between the i^{th} and j^{th} collectors. Equations 6.8 through 6.10 form the optical sensitivity matrix from the instrument position states to the performance metrics.

6.1.4 Attitude Control System

The model of the attitude control system (ACS) is created with the `docs_acs.m` function, a part of the DOCS suite of analysis tools. This function implements a rigid body angular position controller with a bandwidth, f_{acs}, specified as a fraction of the first flexible mode. The bandwith fraction of the ACS is set at $f_{acs} = 0.001$ to keep it well below the first flexible mode of the structure and avoid coupling with the flexible modes. The controller is a proportional-derivative controller with inertia decoupling. The inputs to the controller are three ACS sensor outputs from the plant which are simply the angular motions of the bus node. The controller outputs are torque commands for the RWA and are fed back to the structural plant through a summer element that combines the commanded torques with the disturbance torques from the disturbance model (see Figure 6-1). Since the RWA disturbance model is broadband and therefore not a function of the instantaneous wheel speed the ACS commands simply add to the disturbances and do not drive the vibrations.

6.1.5 Integrated Model Nominal Performance

The individual model components are integrated within MATLAB using the DOCS toolbox. The plant, ACS and RWA isolator component models are transformed to state-space form (Equation 2.6) and integrated to obtain the system model. Since the RWA disturbance model is provided as PSDs, a frequency domain disturbance analysis is run using Equation 2.12 to obtain the RMS of the optical performance metrics over the frequency band of interest. The OPD RMS is the performance used in the PT, RPT and RPTT objective functions. For simplicity, only the RMS OPD between collectors 1 and 4, σ_{14}, is considered in the following sections.

The output PSD from the nominal TPF SCI model is plotted in Figure 6-7 along with the cumulative variance. The RMS OPD between collectors one and four is 15.88 nm. It is clear from the cumulative variance plot that the energy is largely accumulated in the mode at 3.06 Hz. This mode is one of the second bending modes and is shown in Figure 6-8(b). The first bending mode (Figure 6-8(a)) is just under

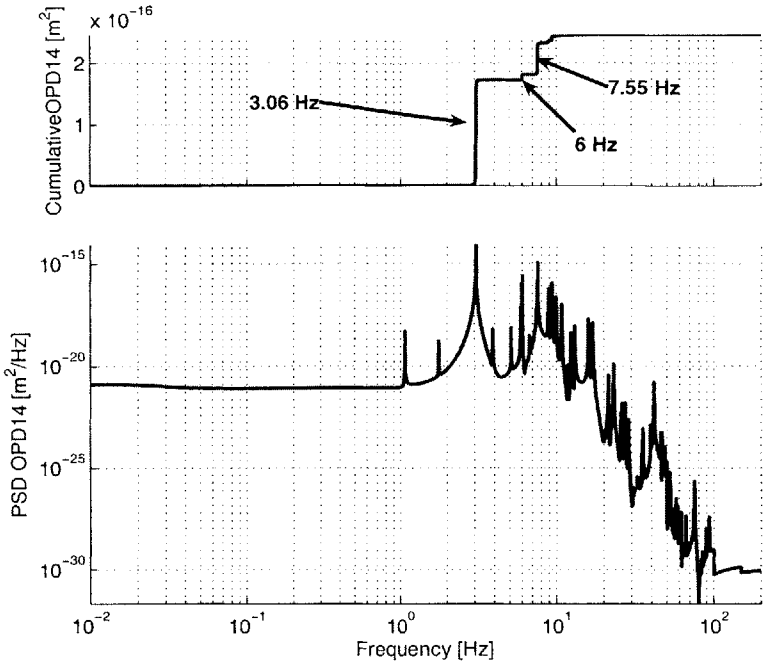
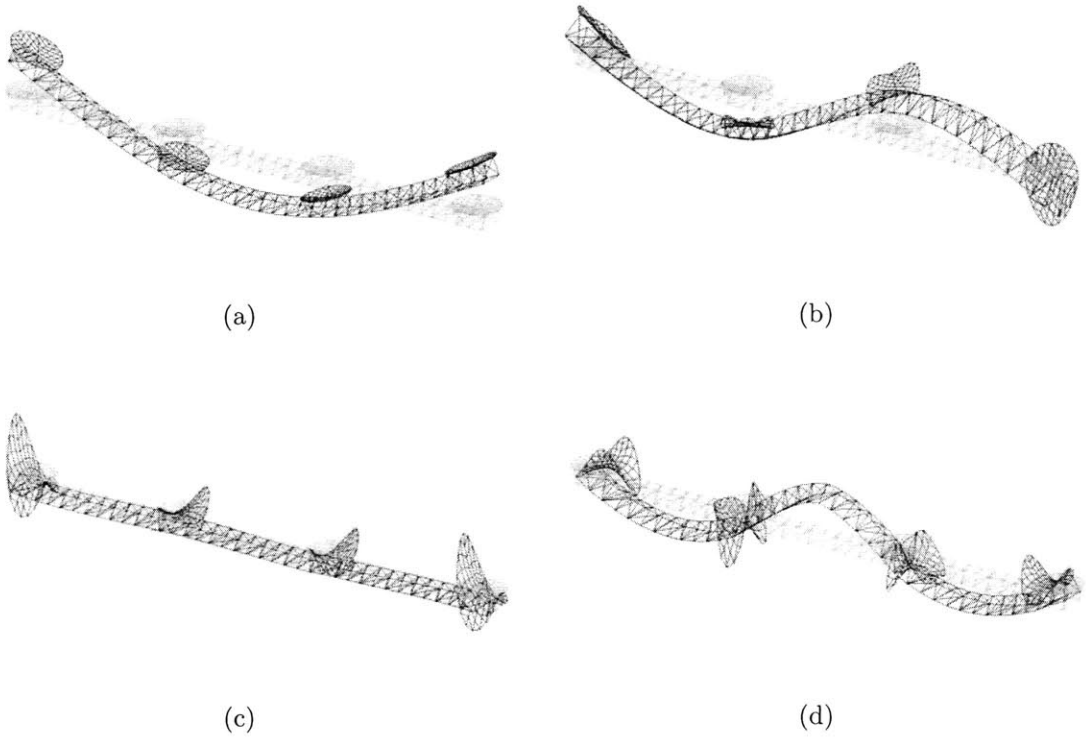


Figure 6-7: Output PSD and cumulative variance from nominal TPF SCI design.

Table 6.9: Critical modes of nominal TPF SCI design.

Mode #	Freq. [Hz]	% Energy (app.).	Description
4	1.07	< 1	truss 1^{st} bending
7	3.06	88.00	truss 2^{nd} bending
13	5.99	2.42	collector 1^{st} bending (potato-chip)
18	7.55	16.7	truss 3^{rd} bending

2 Hz, but does not contribute appreciably to the OPD since it is symmetric and the path lengths change equally on each arm of the interferometer. Additional critical modes under 10 Hz are listed in Table 6.9. The cluster of modes around 6Hz are due to local bending modes of the primary mirrors as shown in Figure 6-8(c). The third bending mode at 7.55 Hz (Figure 6-8(d)) is also a significant contributor. It is interesting to note that, although this model is higher-fidelity than the development model presented in Chapter 2, the global behavior is similar in that the second bending mode is the most critical.



(a) (b)

(c) (d)

Figure 6-8: Mode Shapes of nominal SCI TPF design (a) first bending, 1.07 Hz (b) second bending, 3.06 Hz (c) collector primary mirror, 5.99 Hz (d) third bending, 7.55 Hz.

6.2 Design Parameters

In order to apply the tailoring and tuning design optimizations to the TPF SCI model to improve the RMS OPD, the control, or tailoring and tuning, and noise, or uncertainty, parameters must first be identified. In the following sections these parameters are presented along with appropriate design variable constraints.

6.2.1 Tailoring

Similar to the development model the tailoring parameters allow variation of the truss geometry. However, since the TPF SCI model is a high-fidelity truss structure there are more design options available and a practical and easily visualized tailoring approach is possible. Instead of varying the cross-sectional diameters of each truss segment, the overall geometry of the truss is tailored. To this end, four tailoring parameters are selected as listed in Table 6.10.

Table 6.10: TPF SCI model tailoring parameters.

x	Description	x_0	Units
h_1	outer bay height	1	m
h_2	center bay height	1	m
w_1	outer bay width	2	m
w_2	center bay width	2	m

The values h_1 and h_2 are the height of the batten frames (Figure 6-5(c)) at the end and in the center of the truss, respectively as shown in Figure 6-9. In the nominal configuration these values are set equal to each other so that the batten frames are the same height from bay to bay (Figure 6-2(d)), but for the tailoring optimization they are allowed to change independently so that a configuration such as that shown in Figure 6-9(a) could result. The height of the bays between the end bay and the center bay vary linearly in the x-direction, and the top surface of the truss is always at Z=0 so that the positions of the collectors remain collinear. In order to reduce the number of tailoring parameters and limit the computational effort the truss is assumed to be symmetric about the center so that the batten height at either end is

h_1.

