

**Multivariable Isoperformance Methodology  
for Precision Opto-Mechanical Systems**

by

**Olivier L. de Weck**

Dipl. Ing., Eidgenössische Technische Hochschule, Zurich, Switzerland (1993)  
S.M., Massachusetts Institute of Technology (1999)

Submitted to the Department of Aeronautics and Astronautics  
in partial fulfillment of the requirements for the degree of

**Doctor of Philosophy**

at the

**MASSACHUSETTS INSTITUTE OF TECHNOLOGY**

September 2001

© Massachusetts Institute of Technology 2001. All rights reserved.

Author \_\_\_\_\_  
Department of Aeronautics and Astronautics  
August 10, 2001

Certified by \_\_\_\_\_  
Professor David W. Miller  
Department of Aeronautics and Astronautics  
Thesis Supervisor

Certified by \_\_\_\_\_  
Professor Eric M. Feron  
Department of Aeronautics and Astronautics

Certified by \_\_\_\_\_  
Dr. Marthinus C. van Schoor  
Department of Aeronautics and Astronautics

Certified by \_\_\_\_\_  
Professor Edward F. Crawley  
Department of Aeronautics and Astronautics

Accepted by \_\_\_\_\_  
Professor Wallace E. Vander Velde  
Chairman, Department Graduate Committee



# Multivariable Isoperformance Methodology for Precision Opto-Mechanical Systems

by

Olivier L. de Weck

Submitted to the Department of Aeronautics and Astronautics  
on August 10, 2001, in partial fulfillment of the  
requirements for the degree of  
**Doctor of Philosophy**

## Abstract

Precision opto-mechanical systems, such as space telescopes, combine structures, optics and controls in order to meet stringent pointing and phasing requirements. In this context a novel approach to the design of complex, multi-disciplinary systems is presented in the form of a multivariable isoperformance methodology. The isoperformance approach first finds a point design within a given topology, which meets the performance requirements with sufficient margins. The performance outputs are then treated as equality constraints and the non-uniqueness of the design space is exploited by trading key disturbance, plant, optics and controls parameters with respect to each other.

Three algorithms (branch-and-bound, tangential front following and vector spline approximation) are developed for the bivariate and multivariable problem. The challenges of large order models are addressed by presenting a fast diagonal Lyapunov solver, apriori error bounds for model reduction as well as a governing sensitivity equation for similarity transformed state space realizations. Specific applications developed with this technique are error budgeting and multiobjective design optimization. The goal of the multiobjective design optimization is to achieve a design which is pareto optimal, such that multiple competing objectives can be satisfied within the performance invariant set. Thus, situations are avoided where very costly and hard-to-meet requirements are levied onto one subsystem, while other subsystems hold substantial margins.

An experimental validation is carried out on the DOLCE laboratory testbed. The testbed allows verification of the predictive capability of the isoperformance technique on models of increasing fidelity. A comparison with experimental results, trading excitation amplitude and payload mass, is demonstrated. The predicted performance contours match the experimental data very well at low excitation levels, typical of the disturbance environment on precision opto-mechanical systems. The relevance of isoperformance to space systems engineering is demonstrated with a comprehensive NEXUS spacecraft dynamics and controls analysis. It is suggested that isoperformance is a useful concept in other fields of engineering science such as crack growth calculations in structures. The isoperformance approach enhances the understanding of complex opto-mechanical systems beyond the local neighborhood of a particular point design.

**Thesis committee chairman:** David W. Miller, Sc.D.  
Title: Associate Professor of Aeronautics and Astronautics  
Director Space Systems Laboratory

**Thesis committee member:** Eric M. Feron, Ph.D.

Title: Associate Professor of Aeronautics and Astronautics  
Laboratory for Information and Decision Systems

**Thesis committee member:** Marthinus C. van Schoor, Ph.D.

Title: Senior Lecturer, Aeronautics and Astronautics  
President, Midé Technology Corporation

**Thesis committee member:** Edward F. Crawley, Ph.D.

Title: Professor, Department Head, Aeronautics and Astronautics

## Acknowledgments

I would like to thank my advisor Prof. David W. Miller for his guidance and support in the last four years. He has taught and inspired me in many ways and has become a good friend. Furthermore he has given me insights into academic life and instilled a passion for research, which I plan on pursuing in my own career in the future. My wife Lynn, my daughter Gabrielle and my son Christian can't be thanked enough for their love, patience and encouragement during the time in which this research was accomplished. I would like to particularly acknowledge my mother-in-law Mrs. Clare Hustedde for her unwavering support and sacrifices, which enabled me to focus on this research during the last year. My father, Prof. Dr. Alain de Weck, has supported me patiently in emotional, intellectual and financial ways. Finally, my friends and colleagues at the the M.I.T. Space Systems Laboratory contributed through research discussions and many practical suggestions. Specifically, I would like to mention Dr. Homero Gutierrez, Dr. Greg Mallory, Frederic Bourgault, Scott Uebelhart, Alice Liu, Becky Masterson, Laila Elias, Alissa Clawson, Alvar Saenz-Otero, Cyrus Jilla, Edmund Kong and Soon-Jo Chung. Paul Bauer deserves special credit for his support and experience in the laboratory. The undergraduate students who participated in the fabrication and assembly of the DOLCE testbed are Dusty DeQuine and Cemocan Yesil. Markus Gottier, dipl.Ing. ETHZ, was helpful in posing and reviewing the crack growth sample problem in Appendix C. Sharon-Leah Brown provided encouragement and professional financial management.

This research was supported by the NASA Goddard Space Flight Center under contracts No. NAG5-6079 and No. NAG5-7839 and by the Jet Propulsion Laboratory under the SIM research contract No. JPL 961123. The above research contracts were monitored by Mr. Gary Mosier (GSFC), Dr. Sanjay Joshi (JPL) and Mr. Robert Grogan (JPL), respectively. Financial support to the author was further provided in the form of a Pellegrini Medicus fund scholarship and a Carroll L. Wilson award.

This thesis is dedicated to the memory of my mother, Dr. Christine de Weck (1929-1999).



# Contents

<b>1</b>	<b>Introduction</b>	<b>23</b>
1.1	Research Background . . . . .	24
1.2	Isoperformance Motivation and Analogies . . . . .	28
1.3	Thesis Problem Formulation . . . . .	31
1.3.1	Thesis Objectives . . . . .	31
1.3.2	Mathematical Problem Definition . . . . .	32
1.3.3	Limitation of thesis scope . . . . .	34
1.4	Previous Work . . . . .	35
1.4.1	Literature Review . . . . .	35
1.4.2	Implementation Context (DOCS Framework) . . . . .	41
1.5	Thesis Overview . . . . .	45
<b>2</b>	<b>Sample Problems</b>	<b>49</b>
2.1	Single Degree-of-Freedom Oscillator . . . . .	50
2.1.1	Problem Description . . . . .	50
2.1.2	Disturbance Analysis . . . . .	51
2.1.3	Sensitivity Analysis . . . . .	54
2.1.4	Uncertainty Analysis . . . . .	55
2.1.5	Performance Enhancement . . . . .	56
2.1.6	Conclusion . . . . .	58
2.2	Two Degree-of-Freedom Oscillator . . . . .	58
2.2.1	Problem Description . . . . .	59
2.2.2	Disturbance, Sensitivity and Uncertainty Analyses . . . . .	60
2.2.3	Performance Enhancement . . . . .	63
2.3	Optical Delay Line Design Problem . . . . .	66
2.3.1	Problem Definition . . . . .	66
2.3.2	ODL Integrated Model . . . . .	67
2.3.3	ODL Initial Design . . . . .	71
2.3.4	Stability Margins . . . . .	73
2.3.5	Disturbance, Sensitivity and Uncertainty Analyses . . . . .	75
2.3.6	ODL conclusions . . . . .	77
2.4	Summary . . . . .	77
<b>3</b>	<b>Bivariate Isoperformance Methodology</b>	<b>79</b>
3.1	Bivariate Isoperformance Problem . . . . .	79
3.2	Algorithm Development . . . . .	81
3.2.1	Algorithm I: Exhaustive Search . . . . .	83

3.2.2	Algorithm II: Gradient-based Contour Following . . . . .	86
3.2.3	Algorithm III: Progressive Spline Approximation . . . . .	91
3.3	Algorithm Evaluation . . . . .	96
3.3.1	Quality of Isoperformance Solution . . . . .	96
3.3.2	Results for Exhaustive Search (Algorithm I) . . . . .	96
3.3.3	Results for Gradient-Based Contour Following (Algorithm II) . . . . .	99
3.3.4	Results for Progressive Spline Approximation (Algorithm III) . . . . .	101
3.3.5	Algorithm comparison . . . . .	101
3.4	Application Examples . . . . .	103
3.4.1	Single DOF oscillator . . . . .	103
3.4.2	ODL Design Problem . . . . .	104
3.4.3	SIM Classic (V1.0) . . . . .	106
3.5	Summary . . . . .	113
<b>4</b>	<b>Multivariable Isoperformance Methodology</b>	<b>115</b>
4.1	Branch and Bound Search Algorithm . . . . .	116
4.2	Tangential Front Following Algorithm . . . . .	120
4.3	Vector Spline Approximation . . . . .	126
4.4	Algorithm Comparison . . . . .	130
4.5	Complexity Theory . . . . .	132
4.5.1	Exhaustive Search . . . . .	132
4.5.2	Branch-and-Bound . . . . .	133
4.5.3	Tangential Front Following . . . . .	134
4.5.4	Vector Spline Approximation . . . . .	134
4.6	Graphical Representation of Results . . . . .	135
4.6.1	Contour Plots . . . . .	135
4.6.2	Radar Plots . . . . .	135
4.7	Examples . . . . .	138
4.7.1	Single DOF oscillator . . . . .	138
4.7.2	Two DOF oscillator . . . . .	143
4.7.3	ODL Design Problem . . . . .	147
4.8	Summary . . . . .	150
<b>5</b>	<b>Challenges of Large Order Systems</b>	<b>151</b>
5.1	Efficient Solution of Lyapunov Equation . . . . .	152
5.1.1	Computational Cost of Lyapunov Solution and Balancing . . . . .	152
5.1.2	Fast Lyapunov Solver for Diagonalizable Systems . . . . .	155
5.2	Analyses for Similarity Transformed Systems . . . . .	161
5.2.1	Motivation and Scope . . . . .	161
5.2.2	Internal Balancing Transformation . . . . .	163
5.2.3	Disturbance and Sensitivity Analysis for Similarity Transformed State Space Systems . . . . .	165
5.3	Disturbance and Sensitivity Analysis for Reduced Systems . . . . .	167
5.3.1	RMS and Error Bound for Reduced Systems . . . . .	169
5.3.2	Sensitivity and Error Bound for Reduced System . . . . .	175
5.4	Verification and Examples . . . . .	177
5.4.1	Single Degree of Freedom Oscillator . . . . .	177
5.4.2	Multiple Degree-of-Freedom Problem . . . . .	180



5.4.3	Application to SIM Classic (V1.0)	188
5.5	Summary	193
<b>6</b>	<b>Experimental Validation</b>	<b>195</b>
6.1	Testbed Description	195
6.1.1	Physical Test Setup	196
6.1.2	Finite-Element Model	197
6.1.3	Disturbance Model	197
6.1.4	Performance Model	198
6.2	Experimental Approach	200
6.3	Testbed Characterization	201
6.3.1	System Identification	201
6.3.2	Performance Testing	203
6.4	Isoperformance Results and Interpretation	204
6.5	Summary	207
<b>7</b>	<b>NEXUS Spacecraft Study</b>	<b>209</b>
7.1	NEXUS Description	209
7.2	Integrated Modeling	211
7.2.1	Finite Element Model	211
7.2.2	Optics Model	212
7.2.3	Disturbance Sources	213
7.2.4	Appended Dynamics and Controls	214
7.3	Disturbance Analysis	215
7.4	Sensitivity Analysis	218
7.5	Bivariate Isoperformance	220
7.5.1	Imbalance versus Isolation	220
7.6	Multivariable Isoperformance	222
7.7	Error Budgeting	224
7.8	Summary	225
<b>8</b>	<b>Conclusions and Recommendations</b>	<b>227</b>
8.1	Thesis Summary and Conclusions	227
8.2	Contributions	230
8.3	Limitations	230
8.4	Recommendations for Future Work	231
<b>A</b>	<b>Summary of Disturbance, Sensitivity and Uncertainty Framework</b>	<b>233</b>
A.1	Disturbance Modeling	233
A.2	Integrated Modeling	235
A.3	Performance Assessment	236
A.3.1	Time Domain Analysis	237
A.3.2	Frequency Domain Analysis	238
A.3.3	Lyapunov Analysis	239
A.4	Sensitivity Analysis	241
A.5	Uncertainty Analysis	242

<b>B</b>	<b>Derivations for Large Order Systems</b>	<b>245</b>
B.1	Proof of Similarity Transformation . . . . .	245
B.2	Computation of Balancing Transformation . . . . .	246
B.3	Derivation of Transformed Governing Sensitivity Equation (TGSE) . . . . .	248
B.4	Computation of Sensitivity Error Bound due to Model Reduction . . . . .	254
B.5	Summary . . . . .	259
<b>C</b>	<b>Application of Isoperformance to other Fields of Engineering</b>	<b>261</b>
C.1	Crack Growth in Metallic Structures . . . . .	261
<b>D</b>	<b>DOLCE Finite Element Model</b>	<b>267</b>
	<b>References</b>	<b>277</b>

# List of Figures

1.1	Hubble Space Telescope and proposed successor missions as part of NASA’s space science program. Sample requirements flowdown for SIM. . . . .	25
1.2	Reference problem: Science target observation mode of a space telescope with pointing (RSS LOS) and phasing (RMMS WFE) performances. . . . .	27
1.3	(a) Simple model of axially loaded column in bridge analogy. (b) United States sea level pressure distribution [mbar], 1600 Z time, Tuesday, May 9th, 2000, based on <a href="http://ww2010.atmos.uiuc.edu">http://ww2010.atmos.uiuc.edu</a> [110]. . . . .	29
1.4	Sets related to thesis problem definition. The eigenvalues $\lambda_i$ are obtained by solving the eigenvalue problem $[A_{zd} - \lambda_i I]\phi_i = 0$ . . . . .	34
1.5	DOCS-Framework block diagram. The shaded area in the lower right is in the scope of this thesis. The modules marked with an asterisk are available commercially or through academic licensing. . . . .	42
1.6	Thesis Roadmap: Development (chapter numbers in <b>bold</b> font) . . . . .	46
1.7	Thesis Roadmap: Validation (chapter numbers in <b>bold</b> font) . . . . .	48
2.1	Schematic of single degree-of-freedom (SDOF) oscillator (left) with table of design parameters (right). . . . .	50
2.2	SDOF block diagram. From left: white noise disturbance source, disturbance isolation, oscillator and the “optical control” HPF. . . . .	51
2.3	SDOF time simulation: (Top) disturbance $F(t)$ , (Bottom) output $z(t)$ . Dark curve for $p_o$ , light trace for improved design $p_{nom}$ . . . . .	53
2.4	SDOF Problem: Sensitivities of performance $J_z$ w.r.t. parameters $p_o$ for the initial case (left) and after performance enhancement $p_{nom}$ (right). . . . .	55
2.5	(Bottom) PSD function $S_{zz}(\omega)$ for $p_o$ (dark) and $p_{nom}$ (light). (Top) Cumulative RMS curve. Requirements line is dashed. . . . .	57
2.6	Schematic of two degree-of-freedom (2DOF) oscillator (left) with table of variable parameters (right). . . . .	59
2.7	2DOF time simulation for $J_{z,1}$ (top) and $J_{z,2}$ (bottom). Dark trace for $p_o$ , lighter trace for improved design $p_{nom}$ . . . . .	61
2.8	2DOF sensitivity analysis $\bar{\nabla} J_z$ : (Left) normalized sensitivities w.r.t. $J_{z,1}$ , (Right) w.r.t $J_{z,2}$ . Dark bars show $p_o$ . Light bars show $p_{nom}$ . . . . .	62
2.9	2DOF sample problem PSD analysis. Dashed horizontal line is $J_{z,req}$ . . . . .	65
2.10	Left: Optical Delay Line (ODL) design concept. Right: Table containing performance $J_z$ and other requirements for ODL design. . . . .	66
2.11	ODL block diagram. Signal units in square brackets. . . . .	67
2.12	ODL plant transfer functions: $G_{p1} = X(s)/X_b(s)$ (left) and $G_{p2} = X(s)/F_c(s)$ (right). Suspension mode at 0.5 Hz and $\zeta = 0.073$ as experimentally determined by Park [116]. . . . .	72

2.13	ODL PID controller transfer function: $G_c = U_c(s)/E(s)$ . . . . .	73
2.14	Pole-Zero map for closed loop ODL system. The zoomed region shows the pole-zero cancellation of the plant complex-conjugate pole pair. . . . .	74
2.15	(a) Loop Gain function for ODL, (b) Nyquist diagram for ODL. . . . .	74
2.16	ODL simulation results: Cumulative RMS function for $J_{z,1}$ (top), Power spectral density function (PSD) $S_{zz}(\omega)$ (middle), Sample time realization for $z(t)$ with $T=18$ [sec] (bottom). . . . .	75
2.17	(a) Step response of ODL for $r=1.0$ [mm]; (b) Sensitivities of RMS OPD $\sigma_z$ w.r.t. parameters $p_j$ (normalized). . . . .	76
3.1	Theoretical instances of isoperformance contours $p_j \mapsto J_z(p_j) = J_{z,req}$ in $\mathbb{R}^2$ . (a) regular boundary point on $\mathbf{B}$ , (b) interior solution point $p_{iso}$ in set $\mathbf{I}$ and (c) boundary point on the unstable subset $\mathbf{U}$ . . . . .	82
3.2	(a) Algorithm I: Discretization of $\mathbf{B}$ in a linearly spaced grid with increments $\Delta p = [\Delta p_1, \Delta p_2]^T$ . (b) Linear interpolation between neighboring grid points. The gray shaded area is a zoomed region. . . . .	84
3.3	Flow Diagram for Algorithm I: Exhaustive Search . . . . .	85
3.4	Algorithm II: Depiction of gradient vector $\nabla J_z$ , normal vector $n$ and tangential vector $t$ along the isoperformance contour. . . . .	89
3.5	Flow Diagram for Algorithm II: Gradient-based Contour Following. Note: The correction step from equation (3.31) is omitted for simplicity. . . . .	92
3.6	Progressive (cubic) spline approximation. . . . .	95
3.7	Flow Diagram for algorithm III: Progressive (cubic) spline approximation . . . . .	95
3.8	(a) Algorithm I (Exhaustive Search): Isoperformance contour for single DOF problem $(\omega_d, m)$ with discretization $\Delta p = (1/20)[p_{UB} - p_{LB}]$ and a tolerance of $\tau = 1\%$ . (b) Quality: Contour quality according to (3.40). . . . .	97
3.9	(a) Algorithm I (Exhaustive Search): Surface plot of $J_z(p_1, p_2)$ for SDOF oscillator problem. (b) Family of isoperformance contours in 0.2 mm RMS increments. . . . .	98
3.10	(a) Algorithm II (Gradient-Based Contour Following): Isoperformance contour for single DOF problem $(\omega_d, m)$ with tolerance $\tau = 1\%$ . (b) Quality: Contour solution quality according to (3.40). . . . .	99
3.11	Intercept of isoperformance contour: (a) Iterates from initial point $(p_{nom})$ to isoperformance contour $(p_{iso,1})$ . (b) Convergence history. . . . .	100
3.12	(a) Algorithm III Progressive Spline Approximation: Isoperformance contour for single DOF problem $(\omega_d, m)$ with tolerance $\tau = 1\%$ . (b) Quality: Contour solution quality according to (3.40). . . . .	101
3.13	(a) SDOF isoperformance for $\omega_d$ versus $\omega_o$ with $J_{z,req} = 0.0008$ [m] and $\tau = 1\%$ . (b) SDOF isoperformance for $k$ versus $c$ . . . . .	104
3.14	(a) ODL isoperformance for $S_{dd}$ versus $S_{nn}$ at $J_{z,req} = 100$ [nm] and $\tau = 1\%$ . (b) ODL isoperformance for $S_{dd}$ versus $K_d$ . . . . .	105
3.15	Left(a): Performance $J_z$ (RMS OPD) as a function of $K_d$ alone. Right(b): Root Locus for ODL sample problem (parameter: $K_d$ ). . . . .	106
3.16	Left: (a) Artistic rendering of SIM Classic (V1.0). Right: (b) Corresponding FEM with ray trace for interferometer #1 (version 1.0). . . . .	107
3.17	SIM Classic (version 1.0) Simulink block diagram for bivariate isoperformance problem . . . . .	107
3.18	SIM Classic frequency response functions: (a) wheel disturbance, (b) mechanical isolator and (c) open-loop plant dynamics. . . . .	108

3.19	Left:(a) PSD-based disturbance analysis for SIM Classic (1.0). (bottom) PSD of OPD (total) int. # 1. (top) Cumulative RMS curve. Right: (b) Bar chart of relative % contribution of the three most critical modes to the RMS. . . . .	109
3.20	SIM Classic (version 1.0): Isoperformance analysis for OPD (total) interferometer # 1 fixed at 20 nm. Trading isolator corner frequency $f_i$ vs. reaction wheel static imbalance $U_s$ . . . . .	110
3.21	SIM Classic (version 1.0): Isoperformance quality plot for $\tau = 1.0$ % and the isoperformance solution shown in Figure 3.20. . . . .	111
3.22	SIM Classic (v1.0): Time simulation (240 seconds) for $f_i = 1.5571$ [Hz] and $U_s = 5.0$ [gcm]. RMS value of $z$ is 20.0157 [nm], which is $\approx \sigma_{z,req}$ . . . . .	112
4.1	Multivariable Isoperformance (Ib): Branch-and-Bound graphic representation. Crossed out points are dropped in the next generation. . . . .	117
4.2	Multivariable Isoperformance (Ib): Flow Diagram for Branch-and-Bound Algorithm. . . . .	118
4.3	Multivariable Isoperformance (Ib): Branch-and-Bound results for SDOF problem with three variable parameters: $\omega_d$ , $m$ and $\omega_o$ and required performance $J_{z,req} = 0.8$ [mm]. . . . .	119
4.4	Multivariable Isoperformance (Ib): Quality of solution for SDOF example. Each one of the $\sim 7000$ points corresponds to a point on the performance-invariant surface in Figure 4.3. . . . .	120
4.5	Multivariable Isoperformance (II): Tangential Front Following principle. Note the difference between principal and intermediate front points. Axes $p_1, p_2$ and $p_3$ are values of some generic parameters. . . . .	123
4.6	Tangential Front Following (II) Flow Diagram. . . . .	124
4.7	Multivariable Isoperformance (II): Tangential Front Following results for SDOF problem with three variable parameters: $\omega_d$ , $m$ and $\omega_o$ . . . . .	125
4.8	Multivariable Isoperformance (II): Quality of solution . . . . .	125
4.9	Multivariable Isoperformance (III): Vector Spline Approximation Principle. Boundary splines in dark (blue) and interior splines in light (green). Centroid is $p_{cent}$ . . . . .	128
4.10	Multivariable Isoperformance (III): Flow Diagram . . . . .	128
4.11	Multivariable Isoperformance (III): Vector Spline Approximation for SDOF Problem and performance level $J_{z,req} = 0.8$ [mm]. . . . .	129
4.12	Multivariable Isoperformance (III): Quality of solution . . . . .	130
4.13	Cross-orthogonality matrix (COM) for solution to SDOF problem with Algorithm III (Tangential Front Following). . . . .	136
4.14	Radar plot for SDOF problem solutions . . . . .	137
4.15	(a) Representation of the 32 corners ( $2^5$ ) of the trade space $\mathbf{B}$ for the SDOF problem. (b) Border point search ( $p_{iso,border}$ ) for boundary points of SDOF isoperformance set. . . . .	139
4.16	(a) Vector Spline Approximation of SDOF isoperformance set, $\mathbf{I}$ , projected to the first three parameters, $[\omega_d, m, k]$ . (b) Quality of solution plot for SDOF isoperformance set. . . . .	140
4.17	(a) Radar plot for SDOF problem pareto optimal designs. (b) PSD and cumulative RMS plot for SDOF pareto optimal designs. . . . .	141
4.18	(a) Bivariate Isoperformance analysis for 2DOF problem with $\omega_d$ and $k_2$ . (b) Multivariable Isoperformance analysis for 2DOF problem with three parameters: $\omega_d$ , $k_1$ and $k_2$ . . . . .	144

4.19	(a) 2DOF Isoperformance analysis with all parameters ( $n_p = 8$ ) constraining a single performance , $J_{z,1}$ . (b) 2DOF Isoperformance analysis with all parameters ( $n_p = 8$ ) constraining both performances ( $n_z = 2$ ). (c) Quality plot for 2DOF and $n_z = 1$ corresponding to (a). (d) Quality plot for 2DOF and $n_z = 2$ corresponding to (b). . . . .	145
4.20	Two DOF Problem: Radar plot for pareto optimal designs. . . . .	146
4.21	Two DOF Problem: PSD and cumulative RMS comparison of pareto optimal designs. . . . .	147
4.22	(a) ODL Problem: multivariable isoperformance with four parameters: $S_{dd}$ , $S_{nn}$ , $m$ and $K_d$ . Performance Requirement: $J_{z,1,req} = 100$ [nm]. Isoperformance set, $\mathbf{I}$ projected onto the first three parameters. (b) Quality of solution plot corresponding to (a) with tolerance $\tau = 2.0$ %. . . . .	148
4.23	ODL Problem: Radar plot for pareto optimal designs. . . . .	149
4.24	(a) ODL Problem: Comparison of settling times for a $r = 1.0$ [mm] step input command for both pareto optimal designs. (b) Nyquist diagram showing phase margin (PM) and gain margin (GM) for both designs. . . . .	150
5.1	Computational cost of balancing ( $T$ ) and solving a Lyapunov equation for $\Sigma_q$ as a function of the number of states $n_s$ . . . . .	153
5.2	(a) CPU Time in minutes to solve a Lyapunov equation of size $n_s$ using blocks of size $m$ . (b) Zoom in on optimal region between $m = 20 - 40$ . . . . .	158
5.3	Number of FLOPS for the diagonal Lyapunov solver, along with curves showing the flop count approximation. . . . .	160
5.4	Sparsity structure of (a) the original SIM-Classic A-matrix (308 states) according to equation (5.20); note cross-coupling terms. (b) Balanced A-matrix (308 states, fully populated) and (c) the reduced A-matrix (120 states, fully populated). . . . .	169
5.5	(a) Hankel singular values and model truncation threshold level $n_k$ for RMS error $< 0.5$ %. The original MDOF model has 42 states, of which 14 must be kept in the reduced model. (b) Validity of RMS error bound for various truncation levels according to Inequality (5.65) . . . . .	174
5.6	Comparison of normalized sensitivities for SDOF system parameters (from top to bottom). Modal parameters: $\phi_o$ mode shape, $m_{mod} = \hat{m}$ modal mass, $\zeta$ modal damping ratio, $\omega$ modal frequency. Physical Parameters: $k$ spring stiffness, $c$ damping coefficient and $m$ (physical) mass. . . . .	179
5.7	Mass-spring model of a multi-degree-of-freedom (MDOF) structure representing an interferometer with free-free boundary conditions. . . . .	180
5.8	Disturbance to performance transfer functions for MDOF example. Left column: $z_1$ . Right column: $z_2$ . The reduced transfer function (14 states - solid line) is shown along with the original unreduced system (42 states - dashed line).183	
5.9	(MDOF) Normalized modal sensitivity comparison for performance 1 . . . . .	184
5.10	(a) Normalized Sensitivity comparison for performance 2. (b) Numerical validity of relative sensitivity error bound (5.72) in the MDOF case. The maximum sensitivity error (Max SENS Error) is taken from the set of modal parameter sensitivities for performances $i = 1, 2$ and modes $j = 2, 3, 4$ . . . . .	185
5.11	Normalized Sensitivities for physical parameters (MDOF problem), left shows performance 1 (hub displacement RMS), right shows performance 2 (OPD RMS) sensitivities. . . . .	186