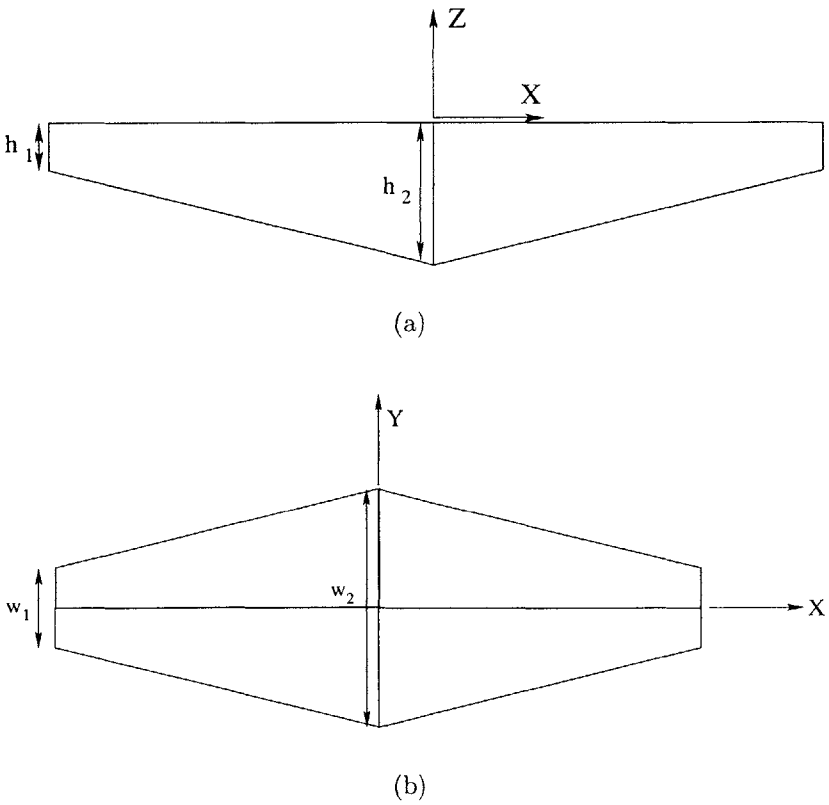


Figure 6-9: Schematic of TPF SCI tailoring parameters (a) XZ-plane (b) XY-plane.

Geometric tailoring is also allowed in the XY-plane through the parameters w_1 and w_2. These parameters are the width of the batten frame at the end and in the center of the array as shown in Figure 6-9(b). The width tailoring is similar to the height tailoring in that the batten widths vary linearly in x and are symmetric about the center of the array. However, in this plane the tailoring is also symmetric about the x-axis, since the positions of the optics are not affected by the batten frame widths.

There are constraints on the tailoring parameters in order to keep the design within practical limits. Both the height and width parameters are subject to a lower bound of 10 cm. In addition, the total mass of the system is constrained to be less than 4000 kg. This number is based on launch vehicle mass limits and ensures that the resulting design is realistic.

6.2.2 Tuning

The tuning parameters are chosen from a small subset of design variables that could practically be adjusted on hardware during component testing or on-orbit operation. Only two tuning parameters are used in order to keep the number of design variables required for the RPTT problem small. The parameters are the cross-sectional radius of the primary mirror supports, r_{PM} and the corner frequency of the RWA isolator, f_{iso}. The nominal tuning parameter values are listed in Table 6.11. Adjusting the parameter r_{PM} is equivalent to changing the stiffness of the optical mount. Although the radii of the support bars are not easily adjusted on the hardware, this parameter is used to model a tunable optical mount stiffness. The isolator corner frequency parameter models a type of active isolator in which the corner frequency can be adjusted through some unspecified mechanism.

Table 6.11: TPF SCI model tuning parameters.

y	Description	y_0	Units
r_{PM}	XS radius of primary mirror support	0.05	m
f_{iso}	RWA isolator corner frequency	8	Hz

The tuning parameters are subject to constraints that keep them within realistic limits. The radius of the primary mirror support structure is constrained by a lower bound of 1 cm to ensure that the primary mirrors stay connected to the truss. Although the radius of the optical mount supports does in fact affect the mass of the system, this parameter is not included in the mass constraint calculation. The reason for this omission is that in reality only the stiffness of the mount will change and not the physical properties of the support. The corner frequency of the RWA isolator is constrained to be between 2 and 10 Hz based on engineering judgment and experience.

6.2.3 Uncertainty

The uncertainty parameters considered for the TPF SCI model are very similar to those in the development model and are listed in Table 6.12. The Young's Modulus of

the truss material is a source of uncertainty and is allowed to vary asymmetrically in the two interferometer arms as in the development model. The resulting asymmetry in the two interferometer arms greatly affects the performance and models the effects of many uncertainty parameters while keeping the computational effort low. A third uncertainty parameter, the modal damping, is added to the parameter set. Damping is a mechanism that is not well-understood and most models of it are conservative approximations to the physical reality. Therefore it is a prime candidate for uncertainty. In this example, a singl modal damping ratio is applied globally to all modes.

Table 6.12: TPF SCI model uncertainty parameters.

p	Description	p_0	Units	Δ [%]
E_1	Young's Modulus of -X truss	111.7	GPa	25
E_2	Young's Modulus of +X truss	111.7	GPa	25
ξ_n	modal damping ratio	0.001	none	40

A bounded uncertainty model of the form in Equation 3.1 is used. The percent bounds on the uncertainty parameters are given by Δ and are listed in Table 6.12. The truss Young's Modulus is allowed to vary $\pm 25\%$ about its nominal value and the modal damping ranges in value $\pm 40\%$ about nominal. The range on the damping parameter is larger than that of the Young's Moduli to capture the high uncertainty inherent in the modal damping model.

6.3 Optimization Implementation

It is shown in the previous chapters that a combination of SA and SQP finds the best design consistently and efficiently in the case of the development model. Unfortunately, running the SQP optimization on the TPF model is not straightforward. Recall from Chapter 2 that the SQP algorithm requires performance gradients, and that the calculation of the gradients requires the gradients of the eigenvalues and eigenvectors. The development model is relatively simple and as a result, these quantities can be derived directly. The TPF model, on the other hand, is much more

complex and the normal modes analysis of the FEM is performed in NASTRAN, not in MATLAB. Therefore, the modal gradients can not be obtained directly. MSC NASTRAN does provide solutions for obtaining eigenvalue and eigenvector gradients, but in the current available NASTRAN version it is difficult to obtain gradient information for all modes. A second option is to use finite-difference gradients (Equation 4.3 or 4.4), however sensitivity of the method to step-size selection leads to inaccurate results. If the step-size is too large, the gradient is beyond the linear regime, but if it is too small the gradient becomes corrupted by numerical errors. Due to these issues, the implementation of a gradient-based optimization method to a model of this size and fidelity is a problem that is left to future study.

As an alternative, simulated annealing alone is used to obtain the PT, RPT and RPTT designs. Although SA does not guarantee an optimum, it has been shown that it can produce a very good design. In order to implement SA it is necessary to evaluate the performance for a random sampling of design variables. The process used to build the model is shown in flow chart form in Figure 6-10. Inputs are indicated by quantities in ovals and the output is an integrated model, SYS($\vec{x}$, $\vec{y}$, $\vec{p}$), that is a function of the tailoring, tuning and uncertainty parameters. To begin, the design parameters are used to generate the finite element model. All design-dependent model data (such as the grid point locations and truss material) is collected in MATLAB and written out to a NASTRAN bulk data deck, `Model.bdf`. This file is combined with the static bulk data found in `Mirrors.bdf` and `Model_Cbar_19TraiBays.bdf`. These files contain model data that remains static, such as the primary mirror elements and the truss connectivity information. A modal analysis of the complete FEM is run in NASTRAN and the results are read into MATLAB by using a custom Perl function to parse the NASTRAN output. The rigid body modes are replaced with geometrically clean modes and the translational rigid body modes are removed. Modal damping is added to the model and the modal quantities are converted to a state-space DOCS structure called PLANT. The PLANT structure is sent to the function `docs_acs.m` to produce an appropriate ACS model. In addition, the tuning parameters are used to generate the isolator model structures, ISO. The RWA disturbance model, DIST,

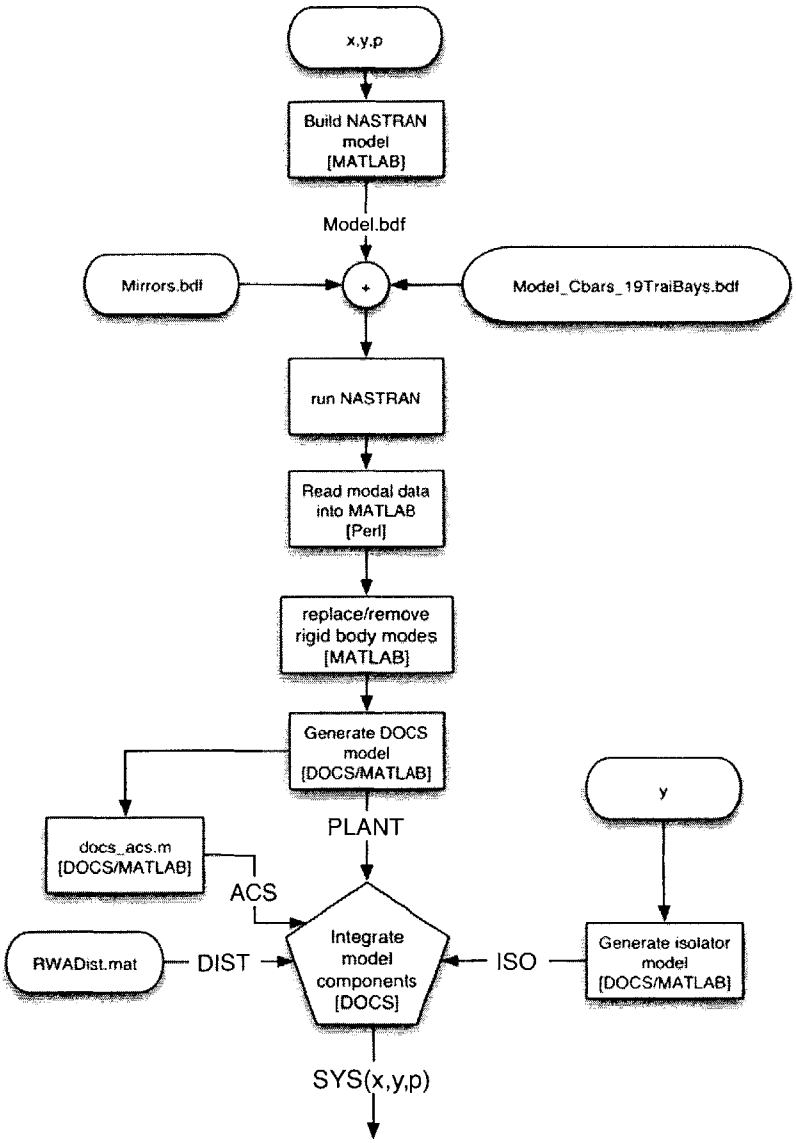


Figure 6-10: Implementation flow for generating integrated TPF model.

does not change with the design parameters, so it is generated a priori and then loaded into the workspace at each iteration. The model components are integrated with DOCS to form the fully-integrated system model and evaluate the performance.