5.12	(a) SIM Classic; (b) FEM with ray trace for interferometer #1; (c) Disturbance PSD's for RWA disturbances; (d) Cumulative RMS plot (top) and PSD (bottom) for the total OPD of interferometer #1, where the dashed line is the reduced model (110 states) and the solid line is the full order model (308 states).	188
5.13	(a) SIM Classic modal parameter sensitivities for 10 modes and full-order system (308 states, 78.2 seconds CPU time), (b) Modal sensitivity results for balanced and reduced model (110 states, 3.3 seconds CPU time).	190
5.14	(a) Very large errors between finite difference approximation (1%) and the analytical modal parameter sensitivities for the full order model; (b) Good approximation with finite differences (1%) in the reduced model case. Mode numbers correspond to modes in Figure 5.13	191
5.15	(a) Erroneous modal parameter sensitivity results for SIM Classic obtained when balancing matrix $T$ is computed without first removing unobservable and uncontrollable states, e.g. via pre-balancing. (b) Hankel singular values for original closed-loop SIM Classic model.	192
6.1	(a) DOLCE testbed setup in laboratory. (b) Finite element model (FEM) of DOLCE using IMOS version 5.0.	196
6.2	(a) Shaker disturbance force, $F_d(t)$ , sample time history for shaker voltage, $V_s = 0.5$ [Vrms]. (b) Averaged disturbance PSD $S_{dd}$ - 25 averages.	198
6.3	(a) Sample time history of displacement, $z(t)$ , for $V_s = 0.5$ [Vrms] and $m_p = 160$ [lbs]. (b) Averaged performance PSD $S_{zz}$ - 25 averages.	199
6.4	DOLCE Testbed Sensors	200
6.5	(a) DOLCE Experimental Approach. (b) Variable system parameters $m_p$ and $V_s$ .	200
6.6	DOLCE transfer function $G_{zd} = Z(s)/F_d(s)$ for $m_p = 0$ , $V_s = 1.0$	202
6.7	DOLCE coherence function $\gamma^2(f)$ corresponding to the transfer function $G_{zd} = Z(s)/F_d(s)$ .	202
6.8	DOLCE Testbed Observable Modes	203
6.9	Waterfall Plot for 1.0 [Vrms] forcing level	204
6.10	DOLCE Experimental Test Matrix ( $V_s, m_p$ ).	204
6.11	DOLCE Testbed: Comparison of experimental versus theoretical isoperformance contours at the levels: $J_z = 7.5, 15$ and $30$ [ $\mu\text{m}$ ] with $p_1 = m_p$ and $p_2 = V_s$ .	205
6.12	(a) Comparison of PSD $S_{zz}$ (bottom) and cumulative RMS plot (top) between experiment and FEM prediction for configuration: $m_p = 0$ [lbs], $V_s = 0.1$ [lbs]. Good Agreement. (b) Comparison of PSD $S_{zz}$ (bottom) and cumulative RMS plot (top) between experiment and FEM prediction for configuration: $m_p = 100$ [lbs], $V_s = 1.0$ [lbs]. Poor Agreement.	206
7.1	NEXUS Spacecraft Concept	210
7.2	NEXUS Finite Element Model. Important I/O grid points (nodes) and variable design parameters are shown.	211

7.3	NEXUS Optical Train modeled with ZEMAX. (a) Side View. (b) Isometric View. Selected mirror surfaces are labeled according to their element number ( <code>iEl<math>\mathbf{t}</math></code> ) in the NEXUS OTA prescription. Key optics data: PM $f/\#=1.25$ , Magnification $M=12$ , back focal length $BFL=0.2$ [m], SM diameter 0.27 [m], $f/15$ beam at Cassegrain focus, $f/24$ telescope at focal plane, same as NGST. Plate scale = 2.06 [masec/ $\mu\text{m}$ ]. . . . .	212
7.4	NEXUS broadband reaction wheel disturbance model. Nominal parameters: $R_u = 3000$ [RPM], $U_s = 0.7160$ [gcm] and $U_d = 29.536$ [gcm <sup>2</sup> ]. . . . .	214
7.5	NEXUS block diagram with 4 disturbance sources (RWA, Cryo, ACS noise, GS noise) and 2 performances (RMMS WFE, RSS LOS). Simulation implemented in Simulink as well as state space. . . . .	215
7.6	LOS Jitter initial disturbance analysis . . . . .	217
7.7	RSS LOS Centroid Jitter Plot on Focal Plane (NEXUS). . . . .	218
7.8	NEXUS normalized sensitivity analysis results at $p^o$ . . . . .	219
7.9	NEXUS Bus design with a 4-wheel symmetric pyramid of ITHACO E-wheels (39.3 [cm] diameter, 16.6 [cm] height, 10.6 [kg] mass each). See Reference [130] for 6-axis Active Vibration Isolator design. . . . .	220
7.10	NEXUS Bivariate Isoperformance analysis with $p_1 = U_d$ , $p_2 = K_r ISO$ and $J_z = \text{RSS LOS}$ . . . . .	221
7.11	NEXUS Multivariable Isoperformance. Radar plot of 3 pareto optimal designs. $J_{c,1}$ is best mid-range design, $J_{c,2}$ is the design with smallest FSM gain, $J_{r,1}$ is the design with smallest performance uncertainty. . . . .	223
7.12	NEXUS Pareto Optimal Designs: RSS LOS power spectral densities (bottom) and cumulative RMS curves (top). . . . .	223
7.13	NEXUS Error Sphere for RSS LOS. Note: ACS sensor noise contributions not shown. . . . .	225
A.1	General block diagram of system dynamics from white noise disturbances $d$ to system performance signals $z$ . . . . .	236
C.1	Crack growth sample problem: Center cracked infinitely long panel. . . . .	261
C.2	(a) Nominal crack growth curve from $a_o$ to $a_c$ with integration step $\Delta a = 0.01''$ . (b) Dependence of predicted critical number of load cycles, $N_c$ , on crack length integration step size, $\Delta a$ . . . . .	262
C.3	(a) Crack growth problem: Gradient search for $J_z = 25,000$ isoperformance contour. (b) Iterations to intercept. . . . .	263
C.4	(a) Crack growth problem: Isoperformance contour for $N_c = 25,000$ and variable parameters $\Delta\sigma$ and $a_o$ . (b) Quality of solution plot for tolerance $\tau = 0.5\%$ . . . . .	264



# List of Tables

2.1	Comparison of $p_o$ and $p_{nom}$ with the ratio of parameter values in the second to last column. A non-zero value in the last column indicates that this parameter is at the upper (+1) or lower bound (-1). . . . .	64
2.2	Results overview for two DOF oscillator problem. . . . .	65
2.3	ODL design parameters. Explanations of the column headings are given in the paragraph below. . . . .	71
3.1	Comparison of algorithms I-III for isoperformance analysis of single degree-of-freedom problem. . . . .	102
4.1	Comparison of multivariable algorithms for SDOF problem: (Ia) Exhaustive Search, (Ib) Branch-and-Bound, (II) Tangential Front Following and (III) Vector Spline Approximation. . . . .	131
4.2	Comparison of pareto optimal designs for SDOF sample problem. . . . .	142
4.3	Comparison of original design with pareto optimal designs for ODL sample problem. . . . .	149
5.1	Time improvements using diagonal Lyapunov solver . . . . .	159
5.2	Improvements in FLOPS using fast Lyapunov solver . . . . .	160
5.3	Comparison of RMS sensitivities for single degree-of-freedom (SDOF) physical parameters (top) and modal parameters (bottom), including a 1% finite difference (fd) approximation. Normalized Sensitivities are $s_{norm}$ and $\tilde{s}_{norm}$ for the balanced system. . . . .	178
5.4	Comparison of RMS results [in $\mu m$ ] for MDOF example. . . . .	182
5.5	RMS Results comparison for SIM Classic . . . . .	189
6.1	Variable parameters of DOLCE testbed. . . . .	195
6.2	DOLCE Testbed sensor specifications. . . . .	199
7.1	NEXUS Variable Design Parameters $p_j, j = 1, \dots, 25$ . . . . .	216
7.2	Initial Performance Analysis Results . . . . .	218
7.3	NEXUS pareto optimal designs . . . . .	224
7.4	NEXUS error budget $\Psi_2$ . . . . .	224



# Nomenclature

## Abbreviations

ACS	attitude control system
AIAA	American Institute of Aeronautics and Astronautics
BW	bandwidth
CDR	Critical Design Review
CPU	central processing unit
CST	controlled structures technologies
DM	deformable mirror
DNC	dynamics navigation and controls
DOCS	Dynamics-Optics-Controls-Structures
DOF	degree of freedom
DPL	differential pathlength
EOM	equation(s) of motion
EVP	eigenvalue problem
FEM	finite-element model; finite-element method
FGS	fine guidance sensor
FSM	fast steering mirror
FWHM	full-width at half maximum
GSE	governing sensitivity equation
GSFC	Goddard Space Flight Center
HPF	highpass filter
HST	Hubble Space Telescope
IMOS	Integrated Modeling of Optical Systems
IR	Infrared
ISE	Intelligent Synthesis Environment
JPL	Jet Propulsion Laboratory
LB	lower bound
LHP	left half plane
LPF	lowpass filter
LQG	linear quadratic gaussian (regulator)
LQR	linear quadratic regulator
LTI	linear time-invariant
MACOS	Modeling and Analysis for Controlled Optical Systems
MATLAB	Matrix Laboratory
MDO	Multidisciplinary Design Optimization
MDOF	multiple degree-of-freedom
MIMO	multiple input - multiple output
MSC	MacNeal Schwendler Corporation

NGST	Next Generation Space Telescope
NICMOS	Near Infrared CMOS
NIR	Near Infrared
ODL	optical delay line
OPD	optical path difference
PID	proportional integral derivative (controller)
PDF	probability density function
PSD	power spectral density
RBE	rigid body element
RHP	right half plane
RMMS	root mean mean square
RMS	root mean square
RPM	revolutions per minute
RSS	root sum square
RWA	reaction wheel assembly
SIM	Space Interferometry Mission
SISO	single input single output
SVD	singular value decomposition
TGSE	transformed governing sensitivity equation
TPF	Terrestrial Planet Finder
URL	Uniform Resource Locator
UB	upper bound
UV	Ultraviolet
WFE	wavefront error
WFPC2	wide field planetary camera number 2
WFT	wavefront tilt
dB	decibel
mbar	milli-bar
mV	milli-volts
pp	piecewise polynomial

### Symbols

$A$	cross-sectional area, sub-block of $A_{zd}$ -matrix
$A_c$	compensator dynamics matrix
$A_d$	disturbance filter dynamics matrix
$A_{zd}$	augmented disturbance/plant/control dynamics matrix
$B_c$	compensator input matrix
$B_d$	disturbance filter input matrix
$B_u$	plant control input matrix
$B_w$	plant disturbance input matrix
$B_{zd}$	augmented disturbance/plant/control input matrix
$C$	physical damping matrix
$C_c$	compensator output matrix
$C_d$	disturbance filter output matrix
$C_{zd}$	augmented disturbance/plant/control performance matrix
$C_{zx}$	plant performance output matrix
$D_c$	compensator feedthrough matrix
$D_d$	disturbance filter feedthrough matrix

$D_{zd}$	augmented disturbance/plant/control feedthrough matrix
$E$	Young's modulus
$G$	shear modulus; transfer function matrix
$I$	cross-sectional bending moment of inertia
$J$	cross-sectional torsional constant
$J_c$	cost (vector) function
$J_r$	risk (vector) function
$J_z$	performance (vector) function
$J_{z,req}$	performance requirements (vector)
$K$	stiffness matrix
$L$	Lagrange multiplier matrix
$M$	mass matrix
$T$	coordinate/state space transformation matrix
$W_c, W_o$	controllability and observability gramians
$d$	unity-intensity white noise
$l$	element length
$m$	block size in diagonal Lyapunov solver
$n_d$	number of disturbance sources
$n_p$	number of (variable) design parameters
$n_s$	number of states in appended state space system
$n_z$	number of performance metrics
$p$	generic design parameter
$p_{iso}$	solution to the isoperformance problem
$q$	augmented disturbance/plant/control state vector
$r$	reference command
$w$	disturbance variable
$x$	FEM physical DOF's
$y$	sensor measurement
$z$	performance variable/channel
$\Omega$	modal frequency matrix
$\Phi$	mode shape matrix
$\Psi$	dynamics error budget
$\Sigma$	gramian matrix (squared Hankel singular values)
$\Sigma_q$	state covariance matrix
$\alpha$	step size
$\beta_u$	plant control influence matrix
$\beta_w$	plant disturbance influence matrix
$\eta$	weight between cost and risk
$\phi$	mode shape vector; phase angle
$\rho$	mass density
$\sigma$	standard deviation (= RMS if zero-mean); Hankel singular value
$\sigma^2$	variance (= mean-square if zero-mean)
$\tau$	isoperformance tolerance in percent
$\zeta$	modal damping ratio

## Subscripts and Superscripts

$(\cdot)^*$	Lagrangian
$(\cdot)^H$	Hermitian (complex-conjugate transpose)
$(\cdot)^T$	transpose
$(\cdot)_{i,j}, (\cdot)_{ij}$	$(i, j)$ entry of a matrix
$(\cdot)_i$	$i$ th entry of a vector

## Units:

Units in this thesis are typically given in the SI-system (système international) and are enclosed in square brackets, e.g. [N/m].

## Definitions

**Isoperformance:** A methodology which treats the performance of a system as an equality constraint and finds the locus or set of parameters that meet this condition, thus allowing tradeoff based on objectives other than performance.

**Precision Opto-Mechanical System:** A system which captures electromagnetic radiation in the form of light via an optical train and a photographic or electronic detector. The optical elements (mirrors, beam splitters, lenses) are mounted to a supporting structure, whose dynamics are excited by disturbance and input commands or non-zero initial conditions. The system is a “precision” system, when the phasing (and pointing) requirements are on the order of a single wavelength of light.

**Pareto Optimal Set:** A pareto optimal set is the solution of a multiobjective optimization problem. The characteristic of these solutions is that an improvement in one of the vector components of the objective can only be achieved by degrading the performance in at least one of the other objective components. The set contains only non-dominated solutions.

**Integrated Model:** State space representation of the dynamics of a system, which captures structural dynamics, optics, controls and disturbance sources in a unified manner in a single computing environment.

# Chapter 1

## Introduction

In designing complex high-performance technical systems, there are typically two conflicting quantities that come into play: resources and system performance. One traditional paradigm fixes the amount of available resources (costs) and attempts to optimize the system performance given this constraint. The other approach is to constrain the system performance to a desired level and to find a design (or a family of designs) that will achieve this performance at minimal cost. This thesis explores the second approach by developing a framework termed the “isoperformance methodology” for dynamic, linear time-invariant (LTI) systems. The word “isoperformance” contains the Latin prefix “iso”, meaning “same”. Thus it refers to a framework where the solutions to a design problem do not distinguish themselves by the performance they achieve but rather by the “cost” and “risk” required to achieve this performance. Note that “cost” is to be understood in a broader sense than monetary cost. The generalized cost metrics can include, among others, hardware costs, system mass, control effort or the amount of energy dissipated in the system. The generalized notion of “risk” is comprised, among others, of margins of stability, stability robustness, performance robustness to uncertainty and system reliability<sup>1</sup>.

The novelty in this approach is that integrated modeling and physical parameter sensitivity analysis for LTI systems are leveraged such that a family of “efficient” solutions is presented to the designer rather than a single point-design as is often the case in pure systems optimization or performance enhancement studies. The term “efficient” relates to a pareto optimal set of solutions. This framework is first developed generically for LTI systems, which can be described

---

<sup>1</sup>There is also technical risk associated with advanced technologies, which have never been implemented before. This type of risk, however, is much more difficult to quantify.

in state space form. It is then applied to dynamics and controls problems of precision optomechanical systems, such as the next generation of space-based observatories. These systems carefully combine structures, optics and controls such that stringent pointing and phasing requirements can be met in the presence of dynamic disturbance sources.

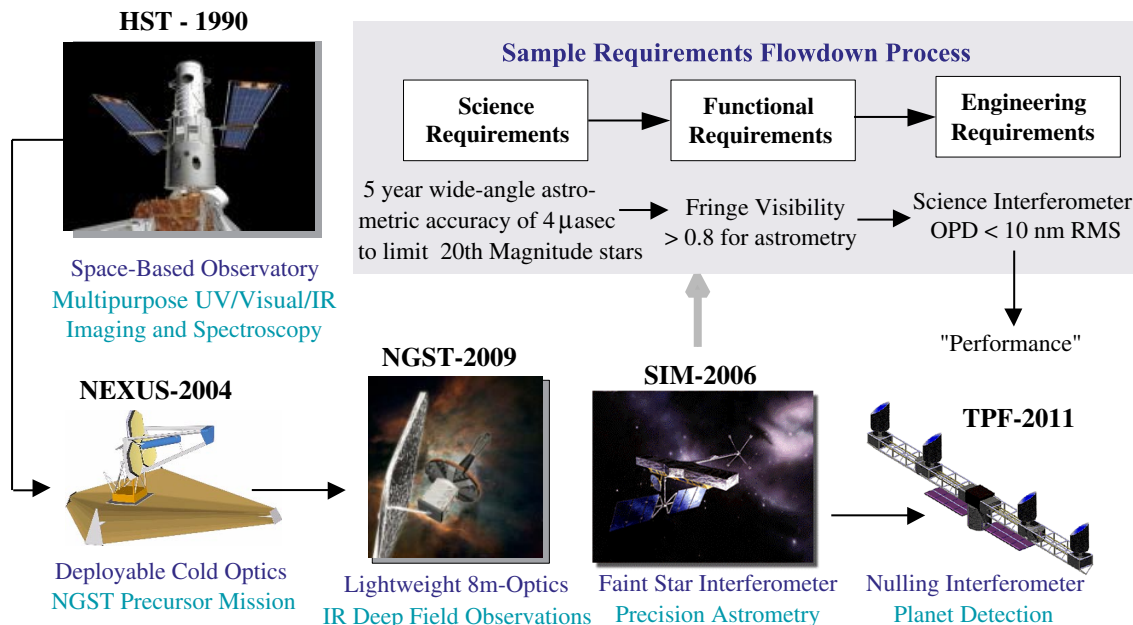
The first step of an isoperformance analysis is to find a nominal system design (starting from an initial guess) that meets all the dynamic performance requirements with sufficient margins. The performance is then treated as an equality constraint (i.e. it is held constant) and those system parameters, which are considered to be variable, independent and important, are traded with respect to each other. The reason the performance is treated as an equality rather than a “smaller-or-equal” inequality is that the latter could lead to solutions which are significantly overdesigned, i.e. they may provide excessive performance margins which will not be used during operations and will waste system resources such as mass and control effort or they may carry unnecessary risk such as small stability margins. A prerequisite for the methodology is that reasonable upper and lower bounds are defined for each variable parameter. Reasonable means that the bounds do not violate physics (e.g. negative mass) and are achievable within the current or foreseeable state-of-the-art. The computation of multiple isoperformance contours (at different performance levels) allows the designer to understand the implications of pursuing increased performance beyond the requirements. The goal is to obtain an isoperformance set, i.e. the locus of design points that yield equal nominal performance in the design space defined by the upper and lower bounds. This information can subsequently be utilized in support of important systems engineering tasks such as multiobjective design optimization and error budgeting.

## 1.1 Research Background

The next generation of space and ground based astronomical observatories such as the Next Generation Space Telescope (NGST), the Space Interferometry Mission (SIM) or the Terrestrial Planet Finder (TPF) will significantly surpass the present generation, for example the Hubble Space Telescope (HST), in terms of their sensitivity, angular resolution, spectral resolution and imaging stability [105, 59, 27]. The present work is motivated by the need to predict the dynamic behavior of these telescopes during the conceptual and preliminary design phases before substantial resources are committed towards a particular system archi-



ecture. Figure 1.1 shows the HST in the upper left corner and a number of proposed successor spacecraft below. The science objectives for these missions are translated into functional requirements. These are further flowed down to engineering system requirements. It is these engineering requirements, specifically relating to dynamics and controls, which constitute the notion of “performance” in the present work.



**Figure 1.1:** Hubble Space Telescope and proposed successor missions as part of NASA’s space science program. Sample requirements flowdown for SIM.

While the HST has performed admirably well over the last decade [27], it is essentially a multi-purpose instrument providing imaging and spectroscopy capabilities in the wavelength range  $0.110\text{-}2.6 \text{ } [\mu\text{m}]$ , i.e. from ultraviolet (UV) to near-infrared (NIR). In order to achieve this large scope of science capabilities, a number of engineering compromises had to be made.<sup>2</sup> The astronomical science community has realized that specialization is necessary such that the ambitious astrophysical research goals of the first half of the 21<sup>st</sup> century (e.g. observation of proto-galaxies at high redshifts  $z$ , direct IR detection of extra-solar earth-sized planets out to 15 parsecs) can be achieved [105]. Consequently a number of successor spacecraft have

<sup>2</sup>The angular resolution for HST is given by the Dawes limit  $\theta_{res} \approx \lambda/D$ , where  $D = 2.4 \text{ [m]}$  is the diameter of the primary mirror and  $\lambda$  is the observing wavelength. Thus at  $\lambda = 0.63 \text{ } [\mu\text{m}]$  (WFPC2) the observatory achieves an angular resolution (FWHM) of 0.053 arcsec. In the infrared regime at  $\lambda = 1.6 \text{ } [\mu\text{m}]$  as seen with the NICMOS camera, however, the achievable angular resolution is reduced to 0.14 arcsec.

been proposed (Figure 1.1).

At first sight it appears impossible to attempt a unified engineering treatment of these various missions due to the large differences in their respective science objectives. Once these objectives have been broken down into tangible engineering requirements, however, the missions can be analyzed with a common set of tools. All missions require that electromagnetic radiation emanating from a science or guide source (e.g. star, proto-galaxy, extra-solar planet...) is collected by an aperture, compressed and redirected to an electronic detector (e.g. spectrophotograph, CCD camera, fringe tracker). During this process it is paramount that the distortion of the wavefront (surface of common phase of light) inside the optical train be kept to a minimum, while the boresight axis of the observatory be held nearly fixed in inertial space<sup>3</sup>.

For interferometers, additional requirements for the angular propagation of the wavefront (wavefront tilt-WFT), the pathlength the light travels in the different arms of the interferometer (optical pathlength difference-OPD) and the amount of overlap the interfering light beams experience at the detector (beamshear-BS) must be formulated. In order to ascertain that these telescopes will meet their stringent phasing and pointing requirements, the path from disturbance sources to the performance metrics of interest must be modeled in detail before construction, integration and testing. Additionally, for a number of light-weight deployable structures, pre-launch tests in a 1-g gravity field are not feasible. Hence, it is paramount that a preliminary design of the system is available, which can be used as a basis for a simulation model.

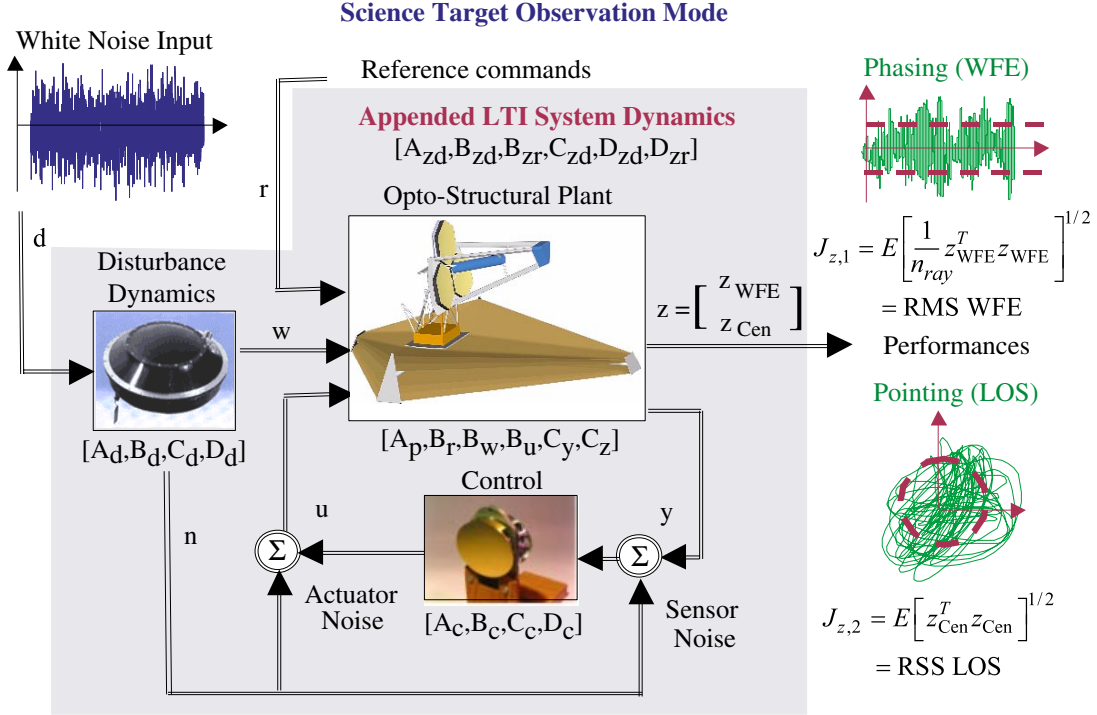
The science target observation mode is in quasi steady-state and is of particular importance. Other modes of interest can be transient such as the slewing and acquisition mode. Figure 1.2 shows a simplified block diagram of the main elements involved in a steady-state dynamics simulation. This is the reference problem setting considered in this thesis. The premise is that a number of disturbance sources (reaction wheel assembly, cryocooler, guide star noise, etc.) are present during the science target observation mode as zero-mean random stochastic processes [11]. Their effect is captured with the help of state space shaping filters<sup>4</sup>, such that the input to the appended system dynamics is assumed to be a vector of

---

<sup>3</sup>A common inertial reference frame is the equatorial system, where the location of a celestial object is defined by its declination  $\delta$  and right ascension  $\alpha$  [65].

<sup>4</sup>Sometimes these are referred to as pre-whitening filters.

unit-intensity white noises  $d$ , which are generally uncorrelated between disturbance sources, see Equation (1.1). Reference input commands are designated as  $r$ . The simplest assumption is that the reference commands are zero, i.e.  $r = 0$ , this, however, is not always the case.