It is important to keep in mind that the process outlined in Figure 6-10 is executed once for each iteration, or random guess, in the SA algorithm. Therefore, the NASTRAN model must be built and a finite element analysis run thousands of times. This operation is computationally expensive and becomes more so as the model fidelity and number of degrees of freedom are increased. In addition, as the design space grows, a greater number of SA iterations are required. It is clear that a brute force method such as the one outlined here may not be appropriate for a model of this fidelity and further investigation into the optimization implementation is warranted. There are known methods for reducing the computational effort required for these types of optimizations [28, 39] that are appropriate for this application, such as reduced-order modeling [37] and response surface approximation [22, 11].

6.4 Results

The TPF SCI structure is designed through a SA stochastic search using the PT, RPT and RPTT cost functions (Equations 2.1, 3.5, and 5.5, respectively). In the RPT and RPTT optimizations the uncertainty space is searched with the vertex method. Since there are three uncertainty parameters, there are eight vertices at which performance must be evaluated during each iteration. The RPTT cost is considered with $\alpha = 0$ to obtain the most tunable design. In the PT and RPT optimizations both the tailoring, $\vec{x}$ and tuning, $\vec{y}$ parameters are used as tailoring parameters to provide for a fair comparison with the RPTT design. In this way all three optimizations have the same design variables available to them and the only difference lies in the manner in which the tuning parameters are incorporated.

The tailoring parameters for the nominal and optimized designs are listed in Table 6.13 along with the mass of the structures. The three designs are distinct from each other and very different from the nominal design. In each case the truss bays

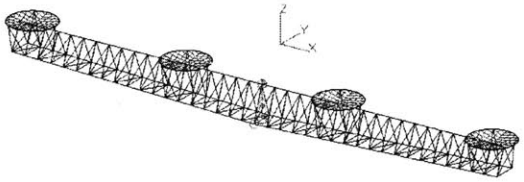
are tailored such that the mass is concentrated towards the center of the array by setting the batten height and width at the ends smaller than at the center. However, the severity of the tailoring is different for each design. The PT design is closet to the nominal design in that the widths are nearly equal along the truss and the bay heights only differ by 0.7 m from the end to the center. The RPT and RPTT designs are tailored more drastically in both bay width and height. The RPTT design has the largest difference between the end and center bay heights (2 m). The mount radius and isolator frequency are both decreased from nominal in the PT and RPT designs. These values are kept at nominal in the RPTT design since tuning is only considered on hardware in this optimization. The finite element models of the three designs are shown in Figures 6-11 through 6-13.

Table 6.13: Design parameters for nominal and optimal TPF SCI designs.

	Tailoring, $\vec{x}$						Total Mass
	h_1 [m]	h_2 [m]	w_1 [m]	w_2 [m]	r_{PM} [m]	f_{iso} [Hz]	[kg]
Nominal	1.0	1.0	2.0	2.0	0.05	8.0	2679.0
PT	1.824	2.522	1.760	1.780	0.03	6.56	2506.6
RPT	0.754	2.52	1.34	2.25	0.02	6.77	2264.2
RPTT	0.50	2.56	1.48	2.34	0.05	8.0	2801.2

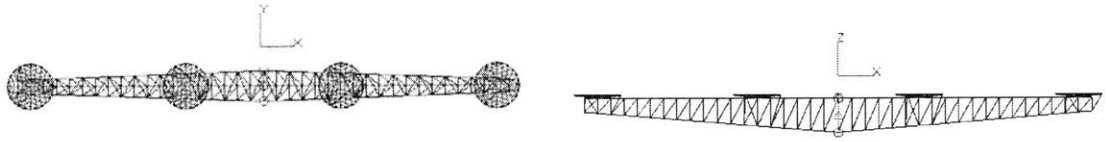
The RMS OPDs between collectors 1 and 4 of each of the three designs are evaluated for the cases of nominal and worst-case uncertainty. The results are listed in Table 6.14 and shown graphically in Figure 6-14. The nominal performance is indicated by circles and the worst-case performance with solid error bars. As in the development model, the PT design exhibits the best nominal performance (2.90 nm), but is sensitive to uncertainty as evidenced by the large performance range between the error bars. The nominal performance of the RPT design is worse than that of the PT design (3.79 nm), but it suffers the least performance degradation at the worst-case uncertainty vertex. The nominal and worst-case performances of the RPTT are worse than those of either the PT or RPT designs. This result is due, in part, to the fact that the tuning parameters are not part of the nominal RPTT design since they are considered explicitly during hardware tuning.

(a) (b)

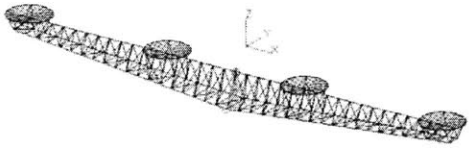


(c)

Figure 6-11: SCI TPF PT design (a) top view (XY) (b) front view (XZ) (c) Isometric view).

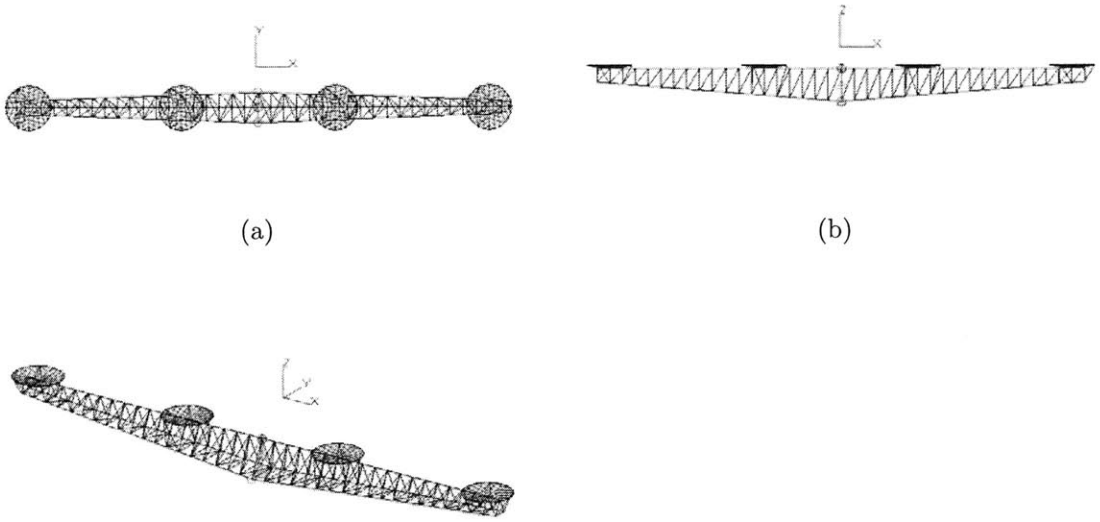


(a) (b)



(c)

Figure 6-12: SCI TPF RPT design (a) top view (XY) (b) front view (XZ) (c) isometric view.



(a) (b)

(c)

Figure 6-13: SCI TPF RPTT design (a) top view (XY) (b) front view (XZ) (c) isometric view.

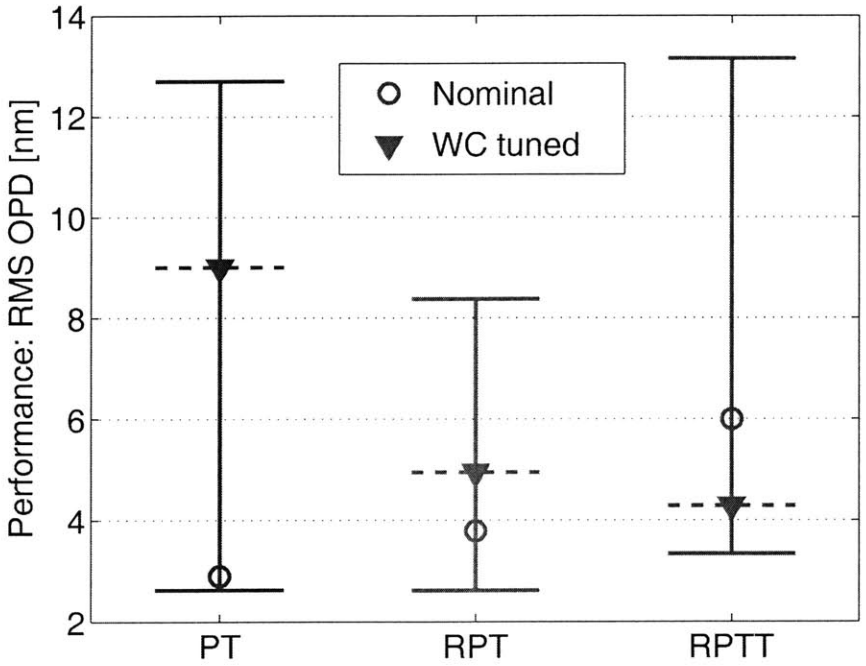


Figure 6-14: Nominal, worst-case and tuned performances for all TPF SCI designs.

Table 6.14: Performance predictions for all TPF SCI designs.

	RMS OPD_{14} [nm]		
	σ_0	σ_{WC}	σ_t
PT	2.90	12.70	9.01
RPT	3.79	8.38	4.95
RPTT	5.99	13.15	4.28

Each design is tuned with the parameters in Table 6.11 at the worst-case uncertainty vertices using Equation 4.2 and SA. This step serves as a simulation of hardware tuning, and the resulting parameter values are listed in Table 6.15. The tuned RMS performances are listed in Table 6.15 and shown graphically in Figure 6-14 with triangles and dotted error bars. As expected, the best tuned performance (4.28 nm) is achieved by the RPTT design. The PT design does not appear to be very tunable as the worst-case performance only improves from 12.8 nm to 9 nm. The RPT design exhibits more favorable tuning properties and can be adjusted to 4.95 nm from the worst-case performance of 8.4 nm. While the overall trend observed is similar to that in the development model (Figure 5-1), it is interesting to note that in the TPF SCI model the RPT design is more tunable than the PT design. In this problem, robustness to uncertainty and tunability are not coupled as they are in the development problem. As a result, the RPTT system is similar in design to the RPT system. However, it still true that explicitly tailoring for tuning improves the performance of the tuned system.

Table 6.15: Tuning parameters for tuned worst-case TPF SCI realizations.

	r_{PM}	f_{iso}
PT	0.07	7.53
RPT	0.213	6.96
RPTT	0.246	6.41

Although the results in Figure 6-14 follow the general trends expected for the PT, RPT and RPTT systems, the design improvement effected by RPTT is not as dramatic as in the development model. One explanation is that the optimization is limited by the lack of gradient information and computational inefficiencies. Recall

from Chapter 2 that SA is only a stochastic search technique, and therefore optimality, local or global, is not guaranteed or even expected. It is simply hoped that randomly searching the design space finds a design that is near-optimal or at least very good. In addition, the performance of the SA algorithm is sensitive to the cooling schedule, or the number of designs that are generated. The design is more likely to be near optimal as the number of iterations increases. However, the complexity of the TPF SCI model leads to large computational expense since a NASTRAN model is generated and analyzed with each performance evaluation. This analysis becomes very costly in the RPT and RPTT optimizations since eight models (one at each uncertainty vertex) must be generated at each design iteration. In fact, evaluating the RPT and RPTT costs for 1000 designs requires over fifteen hours of computation time on a Pentium IV machine with a 3.2 GHz processor. Due to the heavy computational burden the number of SA iterations are limited and the entire design space is not well searched. This problem is especially limiting in the case of the RPTT optimization because the number of design variables is large. A unique set of tuning parameters is allowed for each uncertainty vertex resulting in 30 design variables (4 tailoring parameters plus 2 tuning parameters at each of the eight uncertainty vertices). Therefore, it is possible that RPTT designs better than the one presented here exist.

6.5 Summary

A high-fidelity integrated model of a SCI architecture for TPF is presented in detail. The model consists of a structural model built in NASTRAN, an ACS controller and passive vibration isolation. The model components are integrated in MATLAB with the DOCS tool set and the RMS OPD due to realistic reaction wheel disturbances is evaluated. Appropriate tailoring, tuning and uncertainty parameters are identified, and the model is used to perform PT, RPT and RPTT design optimizations. The performance of the resulting designs is evaluated in the nominal and worst-case uncertainty configurations. The worst-case realizations are then tuned through a dynamic tuning optimization, and the tuned performance is evaluated. It is shown that

performance trends similar to those observed in the development model exist. Although the PT design exhibits the best performance nominally and the RPT design performs best at the worst-case uncertainty realization, it is the RPTT design that obtains the best tuned performance thereby increasing the chance of mission success. However, the RPTT design optimization is hampered by lack of analytical gradients and a large computational burden. It is likely that improvements in the optimization implementation would yield a better RPTT design further extending the capabilities of the TPF SCI system in the presence of model uncertainty.

Chapter 7

Conclusions and Recommendations

7.1 Thesis Summary

Space-based precision optical structures, such as the telescopes in NASA's Origins missions, require high levels of performance to meet science goals. The optics on interferometers, such as the Space Interferometry Mission and the Terrestrial Planet Finder, must meet positional tolerances on the order of nanometers. To further complicate matters, it is not practical to fully test these systems on the ground due to large differences between the testing and operational environments. As a result, models and simulations are expected to play a significant role in the design and testing of these structures. Relying on models to predict future performance is risky as there are many sources of uncertainty that can lead to errors in the model predictions. The design of such high-performance, high-uncertainty systems is challenging because early in the design cycle, when there is maximum design flexibility, the uncertainty is very high. It is often difficult to find a design that can compensate for all of the uncertainty in the model and still meet performance requirements. The uncertainty space is greatly reduced once the design is finalized and hardware is built. However, it may become apparent during testing that the hardware does not meet performance requirements, and only limited adjustments are possible without incurring high cost and/or launch delays. This thesis provides a solution to the problem of system design for high performance and high uncertainty systems that attempts to balance this

trade between design flexibility and prediction accuracy.

First the idea of structural optimization is introduced through a design method termed Performance Tailoring (PT) (Chapter 2). The objective of a PT optimization is to find a set of design parameters that minimize the root mean square (RMS) of the performance metric. A simple model representative of a structurally-connected interferometer (SCI) is introduced as the development model and is described in detail. Methods for evaluating the RMS performance of the structure when subject to a white noise disturbance environment are discussed. Two types of optimization algorithms, a constrained gradient-based search known as sequential quadratic programming (SQP), and a popular heuristic search called simulated annealing (SA) are reviewed. These algorithms are applied to the problem of performance tailoring the development model to minimize the variance of the optical path length difference between the two interferometer arms. The algorithms are compared for performance and efficiency, and the physical mechanisms at work in the tailored design are explored. It is shown that the nominal output RMS can be reduced to one quarter of its value by tailoring the truss diameters such that the truss is very soft and most of the mass is in the collector optics at the ends of the arrays, and the optics are effectively isolated from the disturbance at the array center.

In Chapter 3 the concept of model uncertainty is introduced, and a brief review of model forms and analysis techniques is provided. Uncertainty is added to the development model through the Young's Modulus of the truss segments. An uncertainty analysis is conducted and it is shown that the RMS performance of the PT design degrades dramatically when asymmetry is introduced to the model through the uncertainty parameters. The PT design only performs well at the nominal uncertainty values and covers a large range of performance predictions when the effects of model uncertainty are considered. To address the problem Robust Performance Tailoring (RPT), a subset of the robust design field, is applied. RPT is an design optimization in which the effects of the uncertainty are considered in the objective function through a robustness metric. Three well-developed robust cost functions are considered: anti-optimization, multiple model and a technique that incorporates

a statistical robustness measure such as standard deviation into the objective. The RPT optimizations are applied to the SCI development model, and it is found that anti-optimization provides the most robust design when the uncertainty is modeled as a bounded uniform distribution.

The resulting RPT design is compared to the PT design and it is found that the worst-case performance of the RPT design is significantly lower than that of the PT design. However, the increase in robustness comes at the cost of nominal performance, so that the RPT has a higher RMS performance at the nominal uncertainty values than the PT design. A comparison of the worst-case performance values of the two designs over a range of uncertainty bounds shows that RPT designs can tolerate greater variation in the uncertainty parameters at a specific performance requirement than the PT design. However, since nominal performance is sacrificed, RPT designs cannot meet aggressive performance requirements given high levels of uncertainty.

The RPT designs fail when uncertainty is high because it is not possible to find values of the tailoring parameters that meet requirements under all possible uncertainty realizations. Therefore, in Chapter 4, the concept of dynamic tuning is introduced. It is defined as the adjustment of physical tuning parameters on hardware that brings its performance to within required values. A tuning optimization is formulated and applied to the PT and RPT designs in the worst-case uncertainty realizations. It is shown that in both cases, dynamic tuning succeeds in reducing the performance variance of the worst-case design, however, the effect is more dramatic on the PT design. The worst-case uncertainty realizations of both designs, over a range of uncertainty bounds, are tuned, and it is shown that dynamic tuning extends the range of performance requirements that each design can meet at a given uncertainty level.

The practical issues associated with dynamic tuning are also considered. In reality only limited performance data is available from the hardware and the actual values of the uncertainty parameters are not known. Therefore some type of model updating technique or real-time hardware tuning is necessary to improve the hardware performance. A set of possible hardware tuning methods, ranging from model-only methods that rely on the uncertain model to find successful tuning configurations to

real-time hardware optimizations that do not utilize the model at all, are explored. Hardware simulations are generated with the SCI development model, and it is found that although the model-only methods are attractive in that few costly hardware tests are required, they are not consistently successful, and sometimes the resulting tuning configurations actually degrade the hardware performance. Hardware optimization methods, on the other hand, find a working tuning configuration (given that one exists), but generally require a large number of costly hardware tests. As an alternative, a hybrid method called isoperformance tuning is developed that utilizes the limited hardware performance data that is available to reduce the uncertainty space so that a robust tuning optimization performed on the model results in a tuning configuration that successfully improves the hardware performance. This method is superior to the model-only and hardware optimization methods in that is consistently successful for a large number of hardware simulations and requires only a small number of hardware tests.