**Figure 1.2:** Reference problem: Science target observation mode of a space telescope with pointing (RSS LOS) and phasing (RMMS WFE) performances.

The shaped disturbances  $w$  are then propagated through the opto-structural plant dynamics, which include the structural dynamics of the spacecraft and the linear sensitivity optics matrices [120]. A compensator is often present in order to stabilize the observable rigid body modes (attitude control) and to improve the disturbance rejection or tracking capability (optical control). The sensor outputs  $y$  and actuator inputs  $u$  might also be subject to colored noise  $n$ . The goal of a disturbance analysis is to accurately predict the expected values of the performances  $J_{z,i}$ , where  $i = 1, 2, \dots, n_z$  and  $n_z$  is the number of performance metrics. This has been previously developed and demonstrated by Gutierrez [45]. A summary of the disturbance, sensitivity and uncertainty analysis framework is contained in Appendix A. Outputs of the appended dynamics model are opto-scientific metrics of interest,  $z$ . The performances are typically expressed in terms of the root-mean-square (RMS) of the outputs. Alternatively we can combine channels in a RSS or RMMS metric, see Appendix A for details. Note that  $n_{ray}$

is the number of light rays traced to compute the wavefront error (WFE). Other performance metrics could be the infinity norm  $J_{z,i} = \|z_i\|_\infty$  or settling time  $J_{z,i} = Ts_{z,i}$  of a particular transient signal.

Another objective is to identify the “key” modal and/or physical parameters of the system that strongly drive the system performance. Sensitivity analysis has been previously identified [45] as a useful tool for examining the dependency of the predicted performance values  $J_{z,i}$  on these “key” system parameters  $p_j$ , where  $j = 1, 2, \dots, n_p$  and  $n_p$  is the number of parameters<sup>5</sup>. Some or all of the parameters might be subject to uncertainty. Oftentimes the number of parameters,  $n_p$ , for which a designer has to determine specific values exceeds the number of performance metrics,  $n_z$ , i.e.  $n_p - n_z \geq 1$ . The traditional approach is to first choose reasonable numbers for the system parameters  $p_j$  and to predict the resulting performances  $J_{z,i}$  (initial performance assessment). If all or some of the predicted performances do not initially meet the specified requirements  $J_{z,req,i}$  for  $i = 1, 2, \dots, n_z$ , including margins, a sensitivity analysis can provide partial derivatives  $\partial J_{z,i}/\partial p_j$  which can be used to identify in which direction important parameters  $p_j$  should be changed. This is intended to drive the system to a design point that satisfies all requirements, i.e. a condition where  $J_{z,i} \leq J_{z,req,i}$  for all  $i = 1, 2, \dots, n_z$  is true. This process is called performance enhancement [45]. A  $\pm\%$  uncertainty on the predicted performances,  $\pm\Delta J_{z,i}$ , can be computed based on known or assumed  $\pm\%$  uncertainties,  $v_j$ , of the parameters  $p_j$ . This is useful in establishing performance error bounds.

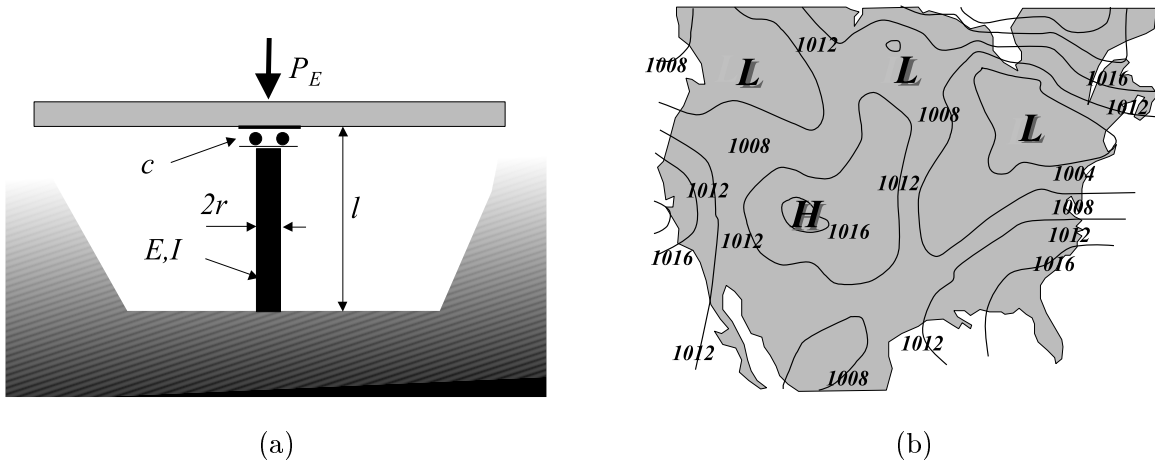
## 1.2 Isoperformance Motivation and Analogies

Once a nominal design has been found that meets all requirements with sufficient margins, it is important to realize that this design is generally not unique. It is likely that different combinations of values for the “key” system parameters,  $p_j$ , will yield the same predicted system performances  $J_{z,i}$ . Consider the following analogy from civil engineering (statics). A metal supporting column for a bridge, see Figure 1.3(a), is to be designed, such that it can sustain an axial buckling load of 500 metric tons<sup>6</sup> with a 100% safety margin (=9.81 [MN]). The buckling load in this case represents the “performance” and the static axial load is the static analog of the dynamic “disturbance”. The critical buckling load,  $P_E$ , is given by Euler’s column formula as  $P_E = c\pi^2 EI/l^2$ , where  $E$  is the modulus of elasticity,  $I$  is

<sup>5</sup>It is assumed that these parameters are continuous over their interval  $p_j \in [p_{LB,j} p_{UB,j}]$

<sup>6</sup>Buckling occurs at the bifurcation point, when the column becomes elastically unstable.

the column moment of inertia,  $l$  is the length of the column and  $c$  is a boundary condition-dependent parameter, the so called “end fixity coefficient” [121, 140]. The bending moment of inertia  $I$  of the circular column is a function of the radius  $r$  according to  $I = \pi r^4/4$ . The end fixity coefficient is  $c = 2.05$ , assuming that the column has a uniform cross section, is axially loaded, has one end fixed (floor) and one end pinned (ceiling). We can reasonably assume that  $l = 15$  [m] and that this length is a constant given by the overall design of the bridge. The remaining variable design parameters are therefore the elastic modulus  $E$  (material) and the column radius  $r$  (geometry). The performance specification can be met if we use steel V2A (74% Fe, 18% Cr, 8% Ni) with  $E = 19.1 \cdot 10^{10}$  [Nm<sup>-2</sup>] and a column radius of  $r = 16.42$  [cm]. Alternatively the same performance can be achieved if we use aluminum (99.99% Al) with  $E = 7.1 \cdot 10^{10}$  [Nm<sup>-2</sup>] and a column radius of  $r = 21.03$  [cm]. In fact, we see that the “contour of equal performance” is given by the equation  $E(r) = (4P_E l^2 / c \pi^3) \cdot r^{-4}$ . The critical buckling load will remain constant as long as the product  $EI$  is kept constant. In practice the choice of materials and geometry in the bridge example will be governed by additional considerations such as cost, corrosion resistance or ease of inspection.



**Figure 1.3:** (a) Simple model of axially loaded column in bridge analogy. (b) United States sea level pressure distribution [mbar], 1600 Z time, Tuesday, May 9th, 2000, based on <http://ww2010.atmos.uiuc.edu> [110].

In the design of high-precision opto-mechanical systems, such as ground and space observatories, it is also important to find the combinations of design solutions that will yield equal performance (thus the term “isoperformance”). In this multi-disciplinary field the disturbance, structural, optical and control parameters,  $p_j$ , interact in complex and sometimes

non-intuitive ways<sup>7</sup>. Knowledge of this interaction can give valuable insight into the design space and predicted behaviour of the system. From an economic perspective an “isoperformance” analysis is important, since it can help to design a product that achieves the required performance at a more affordable cost. Consequently, according to Wehner [139], the system parameters that have small sensitivities are also important in this context, since they can be changed (presumably in the less costly direction) without significantly affecting system performance. This is different from classical optimization of performance, where the goal is to find the design which provides the best performance, i.e. minimization or maximization of a performance objective function given a set of constraints. From a systems engineering and project management perspective an isoperformance analysis is useful, since it supports multiobjective design optimization, error budgeting and to some extent technology roadmapping. For example if the requirements levied on a subsystem in the preliminary design phase cannot be met during detailed design, the “burden” has to be redistributed within the system. Knowledge of the locus of design points that will (theoretically) yield equal performance is useful in such a scenario.

The isoperformance loci are surfaces or contours, which can be compared to the isobars on a weather chart. In this analogy the system “parameters” would be the geographic longitude and latitude and the “performance” would correspond to the atmospheric pressure in millibars at a given altitude. Figure 1.3(b) depicts a pressure chart for the United States at mean sea level as recorded on May 9, 2000. The isobars are the lines of constant pressure. The isobars are generated from mean sea level pressure reports from individual weather stations and are useful in locating areas of high and low pressure, which correspond to the positions of surface cyclones and anticyclones. A map of isobars is also useful in identifying strong pressure gradients which are revealed by a tight packing of the isobars. Stronger winds are associated with larger gradients in pressure [110]. For dynamic opto-mechanical systems, areas where the isoperformance lines are close together are “steep”-gradient areas. This analogy will be exploited in a later chapter and is related to the notion of performance robustness, assuming that some of the variable parameters are uncertain. In these regions of the trade space the performance is very sensitive to small changes in the system parameters.

In summary, the motivation for the isoperformance methodology is to obtain a tradeoff tool in support of the multiobjective design optimization, error budgeting and technology

---

<sup>7</sup>The thermal aspects are also important but are ignored in this thesis, see Section 1.3.3.

roadmapping processes for precision opto-mechanical systems. These processes rely heavily on integrated modeling, model conditioning, as well as disturbance and sensitivity analysis. A tool is needed in order to obtain specifications on the system plant and control parameters and error sources, which will ensure that a required performance level is met, while minimizing the cost and risk of the system. The need for computational tools, which provide such a rapid synthesis and simulation capability has been recognized by NASA [138]. This is the impetus for the following formal thesis problem formulation.

## 1.3 Thesis Problem Formulation

### 1.3.1 Thesis Objectives

The first objective of this thesis is to develop a comprehensive multivariable isoperformance methodology for precision opto-mechanical systems. In other words, given the required system performances  $J_{z,req,i}$ , where  $i = 1, \dots, n_z$ , attempt to find a set of independent solution vectors,  $p_{iso} = [p_1, p_2, \dots, p_{n_p}]^T$ , whose elements are the variable parameters,  $p_j$ , such that an efficient system design can be achieved. The *theoretical* questions to be answered by such a framework are:

1. Can a specific precision opto-mechanical system described by an integrated model in state space form meet all required performance levels  $J_{z,req,i}$  for given variable parameter bounds  $p_{LB,j} \leq p_j \leq p_{UB,j}$  ?
2. What is the locus (set) of points  $p_{iso}$  in  $\mathbb{R}^{n_p}$  that will produce the required performances  $J_{z,i}(p_j) \equiv J_{z,req,i}$  (including margins) within an allowable numerical tolerance,  $\tau$  ?
3. How can the set of isoperformance points,  $p_{iso}$ , be approximated by a functional relationship  $p_{iso} = f_{iso}(t_l)$ , where  $l = 1, 2, \dots, n_p - n_z$  ?

A second objective of the thesis is to demonstrate the usefulness of this methodology in the context of dynamics and controls system engineering problems. Specifically the research attempts to answer the following *applied* questions:

1. What are the specific values for disturbance, plant, optics and control parameters,  $p_j$ , of a given dynamic system that meet the isoperformance condition and at the same time

minimize a scalar or multiobjective cost function  $J_c$  and risk function  $J_r$  ? (Multiobjective Design Optimization)

2. What is a physically realizable error allocation  $\Psi_{l,i}$  for the anticipated disturbance sources  $l = 1, 2, \dots, n_d$  and the  $i$ -th performance of a dynamic LTI system ? (Dynamics Error Budgeting)
3. Assuming that the required performance levels  $J_{z,req,i}$  are not achievable within the specified parameter bounds (state-of-the art), which constraints should be moved (and by what amount) in order to achieve the required performance ? (Technology Roadmapping)

A third objective of the thesis is to implement the isoperformance framework and to experimentally demonstrate its ability to predict correct isoperformance loci given a system model and corresponding laboratory artifact:

1. Implement the isoperformance methodology in a common and user-friendly technical computing environment and thoroughly test it with various system models of increasing complexity.
2. Validate the resulting toolbox with experimental results from a laboratory testbed in a 1-g environment.
3. Obtain insights about the level of model fidelity required to accurately predict the isoperformance surfaces in real world systems.

Finally a fourth objective is to demonstrate if and how the kernel isoperformance algorithms can be applied to engineering problems outside of the dynamics and controls area, e.g. in metal fatigue engineering.

### 1.3.2 Mathematical Problem Definition

An appended state space representation of the dynamics of a closed-loop or open-loop linear time-invariant system is given as

$$\begin{aligned} \dot{q} &= A_{zd}(p_j) q + B_{zd}(p_j) d + B_{zr}(p_j) r \\ z &= C_{zd}(p_j) q + D_{zd}(p_j) d + D_{zr}(p_j) r, \text{ where } j = 1, 2, \dots, n_p \end{aligned} \tag{1.1}$$



where  $A_{zd}$  is the state transition matrix,  $B_{zd}$  and  $B_{zr}$  are the disturbance and reference input coefficient matrices,  $C_{zd}$  is the performance output coefficient matrix,  $D_{zd}$  and  $D_{zr}$  are the disturbance and reference feedthrough matrices,  $d$  are unit-intensity white noise inputs,  $r$  are reference inputs,  $z$  are system performance outputs,  $q$  is the state vector and  $p_j$  are the independent variable system parameters. Given that the functionals

$$J_{z,i}(p_j) = F(z) \text{ , e.g. } J_{z,i} = E [z_i^T z_i]^{1/2} \quad (1.2)$$

where  $i = 1, 2, \dots, n_z$ , are a definition of the performance metrics of interest, find a set of vectors,  $p_{iso}$ , whose elements are the independent variable parameters,  $p_j$ , such that the performance equality (isoperformance) constraint

$$J_{z,i}(p_{iso}) \equiv J_{z,req,i} \quad \forall i = 1, 2, \dots, n_z \quad (1.3)$$

is met, assuming that the number of parameters exceeds the number of performances

$$n_p - n_z \geq 1 \quad (1.4)$$

and that the parameters  $p_j$  are bounded below and above as follows:

$$p_{j,LB} \leq p_j \leq p_{j,UB} \quad \forall j = 1, 2, \dots, n_p \quad (1.5)$$

The isoperformance condition (1.3) has to be met subject to a numerical tolerance,  $\tau$

$$\left| \frac{J_z(p_{iso}) - J_{z,req}}{J_{z,req}} \right| \leq \frac{\tau}{100} \quad (1.6)$$

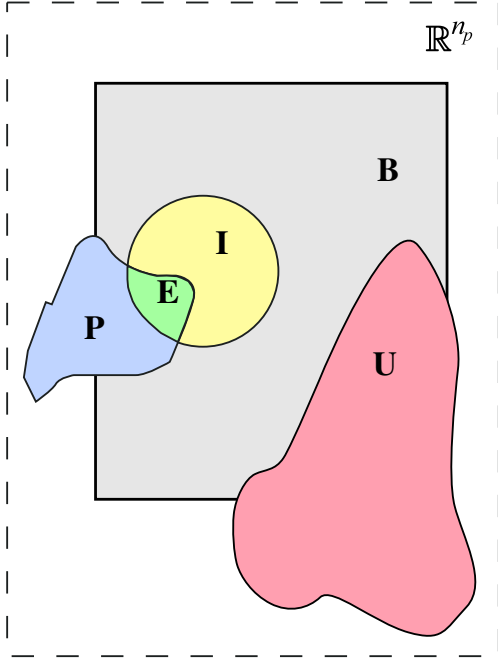
If scalar or vector (multiobjective) cost functions,  $J_c$ , and risk functions,  $J_r$ , are given, solve a constrained non-linear optimization problem such that

$$\begin{aligned} & \text{NLP} \\ & \min [\eta J_c^T Q_{cc} J_c + (1 - \eta) J_r^T Q_{rr} J_r] \\ & \text{such that } p_{iso} \in \mathbf{I} \text{ and } p_{j,LB} \leq p_j \leq p_{j,UB} \\ & \text{and } \eta \in [0 \ 1] \end{aligned} \quad (1.7)$$

where the weight  $\eta$  is used to trade between cost and risk objectives and  $Q_{cc}$  and  $Q_{rr}$  are cost and risk weighting matrices respectively. The set  $\mathbf{I}$  is the performance invariant (isoperformance) set, containing only solutions satisfying (1.3).

Alternatively this can be formulated in terms of set theory. Figure 1.4 shows various sets in the vector space  $p = [p_1 \ p_2 \ \dots \ p_{n_p}]^T$  and their mutual relationship in the general case. The

goal is to first find the elements of the isoperformance set  $\mathbf{I}$ . Note, that since the performance requirements are bounded, i.e.  $|J_{z,req,i}| < \infty \forall i$ , it is true that the intersection  $\mathbf{U} \cap \mathbf{I} = \emptyset$ . In other words only stable solutions can be part of the isoperformance set, thus  $\mathbf{I} \subset \overline{\mathbf{U}}$ , where the overline denotes the stable, complementary set  $\overline{\mathbf{U}} = \{x|x \notin \mathbf{U}\}$ . Our ultimate goal is to find a family of designs  $p_j^*$ ,  $j = 1, 2, \dots, n_p$ , which are elements of the efficient set  $\mathbf{E}$ . The efficient set is the intersection of the isoperformance set  $\mathbf{I}$  and the pareto optimal set  $\mathbf{P}$ , i.e.  $\mathbf{E} = \mathbf{I} \cap \mathbf{P}$ .



set	description
$\mathbb{R}^{n_p}$	$n_p$ -dimensional <b>R</b> eal valued Euclidean vector space
$\mathbf{B} \subset \mathbb{R}^{n_p}$	subset of $\mathbb{R}^{n_p}$ , which is <b>B</b> ounded by (1.5)
$\mathbf{I} \subset \mathbf{B}$	subset of $\mathbf{B}$ , which satisfies <b>I</b> soperformance, see (1.3)
$\mathbf{U} \subset \mathbb{R}^{n_p}$	<b>U</b> nstable subspace, where $\max(\text{Re}(\lambda_i)) > 0$
$\mathbf{P} \subset \mathbb{R}^{n_p}$	<b>P</b> areto optimal subset, satisfies (1.7) without constr.
$\mathbf{E} = \mathbf{I} \cap \mathbf{P}$	<b>E</b> fficient subset, satisfies (1.7) with constraints

**Figure 1.4:** Sets related to thesis problem definition. The eigenvalues  $\lambda_i$  are obtained by solving the eigenvalue problem  $[A_{zd} - \lambda_i I]\phi_i = 0$ .

### 1.3.3 Limitation of thesis scope

It shall be noted that (quasi) static disturbance (=error) sources, such as thermal mirror distortions, manufacturing aberrations or optical misalignments will not be included in this analysis. Thus the disturbance sources  $d$  are assumed to be zero mean:  $\mu_d = E[d] = 0$ , while  $E[d^T d] \neq 0$ . Furthermore this thesis focuses on the science target observation mode, thus representing a steady state condition. This is true while an observatory is phased up and locked onto a desired science target for the duration of the image or fringe integration time. The concept of isoperformance could, however, also be applied to transient modes of operation.

When applying the isoperformance methodology to precision opto-mechanical systems we will always assume linearity and time-invariance (LTI). This entails the assumption that the system behaviour is linear over the entire definition interval of the parameters  $p_j$ , see Equation (1.5). The scope is limited to continuous time problems (s-domain). The effect of D/A or A/D conversion noise and time delay associated with digital control implementations can be included via shaping filters and Padé approximations, respectively. We will exclude discrete or topology-type design parameters such as structural connectivity or actuator/sensor placement and type. Also it is assumed that a baseline controller has been chosen (e.g. PID, LQG) and that it can be parameterized.

Also, some system parameters need to be fixed ahead of time to make the problem formulation tractable for realistic systems. The thesis will provide guidance, however, as to which parameters to select for an isoperformance analysis. The thesis is essentially limited to dynamics and controls problems for opto-mechanical systems, but application of the core isoperformance algorithms to a crack growth problem in metallic structures is presented in Appendix C. This suggests the use of isoperformance in other fields.

## 1.4 Previous Work

### 1.4.1 Literature Review

This section gives a short overview of the scientific literature which is relevant to the development and validation of the isoperformance methodology. The literature search areas begin with papers on the processes and tools used by systems engineers and designers during conceptual and preliminary design. The current state-of-the-art in performance assessment and enhancement of linear time-invariant systems is discussed along with initial work by other researchers in the area of isoperformance methodology. The building blocks of integrated modeling of such systems are structural dynamics, classical and modern control theory, optical ray tracing as well as empirical and analytical modeling of various disturbance sources. In order to leverage these models and simulations in a multidisciplinary design optimization (MDO) context, issues of numerical conditioning, sensitivity analysis and uncertainty analysis cannot be ignored. Finally, past and presents efforts in laboratory testing and implementation on spaceflight missions are briefly discussed.

The **conceptual and preliminary design phases** are important times during a program

in which various system architectures are analyzed and estimates are made of the top level and subsystem functional requirements. Additionally, initial budget allocations are made and enabling technologies are identified. The allocation of design requirements and resources (costs) and an assessment of risk during these early stages of a program is based on preliminary analyses using simplified models that try to capture the behavior of interest [19]. This was a major driver for the development of tools that allow quantitative analysis and design of these preliminary dynamics models early in a program. The kernel of the performance assessment (disturbance analysis), sensitivity and uncertainty analysis framework, which is used as a starting point for developing the isoperformance methodology was established by Gutierrez [45]. The  $\mathcal{H}_2$ -type performances used here are defined in accordance with Zhou, Doyle and Glover [143].

The theory behind the **performance assessment** of linear dynamical systems is well-developed. A special case of the general performance assessment of a dynamical system is given when stochastic random noise processes are present. In this instance we speak of **disturbance analysis** and governing equations and methodologies are presented in random vibration textbooks such as those by Crandall [18] and Wirching [141]. They characterize the response of systems driven by stochastic inputs in the time-domain (using autocorrelation functions) and equivalently in the frequency-domain (using power spectral density functions). The concept of a linear shaping or “pre-whitening” filter whose input is white noise and whose output is “colored” noise, presumably containing more disturbance energy in some frequency bands than in others, is covered by Brown and Hwang [11]. For the case of state-space systems driven by white noise, the output steady-state covariance matrix is known to be the solution of a Lyapunov equation [45].

The idea of holding a performance metric or value of an objective function constant and finding the corresponding contours has been previously explored by researchers in other areas. Gilheany [36] for example presented a methodology for optimally selecting dampers for multidegree of freedom systems [36]. In that particular work (Fig.5) the contours of equal values of the objective function<sup>8</sup> are found as a function of the damping coefficients  $d_{11}$  and  $d_{22}$ . In the field of **isoperformance** methodology, work has been done by Kennedy, Jones and coworkers [70, 71, 69] on the need within the U.S. Department of Defense to improve

---

<sup>8</sup>The objective function in reference [36] is called ITSE = integral of time multiplied by the sum of squares of displacements and velocities of the masses.

systems performance through better integration of men and women into military systems (human factors engineering). They present the application of isoperformance analysis in military and aerospace systems design, by trading off equipment, training variables, and user characteristics. Once the level of operational performance is settled upon (e.g. “pilot will check out all aircraft flight systems within 30 seconds”), tradeoffs among equipment variables, adaptation, training, and individual predisposing factors can be made<sup>9</sup>. A systematic approach to isoperformance in complex, dynamic opto-mechanical systems such as the next generation of space observatories however is lacking at this time.

A field that has received a lot of attention in the last few years is **integrated modeling**. This encompasses research and efforts to simulate complex systems in a unified and multidisciplinary environment. Several important initiatives in this field, like NASA’s Intelligent Synthesis Environment (ISE) described by Venneri, Malone and coworkers [138] are underway. Important contributions to integrated modeling were made by the Jet Propulsion Laboratory (JPL) with the creation of a MATLAB based finite element package and optical modeling software called IMOS (Integrated Modeling of Optical Systems) [62]. This code was developed to assist in the synthesis of initial models of optical instruments and to reduce the model creation, analysis and redesign cycle as described by Laskin and San Martin [76]. The IMOS package is used extensively throughout this thesis for the generation and manipulation of finite element models of spacecraft and a laboratory testbed.

**Structural dynamics** fundamentals, in particular the single degree-of-freedom oscillator used as a sample problem in this thesis, are treated by Craig [17]. More advanced concepts on the finite-element method are presented by Bathe [4] and Cook [16]. The structural dynamics and controls of large flexible spacecraft have been extensively studied by Junkins and Kim [63] as well as Crawley [19]. As will be seen later the stringent pointing and phasing requirements of opto-mechanical systems often require closed-loop attitude and optical **control**. Thus, actuators, sensors, and compensators must be included in the integrated model. Control textbooks by Van de Vegte [137] and Ogata [111] provide an overview of classical control design techniques, while those of Zhou *et al.* [143] and Bélanger [6] emphasize modern control theory (state space based). Typically control systems are implemented on digital computers

---

<sup>9</sup>A company named Isoperformance Inc. was founded as a result of this research. More information can be found at the following URL: <http://www.isoperformance.com>. The company focuses on human factors engineering.

as described by Åström [1] and Franklin and Powell [32]. It is interesting to note that there still is a large discrepancy between theoretical modern control theory based on LQR, LQG and sensitivity weighted LQG, H-infinity and  $\mu$ -synthesis techniques and controllers that are actually implemented in space flight vehicles up to this day. Aerospace contractors, NASA and other space agencies still rely mainly on analog or digital implementations of classical control techniques such as PID and loop-shaping for attitude, thermal, optical and, among others, end effector control. This is likely due to an aversion towards risk and to the ease with which classical controllers are designed, understood, implemented and tested. Additionally the advantage of extensive flight heritage frequently offsets the potential performance benefits of modern controllers. This fundamental disparity was recognized by Mallory and Miller [86]. They proposed, developed and validated a MIMO controller tuning technique which begins with a simple local (often classical) baseline controller, which is then analyzed and iteratively tuned by opening up promising cross-channels and adjusting controller parameter settings based on a gradient search technique.

The fundamental work that allows the computation of **optical** linear sensitivity matrices and their incorporation into dynamics models is attributed to Redding and Breckenridge [120]. The linear sensitivity matrices allow computing optical metrics such as centroid position on the focal plane, wavefront error, wavefront tilt or beam shear as a function of linear and rotational displacements of the points where elements in the optical train (mirrors, beamsplitters, filter wheels etc.) are attached to the structure. The software program called MACOS<sup>10</sup> (Modeling and Analysis for Controlled Optical Systems) [61] creates the sensitivity matrix based on a prescription of optical elements in the system and unit perturbations of the structural degrees-of-freedom. General recommended references for optics are by Born and Wolf [10] as well as Hecht [50]. Telescope optics in particular are described by Rutten and van Venrooij [123].