Finally the concepts of robust performance tailoring and dynamic tuning are combined to create a design methodology called Robust Performance Tailoring for Tuning (RPTT). This design optimization anticipates the fact that hardware tuning may be employed and tailors the design to balance robustness to uncertainty and tuning authority thereby utilizing a two-step uncertainty mitigation scheme . The additional knowledge of the uncertainty provided by the hardware is anticipated by the tailoring optimization and the design variables are augmented to include different tuning parameters for each uncertainty realization in order to minimize the worst-case tuned performance over the uncertainty space. The result is a design that produces a robust system, instead of simply a robust design. Three different mathematical formulations of the RPTT optimization are presented and compared through application to the SCI development problem with both SA and SQP algorithms. The nominal, worst-case and tuned performances of the resulting RPTT design are compared to those of the PT and RPT designs over a range of uncertainty bounds. The nominal and worst-case performances of the RPTT design lie in between those of the PT and RPT designs. The nominal performance RMS is a little higher than that of the PT design,

and the worst-case performance is above that of the RPT design. The tuned performance of the RPTT designs, however, is better than that of either the PT or RPT designs for all uncertainty levels. The RPTT design further extends the performance of the system at a given uncertainty level, and is applicable to all design regimes defined by combinations of PT, RPT and tuning. Random hardware simulations of the three designs are run to evaluate the design performance over the entire uncertainty space. The RPTT design is the clear winner in that it is the only design for which all hardware simulations either meet the requirements nominally or after hardware tuning is employed.

7.2 Contributions

This thesis develops a novel approach to the design of high-performance, high-uncertainty systems in which dynamic tuning is anticipated and formalized in a design optimization. This design methodology, RPTT, is especially applicable to high-precision optical space systems, such as the Space Interferometry Mission (SIM), the James Webb Space Telescope (JWST) and the Terrestrial Planet Finder (TPF). Specific thesis contributions are as follows:

- Development of a design methodology, RPTT, that formalizes a complimentary relationship between dynamic tailoring and tuning. RPTT extends robust design for application to systems that require high levels of performance and exhibit high uncertainty.

- Development of a model updating technique for application to dynamic tailoring that utilizes limited hardware performance data and isoperformance to reduce the parametric uncertainty space so that a robust tuning optimization performed on the model yields tuning parameter values that successfully tune the hardware.

- Study of gradient-based and heuristic optimization techniques for application to tailoring and tuning given a dynamic performance model. Eigenvector and

eigenvalue derivatives are used to obtain RMS gradients with respect to design variables for use in gradient-based optimizations.

- Application of the robust design framework to a simple model of a structurally connected interferometer. Realistic control (tailing and tuning) and noise (uncertainty) factors are identified.

- Application of dynamic tailoring and tuning optimizations to a simple model of a structurally connected interferometer. Three tailoring methods are compared in terms of ability to meet performance requirements given model uncertainty. In addition, three existing robust cost functions are applied and compared in terms of the robustness of the resulting design.

- Application of dynamic tailoring and tuning optimizations to a high-fidelity model of a structurally-connected TPF interferometer. NASTRAN and MATLAB are integrated to perform a heuristic multi-disciplinary design optimization.

7.3 Future Work

The recommendations for future work focus on four main areas: uncertainty modeling, RPTT methodology, isoperformance tuning and application. An itemized list of specific recommendations are provided below:

- Uncertainty Modeling

  - Consider sources of uncertainty other than parametric errors. Explore the ability of the RPTT framework to compensate for discretization errors and unmodelled effects.

  - Consider additional forms of uncertainty models. In particular, examine the effect of RPTT when probabilistic uncertainty models are used in place of bounded.

- Consider uncertainty analysis tools other than the vertex method and Monte Carlo propagation for use in robust tailoring optimizations.
- Evaluate the effects of measurement noise on the dynamic tuning methodologies through simulations.

- Robust Performance Tailoring for Tuning Methodology
  - Repeat the analysis considering coupling among the tailoring, tuning and uncertainty parameters. Assess the effects of uncertainty in the tailoring and tuning parameters on the performance of the RPTT designs. Modify the methodology as necessary to accommodate this coupling.
  - The design space and effectiveness of the RPTT methodology is highly dependant on the tailoring and tuning parameters chosen in a given problem. Formalize the determination of appropriate tailoring and tuning parameters. Develop metrics to measure the tailoring and tuning authority afforded by a design parameter as well as the practical cost of adjusting it. Study these trades and develop a formal method for choosing the most appropriate design parameters for a given system.
  - Apply the framework to additional structures to draw general conclusions regarding tailoring and tuning. For example, further explore the trade inherent in the SCI development model between tunability and robustness by investigating a range of systems to determine under what conditions such a trade exists.
  - Design and conduct experiments to validate the performance of the RPTT methodology in the design of a real system.

- Isoperformance Tuning
  - Extend isoperformance tuning for application to more than two uncertainty parameters using existing n-dimensional isoperformance algorithms. Adjust the tuning method as necessary to account for the additional uncertainty dimensions.

- Design and conduct experiments to test isoperformance tuning using real hardware data and an uncertainty model.

- Application

  - Explore alternate optimization algorithms for application to problems of this type in order to decrease computational effort required for tailoring optimizations. Implement reduced-order modeling or response-surface approximations.

  - Explore the possibility of conducting gradient-based optimizations on high-fidelity integrated models. Consider the complex-step derivative approximation [80] to avoid step-size limitations associated with a finite difference approach. Investigate use of FEM packages such as NASTRAN to obtain the necessary gradients.

Appendix A

Gradient-Based Optimization

The goal of a general unconstrained nonlinear programming optimization problem is to minimize a convex objective function, f, over a set of design variables, x defined in the set of real numbers, $\mathcal{R}^n$:

$$\min_{x \in \mathcal{R}^n} f(x) \tag{A.1}$$

One way to find the optimal set of parameters, x^*, is to begin at some initial iterate, x_0 and search the design space by finding successive iterates, x_k that reduce the objective funcion. A general form for the iterate is:

$$x_{k+1} = x_k + \alpha_k d_k \tag{A.2}$$

where k denotes the iteration number, α_k is the stepsize and d_k is a serach direction. It is the choice of the search direction that distinguishes one optimization algorithm from another. In gradient-based optimizations d is chosen based on the gradient of the cost function at each iteration. There is a large body of work on this subject and many algorithms from which to choose. The choice of algorithm depends on the features of the given problem formulation. Bertsekas provides a detailed overview of many popular gradient methods in [16]. In this appendix, three popular gradient-based algorithms are briefly reviewed.

A.1 Steepest Descent

The simplest gradient-based optimization algorithm is known as steepest descent. The search direction for the k^{th} iterate is simply the value of the cost function gradient at x_k:

$$d_k = -\nabla f(x_k) \tag{A.3}$$

where ∇f is the derivative of f with respect to the design vector x:

$$\nabla f = \frac{\partial f(x)}{\partial x} = \left\{ \begin{array}{cccc} \frac{\partial f(x)}{\partial x_1} & \frac{\partial f(x)}{\partial x_2} & \cdots & \frac{\partial f(x)}{\partial x_n} \end{array} \right\} \tag{A.4}$$

Although Equation A.3 is the simplest form of the steepest descent direction, many modifications, including scaled directions, exist.

Although it is computationally prefereable to compute the cost function gradient analytically, a closed-form solution is not always be availble. In these cases, the gradient can be approximated with one of the following finite difference approximations:

$$\frac{\partial f(x)}{\partial x_i} = \frac{f(x + \Delta x e_i) - f(x)}{\Delta x} \tag{A.5}$$

$$\frac{\partial f(x)}{\partial x_i} = \frac{f(x + \Delta x e_i) - f(x - \Delta x e_i)}{2\Delta x} \tag{A.6}$$

where i denotes an element of x, e_i is a unit vector with a 1 in the i^th location, and Δx is a small change in the design parameter. Equations A.5 and A.6 are known as the forward and central finite difference approximations, respectively. The central difference equation is the more accurate of the two, but requires an additional function evaluation at each step. These approximations are both sensitive to the size of Δx. Large parameter changes may be outside of the linear approximation to the function while very small changes can induce numerical instability.

The main advantage of steepest descent is its ease of implementation. The algorithm gurantees that f decreases at each iteration, but often exhibits slow convergence, especially in the unscaled formulation of Equation A.3. In particular, the algorithm will take a long time to converge if the solution space is relatively flat at the optimum,

such that the gradient direction is almost orthogonal to the direction leading to the minimum.

A.1.1 Stepsize Selection

There many possible choices for the stepsize, α_k. Perhaps one of the simplest is the constant stepsize, in which a fixed stepsize, $s > 0$, is selected for all iterates:

$$\alpha_k = s, \quad k = 0, 1, \ldots \tag{A.7}$$

Although easy to implement, a constant stepsize can cause the algorithm to diverge if s is too large or result in very slow convergence if s is too small.

Another relatively simple rule is the diminishing stepsize in which α_k converges to zero with each successive iteration:

$$\alpha_k \to 0 \tag{A.8}$$

This rule does not guarantee descent and it is possible that the stepsize may become so small that progress is effectively halted before an optimum is reached. To prevent this from happening the following condition is imposed:

$$\sum_{k=0}^{\infty} \alpha_k = \infty \tag{A.9}$$

$$\alpha_k = \frac{\alpha_0}{\sqrt{k}} \tag{A.10}$$

Equation A.10 is a possible choice for a decreasing stepsize rule. Convergence with this rule is also known to be slow.

A more complicated but better-behaved method is the limited minimization rule:

$$f\left(x_k + \alpha_k d_k\right) = \min_{\alpha \in [0,s]} f\left(x_k + \alpha d_k\right) \tag{A.11}$$

In this method, α_k is chosen such that the cost function is minimized along the direction d_k. A fixed positive scalar, s, may be chosen to limit the possible size of

α_k. A one-dimensional line search is generally used to find the minimizing stepsize. In general, this value is not found exactly, instead a termination criteria is used to determine when the line search algorithm has gotten close enough. Bertsekas suggests implementations for the line search in [16].

A.2 Newton's Method

In Newton's method a quadratic approximation to f around the current iterate x_k is minimized. The descent direction, therefore, includes second-order information:

$$d_k = -\left(\nabla^2 f\left(x_k\right)\right)^{-1} \nabla f\left(x_k\right) \tag{A.12}$$

The quantity, $\nabla^2 f$ is the second-order derivate of the objective with respect to the design variables and is known as the Hessian:

$$\nabla^2 f = \frac{\partial^2 f\left(x\right)}{\partial x^2} = \begin{bmatrix} \frac{\partial^2 f(x)}{\partial x_1^2} & \frac{\partial^2 f(x)}{\partial x_1 x_2} & \cdots & \frac{\partial^2 f(x)}{\partial x_1 x_n} \\ \frac{\partial^2 f(x)}{\partial x_2 x_1} & \frac{\partial^2 f(x)}{\partial x_2^2} & \cdots & \frac{\partial^2 f(x)}{\partial x_2 x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f(x)}{\partial x_n x_1} & \frac{\partial^2 f(x)}{\partial x_n x_2} & \cdots & \frac{\partial^2 f(x)}{\partial x_n^2} \end{bmatrix} \tag{A.13}$$

As with the gradients, the Hessian may be calculated ananlytically or approximated through finite-difference or other methods. If the central difference equation (Equation 4.4) is used to calculate gradients, then the diagonal elements of the Hessian can be obtained at very little additional expesne:

$$\frac{\partial^2 f\left(x\right)}{\partial x_i^2} = \frac{f\left(x + \Delta x \vec{e_i}\right) - f\left(x - \Delta x e_i\right) - 2f\left(x\right)}{\Delta x^2} \tag{A.14}$$

Newton's method is very popular due to its fast convergence properties. In fact, the method finds the global minimum of a positive definite quadratic function in only one iteration. For this reason, many modified algorithms are based on Newton's method. One drawback of the tecqhnique is that the Hessian is required to determine

the descent direction. In some cases, calculation of the Hessian may be impossible or require prohibitively expensive computation. In addition, if the Hessian matrix is ill-conditioned then the optimization may have trouble converging.

A.3 Conjugate Gradient

Conjugate direction methods were originally developed for solving quadratic problems and were motivated by a desire to speed up the convergence of steepest descent while avoiding the exta storage and computation necessary for Newton's method. The algorithm can also be applied to the general non-quadratic problem (Equation A.1) and proceeds with successive iterations as in Equation A.2. The stepsize, α_k, must be obtained through line minimization ((A.11)), and the search directions for this method are functions of the gradient at the current iteration as well as the previous search direction:

$$d_k = -\nabla f(x_k)) + \beta_k d_{k-1} \tag{A.15}$$

There are multiple forms of the quantity β_k. One form, known as the Fletcher-Reeves conjugate direction [45], requires a quadratic objective function $f(x)$ and a perfect line search for the stepsize. Another common form is the Polak-Ribiere conjugate direction which relaxes the quadratic requirement on $f(x)$, but only prodcues conjugate directions if the line search is accurate. A more general direction, requiring neither a quadratic objective nor a perfect line search, is proposed by Perry[101]:

$$\beta_k = \frac{(\nabla f(x_k) - \nabla f(x_{k-1}) - \alpha_{k-1} d_{k-1})^T \nabla f(x_k)}{(\nabla f(x_k) - \nabla f(x_{k-1}))^T d_{k-1}} \tag{A.16}$$

The conjugate gradient method is essentially a compromise between the simplicity of steepest descent and fast convergence of Newton's method. When there are non-quadratic terms in the cost function there is the danger for loss of conjugancy as the algorithm progresses. Therefore it is common practice to periodically restart the algorithm with a steepest descent direction.

Bibliography

[1] Jpl planet quest website: Space interferometry mission [online, cited February 13, 2004]. Available from: http://planetquest.jpl.nasa.gov/SIM/sim_index.html.

[2] Jpl planet quest website: Terrestrial planet finder [online, cited February 13, 2004]. Available from: http://planetquest.jpl.nasa.gov/TPF/tpf_index.html.

[3] Nasa origins website [online, cited February 13, 2004]. Available from: http://origins.jpl.nasa.gov/missions/index.html.

[4] Northern lights optics [online, cited September 20, 2004]. Available from: http://www.minerals.sk.ca/atm_design/materials/ule.html.

[5] E. Anderson. *Robust Placement of Actuators and Dampers for Structural Control.* PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, October 1993.

[6] J.R.P. Angel and N. J. Woolf. An imaging nulling interferometer to study extrasolar planets. *The Astrophysics Journal*, 475(1):373–379, January 1997.

[7] D. K. Anthony and S. J. Elliott. On reducing vibration transmission in a two-dimensional cantilever truss structure using geometric optimization and active vibration control techniques. *Journal of Accoustical Society of America*, 110(2):1191–1194, 2001.

[8] J. T. Armstrong, D. J. Hutter, K. J. Johnston, and D. Mozurkewich. Stellar optical interferometry in the 1990s. *Physics Today*, 48:42–49, May 1995.

[9] J.T. Armstrong, D. Mozurkewich, L. J. Rickard, D. J. Hunter, J. A. Benson, P. F. Bowers, N. M. Elias II, C. A. Hummel, K. J. Johnston, D. F. Buscher, J. H. Clark III, L. Ha, L.-C. Ling, N. M. White, and R. S. Simon. The navy prototype optical interferometer. *The Astrophysical Journal*, 496:550–571, March 1998.

[10] A. Ashkenazi and A. E. Bryson. Control logic for parameter insensitivity and disturbance attenuation. *Journal of Guidance, Control and Dynamics*, 5(4):383–388, July-August 1982.

[11] P. J. Attar and E. H. Dowell. A stochastic analysis of the limit cycle behavior of a nonlinear aeroelastic model using the response surface method. In *AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials*, Austin, TX, April 2005. AIAA. accepted for presentation.

[12] P. Avitabile. Model updating: Endlesss possbilities. In *Proceedings IMAC-XVIII: A Conference on Structural Dynamics*, volume 1, pages 562–570, San Antonio, TX, 2000. Society for Experimental Mechanics/SPIE.

[13] C. A. Beichman, N. J. Woolf, and C. A. Lindensmith. *Terrestrial Planet Finder.* Jet Propulsion Laboratories, May 1999. Prepared by: TPF Science Working Group.

[14] W. K. Belvin and K. C. Park. Structural tailoring and feedback control synthesis: An interdisciplinary approach. *J. Guidance*, 13(3):424–429, May-June 1990.

[15] Y. Ben-Haim and I Elishakoff. *Convex Methods of Uncertainty in Applied Mechanics.* Elsevier Science Publishers, New York, NY, 1990.

[16] D. P. Bertsekas. *Nonlinear Programming.* Athena Scientific, Belmont, MA, 1999. Second Edition.

[17] W. B. Bickford. *Mechanics of Solids: Concepts and Applications.* Richard D. Irwin, Inc., 1993.

[18] M. A. Blair, Jr. J. W. Sills, and A. Semple. Determination of model uncertainty factor using cross-orthogonality and overall load factor decomposition. In *Proceedings of the 12th International Modal Analysis Conference*, volume 1, pages 613–618, Honolulu, Hawaii, 1994. Society for Experimental Mechanics, Inc.

[19] F. Bourgault. Model uncertainty and performance analysis for precision controlled space structures. Master's thesis, Massachusetts Institute of Technology, Cambridge, MA, Deceber 2000.

[20] R. N. Bracewell. Detecting nonsolar planets by spinning infrared interferometer. *Nature*, 274(24):780–781, August 1978. letter.

[21] R. N. Bracewell and R. H. MacPhie. Searching for nonsolar planets. *Icarus*, 38:136–147, 1979.

[22] A. J. Bronowicki. Structural modification to minimize response of electro-optical systems to random excitation. In *Structural mechanics of optical systems II*, pages 68–80, Los Angeles, CA, Jan. 13-15 1987. Society of Photo-Optical Instrumentation Engineers. A88-34495.

[23] M. Campbell and E. F. Crawley. Devlopement of structural uncertainty models. *Journal of Guidance Control and Dynamics*, 20(5):841–849, Septmeber-October 1997.

[24] Nightsky Systems Carl Blaurock. Docs (disturbance optics controls structures) users manual. Version 1.7, 2004.

[25] P. D. Cha and L. G. Pillis. Model updating by adding known masses. *International Journal for Numerical Methods in Engineering*, 50:2547–2571, 2001.