When considering a disturbance analysis it is important to enumerate and characterize all potential energy sources that might interfere with the opto-mechanical performance of the system. Eyerman and Shea [30] provide a very complete overview of **spacecraft disturbances**. Reaction wheel disturbances are often expected to be the dominant source and Bialke [7], Davis, Wilson, Jewell and Roden [21], Melody [91] as well as Masterson [89] have contributed to this field. Reaction wheel disturbance models are also included in this thesis and an attempt is made to derive performance derivatives with respect to physical parame-

---

<sup>10</sup>formerly known as COMP

ters of the wheels. These are needed for the computation of isoperformance solutions that involve reaction wheel parameters. Other likely disturbance sources are linear Sterling cryocoolers, attitude determination sensors, guide star noise in a CCD guider camera or A/D, D/A conversion noise.

Once an integrated model has been assembled into a large appended state space model **numerical conditioning** becomes important, before it can be useful in a disturbance, sensitivity or isoperformance analysis. This is due to that fact that numerically ill-conditioned models are likely to produce inaccurate results in time simulations or when solving Riccati or Lyapunov equations. Additionally it is often true that the dynamic response of a complex opto-mechanical system is often dominated by a small subset of states or modes. Large computational savings can be obtained if this subset can be identified and isolated apriori via model conditioning and reduction. Creating a balanced state-space realization according to Moore [102] can not only improve the numerical conditioning properties of a model, but it also serves as a technique for identifying states that can be eliminated from the model due to low observability and controllability [143]. Additional techniques for balancing large state space models are presented by Laub and coworkers [77, 78]. For the case of systems dominated by lightly-damped modes, the method proposed by Gregory [39] is an efficient approach for ranking the importance of these modes. An effective pre-balancing technique was proposed by Mallory [86]. A comprehensive view of model quality management was taken by Uebelhart [136].

A **sensitivity** analysis provides useful information related as to how dependent certain outputs of a model are with respect to modal or physical parameters of the model. A key aspect of the isoperformance methodology is the ability to compute exact analytical sensitivities, as opposed to inexact and inefficient finite difference based gradients which are strongly affected by the chosen perturbation size. These are then assembled into a gradient vector or Jacobian matrix in the multivariable case. The isoperformance technique operates in the nullspace of the Jacobian. A good overview of linear algebra in this context, describing the nullspace and singular value decomposition of a matrix, is provided by Strang [132]. Analytical expressions for the sensitivities of performance metrics previously derived by Gutierrez [45] are used throughout this thesis. A Lagrange multiplier approach was proposed by Jacques [56] to obtain analytical sensitivities of a system's outputs with respect to various parameters. The calculation of sensitivities requires mode shape and frequency derivatives, which fall under the

category of eigenvalue and eigenvector derivatives. A good survey of various eigenderivative methods is provided by Murthy and Haftka [104]. When the parameters are element mass and stiffness properties of a finite-element model, these derivatives can be computed exactly using methods developed by Fox and Kapoor [31] and Nelson [107]. Practical implementation of these methods is done by Kenny [72], and this work is extended by Gutierrez [45]. Previous work by Hou and Koganti [52] in the context of integrated controls-structure design also makes extensive use of sensitivities.

Once an isoperformance set  $\mathbf{I}$  has been computed we are looking for other metrics, which can be assembled into a cost objective function  $J_c$  and/or a risk objective function  $J_r$ , respectively, in order to discriminate between competing solutions. Since not all parameters in a system are known to the same degree, parametric uncertainty represents an important element of risk. **Uncertainty analysis** of the performances (performance robustness) and stability margins (stability robustness) are the topic of publications by Yang [142], How [53], Skelton [126], Zhou *et al.* [143] and the work by Campbell and Crawley [14]. An approximate method for predicting worst-case performance RMS values due to parametric uncertainties is that used by Bryson and Mills [12]. One particular method is the first-order approach that relates the covariance matrix of output quantities in terms of the covariance matrix of uncertain parameters and the sensitivity matrix of the outputs with respect to the parameters.

The multiobjective design optimization aspects of this work draw on previous research results in **multidisciplinary design optimization**. A fundamental book on the theory of multiobjective optimization was published by Sawaragi, Nakayama and Tanino [124]. An important application of multiobjective optimization is concurrent control/structure optimization. The objective is to develop structure and control designs simultaneously such that the overall system design has improved properties compared to a system design obtained through a traditional, sequential approach. This allows the same performance to be achieved with less control effort or less structural mass, for example. This philosophy is carried on in this thesis by additionally including disturbance and optical design parameters. Solutions of these multi-disciplinary optimizations are dependent on the type of objective functionals specified and the programming techniques employed. The method developed by Milman *et al.* [100] does not seek the global optimal design, but rather generates a series of Pareto-optimal designs that can help identify the characteristics of better system designs. This work comes closest to the spirit followed in this thesis. Masters and Crawley use Genetic Algorithms to



identify member cross-sectional properties and actuator/sensor locations that minimize an optical performance metric of an interferometer concept [88]. Performance enhancement on systems with uncertain parameters is treated by Parkinson *et al.* [117] and Pritchard *et al.* [119]. A good overview of structural and multidisciplinary optimization research is given in the volume “Structural Optimization: Status and Promise” edited by Kamat [64]. Crawley *et al.* present a methodology for the conceptual design of controlled structural systems [19]. It applies a controlled structures technology (CST) framework in a consistent level of modeling detail in order to identify and modify the critical disturbance-to-performance transmission paths.

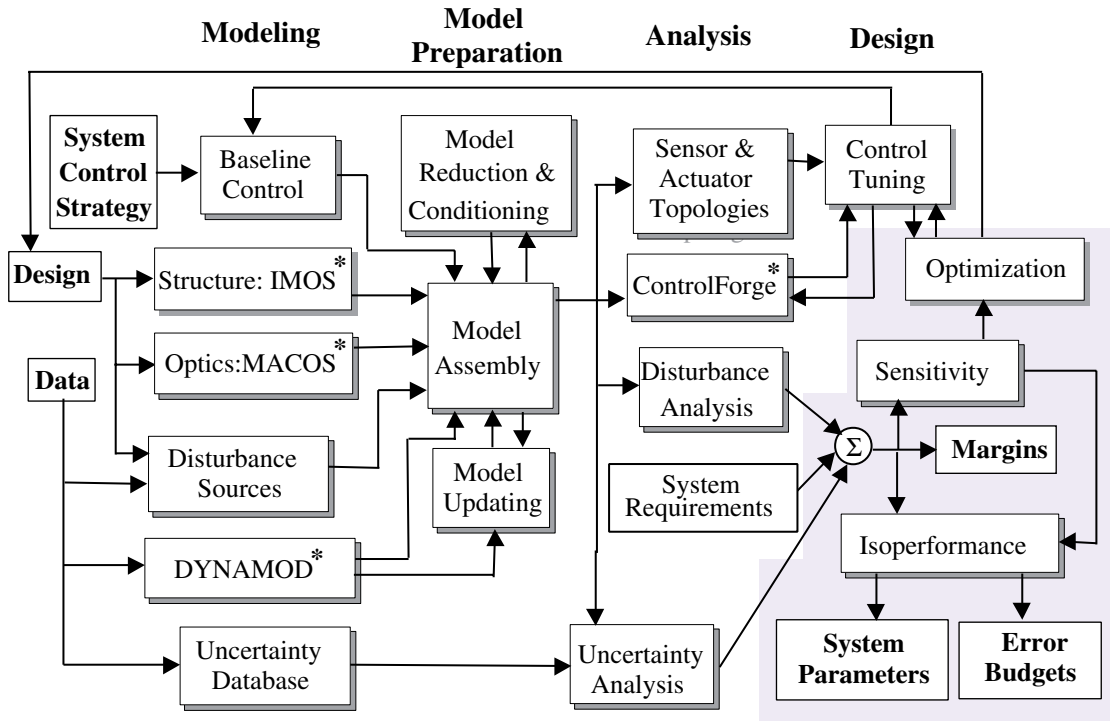
It is interesting to note that the research presented on dynamics and controls of actual **precision spacecraft or laboratory testbeds** is much less voluminous than the theoretical literature in the field. The integrated modeling process is demonstrated by Melody and Neat [94] on JPL’s Micro-Precision Interferometer (MPI) testbed and is experimentally validated based on the comparison of predicted and measured closed-loop transfer functions. Optical pathlength control of a JPL Phase B interferometer testbed was presented by O’Neal and Spanos [113]. Another example is the work done at NASA Langley Research Center by Maghami *et al.* [84]. The pointing performance of a large, laboratory testbed was successfully maintained while control effort was decreased. The genetic algorithm approach to evolutionary design of flexible structures developed by Masters *et al.* was experimentally validated on a closed-loop, truss-like testbed [87]. Mallory [86] presents experiments of slewing, pointing and phasing control of the ORIGINS space telescope testbed.

Apart from the more generic literature in the field we can also find publications devoted to specific **missions**. Disturbance analysis results for space optical systems have been presented for SIM by Grogan and Gutierrez [42],[46], as well as by de Weck along with Miller, Uebelhart, Grogan and Basdogan [98]. Open loop and closed loop NGST results have been published by de Weck, Miller and Mosier [24]. A preliminary analysis of the dynamics of the Terrestrial Planet Finder mission was also prepared by Miller, de Weck and co-workers [96].

#### 1.4.2 Implementation Context (DOCS Framework)

The isoperformance methodology developed in this thesis is implemented in the MATLAB technical computing environment. As such it represents a module of the DOCS analysis framework. The DOCS (Dynamics-Optics-Controls-Structures) framework is a powerful tool

set for the modeling and analysis of precision opto-mechanical space systems. Its development spans over the last 10 years at MIT's Space Systems Laboratory and the former Space Engineering Research Center (SERC). Within the MATLAB environment a model of the spacecraft (or ground system) can be created, which simulates the dynamic behavior of the structure, the optical train, the control systems and the expected disturbance sources in an integrated fashion (Figure 1.5). The existing toolbox is compatible with IMOS (version 5.0) and MACOS<sup>11</sup>, MSC/NASTRAN as well as the packages DynaMod and ControlForge<sup>12</sup>.



**Figure 1.5:** DOCS-Framework block diagram. The shaded area in the lower right is in the scope of this thesis. The modules marked with an asterisk are available commercially or through academic licensing.

The modules in DOCS can be subdivided into four categories, which correspond to the columns of the block diagram in Figure 1.5. The first column contains modules which are useful in the initial **modeling** of the dynamic system under investigation. This system can be a conceptual or preliminary design model of a scientific spacecraft, a spaceborne or ground based telescope or some other high performance dynamic system. The initial design is the

<sup>11</sup> Available for academic licensing from the Jet Propulsion Laboratory (JPL).

<sup>12</sup> Available commercially from Midé Technology Corporation.

starting point for the creation of a structural model, which provides the stiffness and mass distribution of the system. This is typically done via the finite element method (FEM) and produces the natural frequencies,  $\Omega$ , and mode shapes,  $\Phi$ , which are used to obtain a state space representation of the structural dynamics. This can be done via commercial codes such as FEMAP and MSC/NASTRAN or with the help of IMOS. The corresponding optical model typically traces a large number of on-axis and off-axis rays through the system to obtain the linear sensitivity matrices. These matrices relate motions of the attachment points of optical elements to the optical performance metrics (wavefront error, optical path difference, wavefront tilt, beam shear etc.). DOCS has provisions for modeling a large variety of on-board and external disturbance sources, such as reaction wheel noise, cryocooler disturbances, thermal snap, D/A and A/D noise and various sensor noises. In some cases, if disturbance measurement data is available, these disturbance models are empirical and based on actual disturbance data from spacecraft operations or laboratory tests. The module DYNAMOD provides an alternative to finite element modeling, if physical measurements can be made on the system or subsystem. In that case model fitting techniques are used to obtain a MIMO measurement model which is a good fit to the experimentally determined transfer function matrix of the actual system (measurement model). Most high performance dynamic systems rely on feedback control systems to achieve the required performance levels. Thus, a baseline controller (compensator) must be synthesized based on a chosen system control strategy (e.g. local versus global control as in decoupled control loops for attitude and optical control, classical control versus modern control). Finally, it is likely that physical or modal parameters of the system model are uncertain. This uncertainty can be estimated apriori by consulting an uncertainty database of past model errors of similar systems, such as the one provided by Hasselman [47, 48, 49].

The second category of tools in the DOCS framework is dedicated to **model preparation and conditioning**. Model assembly is the process of aggregating the disciplinary sub-models into an overall, integrated model of the system dynamics. This assembled model will be open or closed-loop and can be mathematically represented in transfer matrix form  $G_{zd}(j\omega)$ , as an appended state space system,  $S_{zd}$ , or in system block diagram form (e.g. in Simulink). Typically the integrated model will initially be numerically ill-conditioned. A number of tools are provided to facilitate model “quality management”. This includes verification of absolute and relative stability (margins), extraction of minimal representations, model balancing and

reduction as well as model diagonalization. Model reduction is a critical step, since it reduces the model size leading to faster model evaluation, while maintaining the critical information. The recommended model conditioning in DOCS is a two step procedure with an initial conditioning of the opto-structural plant dynamics alone. The objective of this first step is to eliminate pole-zero cancellations and obtain a well conditioned plant model. The second conditioning and reduction step is more aggressive and is executed on the aggregated model, which contains the frequency weighting introduced by control and disturbance dynamics. Finally a number of software routines are emerging in DOCS which allow model updating of the plant model with measurement data.

After an integrated model has been created and numerically conditioned, the performances such as root-mean-square (RMS) values of opto-mechanical metrics (e.g. pathlength difference, pointing jitter, fringe visibility, null-depth) can be predicted. This capability is provided by the disturbance **analysis** module, which appears in the third column of the DOCS block diagram. The disturbance analysis can be conducted in the time domain, frequency domain or as a Lyapunov analysis. The predicted performance values are compared with the requirements. The goal of the uncertainty analysis is to associate error bars with the predicted performance values, which are based on an uncertainty database resulting from past ground and flight experience. The actuator-sensor topology of the system can be analyzed numerically to ensure that the control system uses the actuator-sensor pairs that will ensure maximum disturbance rejection or tracking performance. This gives an indication if additional sensor-actuator channels should be included in the compensator beyond the ones in the baseline controller. The controller can also be analyzed and modified with ControlForge. This module is the product of control synthesis and analysis work on MACE and other experimental programs and allows implementation of modern control techniques such as sensitivity-weighted LQG.

An initial design will usually not be satisfactory without some amount of iteration. The exact analytical sensitivities of the performance with respect to modal or physical design parameters can be useful in this instance. These sensitivities are essential for conducting gradient-based optimization, redesign or uncertainty analyses. This capability is used in the model updating, ControlForge, uncertainty analysis, optimization, sensitivity and isoperformance modules shown in Figure 1.5. These activities are part of the “**design**” category of tools in DOCS. A control tuning engine was developed by Mallory [86] such that a parameterized

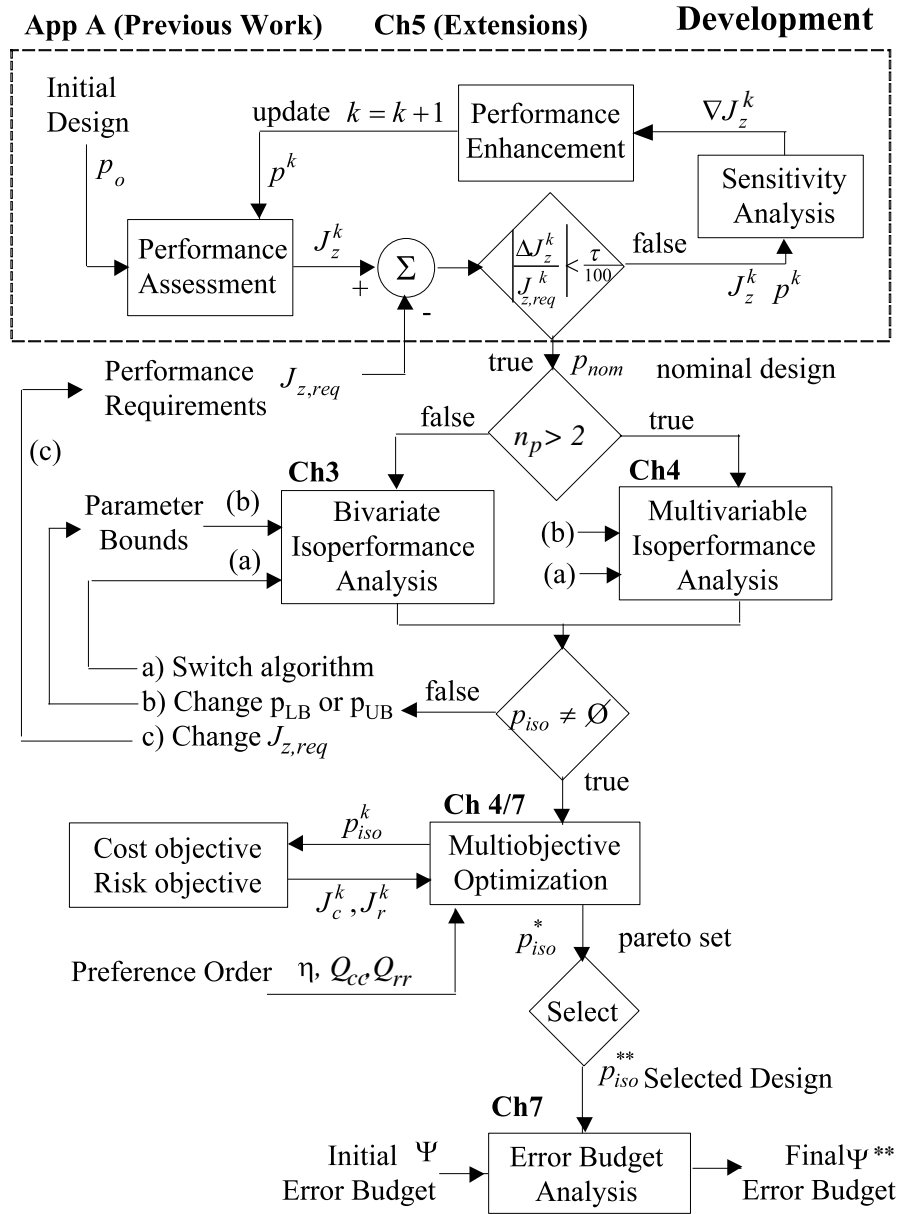
controller is tweaked to provide improved performance with satisfactory stability margins for a fixed plant. An outer optimization loop is carried out during performance enhancement. Here insights from Controlled Structures Technology (CST) are levied such that a combination of performance enhancement steps (disturbance isolation, output isolation, plant redesign, low-authority control (LAC), high-authority control (HAC)) is implemented in the system design. Once a design has been found that meets all requirements with sufficient margins, an isoperformance analysis can be conducted. This module, shown in the lower right hand corner of Figure 1.5, is the main contribution of this thesis. Treating the performance as an equality constraint, the expected error sources (error budgeting) or key design parameters (multiobjective design optimization) can be traded with respect to each other. This is useful in ensuring that the level of difficulty in implementing the design is similar for all subsystems.

If hardware exists, the experimental transfer functions can be used to update the dynamics models throughout the life of the program to achieve a convergent design that will achieve mission success. Preliminary versions of the framework have been successfully applied to conceptual designs of SIM, NGST, the Terrestrial Planet Finder (TPF) mission and NEXUS. Substantial contributions to the framework were made by Blaurock, Gutierrez [45], Jacques [57], Mallory [85, 86], Uebelhart [136], Masterson [89] as well as Miller and co-workers.

## 1.5 Thesis Overview

A thesis roadmap is shown in Figures 1.6 and 1.7. The purpose of this roadmap is to logically organize the thesis research, which solves the problem definition given in Section 1.3. The flow diagram in Figure 1.6 comprises the **development** of the methodology and its implementation. The dashed box depicts essentially the performance assessment and enhancement framework developed by Gutierrez [45], which is briefly summarized in Appendix A.

In order to solve realistic problems with a large number of parameters,  $n_p$ , performance metrics,  $n_z$ , and states,  $n_s$ , it is necessary to solve hundreds of Lyapunov equations involving hundreds to thousands of states. These challenges are addressed in Chapter 5 by presenting numerical methods, which allow efficient solutions of Lyapunov equations for diagonalizable systems, sensitivity computations for similarity state space transformations and a priori error bounds for model reduction. These contributions represent extensions of the research presented by Gutierrez [45] and Uebelhart [136].



**Figure 1.6:** Thesis Roadmap: Development (chapter numbers in bold font)

As shown in Figure 1.6, the analysis process starts with an initial design vector  $p_o$ . The performance assessment calculates the performance vector  $J_z$  and compares it to the requirements  $J_{z,req}$ . If the absolute value of the relative difference  $\Delta J_z^k = J_z^k - J_{z,req}$  between the actual performance vector  $J_z^k$  and the requirement  $J_{z,req}$ , divided by  $J_{z,req}^k$ , is smaller than a numerical tolerance  $\tau/100$ , we have found a solution that satisfies the isoperformance condition, see Equation (1.6). We will call this solution our nominal design,  $p_{nom}$ . If the relative error is larger than  $\tau/100$  we perform a sensitivity analysis, which yields the gradient vector (Jacobian)  $\nabla J_z^k$  at the k-th iteration. This is used in a performance enhancement algorithm, which attempts to drive all performances to the isoperformance condition by updating  $p^k$ . Once  $p_{nom}$  is found we begin the actual isoperformance analysis.

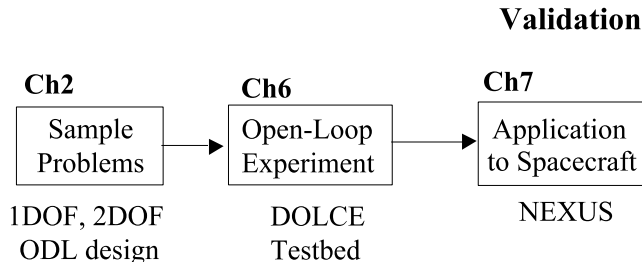
Before trying to attack the full multivariable isoperformance problem, the problem space is restricted to only two parameters  $p_j$ ,  $j = 1, 2$  and one performance  $n_z = 1$ . Thus, if the number of variable parameters is  $n_p = 2$  we perform a bivariate isoperformance analysis, which is developed in Chapter 3. The generalization of the methodology to the multivariable case with  $n_p > 2$  is the topic of Chapter 4. The main result from the isoperformance analysis is a set of points,  $p_{iso}$ , which approximates the isoperformance surfaces in  $\mathbb{R}^{n_p}$ .

If this set is empty, the recommended procedure is to (a) switch to a more general algorithm, (b) modify the upper or lower parameter bounds  $p_{LB}$  or  $p_{UB}$  as indicated by the active constraints or (c) to modify the requirements  $J_{z,req}$ . If this does not lead to a non-zero isoperformance set it is likely that not all performance requirements are achievable with the proposed system architecture.

If an isoperformance solution was found the methodology proceeds to the multiobjective optimization module as described in Chapter 2 for the sample problems and in Chapter 7 for the NEXUS spacecraft. The solutions in the isoperformance set  $p_{iso}$  are evaluated for the cost objective function  $J_c$  and the risk objective function  $J_r$ . A preference order is quantified by the scalar  $\eta$ , which trades cost versus risk and by the normalization/weighting matrices  $Q_{cc}$  and  $Q_{rr}$ , which rank the cost and risk objectives within  $J_c$  and  $J_r$  with respect to each other. The solution is not a single “optimal” point design, but rather a family of pareto optimal designs,  $p_{iso}^*$ , which form the “efficient” set  $\mathbf{E}$ , shown in Figure 1.4. At this point a specific design vector,  $p_{iso}^{**}$ , has to be selected from the efficient set using engineering judgement. This design is then used for an error budget analysis.

The application of the isoperformance methodology to dynamics error budgeting is de-

scribed in Chapter 7. An initial error budget  $\Psi$  is usually established. The (l,i)-entry of the matrix  $\Psi$  is  $\Psi_{l,i}$  and represents the fractional contribution of the l-th disturbance source to the i-th performance metric. This initial budget is input into the error budget analysis module. The module returns a final error budget  $\Psi^{**}$  by finding the actual error contributions for  $p_{iso}^{**}$ , thus insuring that the error budget is physically achievable within the given parameter constraints and underlying integrated model.



**Figure 1.7:** Thesis Roadmap: Validation (chapter numbers in **bold** font)

Figure 1.7 contains the sequential steps used for the **validation** of the isoperformance methodology. In Chapter 2 we introduce three sample problems of increasing complexity, which help in gaining intuitive understanding and confidence in the correct implementation of the governing equations. An experimental investigation is presented in Chapter 6. The experiment uses a structural dynamics testbed called DOLCE with a uniaxial vibration exciter as the surrogate mechanical noise source. The goal of the experiment is to demonstrate the ability of the isoperformance analysis code to predict the shape and locations of isoperformance curves for a physical system, using payload mass and excitation amplitude as parameters. Once confidence has been obtained that the methodology can yield useful results on physical systems it is applied to a realistic spacecraft model. The NGST precursor mission NEXUS was chosen for an in-depth analysis including performance, sensitivity and isoperformance analyses (Chapter 7). The error budgeting process is also briefly discussed. A thesis summary, contributions and recommendations for future work can be found in Chapter 8.



## Chapter 2

# Sample Problems

Throughout the thesis a number of sample problems are used to demonstrate the theory and to explore features of the isoperformance methodology. These sample problems are introduced early in the thesis to avoid cluttering the subsequent presentation and to anchor the development in concrete examples. The sample problems attempt to capture some of the complexity and interesting relationships of larger dynamic systems, with a small number of states and design parameters.

The first sample problem is the classical single degree-of-freedom (SDOF) oscillator problem described by Craig [17]. The oscillator is acted upon by a colored noise random disturbance force and the displacement is highpass-filtered, thus simulating the effect of optical control. The five parameters in this problem are the mass,  $m$ , the linear spring stiffness,  $k$ , the damping coefficient,  $c$ , the disturbance corner frequency,  $\omega_d$ , and the optical control corner frequency,  $\omega_o$ . This set allows conducting some preliminary isoperformance analyses with two or more parameters. The second sample problem is the two degree-of-freedom (2DOF) oscillator problem. It adds the difficulty of dealing with multi-mode systems and multiple performances to the previous problem. The third problem is a simplified single-stage optical delay line (ODL) design. Here issues of trading sensor noise versus process noise, while keeping the resulting OPD performance constant, can be investigated. Also the ODL problem is set up such that the design can go unstable for some combinations of design parameters (e.g. involving the derivative control gain,  $K_d$ ). This corresponds to the unstable subset,  $\mathbf{U}$ , shown in Figure 1.4.