[26] G Chen, R Bruno, and M. Salama. Optimal placement of active/passive members in truss structures using simulated annealing. *AIAA Journal*, 29(8):1327–1334, August 1991.

[27] G Chen and D. J. Ewins. A perspective on model updating performance. In *Proceedings IMAC-XVIII: A Conference on Sturctural Dynamics*, volume 2, pages 999–1005, San Antonio, TX, 2000. Society for Experimental Mechanics/SPIE.

[28] T. Y. Chen and J. J. Su. Efficiency improvement of simulated annealing in optimal structural designs. *Advances in Engineering Software*, 33:675–680, 2002.

[29] M. M. Colavita, A. F. Boden, S. L. Crawford, A. B. Meinel, M. Shao, P. N. Swanson, G. T. van Belle, G. Vasisht, J. M. Walker, J. K. Wallace, and P. L. Wizinowich. The keck interferometer. In *SPIE Conference on Astronomical Interferometry*, pages 776–784, Kona, Hawaii, March 1998. SPIE.

[30] M. M. Colavita and P. L. Wizinowich. Keck interferometer: progress report. In Pierre J. Lena and Andreas Quirrenbach, editors, *Optical Astronomy, Proceedings of the SPIE*, pages 310–3203. SPIE, 2000.

[31] E. F. Crawley, B. P. Masters, and T. T. Hyde. Conceptual design methodology for high performance dynamic structures. In *AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference and Exhibit*, pages 2060–2073, New Orleans, LA, April 10-13 1995. AIAA. AIAA-95-1407.

[32] J. Davis, A. Mendez, J. W. O'Byrne, B. Seneta, W. J. Tango, and P. G. Tuthill. The sydney university stellar interferometer program. In Pierre J. Lena and Andreas Quirrenbach, editors, *Optical Astronomy, Proceedings of the SPIE*, pages 444–451. SPIE, 2000.

[33] J. Davis, W. J. Tango, A. J. Booth, T. A. ten Brummelaar, R. A. Minard, and S. M. Owens. The sydney university stellar interferometer - i. the instrument. *Mon. Not. R. Astron. Soc.*, 303:773–782, 1999.

[34] J. Davis, W. J. Tango, A. J. Booth, E. D. Thorvaldson, and J. Giovannis. The sydney university stellar interferometer - ii. commissioning. *Mon. Not. R. Astron. Soc.*, 303:783–791, 1999.

[35] O. L. de Weck. Integrated modeling and dynamics simulation for the next generation space telescope. Master's thesis, Massachusetts Institute of Technology, Cambridge, MA, 1999.

[36] O. L. de Weck. *Multivariable Isoperformance Methodology for Precision Opto-Mechanical Systems.* PhD thesis, Massachuestts Institute of Technology, Cambridge, MA, September 2001.

[37] O. L. de Weck, S. A. Uebelhart, H. L. Gutierrez, and D. W. Miller. Performance and sensitivity analysis for large order linear time-invariant systems. In *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Atlanta, GA, Sept. 4-6 2002. American Institute of Aeronautics and Astronautics. AIAA Paper 2002-5437.

[38] M. Delbo, A. W. Harris, R. P. Binzel, P. Pravec, and J. K. Davies. Keck observations of near-earth asteroids in the thermal infrared. *Icarus*, 116:116–130, 2003.

[39] I. N. Egorov, G. V. Kretinin, and I. A. Leshchenko. Two approaches to multidisciplinary optimization problems. In *European Congress on Computational Methods in Applied Sciences and Engineering*, Barcelona, Spain, Septmber 2000.

[40] Igor N. Egorov, G. V. Kretinin, and I. A. Leshchenko. How to execute robust design optimization. In *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimzation*, page 7, Atlanta, Georgia, 4-6 September 2002. American Institute of Aeronautics and Astronautics. AIAA Paper 2002-5670.

[41] I. Elishakoff. Convex versus probabilistic modelling of uncertainty in structural dynamics. In M. Petyt, H.F. Wolfe, and C. Mei, editors, *Structural Dynamics: Recent Advances*, pages 3–21. Elsevier Applied Science Publishers, 1991.

[42] I Elishakoff, R. T. Haftka, and J. Fang. Structural design under bounded uncertainty – optimization with anti-optimization. *Computers & Structures*, 53(6):1401–1405, 1994.

[43] R. P. Feynman. *What do You Care What Other People Think?* Bantam Books, New York, New York, 1988.

[44] R. Fletcher. *Practical Methods of Optimization.* John Wiley and Sons, 1987.

[45] R. Fletcher and C. M. Reeves. Function minimization by conjugate gradients. *Computer Journal*, 7(2):149–154, July 1964.

[46] R. M. Glaese and G. L. Bales. Demonstration of dynamic tailoring for gossamer structures. In *45th AIAA/ASME/AHS/ASC Structures, Structural Dynamics and Materials Conference*, page 12, Palm Springs, CA, April 19-22 2004. AIAA. 2004-1824.

[47] S. C. O. Grocott, J. P. How, and D. W. Miller. A comparison of robust control technqiues for uncertain structural systems. In *Proceedings of AIAA Guidance, Navigation and Control Conference*, pages 261–271, Scottsdale, AZ, August 1994. AIAA.

[48] S. C. O. Grocott, J. P. How, and D. W. Miller. Experimental comparison of robust h2 control techniques for uncertain structural systems. *Journal of Guidance, Control and Dynamics*, 20(3):611–614, May 1997. Engineering Note.

[49] R. Grogan and R. Laskin. On multidisciplinary modeling of the space interferometry mission. In *Proccedings of the American Control Conference*, pages 1558–1562, Philadelphia, Pennsylvania, June 24-26 1998. Institute of Electrical and Electronics Engineers.

[50] H. L. Gutierrez. *Performance Assessment and Enhancement of Precision Controlled Strcutures During Conceptual Design.* PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, February 1999.

[51] O. Hasancebi and F. Erbatur. Layout optimisation of trusses using simulated annealing. *Advances in Engineering Software*, 33:681–696, 2002.

[52] T. Hasselman. Quantification of uncertainty in structural dynamic models. *Journal of Aerospace Engineering*, 14(4):158 –165, October 2001.

[53] T. Hasselman, J. D Chrostowski, and T. J. Ross. Interval prediction in structural dynamics analysis. In *Proceedings of 33rd AIAA/ASME/AHS/ASC Structures, Structural Dynamics and Materials Conference*, pages 1272–1284, Reston, VA, 1992. AIAA. AIAA-92-2215.

[54] E. Hecht. *Optics.* Addison Wesley, 2002. Fourth Edition.

[55] F. M. Hemez. Can model updating tell the truth. In *Proceedings 16th Intermational Modal Analysis Conference*, volume 1, pages 7–Jan, Santa Barbara, CA, 1998. Society for Experimental Mechanics/SPIE.

[56] L. Holzleitner and K. G. Mahmoud. Structural shape optimization using msc/nastran and sequential quadratic programming. *Computers and Structures*, 70:487–514, 1999.

[57] D. J. Howell. Mutli-configuration model tuning for precision opto-mechanical systems. Master's thesis, Massachusetts Institute of Technology, Cambrdge, MA, February 2004.

[58] C. A. Hummel. The practice of interferometry with npoi. In Pierre J. Lena and Andreas Quirrenbach, editors, *Interferometry in Optical Astronomy, Proceedings of SPIE*, pages 459–464. SPIE, 2000.

[59] D. J. Hutter, N. M. Elias II, and C. A. Hummel. First astrometric results from the npoi. In *Proccedings of the SPIE Conference on Astrononmical Interferometry*, pages 452–460, Kona, Hawaii, March 1998. SPIE.

[60] T. Janter and P. Sas. Uniqueness aspects of model updating procedures. *AIAA Journal*, 28:538–543, 1990.

[61] C. D. Jilla. *A Multiobjective, Multidisciplinary Design Optimization Methodology for the Conceptual Design of Distributed Satellite Systems.* PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, May 2002.

[62] S. S. Joshi, M. M. Milman, and J. W. Melody. Optimal passive damper placement methodology for interferometers using integrated structures/optics modeling. In *AIAA Guidance, Navigation, and Control Conference*, pages 106–113, New Orleans, LA,, Aug. 11-13 1997. American Institute of Aeronautics and Astronautics.

[63] A. J. Keane. Passive vibration control via unusual geometries: The application of genetic algorithm optimization to structural design. *Journal of Sound and Vibration*, pages 441–453, 1995.

[64] H. Kim, O. M. Querin, and G. P. Steven. On the development of structural optimisation and its relevance in engineering design. *Design Studies*, 23(1):85–102, Januray 2002.

[65] S. Kirkpatrick, M. P. Vecchi, and C. D. Gelatt. Optimization by simulated annealing. *Science*, 220(4598):671–680, May 1983.

[66] U. Kirsch. *Optimum Structural Design.* McGraw-Hill Inc., 1981.

[67] N. D. Lagaros, M. Papadrakakis, and G. Kokossalakis. Structural optimization using evolutionary algorithms. *Computers and Structures*, 80:571–589, 2002.

[68] R. S. Langley, N. S. Bardell, and P. M Loasby. The optimal design of near-periodic structures to minimize vibration transmission and stress levels. *Journal of Sound and Vibration*, 207(5):627–646, 1997.

[69] R. Laskin. Sim technology development overview - light at the end of the tunnel. In *Interferometry in Space*, pages 16–32, Waikoloa, HI, August 26-28 2002. Society of Photo-Optical Instrumentation.