Sample problems have been used by Jacques (“typical sections”) [56] and others to derive insights and draw conclusions about controlled, dynamic structures. Additionally the sample

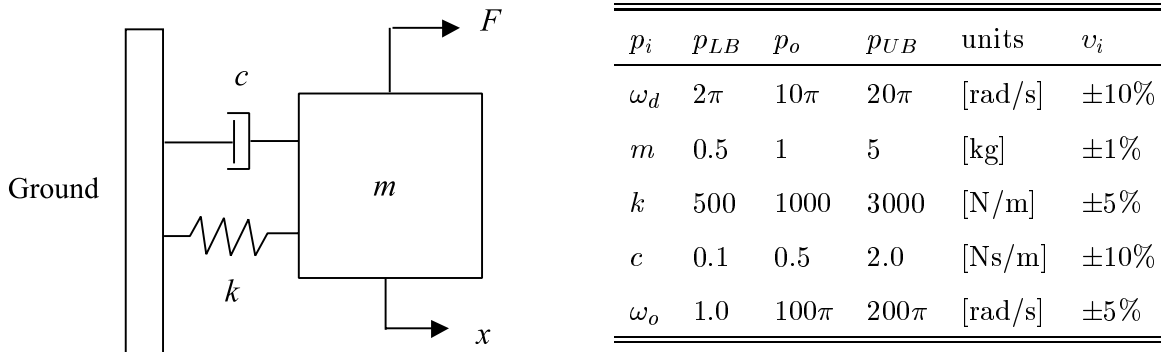
problems are used to ensure that the governing equations have been implemented correctly. While these problems are useful in an academic sense, they do not reflect the complexities, numerical conditioning problems and computational challenges of large order, real-world systems. Chapters 5 and 7 are devoted specifically to large order systems and a case study of a conceptual spacecraft design.

## 2.1 Single Degree-of-Freedom Oscillator

As a simple introduction, the single degree-of-freedom (SDOF) oscillator problem will be considered. This represents the simplest mechanical oscillatory system and has been extensively analyzed by Craig [17] and others.

### 2.1.1 Problem Description

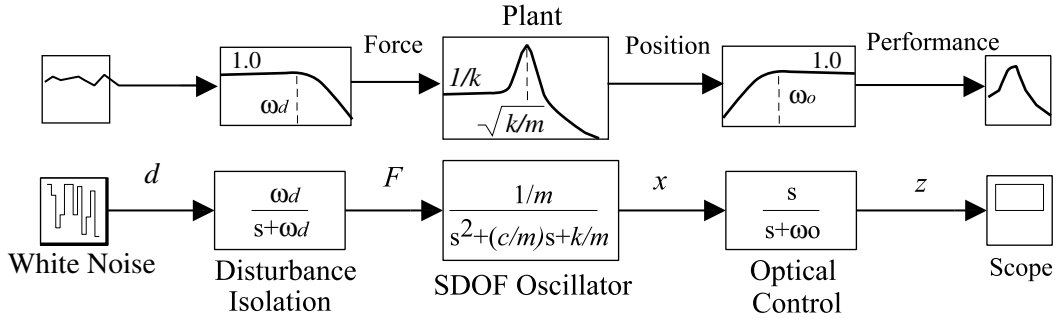
Figure 2.1 shows a schematic representation of the oscillator, which is composed of a mass  $m$  [kg], a linear spring of stiffness  $k$  [N/m] and a linear damper (dashpot) with coefficient  $c$  [Ns/m]. The oscillator is excited by a zero-mean white-noise disturbance force  $F$  [N], which has been passed through a first order low-pass filter (LPF) with unity DC-gain and a corner frequency  $\omega_d$  [rad/sec]. This represents the amount of disturbance isolation present in the system. The displacement  $x$  [m] of the mass is passed through a first order highpass filter (HPF) with corner frequency  $\omega_o$  [rad/sec], thus simulating the effect of an optical controller. This is a simplification, but it is generally true that closing optical control loops is only effective in suppressing disturbances at low frequency due to limitations on the optical sensor sampling frequency. The table on the right side of Figure 2.1 contains the initial parameter values,  $p_o$ ,



**Figure 2.1:** Schematic of single degree-of-freedom (SDOF) oscillator (left) with table of design parameters (right).

as well as upper and lower bounds (range of allowable values) for  $p_i$ , where  $i = 1, 2, \dots, 5$ , as well as a  $\pm$  percentage uncertainty,  $v_i$ , in the true value of the  $i$ -th parameter.

The resulting output  $z$  [m], i.e. the filtered displacement,  $x$ , of the mass, is used to compute the performance. The performance is the RMS of  $z(t)$ , specifically  $J_z = (E[z^T z])^{1/2}$ , where  $E[\cdot]$  denotes the expectation operator [11]. One can also write  $J_z = \sigma_z$ , since for zero-mean stochastic processes the RMS and the standard deviation  $\sigma$  are equal to each other [45]. This system is shown in the block diagram of Figure 2.2.



**Figure 2.2:** SDOF block diagram. From left: white noise disturbance source, disturbance isolation, oscillator and the “optical control” HPF.

The goal is to understand how this performance depends on the variable design parameters, i.e.  $p_j \mapsto J_z(p_j)$  for  $j = 1, 2, \dots, 5$ , where  $p = [\omega_d \ m \ k \ c \ \omega_o]^T$ .

### 2.1.2 Disturbance Analysis

The first step is to compute the performance  $J_{z,o}$  for the initial parameter values,  $p_o$ , given in the table of Figure 2.1. This process is typically called a “disturbance analysis” or “initial performance assessment” [45]. The result,  $J_{z,o}$ , will be compared to a requirement,  $J_{z,req} \leq 0.001$  [m], which has been set based on “customer” needs. Note that  $J_{z,req}$  corresponds to 1 mm of RMS motion of the (filtered) signal  $z$  for the mass  $m$  and about 6 mm of motion peak-to-peak<sup>1</sup>. The three most common methods for carrying out such a disturbance analysis are a time domain analysis with forward time integration of the equations of motion, a frequency domain analysis using power spectral densities (PSD) and a Lyapunov analysis, assuming that the system is driven by white noise. The mathematics behind these three methods are

<sup>1</sup>A 1 Volt RMS random electrical signal will exhibit a waveform amplitude of roughly  $\pm 3$  volts, i.e. 6 volts peak-to-peak on an oscilloscope.

summarized in Appendix A.

One can first write the (linear) equation of motion of the oscillator as:

$$m\ddot{x} + c\dot{x} + kx = F \quad (2.1)$$

For small values of  $c$  (lightly damped plant), the natural frequency of the oscillator is approximately  $\omega_p \simeq \sqrt{k/m}$ . Rewriting (2.1) in state space form and appending the filter dynamics results in the following “integrated model” of the system:

$$\begin{aligned} \dot{q} &= \underbrace{\begin{bmatrix} -\omega_d & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \omega_d/m & -k/m & -c/m & 0 \\ 0 & 1 & 0 & -\omega_o \end{bmatrix}}_{A_{zd}} \underbrace{\begin{bmatrix} q_d \\ x \\ \dot{x} \\ q_c \end{bmatrix}}_q + \underbrace{\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{B_{zd}} d \\ z &= \underbrace{\begin{bmatrix} 0 & 1 & 0 & -\omega_o \end{bmatrix}}_{C_{zd}} q + \underbrace{\begin{bmatrix} 0 \end{bmatrix}}_{D_{zd}} d \end{aligned} \quad (2.2)$$

A numerical time domain simulation was conducted with this system for the initial values  $p_o$  and zero initial state conditions. The resulting time history for a  $T = 40$  [sec] simulation is shown in Figure 2.3. The dark trace corresponds to the initial case  $p_o$ , while the light trace is obtained for the improved design  $p_{nom}$ , which is developed below. The performance  $J_z$  is computed from the output  $z(t)$  as:

$$J_z = \left[ \frac{1}{T} \int_0^T z^2(t) dt \right]^{1/2} \quad (2.3)$$

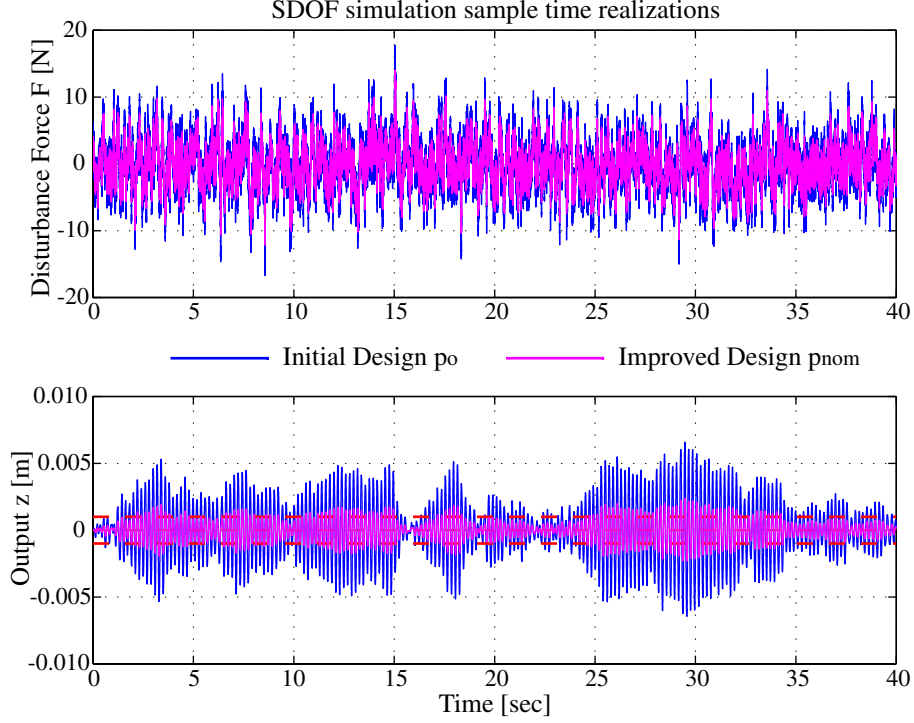
Since this random process is simulated with a digital computer and sampling time  $t_k = k \cdot dt$ , where  $T = N \cdot dt$  and  $N$  is the number of samples,  $J_z$  is obtained as:

$$J_z = \left[ \frac{1}{N} \sum_{k=1}^N z_k^2 \right]^{1/2} \quad (2.4)$$

The resulting performance is  $J_{z,o} = J_z(p_o) = 2.3281 \cdot 10^{-3}$  [m] = 2.3291 [mm], which does not meet the requirement  $J_{z,req}^2$ .

A disturbance analysis was also conducted with the second method (PSD analysis) and the third method (Lyapunov analysis), yielding RMS values of  $2.2232 \cdot 10^{-3}$  and  $2.2231 \cdot 10^{-3}$

<sup>2</sup>In the fields of vibration suppression, precision optics, micro-positioning devices etc... numbers larger than the requirement, say the RMS, usually mean that the performance is *not* met.



**Figure 2.3:** SDOF time simulation: (Top) disturbance  $F(t)$ , (Bottom) output  $z(t)$ . Dark curve for  $p_o$ , light trace for improved design  $p_{nom}$ .

[m], respectively. This corroborates the results obtained from the time simulation and demonstrates the equivalence of the three methods. For large order models, significant discrepancies between the three methods may occur due to different integration schemes (Dormand-Prince [26] was used here), frequency resolution issues and/or numerical ill-conditioning as discussed by Gutierrez [45].

While these three methods are valuable for predicting performance, they do not explicitly reveal how parameters should be changed to improve performance. One way to reveal these trends is to algebraically solve for the performance,  $J_{z,o}$ , via the integral of the PSD  $S_{zz}(\omega)$ , as discussed in earlier work by the author [23]. This results in a closed form expression for the performance,  $J_z$ , as a function of the variable system parameters  $J_z = f(\omega_d, m, k, c, \omega_o)$  as follows:

$$J_z = \sigma_z = \sqrt{\frac{\omega_d^2 (m\omega_o + \omega_d m + c)}{2c(\omega_o + \omega_d) (m\omega_o^2 + k + c\omega_o) (\omega_d^2 m + c\omega_d + k)}} \quad (2.5)$$

For larger systems with many degrees-of-freedom these equations become intractable, such

that a closed form solution cannot generally be obtained. A dimensionality analysis can be done by looking at the magnitude and sign of the largest exponent, with which a specified parameter occurs in Equation (2.5). This suggests that the performance  $J_z$  should be dependent on  $\sim m^{-1/2}$ , thus *increasing* the mass  $m$  of the oscillator should *decrease* the RMS and improve performance. The same is true for  $k, c$  and  $\omega_o$ , which should influence  $J_z$  as  $\sim k^{-1/2}, c^{-1}$  and  $\omega_o^{-1}$ , respectively. For  $\omega_d$  the highest exponents cancel in the numerator and denominator of (2.5) such that it is not directly clear whether an increase or decrease in  $\omega_d$  will improve performance. Intuitively, however, we expect that a *decrease* in the lowpass filter corner frequency should also *decrease* the RMS, since less disturbance energy is entering the system. This is analogous to disturbance isolation.

### 2.1.3 Sensitivity Analysis

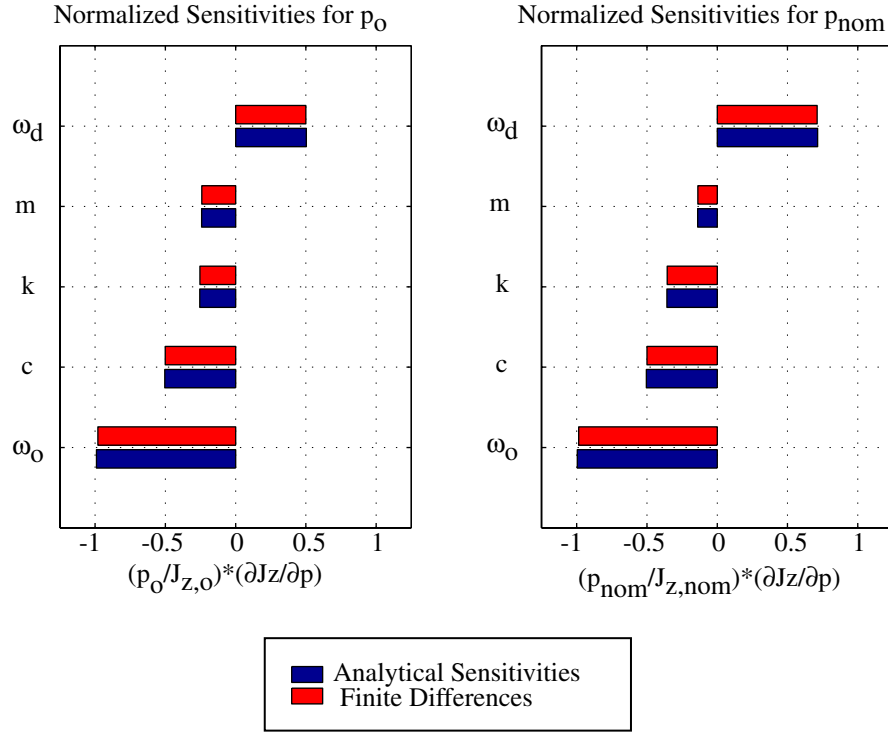
In order to quantify these dependencies of  $J_z$  on  $p$ , a *sensitivity analysis* is performed, which is the second step in the process. This means computing the gradient vector  $\nabla J_z$ , evaluated at  $p_o$ . The governing sensitivity equation (A.32) presented in Appendix A is used to obtain the gradient vector  $\nabla J_z$ , whose elements are the partial derivatives  $\partial J_z / \partial p_j$ . The gradient vector,  $\overline{\nabla J_{z,j}}$ , is normalized.<sup>3</sup> The normalized gradient vector,  $\overline{\nabla J_{z,j}}$ , facilitates comparing the elements of the gradient vector (i.e. the sensitivities) for different parameters.

$$\nabla J_z|_{p_o} = \begin{bmatrix} \partial J_z / \partial \omega_d \\ \partial J_z / \partial m \\ \partial J_z / \partial k \\ \partial J_z / \partial c \\ \partial J_z / \partial \omega_o \end{bmatrix} = \begin{bmatrix} 3.56 \cdot 10^{-5} \\ -5.36 \cdot 10^{-4} \\ -5.66 \cdot 10^{-7} \\ -2.24 \cdot 10^{-3} \\ -7.00 \cdot 10^{-6} \end{bmatrix} \quad \text{and} \quad \overline{\nabla J_z}|_{p_o} = \begin{bmatrix} 0.5032 \\ -0.2413 \\ -0.2547 \\ -0.5040 \\ -0.9899 \end{bmatrix} \quad (2.6)$$

The right vector of normalized sensitivities in Equation (2.6) is easiest to interpret. Even though the gradient vector is only valid in the infinitesimal neighborhood of  $p_o$ , one can say that a 1% increase in the disturbance corner frequency  $\omega_d$ , should lead to approximately a 0.5% increase in the RMS. On the other hand a 1% increase in the optical control “bandwidth”  $\omega_o$  would cause a  $\sim 1\%$  decrease (negative sensitivity) in the RMS. This last result is not

<sup>3</sup>The normalized sensitivity is computed as  $\overline{\nabla J_{z,j}} = (p_{j,o} / J_{z,o}) \cdot (\partial J_z / \partial p_j)$ , where the overbar denotes that the sensitivity is normalized. It is a measure of the relative change in performance per relative change in the parameter value.

surprising, since in precision opto-mechanical systems the performance is often most sensitive to optical control parameters. The normalized sensitivities are graphically shown in Figure 2.4. These sensitivities were confirmed via a finite difference approximation with a 1% perturbation and by analytically taking the derivatives of expression (2.5).



**Figure 2.4:** SDOF Problem: Sensitivities of performance  $J_z$  w.r.t. parameters  $p_o$  for the initial case (left) and after performance enhancement  $p_{nom}$  (right).

#### 2.1.4 Uncertainty Analysis

Given the slope around  $p_o$ , one can estimate reasonable *error bounds* on the predicted performance. This computation has been denoted as an uncertainty analysis. From the Table in Figure 2.1 one finds the  $\pm\%$  uncertainties,  $v_i$ , in the knowledge of the exact value of a design parameter. Note that these values do not take into account non-parametric uncertainty, which is most likely present in opto-mechanical systems as reported by Campbell [13]. In order to get an estimate of the uncertainty in the performance,  $\Delta J_z^v$ , due to the uncertainty in the individual  $p_j$ , the first-order approach introduced by Gutierrez [45] is used. More formal approaches, taking into account the probability density function (pdf) for each parameter, are

also available but are beyond the scope of this thesis. The uncertainty  $\Delta J_z^v$  is then

$$\Delta J_z^v(p) = \sum_{j=1}^{n_p} \left| \frac{\partial J_z}{\partial p_j} \cdot (v_j p_j) \right| \quad (2.7)$$

For the initial solution  $p_o$ , an uncertainty of  $\Delta J_z = 3.676 \cdot 10^{-4}$  [m] is obtained, which corresponds to  $\pm 16.5\%$  of the nominal performance value  $J_{z,o}$ . Thus the (theoretically) predicted RMS performance of the initial system should be quoted as  $J_{z,o} = 2.2231 \pm 0.3676$  [mm]. Even in the best case, it does not meet performance.

### 2.1.5 Performance Enhancement

An attempt to modify the system parameters  $p_o$  is made, such that the system performance falls below the requirement  $J_{z,req}$  with some margin. This will ensure that even the upper uncertainty bound will be below  $J_{z,req}$ . This exercise has been termed “performance enhancement.” A new working requirement of  $0.8 \cdot J_{z,req}$  is adopted, which should provide roughly 20% of performance margin. The gradient vector  $\nabla J_z$  is used in a line search algorithm as follows:

**while**

$$\left| \frac{0.8 J_{z,req} - J_{z,k}}{0.8 J_{z,req}} \right| > \tau/100$$

1. **step size:**  $\alpha_k = \left[ \sum_j \nabla J_{z_j,k} ((-1)\text{sgn}(\nabla J_{z_j,k}) p_{j,k}) \right]^{-1} \cdot \Delta J_{z_k}$

2. **step direction:**  $\Delta p_k = \alpha_k \cdot (-1)\text{sgn}(\nabla J_{z_k}) p_k$

3. **update iterand:**  $p_{k+1} = p_k + \Delta p_k$

4. **update k:**  $k = k + 1$

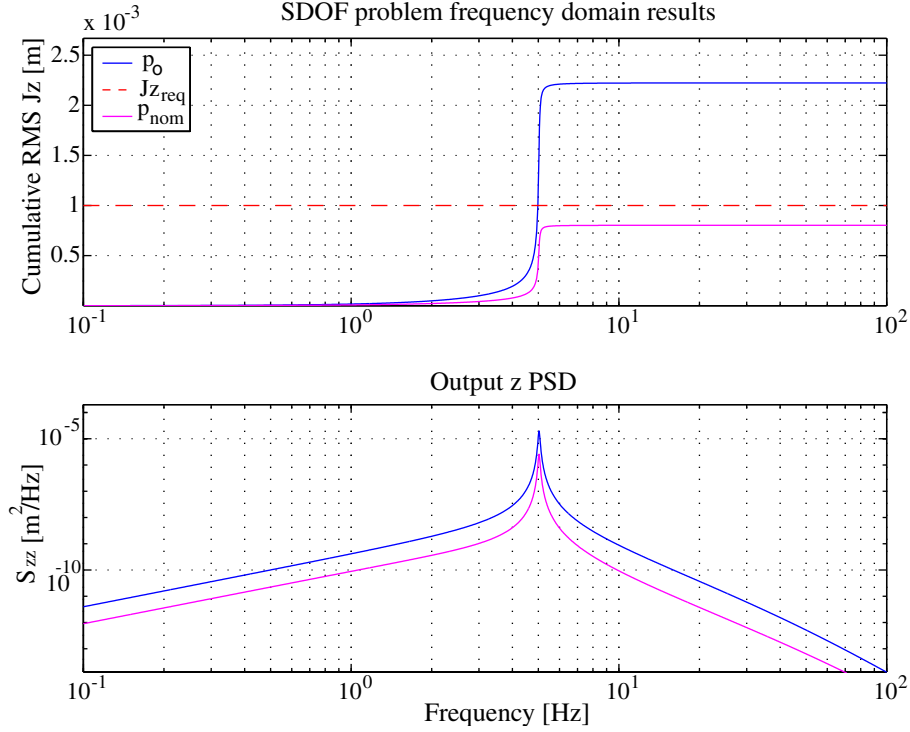
5. **recompute:**  $J_{z_k}$  and  $\nabla J_{z_k}$

**end**

The tolerance  $\tau/100$  was set to a value of 0.01, i.e. the enhanced performance is expected to be within 1% of  $0.8 \cdot J_{z,req}$ . After three iterations, the enhanced solution  $p_{nom} =$



$[\omega_d \ m \ k \ c \ \omega_o]^T = [19.92 \ , \ 1.45 \ , \ 1450 \ , \ 0.725 \ , \ 455.4]^T$  is obtained. It is verified that this solution is still within the upper and lower parameter bounds. The improved performance, including uncertainty, is  $J_{z,nom} = 0.802 \pm 0.153 \text{ mm}$  ( $\pm 19\%$ ), thus remaining under  $J_{z,req}$  even in the worst case.



**Figure 2.5:** (Bottom) PSD function  $S_{zz}(\omega)$  for  $p_o$  (dark) and  $p_{nom}$  (light). (Top) Cumulative RMS curve. Requirements line is dashed.

Comparing this solution,  $p_{nom}$ , to  $p_o$ , one sees that the disturbance corner frequency  $\omega_d$  is smaller, the mass  $m$  and stiffness  $k$  are both increased (thus  $\omega_p$  remains constant), the damping  $c$  is increased (decreases response magnitude), and the optical control bandwidth  $\omega_o$  is increased. The combination of these changes leads to a significant improvement in performance as shown in Figure 2.5. The lower part of the plot shows the power spectral density (PSD) of the performance,  $S_{zz}$ , as a function of frequency. Taking the square-root of the cumulative integral under this function, according to Equation (A.23), leads to the cumulative RMS curve in the top part of Figure 2.5. As expected the large step in the cumulative RMS is caused by the lightly damped resonance of the oscillator which is excited by the broadband disturbance force  $F(t)$ .

Figure 2.4 depicts the normalized sensitivities for the improved design on the right side.

The signs of the sensitivities have not changed, which indicates that the performance function  $J_z(p)$  appears to be monotonic in this neighborhood. The magnitudes of the sensitivities are similar, except that  $J_{z,nom}$  is more sensitive to  $\omega_d$ . This can be explained by the fact that the oscillator natural frequency  $\omega_p$  is at 5 Hz and the disturbance corner frequency  $\omega_d$  is now at 3.2 Hz, thus the resonance is in the rolloff region of the disturbance filter and therefore more sensitive to changes in the filter corner frequency. Also the design  $p_{nom}$  is somewhat less sensitive to changes in the mass  $m$ . A time simulation for the improved design was carried out and the results are shown as the light trace in Figure 2.3. As expected the (filtered) displacements  $z(t)$  of the mass  $m$  are significantly smaller than in the initial case and remain below the requirements line in an RMS sense.

### 2.1.6 Conclusion

In a traditional analysis framework and using the tools presented by Gutierrez [45] one would stop at this point. The requirements are met by the improved design  $p_{nom}$ , it is feasible since the condition  $p_{LB} \leq p_{nom} \leq p_{UB}$  is met and the uncertainty bounds also remain within  $J_{z,req}$ . At first glance there does not seem to be a reason for continued analysis and the improved design could indeed be implemented as is.

The design  $p_{nom}$  is, however, the starting point for the isoperformance analysis, which is the main subject of this thesis. The motivation for continued analysis is the fact that *the improved design is not unique*. Other combinations of  $p_j$ 's will likely achieve the same performance, but perhaps at a lower "cost". Thus other cost criteria such as hardware (implementation) cost or performance robustness to uncertainty can be used to come up with an alternative design. The first step is to find the set (or locus) of design points that yield the same performance as  $J_{z,nom}$ . This is the focus of Chapters 3 and 4.

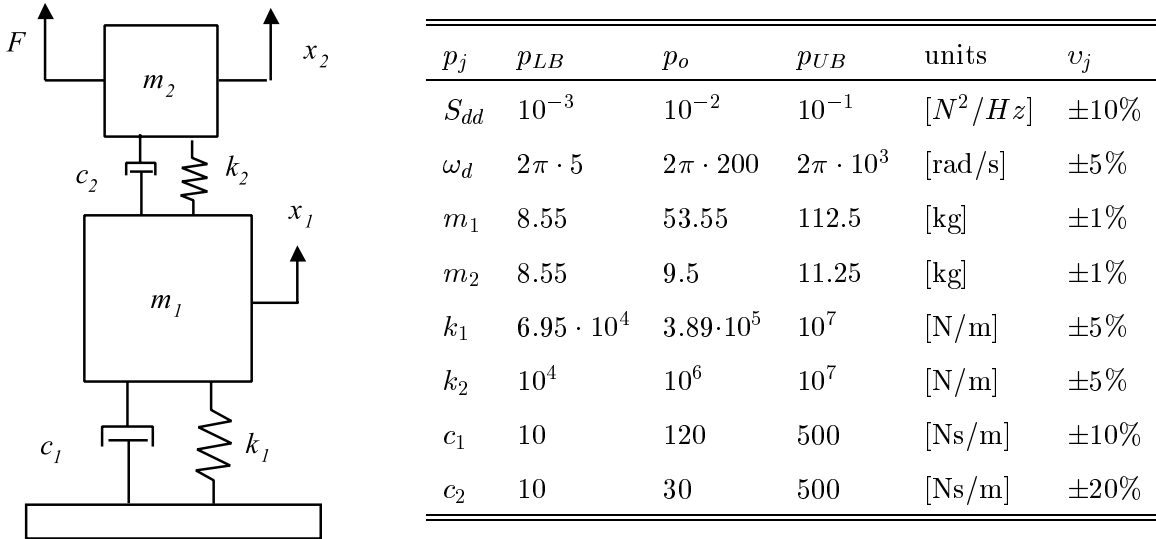
## 2.2 Two Degree-of-Freedom Oscillator

A sample two degree-of-freedom (2DOF) problem was chosen as an extension of the previous example. The two degrees-of-freedom mimic the multi-mode behavior of more complex systems and allow the formulation of an additional performance metric, i.e.  $n_z = 2$ .

### 2.2.1 Problem Description

The system is depicted in Figure 2.6 and consists of two masses  $m_1$  and  $m_2$ , which are connected to ground and to each other by linear springs  $k_1$  and  $k_2$ , respectively. Also there are two linear dampers  $c_1$  and  $c_2$ , which cause velocity-dependent damping forces. Interpreting this problem as a simple spacecraft model, we can view the mass  $m_1$  as the “base” mass (spacecraft bus) and  $m_2$  as an “appendage” mass (payload). The parameters  $k_1$  and  $c_1$  would then represent the effects of a low frequency attitude control system (ACS), which controls the rigid body rotations. The parameters  $k_2$  and  $c_2$  represent the internal damping and compliance of the vehicle. The parameter  $k_2$  will be called the “appendage stiffness”. A colored noise disturbance force  $F$  of intensity  $S_{dd}$  [ $N^2/Hz$ ] and rolloff frequency  $\omega_d$  [rad/sec] is entering at the appendage mass,  $m_2$ , representing a mechanical disturbance generated by onboard machinery.

The two performance metrics of interest are the RMS of the relative displacement (DPL)  $x_2 - x_1$ , i.e.  $J_{z,1} = (E[(x_2 - x_1)^2])^{1/2}$  and the RMS displacement of the first mass,  $J_{z,2} = (E[x_1^2])^{1/2}$ . Figure 2.6 shows a schematic representation of the two DOF oscillator and a table of variable system parameters. This table contains the initial parameter values,  $p_o$ , as well as upper and lower bounds (range of allowable values) for  $p_j$ , where  $j = 1, \dots, 8$ , as well as a percentage of uncertainty,  $v_j$ , in the knowledge of the true value of the  $j$ -th parameter.



**Figure 2.6:** Schematic of two degree-of-freedom (2DOF) oscillator (left) with table of variable parameters (right).

## 2.2.2 Disturbance, Sensitivity and Uncertainty Analyses

As in the previous section, we perform a disturbance, sensitivity and uncertainty analysis. The equations of motion (EOM) can be formulated directly by looking at the freebody diagram for each mass and writing the EOM in matrix form as shown in equation (2.8):

$$\underbrace{\begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix}}_M \begin{bmatrix} \ddot{x}_1 \\ \ddot{x}_2 \end{bmatrix} + \underbrace{\begin{bmatrix} c_1 + c_2 & -c_2 \\ -c_2 & c_2 \end{bmatrix}}_C \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} + \underbrace{\begin{bmatrix} k_1 + k_2 & -k_2 \\ -k_2 & k_2 \end{bmatrix}}_K \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ F \end{bmatrix} \quad (2.8)$$

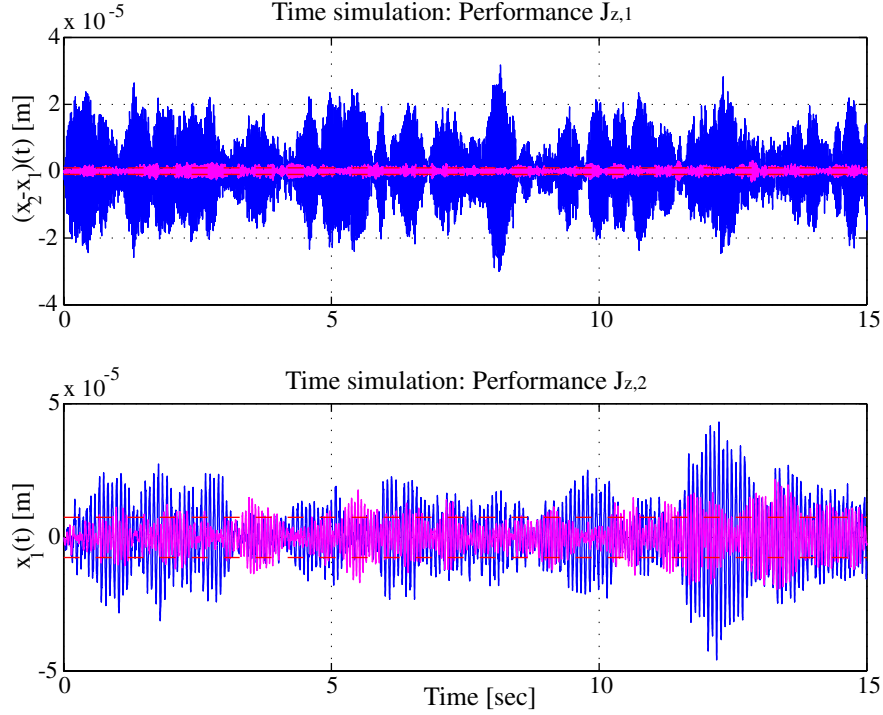
We can then solve the generalized eigenvalue problem for the undamped system or write the state space system directly in physical coordinates. This second approach was chosen here, since it will be easier to compute the Jacobian (matrix of partial derivatives) for this small order model. With the disturbance isolation filter given as  $A_d = [-\omega_d]$ ,  $B_d = [1]$ ,  $C_d = [\sqrt{S_{dd}\omega_d}]$  and  $D_d = [0]$ , we obtain an “integrated model” of the system from (2.8) by pre-multiplying with the inverted mass matrix:

$$\begin{aligned} \dot{q} &= \underbrace{\begin{bmatrix} A_d & 0 & 0 \\ 0 & 0 & I \\ M^{-1}\beta_d C_d & -M^{-1}K & -M^{-1}C \end{bmatrix}}_{A_{zd}} q + \underbrace{\begin{bmatrix} B_d \\ 0 \\ 0 \end{bmatrix}}_{B_{zd}} d \\ z &= \underbrace{\begin{bmatrix} 0 & C_{zx} & 0 \end{bmatrix}}_{C_{zd}} q + \underbrace{[0]}_{D_{zd}} d \end{aligned} \quad (2.9)$$

Note that the system is open loop and that the output coefficient matrix is  $C_{zx} = [-1 \ 1 \ ; \ 1 \ 0]$  and the disturbance influence coefficient matrix is  $\beta_d = [0 \ 1]^T$ . This allows computing the performances  $J_{z,1}$  and  $J_{z,2}$  with the three methods presented in Appendix A. We set the RMS requirements to  $J_{z,req,1} = 10^{-6} \text{ [m]} = 1 \text{ [\mu m]}$  and  $J_{z,req,2} = 7.5 \cdot 10^{-6} \text{ [m]} = 7.5 \text{ [\mu m]}$ . As will be seen in Chapter 6 these requirements are realistic and close to the experimental observations made on the DOLCE isoperformance testbed.

A time simulation was carried out with the Dormand-Prince [26] integrator and an integration step size of 0.001 seconds. The results shown in Figure 2.7 were obtained with Equation (2.4) and parameter vector  $p_o$  and indicate that  $J_{z,1} = 9.37 \text{ [\mu m]}$  and  $J_{z,2} = 11.80 \text{ [\mu m]}$ , which exceeds (i.e. does not meet) the RMS requirement for both performances.

The next step consists of computing the sensitivities of the parameters with respect to the two performances. Note that in this case  $\nabla J_z$  is the Jacobian, which is defined as follows:



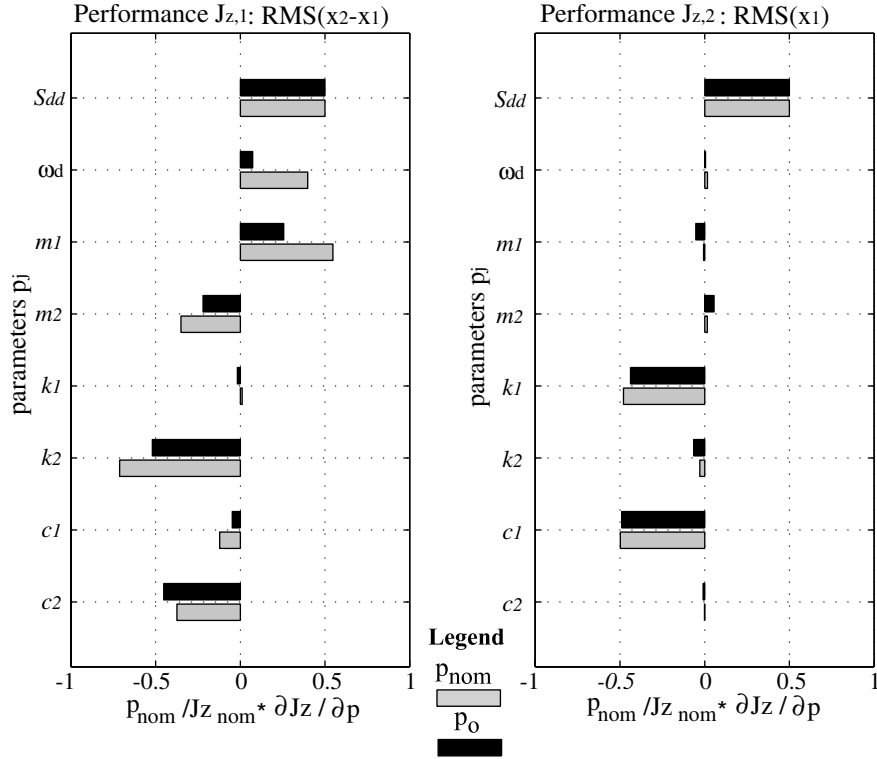
**Figure 2.7:** 2DOF time simulation for  $J_{z,1}$  (top) and  $J_{z,2}$  (bottom). Dark trace for  $p_o$ , lighter trace for improved design  $p_{nom}$ .

$$\nabla J_z = \begin{bmatrix} \partial J_{z,1}/\partial p_1 & \partial J_{z,2}/\partial p_1 \\ \partial J_{z,1}/\partial p_2 & \partial J_{z,2}/\partial p_2 \\ \vdots & \vdots \\ \partial J_{z,1}/\partial p_8 & \partial J_{z,2}/\partial p_8 \end{bmatrix} = \begin{bmatrix} \partial J_{z,1}/\partial S_{dd} & \partial J_{z,2}/\partial S_{dd} \\ \partial J_{z,1}/\partial \omega_d & \partial J_{z,2}/\partial \omega_d \\ \vdots & \vdots \\ \partial J_{z,1}/\partial c_2 & \partial J_{z,2}/\partial c_2 \end{bmatrix} \quad (2.10)$$

The sensitivity analysis is accomplished by using the governing sensitivity equation (GSE) presented in Appendix A. Note that partial derivatives of the inverted stiffness matrix have to be taken. The relationship

$$\frac{dM^{-1}}{dp} = -M^{-1} \frac{dM}{dp} M^{-1}$$

was used to facilitate these calculations [143]. Figure 2.8 shows the normalized sensitivities for the 2DOF sample problem. Note that the left subplot shows the results for performance  $J_{z,1}$  (the first column of the Jacobian) and the right subplot shows the sensitivities for  $J_{z,2}$ , i.e. the second column of the Jacobian.



**Figure 2.8:** 2DOF sensitivity analysis  $\bar{\nabla} J_z$ : (Left) normalized sensitivities w.r.t.  $J_{z,1}$ , (Right) w.r.t.  $J_{z,2}$ . Dark bars show  $p_o$ . Light bars show  $p_{nom}$ .

Decreasing the disturbance force intensity,  $S_{dd}$  will result in a decrease of the RMS of both performances. The disturbance corner frequency,  $\omega_d$ , affects primarily performance  $J_{z,1}$ , but not  $J_{z,2}$ . The reason for this will become apparent shortly. The mass sensitivities are interesting since they have opposite signs for both performances. This means that decreasing  $m_1$  for example will improve  $J_{z,1}$  but worsen  $J_{z,2}$ . These parameters are interesting, since they cannot simply be tuned in one direction to improve all performances at once. This has to be taken into account in the performance enhancement step. While the stiffness sensitivities have the same sign it is noteworthy that  $J_{z,1}$  primarily responds to changes in  $k_2$ , while  $J_{z,2}$  is more sensitive to  $k_1$ . This is intuitively satisfactory, since  $J_{z,1}$ , the RMS of the gap distance between  $m_1$  and  $m_2$ , strongly depends on the compliance between the two masses, whereas  $x_1$  is strongly driven by the suspension mode and thus the value of  $k_1$ . Increasing the damping coefficients also has a positive impact on the performances.

Now that the sensitivities are known, one may again obtain an estimate of the performance uncertainties due to parametric uncertainty, as shown in Equation (2.7). The resulting un-

certain performances are  $J_{z,1} = 9.37 \pm 1.79$  [ $\mu\text{m}$ ] ( $\pm 18.04\%$ ) and  $J_{z,2} = 11.80 \pm 1.404$  [ $\mu\text{m}$ ] ( $\pm 12.74\%$ ), respectively. The requirements were set at  $J_{z,req,1} = 1.0$  and  $J_{z,req,2} = 7.5$  [ $\mu\text{m}$ ]. Thus, even in the best case the system does not meet the performance requirements.

### 2.2.3 Performance Enhancement

The computed sensitivities are used for a performance enhancement exercise. One difference with the SDOF problem is that in this performance enhancement cycle both performance objectives  $J_{z,req,1}$  and  $J_{z,req,2}$  have to be met simultaneously. The simple line search from the SDOF example will not work here, since two performances  $J_{z,1}$  and  $J_{z,2}$  have to be simultaneously driven to the working requirement of  $0.8 \cdot J_{z,req}$  (includes 20% performance margin). The algorithm in this case uses the singular value decomposition (SVD) of the Jacobian as discussed in Chapter 4.

while

$$\max \left( \left| \frac{0.8J_{z,req} - J_{z,k}}{0.8J_{z,req}} \right| \right) > \tau/100$$

1. SVD of Jacobian:  $USV^T = \overline{\nabla J_z^T}$
2. Normalize column space:  $V_n = p_k * V_{1,2}$  (elementwise multiplication)
3. step size vector:  $\alpha_k = [\nabla J_z^T \cdot V_n]^{-1} \cdot \Delta J_{z,k}$
4. step direction:  $\Delta p_k = V_n \cdot \alpha_k$
5. update iterand:  $p_{k+1} = p_k + \Delta p_k$
6. update k:  $k = k + 1$
7. recompute:  $J_{z,k}$  and  $\nabla J_{z,k}, \overline{\nabla J_{z,k}^T}$

end

After five iterations an improved design,  $p_{nom}$ , is found, which meets both working requirements. It turns out that the requirement  $J_{z,req,1}$  is harder to meet than  $J_{z,req,2}$ . The resulting design vector,  $p_{nom}$ , and the initial design,  $p_o$ , are contrasted in Table 2.1

From this comparison one sees that the performance enhancement algorithm has found a viable design  $p_{nom}$ , which is not “optimal”<sup>4</sup>, but rather a “good” design that meets the requirements,  $J_{z,req}$ . As expected, the improved solution requires a reduction in the disturbance

<sup>4</sup>An optimal design would satisfy the Karush-Kuhn-Tucker [33] optimality conditions.

**Table 2.1:** Comparison of  $p_o$  and  $p_{nom}$  with the ratio of parameter values in the second to last column. A non-zero value in the last column indicates that this parameter is at the upper (+1) or lower bound (-1).

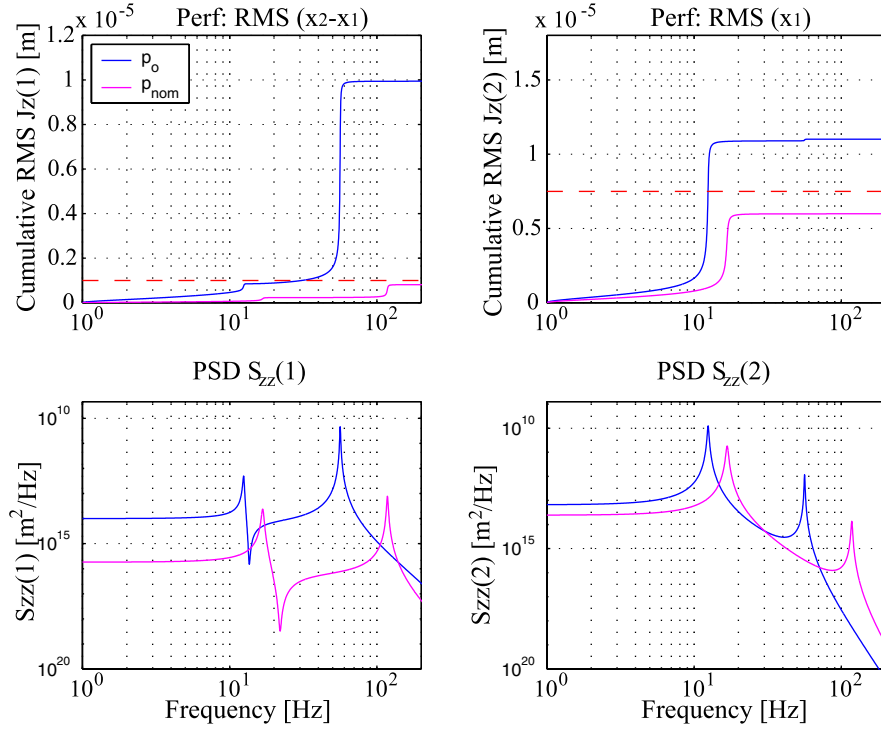
i	$p_i$	Parameter Name	$p_o$	$p_{nom}$	units	ratio	<b>B</b>
1	$S_{dd}$	Disturbance Intensity	$10^{-2}$	$2.36 \cdot 10^{-3}$	[N <sup>2</sup> /Hz]	0.24	0
2	$\omega_d$	Disturbance Bandwidth	$2\pi \cdot 200$	$2\pi \cdot 133.8$	[rad/sec]	0.67	0
3	$m_1$	Base Mass	53.55	16.26	[kg]	0.30	0
4	$m_2$	Appendage Mass	9.5	11.25	[kg]	1.18	+1
5	$k_1$	Base Stiffness	$3.89 \cdot 10^5$	$3.10 \cdot 10^5$	[N/m]	0.80	0
6	$k_2$	Appendage Stiffness	$10^6$	$3.6 \cdot 10^6$	[N/m]	3.59	0
7	$c_1$	Damping Coefficient	120	110.3	[Ns/m]	0.92	0
8	$c_2$	Damping Coefficient	30	83.2	[Ns/m]	2.78	0

force intensity  $S_{dd}$  and a somewhat reduced disturbance corner frequency  $\omega_d$ , which probably only benefited  $J_{z,1}$ . Furthermore mass  $m_1$  was decreased, while mass  $m_2$  was increased to its upper constraint (see +1 in the last column of Table 2.1). This again suggests that  $J_{z,1}$  is the “driving” performance, since these are the changes that would have been indicated by its sensitivities alone, see the left side of Figure 2.8. The same pattern is apparent in the stiffnesses, where increasing the inter-mass stiffness  $k_2$  reduces the RMS of  $x_2 - x_1$ , i.e.  $J_{z,1}$ . Finally increasing the damping coefficient  $c_2$  also reduces  $J_{z,1}$  as expected. The only unexpected change is the decrease in the damping  $c_1$ , since all its sensitivities are negative. A likely explanation is that the exit criterion for the performance enhancement loop on the previous page enforces the “isoperformance condition”. The damping coefficient  $c_1$  is actually decreased (thus worsening  $J_{z,2}$ ) to make the performance  $J_{z,2}$  fall in the performance band dictated by the “while” condition and the tolerance of  $\tau/100 = 0.01$ .

A more intuitive insight into the effects of these changes can be gained by considering the performance PSD’s and their cumulative RMS curves in Figure 2.9. This is a bimodal system and for the first performance,  $J_{z,1}$ , both modes contribute to the RMS, with the second mode being dominant. The second performance,  $J_{z,2}$ , on the other hand is driven by the first mode alone. The performance improvement is primarily achieved by reducing the disturbance magnitude and by stiffening the system, which shifts the modes to higher frequencies. The



first mode goes from 12.5 Hz to 16.8 Hz and the second mode goes from 56.3 to 117.8 Hz.



**Figure 2.9:** 2DOF sample problem PSD analysis. Dashed horizontal line is  $J_{z,req}$ .

A summary of the RMS performance results is shown in Table 2.2.

**Table 2.2:** Results overview for two DOF oscillator problem.

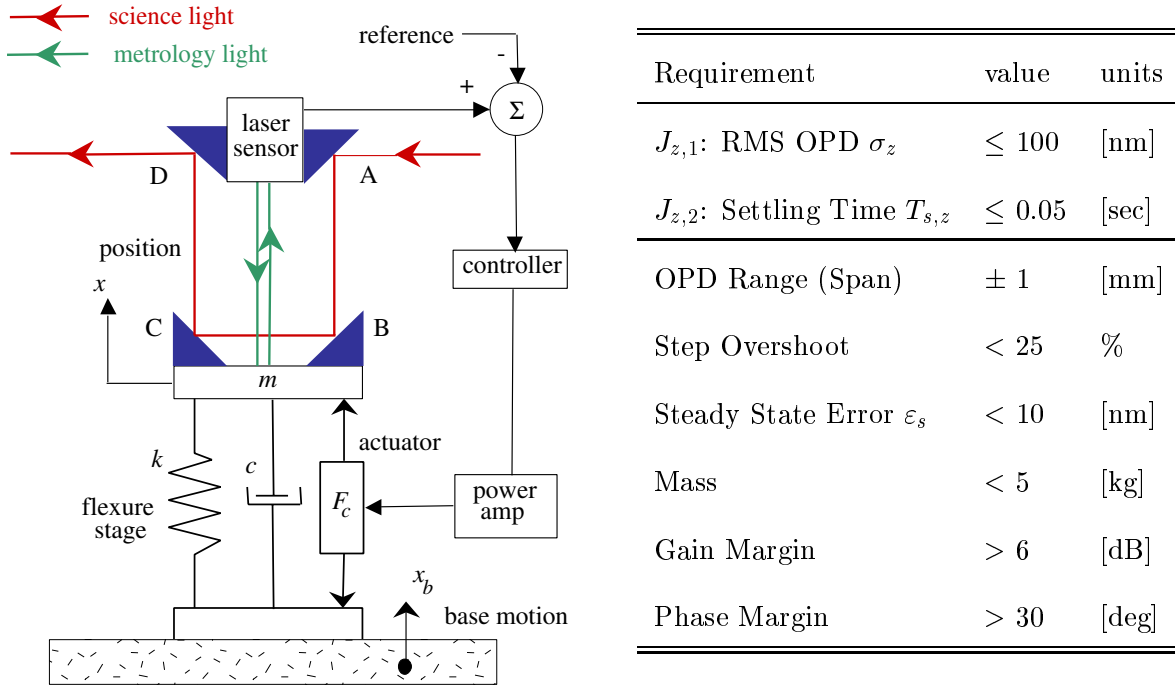
		Initial Design		Improved Design		Requirement
$J_z$	Method	$J_{z,o}$ [ $\mu\text{m}$ ]	$\pm\%$	$J_{z,nom}$ [ $\mu\text{m}$ ]	$\pm\%$	$J_{z,req}$ [ $\mu\text{m}$ ]
1	time	9.3674	18.04	0.80497	20.21	1.0
	psd	9.9431		0.80961		
	lyap	9.9442		0.80983		
2	time	11.797	12.74	6.2518	12.66	7.5
	psd	11.011		5.9939		
	lyap	11.017		5.998		

One could stop at this point and implement  $p_{nom}$ . It is, however, expected that the improved design  $p_{nom}$  is not unique and other designs might perform as well, but at a lower “cost”. This is explored in Chapter 4.

## 2.3 Optical Delay Line Design Problem

### 2.3.1 Problem Definition

The third sample problem presented in this thesis is the design of a single-stage optical delay line (ODL). A schematic of the ODL design problem considered is shown in Figure 2.10.



**Figure 2.10:** Left: Optical Delay Line (ODL) design concept. Right: Table containing performance  $J_z$  and other requirements for ODL design.

Optical delay lines are actively controlled devices which are used to precisely control the path-length of light. A typical application is as an actuator in a stellar Michelson interferometer. A detailed engineering prototype of a three stage ODL for interferometry was built and tested by the Jet Propulsion Laboratory. The delay line requirements (see Figure 2.10) levied on this sample problem are most similar to the intermediate (voice coil) stage. Delay line design was studied in detail by Park and Miller [116]. Some of the aspects of that study are included here such as stability and performance considerations. Others are neglected such as the use of multiple stages or the noise floor introduced by D/A-conversion actuator noise. The goal is to investigate a physically meaningful isoperformance sample problem, where instability (presence of a subset  $\mathbf{U}$ ) can be an issue.