[70] P Lawson. The terrestrial planet finder. In *Aerospace Conference*, volume 4, pages 2005–2011, Big Sky, MT, March 10-17 2001. IEEE.

[71] A. Leger, J. M. Mariotti, B. Mennesson, M. Ollivier, J. L. Puget, D. Rouan, and J. Schneider. Could we search for primitive life on extrasolar planets in the near future? the darwin project. *ICARUS*, 123(2):249–255, October 1996.

[72] J. Leitner, F. Bauer, R. Carpenter, M. Moreau, and J. P. How. Formation flight in space. *GPS World*, pages 22–31, February 2002.

[73] J. P. Levia, B. P. Watson, and I. Kosaka. Modern structural optimization concepts applied to topology optimization. In *30th AIAA Plasmadynamics and Lasers Conference*, Norfolk, VA, June 28-July 1 1999. AIAA.

[74] M. Lieber. Development of ball integrated telescope model. In Torben Andersen, editor, *Proceedings of SPIE: Integrated Modeling of Telescopes*, volume 4757, pages 19–30. SPIE, July 2002.

[75] M. Lieber, C. Noecker, and S. Kilston. Integrated systems modeling for evaluating the coronograph approach to planet detection. In Alfred B. Schultz and Richard G. Lyon, editors, *Proceedings of SPIE: High-Contrast Imaging for Exo-Planet Detection*, volume 4860, pages 108–119. SPIE, February 2003.

[76] D. M. Lobosco. Integrated modeling of optical performance for the terrestrial planet finder structurally connected interferometer. Master's thesis, Massachusetts Institute of Technology, June 2004.

[77] B. A. Macintosh, D. Gavel, S. G. Gibbard, C. E. Max, M. Eckart, I. dePater, A. M. Ghez, and J. Spencer. Speckle imaging of volcanic hotspots on io with the keck telescope. *Icarus*, 165:137–143, 2003.

[78] Emmanuel Manil and M. Lieber. End-to-end system modeling of the vlt. In Arne L. Ardeberg, editor, *Optical Telescopes of Today and Tomorrow*, volume 2871, pages 181–192. SPIE, March 1997.

[79] S Manoharan and S Shanmuganathan. A comparison of search mechanicsms for structural optimization. *Computers and Structures*, 73:363–372, 1999.

[80] J. R. R. A. Martins, P. Sturdza, and J. J. Alonso. The complex-step derivative approximation. *ACM Transactions on Mathematical Software*, 29(3):245–262, September 2003.

[81] B. P. Masters and E. F. Crawley. Evolutionary design of controlled structures systems. In *AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference and Exhibit*, pages 1556 – 1569, Kissimmee, FL, Apr. 7-10 1997. AIAA. AIAA-97-1263.

[82] B. P. Masters and E. F. Crawley. An experimental investigation of optimized precision optical controlled-structures. In Mark E. Regelbrugge, editor, *Smart Structures and Materials 1997: Smart Structures and Integrated Systems*, volume 3041, pages 348–359. SPIE, June 1997.

[83] R. A. Masterson, D. W. Miller, and R. L. Grogan. Development and validation of reaction wheel disturbance models: Empirical model,. *Journal of Sound and Vibration*, 249(3):575–598, 2002.

[84] J. W. Melody and G. W. Neat. Integrated modeling methodology validation using micro-precision interferometer testbed. In *Proceedings of the 35th Conference on Decision and Control*, volume 4, pages 4222–4227, Kobe, Japan, December 1996. IEEE.

[85] J. W. Melody and G. W. Neat. Integrated modeling methodolog validation using the micro-precision interferometer testbed: Assessment of closed-loop performance capability. In *Proccedings of the American Control Conference*, volume 6, pages 3531–3536, Albuquerque, NM, June 1997. SPIE.

[86] B. Mennesson and J. M. Mariotti. Array configurations for a space infrared nulling interferometer dedicated to the search for earthlike extrasolar planets. *ICARUS*, 128(1):202–212, July 1997.

[87] D. F. Miller and J. Shim. Graident-based combined structural and control optimization. *J. Guidance*, 10(3):291–298, May-June 1987.

[88] D. W. Miller, O. L. de Weck, and G. E. Mosier. Framework for multidisciplinary integrated modeling and analysis of space telescopes. In Torben Andersen, editor, *SPIE Proceedings: Integrated Modeling of Telescopes*, volume 4757, pages 1–18. SPIE, July 2002.

[89] M. M. Milman, M. Salama, and M. Wette. Integrated control, structure and optical design. In *Proceedings of the American Control Conference*, number 2, pages 1448–1452. American Automatic Control Council, 1991.

[90] S. K. Morrison, Ye Yinyu, C. Z. Gregory Jr., G. R. Kosut, and M. E. Regelbrugge. Integrated structural/controller optimization of large space structures. In *AIAA Guidance, Navigation and Controls Conference*, pages 681–688, Minneapolis, MN, 1988. AIAA.

[91] M. Moshrefi-Torbati, A. J. Keane, S. J. Elliott, M. J. Brennan, and E. Rogers. Passive vibration control of a satellite boom structure by geometric optimization using genetic algorithm. *Journal of Sound and Vibration*, 267(4):879–892, October 2003.

[92] G. E. Mosier, M. Femiano, Kong Ha, et al. An integrated modeling environment for systems-level performance analysis of the next generation space telescope. In *SPIE Conference on Space Telescopes and Instruments V*, volume 3356, pages 89–97, Kona, Hawaii, March 1998. SPIE.

[93] J. E. Mottershead and M. I. Friswell. Mdoel updating in structural dynamics: A survey. *Journal of Sound and Vibration*, 167(2):347–375, 1993.

[94] MSC.Software. *MSC.NASTRAN 2004 Quick Reference Guide.* MSC.Software Corporation, 2003.

[95] P. B. Nair and A. J. Keane. Passive vibration suppression of flexible space structures via optimal geometric redesign. *AIAA Journal*, 39(7):1338–1346, 2001.

[96] P. B. Nair, A. J. Keane, and R. S. Langley. Design optimization of space structures with nonperiodic geometries for vibration supression. In

AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials, St Louis, MO, April 12-15 1999. AIAA. AIAA-99-1260.

[97] Gregory W. Neat, James W. Melody, and Boris J. Lurie. Vibration attenuation approach for spaceborne optical interferometers. In *IEEE Transactions on Control Systems Technology*, volume 6, November 1998.

[98] R. B. Nelson. Simplified calculations of eigenvector derivatives. *AIAA Journal*, 14:1201–1205, 1976.

[99] Gyung-Jin Park, Woo-Jeong Hwang, and Wan-Ik Lee. Structural optimization post-process using taguchi method. *JSME International Journal Series A*, 37(2):166–172, 1994.

[100] Gyung-Jin Park, Kwon-Hee Lee, and Wan-Ik Lee. Robust design for unconstrained optimization problems using the taguchi method. *AIAA Journal*, 34(5):1059–1063, 1996.

[101] A. Perry. A modified conjugate gradient algorithm. *Operations Research*, 26(6):1073–1078, November-December 1978. Technical Note.

[102] M. S. Phadke. *Quality Engineering Using Robust Design.* Prentice Hall PTR, Englewood Cliffs, NJ, 1989. AT&T Bell Laboratoris.

[103] E Sandgren and T. M. Cameron. Robust design optimization of structures through consideration of variation. *Computers and Structures*, 80:1605–1613, 2002.

[104] P. O. M. Scokaert and D. Q. Mayne. Min-max feedback model predictive control for constrained linear systems. *IEEE Transancations on Automatic Control*, 43(8):1136–1142, 1998.

[105] M. Shao and M. M. Colavita. Potential of long-baseline infrared interferometry for narrow-angle astrometry. *Astronomy and Astophysics*, 262:353–358, 1991.

[106] S. Simonian. Survey of spacecraft damping measurements: Applications to electro-optic jitter problems. In *The role of damping in vibration and noise control in 11th Biennial Conference on Mechanical Vibration and Noise*, volume 5, pages 287–292, Boston, MA, September 27-30 1987. ASME.

[107] G. Taguchi. Qulaity engineering (taguchi methods) for the development of electronic circuit technology. *IEEE Transactions on Reliability*, 44(2):225–229, June 1995.

[108] Inc. The MathWorks. *Optimization Toolbox User's Guide.* The MathWorks, Inc., Natick, MA, 2003.

[109] H. Uchida and J. Onoda. Simultaneous optimization of structural stiffness/passive damping/active controller for flexible space structures. In *AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials*, pages 1928–1937, Salt Lake City, Utah, 1996. American Institute of Aeronautics and Astronautics. AIAA 96-1544.

[110] A. Vinkler and L. J. Wood. A comparison of several techniques for designing controllers of uncertain dynamic systems. In *Proceedings of the 17th IEEE Conference on Decision and Control*, pages 31–38. IEEE, 1978.

[111] A. von Flowtow. Control-motivated dynamic tailoring of truss-work structures. In *AIAA Guidance, Navigation and Control Conference*, pages 622–628, Willimasburg, Virginia, 1986. AIAA.

[112] P. H. Wirsching, T. L. Paez, and H. Ortiz. *Random Vibrations: Theory and Practice*. John Wiley & Sons, Inc., 1995.

[113] H. Yamakawa. Optimum structual designs for dynamic response. *New Directions in Optimum Structural Design*, pages 249–266, 1984. ed. E. Atrek, et. al.

[114] R. J. Yang. Multidiscipline topology optimization. *Computers and Structures*, 63(6):1205–1212, 1997.