The primary performance goal of the ODL is to control the pathlength of science light



Power  
Amp  
Actuator  
 $G_a(s)$

$U_f(s)$   
+  
 $U_c(s)$

$\longrightarrow$

$F_c(s)$

$\longrightarrow$

$$F_c = G_a(s) \cdot (U_f + U_c) = \frac{K_a \omega_a}{s + \omega_a} \cdot (U_f + U_c)$$

$$\dot{q}_a = \underbrace{[-\omega_a]}_{A_a} q_a + \underbrace{[1 \ 1]}_{B_a} \begin{bmatrix} u_f & u_c \end{bmatrix}^T$$

$$F_c = \underbrace{[K_a \omega_a]}_{C_a} q_a + \underbrace{[0 \ 0]}_{D_a} \begin{bmatrix} u_f & u_c \end{bmatrix}^T$$

(2.11)

The control force,  $F_c$  [N], produced by the actuator acts on the flexure stage and results in a displacement  $x$  [m]. The other input into the flexure stage dynamics is a base motion  $x_b(t)$  [m], which is considered a process noise. This disturbance is driven by random white noise of intensity  $S_{dd}$  [m/ $\sqrt{\text{Hz}}$ ] and corner frequency  $\omega_d$  [rad/sec]. The disturbance dynamics of the base motion are assumed as:

Base  
Motion  
 $G_d(s)$

$D(s)$

$\longrightarrow$

$X_b(s)$

$\longrightarrow$

$$X_b(s) = G_d(s) \cdot D = \frac{S_{dd} \omega_d}{s + \omega_d} \cdot D(s)$$

$$\dot{q}_d = \underbrace{[-\omega_d]}_{A_d} q_d + \underbrace{[1]}_{B_d} d(t)$$

$$x_b(t) = \underbrace{[S_{dd} \omega_d]}_{C_d} q_d + \underbrace{[0]}_{D_d} d(t)$$

(2.12)

The dynamics of the flexure stage are fully determined by the mass  $m$  [kg], the stiffness  $k$  [N/m] and the damping coefficient  $c$  [Ns/m]. The dynamics in transfer function form are as follows:

Plant  
 $G_p(s)$

$X_b(s)$

$\longrightarrow$

$X(s)$

$\longrightarrow$

$$X(s) = G_p(s) \cdot \begin{bmatrix} X_b(s) & F_c(s) \end{bmatrix}^T =$$

$$\begin{bmatrix} \frac{c}{m}s + \frac{k}{m} & \frac{1}{m} \\ s^2 + \frac{c}{m}s + \frac{k}{m} & s^2 + \frac{c}{m}s + \frac{k}{m} \end{bmatrix} \cdot \begin{bmatrix} X_b \\ F_c \end{bmatrix}$$

(2.13)

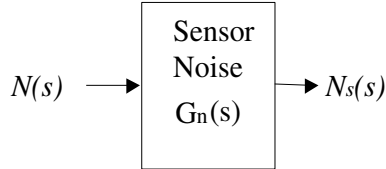
This can be written in state space form as:

$$\begin{aligned}
\dot{q}_p &= \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{k}{m} & -\frac{c}{m} & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\frac{k}{m} & -\frac{c}{m} \end{bmatrix}}_{A_p} q_p + \underbrace{\begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}}_{B_p} \begin{bmatrix} x_b \\ F_c \end{bmatrix} \\
x(t) &= \underbrace{\begin{bmatrix} k/m & c/m & 1/m & 0 \end{bmatrix}}_{C_p} q_p + \underbrace{\begin{bmatrix} 0 & 0 \end{bmatrix}}_{D_p} \begin{bmatrix} x_b \\ F_c \end{bmatrix}
\end{aligned} \tag{2.14}$$

The ODL motion  $x(t)$  causes an optical pathlength difference  $z(t) = K_o \cdot x(t)$ , where  $K_o$  is the linear optical sensitivity. In this case flat (reflective) mirrors are mounted on the flexure stage such that the pathlength is shortened by twice the amount of positive displacement of  $x$  as defined in Figure 2.10, hence  $K_o = -2$ . The equation for the optical pathlength in the ODL is:

$$\text{OPL} = \overline{\text{ABCD}} = L_o + \Delta L = L_o \underbrace{-2}_{K_o} x \tag{2.15}$$

The OPD  $z(t)$  is measured by a laser sensor with gain  $K_s$  [V/m]. This gain converts a measurement in meters to volts, thus  $y(t) = K_s \cdot z(t)$ . The sensor signal contains sensor noise, such that  $y_{OPD} = y + n_s$ . Thus the sensed performance  $J_{z,1}$  of the system is corrupted by noise. This disturbance is described as colored white noise according to:



$$\begin{aligned}
N_s(s) &= G_n(s) \cdot N(s) = \frac{S_{nn}\omega_n}{s + \omega_n} N(s) \\
\dot{q}_n &= \underbrace{[-\omega_n]}_{A_n} q_n + \underbrace{[1]}_{B_n} n(t) \\
n_s(t) &= \underbrace{[S_{nn}\omega_n]}_{C_n} q_n + \underbrace{[0]}_{D_n} n(t)
\end{aligned} \tag{2.16}$$

where  $S_{nn}$  [V/ $\sqrt{\text{Hz}}$ ] is the sensor noise intensity and  $\omega_n$  [rad/sec] is the sensor noise corner frequency. The sensor noise RMS voltage can be computed as  $\sigma_{n_s} = [(1/2)S_{nn}^2\omega_n]^{1/2}$ . The measurement,  $y_{OPD}$ , is then compared to the reference signal  $r$  multiplied by the sensor gain  $K_s$ . The error signal  $e = y_{OPD} - K_s r$  is used as the input to the controller. A PID controller was chosen for this application. The SISO controller dynamics are:

PID  
Control  
G<sub>c</sub>(s)

$E(s) \longrightarrow$

$\longrightarrow U_c(s)$

$$U_c = G_c(s) \cdot E = \frac{K_d \omega_c \left[ s^2 + \frac{K_p}{K_d} s + \frac{K_i}{K_d} \right]}{s(s + \omega_c)} E$$

$$\dot{q}_c = \underbrace{\begin{bmatrix} 0 & 1 \\ 0 & -\omega_c \end{bmatrix}}_{A_c} q_c + \underbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}_{B_c} e(t) \quad (2.17)$$

$$u_c(t) = \underbrace{\begin{bmatrix} K_i \omega_c & K_p \omega_c & -K_d \omega_c^2 \end{bmatrix}}_{C_c} q_c + \underbrace{\begin{bmatrix} K_d \omega_c \end{bmatrix}}_{D_c} e$$

where  $K_p$ ,  $K_i$  and  $K_d$  are the proportional, integral and derivative gain, respectively. The PID control has been augmented by a first order lowpass filter with corner frequency  $\omega_c$ , which allows writing the controller in proper form<sup>5</sup> and makes it physically realizable. The output of the controller is the signal  $u_c$ , which completes the loop.

The dynamics described above are shown in Figure 2.11 and are assembled into an appended state space realization (2.18) of the form shown in Equation (1.1), which will be referred to as the “integrated model” of the ODL.

$$\dot{q} = \begin{bmatrix} \dot{q}_d \\ \dot{q}_n \\ \dot{q}_p \\ \dot{q}_c \\ \dot{q}_a \end{bmatrix} = \underbrace{\begin{bmatrix} A_d & 0 & 0 & 0 & 0 \\ 0 & A_n & 0 & 0 & 0 \\ B_{p1}C_d & 0 & A_p & 0 & B_{p2}C_a \\ 0 & B_c C_n & B_c K_s K_o C_p & A_c & 0 \\ 0 & B_{a1} D_c C_n & B_{a1} D_c K_s K_o C_p & B_{a1} C_c & A_a \end{bmatrix}}_{A_{zd}} \begin{bmatrix} q_d \\ q_n \\ q_p \\ q_c \\ q_a \end{bmatrix} \quad (2.18)$$

$$+ \underbrace{\begin{bmatrix} B_d & 0 \\ 0 & B_n \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}}_{B_{zd}} \begin{bmatrix} d \\ n \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ -B_c K_s \\ B_{a2} K_f - B_{a1} D_c K_s \end{bmatrix}}_{B_{zr}} [r]$$

The performance output equation for  $z(t)$  is obtained as:

$$z(t) = \underbrace{\begin{bmatrix} 0 & 0 & K_o C_p & 0 & 0 \end{bmatrix}}_{C_{zd}} q + \underbrace{\begin{bmatrix} 0 & 0 \end{bmatrix}}_{D_{zd}} \begin{bmatrix} d \\ n \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \end{bmatrix}}_{D_{zr}} r \quad (2.19)$$

From this signal the RMS OPD is computed as:

<sup>5</sup>Proper form means that numerator polynomial degree is equal or smaller than the denominator degree.

$$J_{z,1} = \sigma_z = \text{RMS OPD} = \left[ \frac{1}{T} \int_0^T z^2(t) dt \right]^{1/2} \quad (2.20)$$

The OPD settling time (using a  $\pm 2\%$  threshold) is computed as:

$$J_{z,2} = T_{s,z} = t > 0 \cap \min(t) \text{ such that } \left| \frac{z(t) - r}{r} \right| < 0.02 \quad (2.21)$$

This allows computation of the performances,  $J_z$ , that are specified in the requirements of Figure 2.10. The next subsection will define and analyze an initial design for the optical delay line.

### 2.3.3 ODL Initial Design

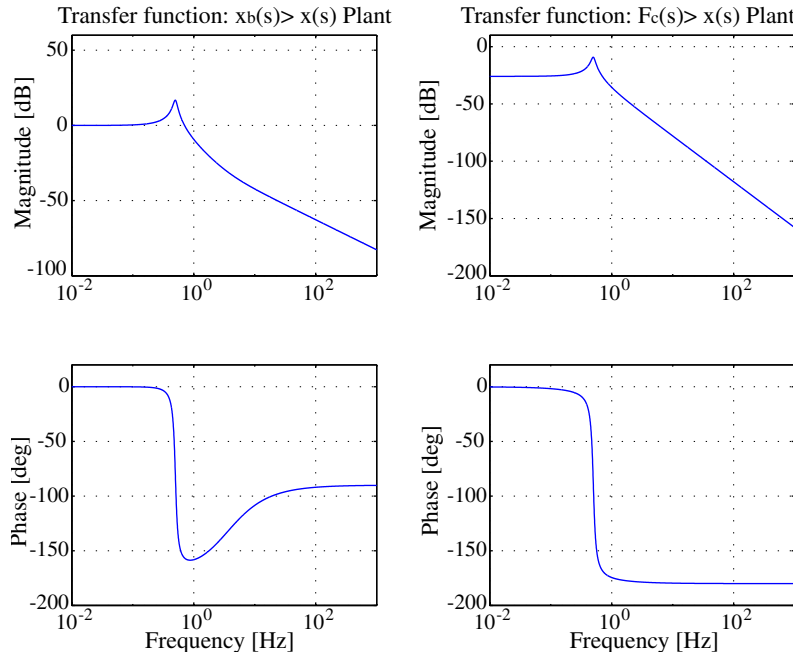
At this point initial values, bounds and uncertainties for the ODL design parameters,  $p_j$ , have to be specified. There is a distinction between *constant*, *dependent* and *variable* parameters as shown in Table 2.3.

**Table 2.3:** ODL design parameters. Explanations of the column headings are given in the paragraph below.

sys	$j$	nom	description	$p_{j,LB}$	$p_{j,o}$	$p_{j,UB}$	$v_j$	units	type
(D)	1	$S_{dd}$	base motion int.	$E-7$	$5.64E-7$	$2E-6$	5.0	$[\text{m}/\sqrt{Hz}]$	var
(D)	-	$\omega_d$	base motion cut.	-	$2\pi 100$	-	-	$[\text{rad}/\text{sec}]$	fix
(D)	2	$S_{nn}$	sensor noise int.	$1.5E-5$	$5.15E-5$	$E-4$	5.0	$[\text{V}/\sqrt{Hz}]$	var
(D)	-	$\omega_n$	sensor noise cut.	-	$2\pi 120$	-	-	$[\text{rad}/\text{sec}]$	fix
(P)	3	$m$	mass	1.0	2.0	5.0	0.1	$[\text{kg}]$	var
(P)	4	$k$	flexure stiffness	10	19.74	500	1.0	$[\text{N}/\text{m}]$	var
(P)	5	$c$	damping coeff.	0.1	0.9173	2.0	2.5	$[\text{Ns}/\text{m}]$	var
(O)	-	$K_o$	optical sensitivity	-	-2.0	-	-	$[\text{m}/\text{m}]$	fix
(C)	6	$K_d$	derivative gain	0.0	0.5	5	1.0	$[\text{V}/\text{V}]$	var
(C)	7	$\omega_c$	controller cutoff	$2\pi 100$	$2\pi 200$	$2\pi 500$	1.0	$[\text{rad}/\text{sec}]$	var
(C)	-	$K_f$	feedforward gain	-	$k/(K_o K_a)$	-	-	$[\text{V}/\text{m}]$	dep
(C)	-	$K_p$	proportional gain	-	$K_d(c/m)$	-	-	$[\text{V}/\text{V}]$	dep
(C)	-	$K_i$	integral gain	-	$K_d(k/m)$	-	-	$[\text{V}/\text{V}]$	dep
(S)	-	$K_s$	sensor gain	-	$10^4$	-	-	$[\text{V}/\text{m}]$	fix
(A)	-	$K_a$	amplifier gain	-	0.1	-	-	$[\text{N}/\text{V}]$	fix
(A)	8	$\omega_a$	actuator BW	$2\pi 50$	$2\pi 98$	$2\pi 500$	5.0	$[\text{rad}/\text{sec}]$	var

In Table 2.3 “sys” denotes the subsystem that the parameter belongs to: (D) disturbance sources, (P) structural plant, (O) optics, (C) controls, (S) sensors, (A) actuators; “j” is the number of the j-th variable parameter; “nom” is a nomenclature symbol; “description” is the parameter name; “ $p_{j,LB}$ ” is the lower bound; “ $p_{j,o}$ ” is the initial value; “ $p_{j,UB}$ ” is the upper bound and “ $v_j$ ” is a parameter uncertainty in  $\pm\%$ . The column “units” shows the SI-units of the parameter in model and “type” designates if the parameter is constant (fix), dependent (dep) or independently variable (var).

Constant parameters such as  $\omega_d$ ,  $\omega_n$  and  $K_o$  can typically not be changed and are dictated by constraints such as environmental factors, physical limitations of the electronics or the specific optical design chosen. Dependent parameters are not constant, since they change as a function of the truly independent (variable) design parameters of the system. An example of depended parameters are the feedforward gain  $K_f$  or the control gains  $K_p$  and  $K_i$ . Once these values are substituted in Equations (2.11-2.18) the dynamics of the system can be evaluated.



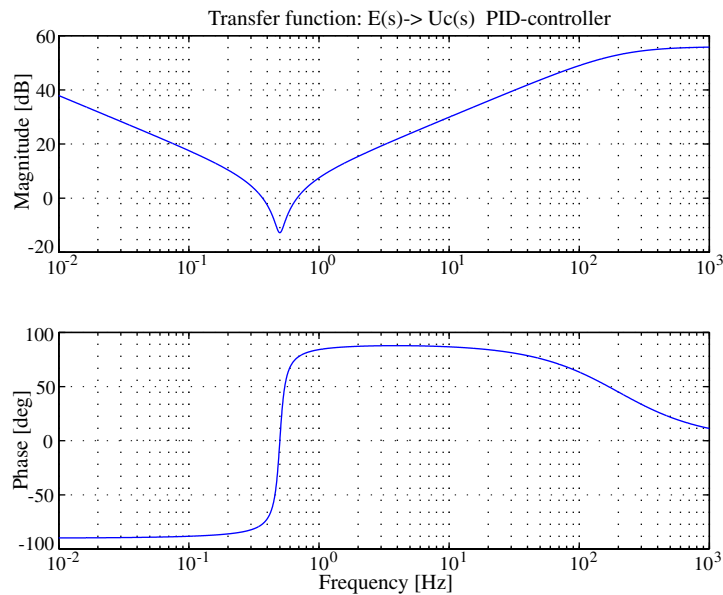
**Figure 2.12:** ODL plant transfer functions:  $G_{p1} = X(s)/X_b(s)$  (left) and  $G_{p2} = X(s)/F_c(s)$  (right). Suspension mode at 0.5 Hz and  $\zeta = 0.073$  as experimentally determined by Park [116].

The plant transfer functions from  $X_b(s)$  to  $X(s)$  and  $F_c(s)$  to  $X(s)$  are shown in Figure 2.12. The fundamental “suspension” mode is at 0.5 Hz as recommended by Park [116].



The DC gain for the  $X/F_c$  transfer is  $1/k$ , which corresponds to Hooke's law. There is a phase loss of -180 degrees at frequencies above the suspension mode, which will influence the stability of the loop gain function.

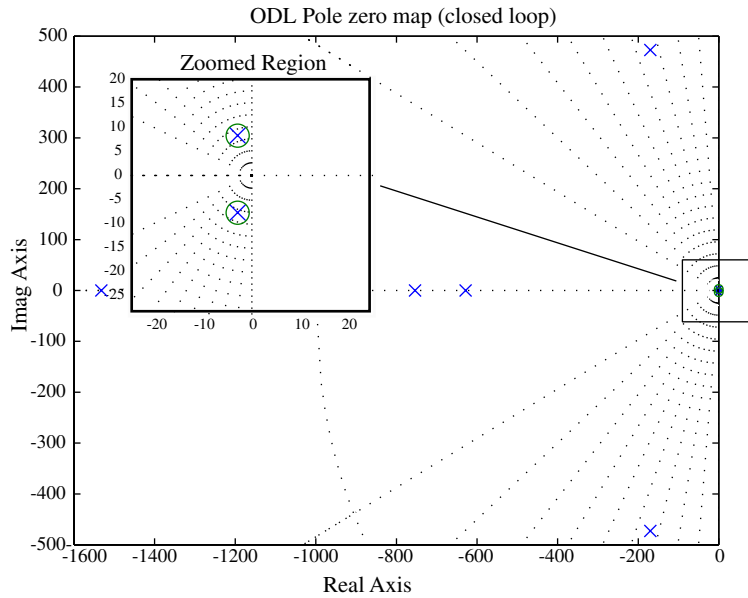
The PID control strategy is to set  $\omega_c$  to a frequency, which is limited by the capability of the electronics to provide gain at high frequency. It is attempted to achieve a pole-zero cancellation of one of the complex conjugate pole pairs given by the structural plant. From (2.17) one obtains the relationships  $K_p/K_d = c/m$  and  $K_i/K_d = k/m$ , which leaves the derivative gain  $K_d$  as the only variable control parameter. The PID controller transfer function is expected to look similar to a notch filter. This is confirmed by Figure 2.13.



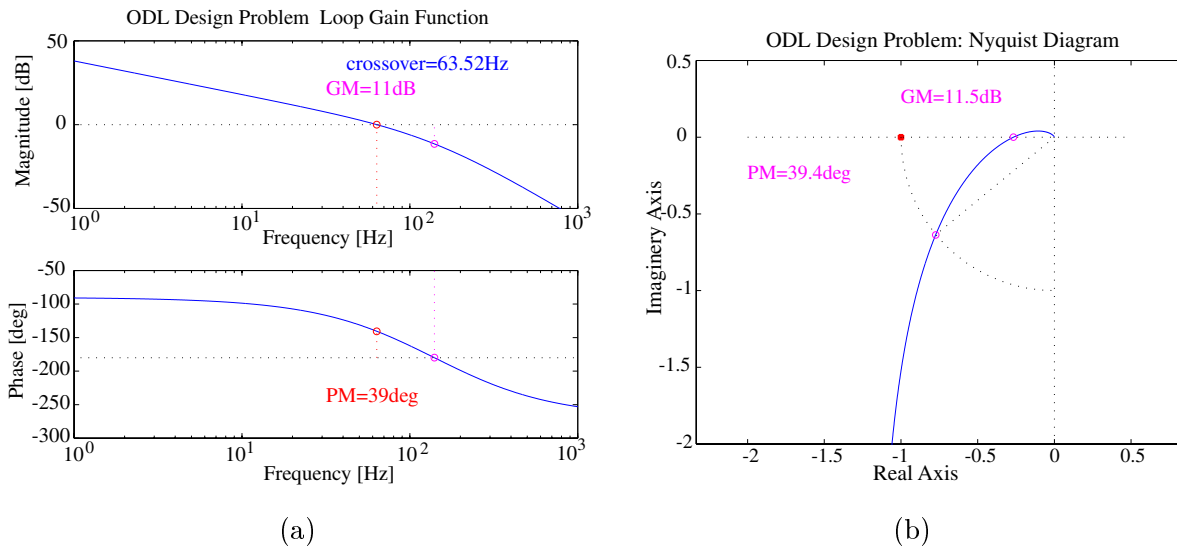
**Figure 2.13:** ODL PID controller transfer function:  $G_c = U_c(s)/E(s)$ .

### 2.3.4 Stability Margins

Absolute and relative stability must be assessed before proceeding. Absolute (asymptotic) stability is met, since all closed loop poles of the system lie in the left half of the s-plane as shown in Figure 2.14. The relative stability is examined via the loop gain function, see Figure 2.15. The crossover (0 [dB]) occurs at 63.5 Hz. The phase margin is 39.4 [deg] (requirement:  $PM > 30$  [deg]), the gain margin frequency is 140.0819 [Hz] and the gain margin is 11.5 [dB] (requirement:  $GM > 6$  [dB]). These margins are confirmed by the Nyquist diagram in Figure 2.15(b) and meet the requirements laid out in Figure 2.10.



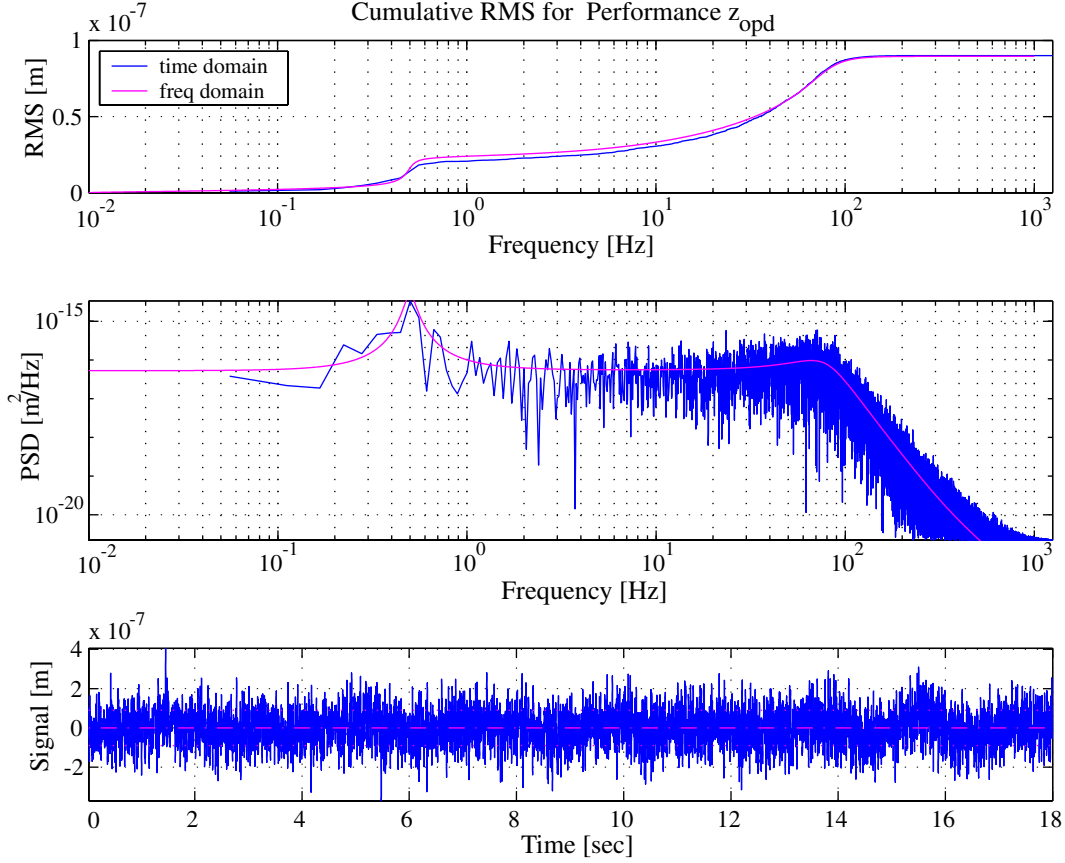
**Figure 2.14:** Pole-Zero map for closed loop ODL system. The zoomed region shows the pole-zero cancellation of the plant complex-conjugate pole pair.



**Figure 2.15:** (a) Loop Gain function for ODL, (b) Nyquist diagram for ODL.

### 2.3.5 Disturbance, Sensitivity and Uncertainty Analyses

The performance  $J_{z,1}$  = RMS OPD of the system is computed using the methods presented in Appendix A. An analysis of  $z(t)$  as well as its power spectral density function  $S_{zz}(\omega)$  and cumulative RMS curves are shown in Figure 2.16.



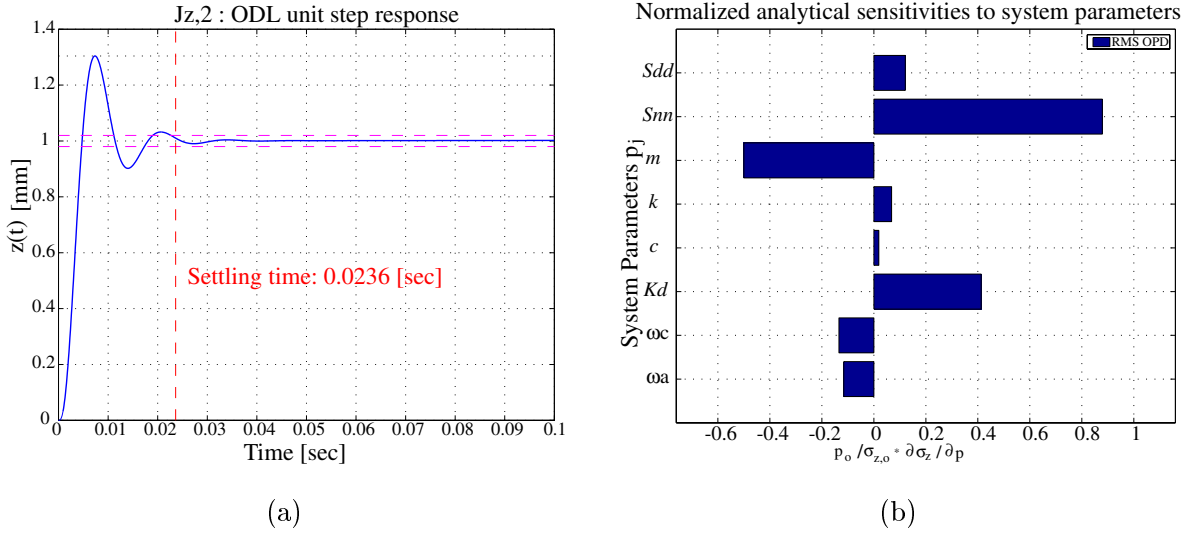
**Figure 2.16:** ODL simulation results: Cumulative RMS function for  $J_{z,1}$  (top), Power spectral density function (PSD)  $S_{zz}(\omega)$  (middle), Sample time realization for  $z(t)$  with  $T=18$  [sec] (bottom).

Time simulation results in a performance of  $\sigma_z = 90.482$  [nm]. This is close to the performances computed via the PSD method ( $\sigma_z = 89.581$  [nm]) and the Lyapunov approach ( $\sigma_z = 89.584$  [nm]). Since  $J_{z,1} < J_{z,req,1}$ , i.e the RMS OPD is smaller than 100 [nm], the initial system meets the steady state performance requirement. It is interesting to note that the contributions to  $\sigma_z$  come from the suspension mode at 0.5 Hz and, somewhat unexpectedly, from higher frequency contributions up to roughly 100 Hz. The reason for this will become clear in the ODL isoperformance analysis of Chapter 3.

Another observation is that the “sensed” performance RMS,  $y_{OPD}$ , has a value of 122 [nm] and is significantly different from the actual performance RMS value of 90 [nm]. By consulting the block diagram in Figure 2.11, we see that the difference between the performance,  $z$ , and the sensed performance,  $y_{opd}$ , can only be due to sensor noise,  $n_s$ . The disturbance signals for this analysis were generated with the parameters  $p_o$  from Table 2.3 and resulted in a base motion with RMS  $x_b = 10$  [ $\mu\text{m}$ ] and a sensor noise with a 1 [mV] RMS voltage.

The step performance of the system,  $J_{z,2} = T_{s,z}$  was computed using the MATLAB `step.m` command. The result is shown in Figure 2.17(a) and indicates that the ODL settles from the initial position at 0 [mm] to a new position at  $r = +1$  [mm] in 0.0236 [sec], assuming a 2% threshold. This suggests that the transient performance  $J_{z,2}$  also meets requirements, since  $T_{s,z} < 0.05$  [sec]. The overshoot, however, is 30.4% and somewhat larger than the maximum of 25% stipulated earlier.

A sensitivity analysis for the design was conducted with respect to the first performance  $J_{z,1}$ . The results are shown in Figure 2.17(b).



**Figure 2.17:** (a) Step response of ODL for  $r=1.0$  [mm]; (b) Sensitivities of RMS OPD  $\sigma_z$  w.r.t. parameters  $p_j$  (normalized).

The sensitivities with respect to the noise intensities  $S_{dd}$  and  $S_{nn}$  are positive. The large positive sensitivity of  $S_{nn}$  suggests that excessive sensor noise is present in the control signal or that the control gain,  $K_d$ , is too large. This is likely, since the sensitivity with respect to the control gain,  $K_d$ , is positive. Thus, increasing gain will increase the RMS OPD. The issue of balancing sensor noise versus process noise will be explored in Chapter 3.

The sensitivity analysis allows computing an uncertainty estimate on the performance via equation (2.7). The estimated uncertainty on the RMS OPD is  $89.6 \pm 5.64$  [nm], which corresponds to  $\pm 6.3$  %. Even in the worst case the performance  $J_{z,1}$  is expected to remain within the requirement,  $J_{z,req,1} \leq 100$  [nm].

### 2.3.6 ODL conclusions

At this point a satisfactory design has been achieved since the two performance requirements: RMS OPD  $J_{z,1} = 89.6$  [nm] and settling time  $J_{z,2} = 0.0236$  [sec] have been met. The other requirements such as mass  $m \leq 2$  [kg] and zero steady state error  $\varepsilon_s \leq 10$  [nm] are also met. The only exception is the step overshoot of 30.4%, which slightly exceeds the requirement of 25%. Note that the span (OPD range) requirement cannot be verified with this simplified linear model. In contrast to the oscillator problems,  $p_{nom} = p_o$ , will be accepted as the nominal design without further performance enhancement. Note that the RMS control force,  $F_c$ , to achieve this performance (with  $r=0$ ) was 0.02 [N]. A search for ODL design vectors that might also meet the performance requirements, albeit at different “cost” and “risk” will be discussed in subsequent chapters.

## 2.4 Summary

This chapter introduced three sample problems. These are the single degree-of-freedom (SDOF) oscillator, the two degree-of-freedom (2DOF) oscillator and the optical delay line (ODL) design problem. The motivation of the sample problems is to provide specific examples for illustrating isoperformance related concepts in subsequent chapters, without cluttering the discussion. For each of the problems a disturbance, sensitivity and uncertainty analysis is conducted. If necessary a performance enhancement (line or gradient search) algorithm was applied such that a satisfactory design,  $p_{nom}$ , was obtained starting from an initial guess,  $p_o$ . The nominal designs,  $p_{nom}$ , are not unique. For the SDOF problem for example it is expected that the same performance could be achieved by simultaneously increasing the mass,  $m$ , and decreasing the disturbance isolation corner frequency,  $\omega_d$ . This will be explored in Chapter 3 for bivariate problems and in Chapter 4 for multivariable problems.



## Chapter 3

# Bivariate Isoperformance

## Methodology

This chapter solves the bivariate isoperformance problem for two independent variable parameters  $p_j$ , where  $j = 1, 2$ , and one (scalar) performance objective  $p_j \mapsto J_z(p_j)$ . Three alternative algorithms (exhaustive search, gradient-based contour following and progressive spline approximation) are developed, implemented and evaluated. The solutions,  $p_{iso}$ , form the isoperformance set,  $\mathbf{I}$ . This set meets the isoperformance condition  $J_z(p_{iso}) = J_{z,req}$ . Applications of the bivariate isoperformance methodology are presented for the sample problems introduced in Chapter 2 and some realistic spacecraft dynamics examples. The algorithms have been implemented in a MATLAB module function called `isoperf_2var.m`.

### 3.1 Bivariate Isoperformance Problem

Once a viable initial design,  $p_{nom}$ , has been found, it is important to determine the combinations of values for the independent parameters  $p_j$ ,  $j = 1, 2, \dots, n_p$ , of the system, that will result in the same (predicted) dynamical performance  $J_z(p_j) = J_{z,req}$ . A first step towards solving the problem stated in Section 1.3 is undertaken by considering two variable parameters,  $n_p = 2$  and one (scalar) performance metric  $J_z$ , where  $n_z = 1$ . This gives the impetus for the following task formulation:

**Problem:** Find algorithms, which are able to efficiently compute the isocontours  $p_j \mapsto J_z(p_j) = J_{z,req}$  for a dynamic LTI system. Develop these algorithms for two system parameters  $p = [p_1 \ p_2]^T$  and one (scalar) performance,  $J_z(p)$ , first. The methodology must be generalizable to  $n_z > 1$  performance metrics and  $n_p > 2$  parameters.

The extension to the full multivariable case is presented in Chapter 4. Formally, an appended state space representation of the dynamics of a closed-loop or open-loop linear time-invariant system is given as

$$\begin{aligned} \dot{q} &= A_{zd}(p_j)q + B_{zd}(p_j)d + B_{zr}(p_j)r \\ z &= C_{zd}(p_j)q + D_{zd}(p_j)d + D_{zr}(p_j)r, \text{ where } j = 1, 2 \end{aligned} \quad (3.1)$$

where  $A_{zd}$  is the state transition matrix,  $B_{zd}$  and  $B_{zr}$  are the disturbance and reference input coefficient matrices,  $C_{zd}$  is the performance output coefficient matrix,  $D_{zd}$  and  $D_{zr}$  are the disturbance and reference feedthrough matrices,  $d$  contains unit-intensity white noise inputs,  $r$  represents reference inputs,  $z$  contains system performance outputs,  $q$  is the state vector and  $p_j$  is the  $j$ -th variable, independent system parameter. Additionally the functional

$$J_z(p_j) = F(z) \quad (3.2)$$

represents a definition of the (scalar) performance metric of interest. See Appendix A for a list of typical performance specifications such as RMS or RSS. Regardless of which metric is used, the resulting  $J_z$ , will always be a scalar quantity.

We want to find solution vectors,  $p_{iso}$ , such that the performance equality constraint

$$J_z(p_{iso}) \equiv J_{z,req} \quad (3.3)$$

is met, assuming that the parameters  $p_j$  are bounded below and above as

$$p_{j,LB} \leq p_j \leq p_{j,UB} \quad \forall j = 1, 2 \quad (3.4)$$

The isoperformance condition (3.3) has to be met subject to a numerical tolerance,  $\tau$

$$\left| \frac{J_z(p_{iso}) - J_{z,req}}{J_{z,req}} \right| \leq \frac{\tau}{100} \quad (3.5)$$

In the case of spline approximation a functional approximation to the isoperformance contours,  $t \mapsto f_{iso}(t) = p_{iso}$ , where  $\mathbb{R} \mapsto \mathbb{R}^2$ , may be obtained. Note that in this case the parameter  $t$  is the analog of the parameter  $t$  in a parametric curve description in  $\mathbb{R}^2$  or  $\mathbb{R}^3$ . This



would be describing the time  $t$  in the curved path of a particle in three dimensional Euclidean space. For example the position vector  $\vec{r}(t) = [x(t), y(t), z(t)]$  of a helical trajectory is  $t \mapsto [\cos(t), \sin(t), t]$ . In the isoperformance context the coordinates  $x, y, z$  correspond to the coordinates  $p_1, p_2, \dots, p_{n_p}$  into the parameter space  $\mathbf{B}$  in  $\mathbb{R}^{n_p}$ .

## 3.2 Algorithm Development

The purpose of this section is to work out the fundamental mathematics and algorithms for determining the isoperformance contours for the case of a single performance metric,  $J_z$ , and two parameters  $p_j$ ,  $j = 1, 2$ . The performance is a function of an opto-mechanical output quantity,  $z$ , of the system. The parameters are structural, optical, disturbance or control parameters of the system (e.g. beam moment of inertia, detector quantum efficiency, reaction wheel static imbalance, derivative control gain). Note that physical parameters are typically used for an isoperformance analysis, as opposed to modal parameters, since they relate more directly to engineering design quantities of interest and are more readily and independently alterable by the designer. On a conceptual level there are three fundamentally different algorithms, which will be considered:

1. Exhaustive Search
2. Gradient-based Contour Following
3. Progressive Spline Approximation

There is no claim that this set is complete and that no other viable algorithms exist. Note that a fourth approach, using Newton's method for finding the roots of a multivariable function numerically, is conceivable if the performance objective  $J_z(p_j)$  is known in closed form. This, however, is only true for relatively simplistic low-order problems, such that the requirement: "generalizable to the multivariable case" is not met. The next section quantitatively evaluates and compares these three approaches by solving the single degree-of-freedom (SDOF) oscillator problem introduced in Section 2.1.

Before developing each algorithm, it shall be noted that three quantities must be specified *a priori*, regardless of the algorithm used. First, upper and lower bounds for each variable system parameter  $p_j$ , where  $j = 1, 2$ , must be defined such that:

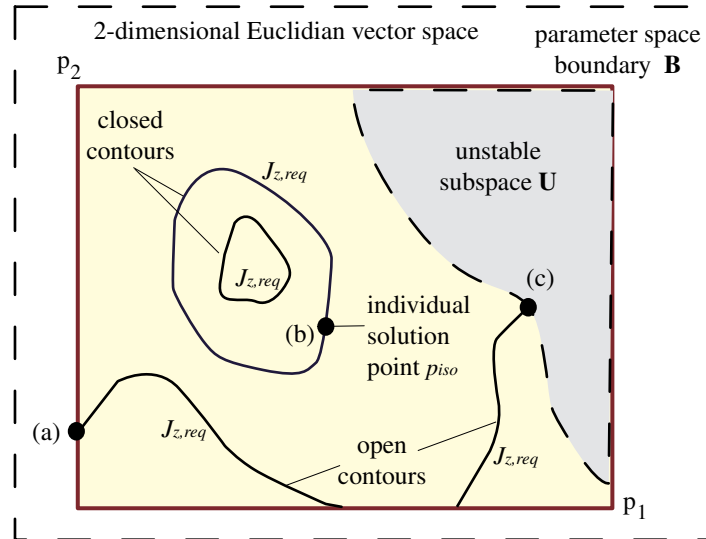
$$p_{j,LB} \leq p_j \leq p_{j,UB} \tag{3.6}$$

Here the subscripts “LB”, “UB” identify a lower bound and upper bound for the  $j$ -th parameter, respectively. In practice these bounds are chosen not to violate physics (e.g. negative mass) and are based on empirical experience regarding the range which is physically realizable within the current or foreseeable state-of-the-art for a given parameter. For example a realistic lower bound for the noise voltage RMS produced by off-the-shelf 12-bit electronics is 4 [mV].

Furthermore an allowable tolerance,  $\tau$ , on the isoperformance (contour) solution must be specified, since a digital computer with a limited floating-point accuracy is used to determine individual solution points,  $p_{iso}$ . If this tolerance,  $\tau$ , is expressed as a permissible  $\pm$  percent deviation from the required performance,  $J_{z,req}$ , one expects the  $k$ -th solution  $p_{iso,k} \mapsto J_{z,k}$  to be “on” the isoperformance contour such that

$$J_{z,req} \cdot [1 - \tau/100] \leq J_{z,k} \leq J_{z,req} \cdot [1 + \tau/100] \quad (3.7)$$

Finally, the required performance,  $J_{z,req}$ , which corresponds to the “level” or “height” of the isoperformance contours, needs to be defined. This is analogous to looking at the pressure chart in Figure 1.3(b) and asking for the locus of the 1006 [mb] isobars, as an example. It is possible that no point in the parameter space  $\mathbb{R}^2$  bounded by (3.6) will yield  $J_{z,req}$ , i.e. the isoperformance set is empty  $\mathbf{I}=\emptyset$ , or that there are several closed or open contours as shown in Figure 3.1. These contours are sometimes referred to as *level curves* [28].



**Figure 3.1:** Theoretical instances of isoperformance contours  $p_j \mapsto J_z(p_j) = J_{z,req}$  in  $\mathbb{R}^2$ . (a) regular boundary point on  $\mathbf{B}$ , (b) interior solution point  $p_{iso}$  in set  $\mathbf{I}$  and (c) boundary point on the unstable subset  $\mathbf{U}$ .

It is also true that isoperformance contours will never cross, since a particular design point  $p_j$ ,  $j = 1, 2$ , will always map to a unique performance value  $p_j \mapsto J_z(p_j)$ , where  $j = 1, 2$ . The multivariable function  $p_j \mapsto J_z(p_j)$  is said to be unidirectionally unique. Note that the isoperformance algorithms should be as general as possible to efficiently compute all of the cases presented in Figure 3.1. Specifically, there can be multiple instances of open and closed contours. Closed contours, by definition, close on themselves (non-zero curl in vector calculus) and can be circumscribing other closed contours. Closed contours can only be tangent to the parameter space boundary  $\mathbf{B}$ , but never intersect it. Open contours always have a start and end point on a parameter space boundary of  $\mathbf{B}$  or at the absolute stability boundary of the unstable subset  $\mathbf{U}$ . The next three subsections develop the bivariate isoperformance algorithms, which solve the problem posed above.

### 3.2.1 Algorithm I: Exhaustive Search

This method discretizes the parameter space, defined by the lower and upper bounds  $p_{j,LB}, p_{j,UB}$ , where  $j = 1, 2$ , by overlaying a fine grid and completely evaluating all grid points. The subdivisions of the grid are defined by means of uniform parameter increments  $\Delta p_1, \Delta p_2$ . The size of the increments should be small enough to capture details of the isoperformance contours. This is related to the smoothness of  $J_z(p_j)$ . Small increments are desirable as this will allow to capture a large number of points,  $p_{iso}$ , on the isoperformance contours. On the other hand the computational expense grows significantly with smaller increments.

Each grid point on the grid represents a unique parameter combination  $p_{k,l} = [p_{1,k} \ p_{2,l}]^T$ . The parameter values are obtained from  $p_{1,k} = p_{1,LB} + (k-1)\Delta p_1$  and  $p_{2,l} = p_{2,LB} + (l-1)\Delta p_2$ , respectively, which leads to a linearly spaced grid. The performance  $(J_z)_{k,l} = J_z(p_{k,l})$  is evaluated for all parameter combinations (complete enumeration). The number of increments in each parameter axis is obtained as<sup>1</sup>:

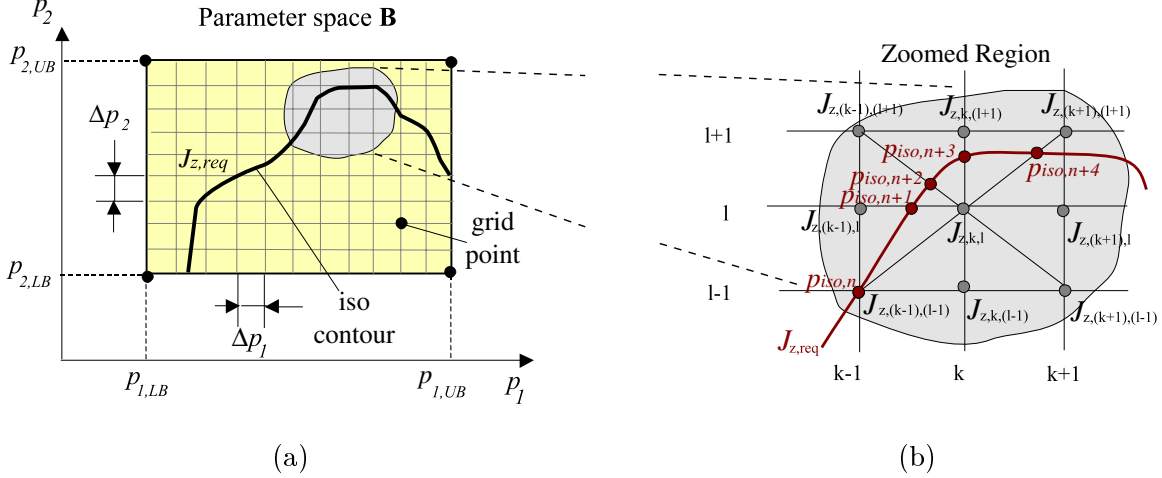
$$n_1 = \left\lceil \frac{p_{1,UB} - p_{1,LB}}{\Delta p_1} \right\rceil \quad \text{and} \quad n_2 = \left\lceil \frac{p_{2,UB} - p_{2,LB}}{\Delta p_2} \right\rceil \quad (3.8)$$

The index  $k$  on the first parameter runs from 1 to  $n_1+1$ , the index  $l$  runs from 1 to  $n_2+1$ <sup>2</sup>. Thus a total number of  $(n_1+1) \times (n_2+1)$  combinations has to be evaluated. This is algorithmically

<sup>1</sup>The  $\lceil \cdot \rceil$  operator denotes the ceiling function.

<sup>2</sup>If  $k = n_1 + 1$  then  $p_{1,k} = p_{1,UB}$  and if  $l = n_2 + 1$  then  $p_{2,l} = p_{2,UB}$ .

achieved by means of two nested for loops. The resulting performances  $(J_z)_{k,l}$  are stored in a  $(n_1 + 1) \times (n_2 + 1)$  matrix. A representation of the parameter space  $\mathbf{B}$  discretization is shown in Figure 3.2(a) along with a zoomed region in Figure 3.2(b).



**Figure 3.2:** (a) Algorithm I: Discretization of  $\mathbf{B}$  in a linearly spaced grid with increments  $\Delta p = [\Delta p_1, \Delta p_2]^T$ . (b) Linear interpolation between neighboring grid points. The gray shaded area is a zoomed region.

Note that the result of a particular parameter combination,  $p_{k,l}$ , does not affect the computation of the next point. The algorithm will be shown to be computationally expensive, particularly for more than two parameters and fine discretization, see Chapter 4. Once all the parameter combinations  $p_{k,l}$  have been evaluated, linear interpolation between neighboring grid points, as shown in Figure 3.2(b), is used to find isoperformance points  $p_{iso,r}$ . The linear interpolation algorithm uses the following equation to find the  $r$ -th isoperformance point:

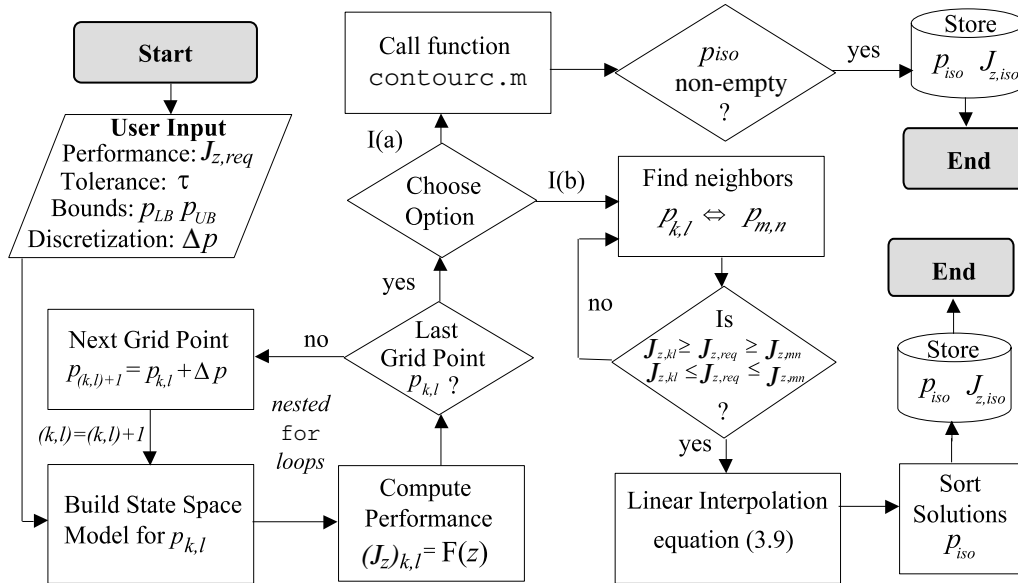
$$p_{iso,r} = \begin{bmatrix} p_{1,iso,r} \\ p_{2,iso,r} \end{bmatrix} = \begin{bmatrix} p_{1,k} \\ p_{2,l} \end{bmatrix} + \frac{(J_z)_{k,l} - J_{z,req}}{(J_z)_{k,l} - (J_z)_{m,n}} \cdot \begin{bmatrix} p_{1,m} - p_{1,k} \\ p_{2,n} - p_{2,l} \end{bmatrix} \quad (3.9)$$

The above equation is invoked if it is found that either  $(J_z)_{k,l} \geq J_{z,req} \geq (J_z)_{m,n}$  or  $(J_z)_{k,l} \leq J_{z,req} \leq (J_z)_{m,n}$ , assuming continuity of  $J_z(p)$ . This requires that the predicted performance at each grid point  $(J_z)_{k,l}$  is compared to the performance of each neighboring grid point  $(J_z)_{m,n}$ . Note that  $(J_z)_{m,n}$  is the performance at a neighboring point such that  $m \in [k - 1, k, k + 1]$  and  $n \in [l - 1, l, l + 1]$ . The point  $m = k, n = l$  is not tested, since it represents the center grid point  $p_{k,l}$  itself as shown in Figure 3.2(b).

This procedure is similar to the process that is used to obtain the pressure chart shown in

Figure 1.3(b). Consider the following example: We are interested in finding the 1006 [mbar] isobar. We know that the atmospheric pressure is 1003.3 [mbar] in Boston and 1007.2 [mbar] in New York City at 5:00 p.m. on a given day. Since the distance between Boston and New York is 216 miles, we expect the point, which is at a distance of 66.5 miles from New York on the straight-line between the two cities to be on or close to the 1006 mbar isobar. This assumes that the pressure varies linearly and that no other reporting station exists between the two cities.

An alternate option replaces the linear interpolation step with a call to the MATLAB built-in function `contourc.m` for contouring. This allows displaying a family of several performance levels at once. The flow diagram used to implement Algorithm I is shown in Figure 3.3.



**Figure 3.3:** Flow Diagram for Algorithm I: Exhaustive Search

The algorithm first requires the user to input a performance requirement  $J_{z,req}$ , a tolerance  $\tau$ , the upper and lower bounds  $p_{LB}$  and  $p_{UB}$ , as well as the desired discretization  $\Delta p$ . Next it loops over all points in the parameter space  $\mathbf{B}$  and builds a state space model according to (1.1) for each combination  $p_{k,l}$ . The corresponding performance  $(J_z)_{k,l}$  is evaluated for each combination. After evaluating all grid points (complete enumeration) the algorithm branches into two separate options. Option I(a) calls the built-in MATLAB function `contourc.m` for contouring. If the matrix of isoperformance points,  $p_{iso}$ , is non-zero, it is stored and the

algorithm ends. Option I(b) searches for all closest neighboring grid points and checks whether the isoperformance contour passes in between them. If it does, an intermediate point,  $p_{iso,r}$ , is obtained using linear interpolation according to (3.9). The resulting points have to be sorted with a “closest neighbor” algorithm, which is accomplished by the function `find_segments.m`.

In summary, the exhaustive search algorithm attempts to find contours of equal performance by laying a fine grid over the trade space  $\mathbf{B}$  in  $\mathbb{R}^2$  and evaluating all grid points. It obtains an estimate of the isoperformance contours via linear interpolation.

### 3.2.2 Algorithm II: Gradient-based Contour Following

The exhaustive search algorithm does not exploit knowledge about previous points  $p_{iso,k} = [p_{iso,1,k}, p_{iso,2,k}]^T$  found on an isoperformance contour in any way. The basic idea of the gradient-based contour following scheme is to first find an “isopoint”  $p_{iso,1}$ , which is known to yield the required performance  $J_{z,req}$ , i.e. it lies on an isoperformance contour. Once such a point is found a neighboring point  $p_{iso,k+1}$  on the same isoperformance contour is found by means of a direction derived from the gradient vector,  $\nabla J_z(p_1, p_2)$ . Thus, a prerequisite is that  $J_z(p_j)$  is continuous and differentiable at all points in the parameter space  $p = [p_1, p_2]^T \in \mathbf{B}$ . It will be shown that this step direction is colinear with the tangent vector  $t_k$  to the isoperformance contour. This step-wise “contour following” is repeated until the isoperformance contour intersects the parameter space boundary  $\mathbf{B}$ , the absolute stability boundary  $\mathbf{U}$  or it closes on itself.

If several segments (i.e. separate contours as in Figure 3.1) are present, different “random” initial trial points are necessary in order to detect all segments. This algorithm promises to be significantly more efficient than exhaustive search, since it only computes points on the contours and takes advantage of previously computed solutions. The derivation starts by considering the bivariate function

$$p_1, p_2 \mapsto J_z(p_1, p_2), \text{ where } \mathbb{R}^2 \mapsto \mathbb{R} \text{ and } p_j \in \mathbf{B} \quad (3.10)$$

The vector of parameters can be stacked as follows:

$$p = \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} \text{ and } \Delta p = \begin{bmatrix} \Delta p_1 \\ \Delta p_2 \end{bmatrix} \quad (3.11)$$

Next a Taylor series expansion of the vector function  $J_z(p)$  is performed around a nominal

point,  $p_{nom}$ , where  $p_{nom} \in \mathbf{B}$ , as follows:

$$J_z(p) = J_z(p_{nom}) + (\nabla J_z)^T \Big|_{p_{nom}} \cdot \Delta p + \frac{1}{2} \Delta p^T H \Big|_{p_{nom}} \Delta p + \text{H.O.T.} \quad (3.12)$$

Note that  $p = p_{nom} + \Delta p$  and that  $\nabla J_z$  and  $H$  are the gradient vector and Hessian matrix, respectively.

$$\nabla J_z = \begin{bmatrix} \frac{\partial J_z}{\partial p_1} \\ \frac{\partial J_z}{\partial p_2} \end{bmatrix} \quad \text{and} \quad H = \begin{bmatrix} \frac{\partial^2 J_z}{\partial p_1^2} & \frac{\partial^2 J_z}{\partial p_1 \partial p_2} \\ \frac{\partial^2 J_z}{\partial p_2 \partial p_1} & \frac{\partial^2 J_z}{\partial p_2^2} \end{bmatrix} \quad (3.13)$$

From multivariable calculus [28] we know that the cross-derivatives are equal to each other and therefore the Hessian is symmetric:  $H = H^T$ . The parameter vector increment  $\Delta p$  can be written as the product of a step size  $\alpha$  and a step direction (vector)  $d$ . Note that  $d$  is normalized to unit length

$$\Delta p = \alpha \cdot d \quad (3.14)$$

The starting point of Algorithm II is an initial guess  $p_o = [p_{1,o}, p_{2,o}]^T$ , which is in the ‘‘vicinity’’ of, but not necessarily exactly on the isoperformance contour. A steepest descent algorithm [33] is used to obtain a first isopoint,  $p_{iso,1}$ , on the isoperformance contour. This was demonstrated for the oscillator sample problems in sections 2.1 and 2.2.

A direction  $d$  of  $J_z(p_1, p_2)$ , where  $\mathbb{R}^2 \mapsto \mathbb{R}$  at  $p = p_o$  is a descent direction if

$$J_z(p_o + \alpha \cdot d) < J_z(p_o) \quad (3.15)$$

for all sufficiently small positive values of  $\alpha$ . The step size  $\alpha$  is a scalar value and is chosen to be positive if the initial guess  $p_o$  lies ‘‘above’’ the isoperformance contour (e.g. yields a larger RMS value). Conversely if the initial guess  $p_o$  or any subsequent iterate is ‘‘below’’ the isoperformance level,  $\alpha$  will be a negative scalar. The next iterate is then obtained as  $p_{o+1} = p_o + \alpha_o \cdot n_o$ , where  $n_o$  is the unit-length vector of steepest descent<sup>3</sup>. Thus, one can write the first order approximation at the point  $p_o$  as:

$$J_z(p_o + \alpha_o \cdot n_o) \cong J_z(p_o) + \nabla J_z(p_o)^T \cdot \alpha_o n_o \quad (3.16)$$

Recall from the Cauchy-Schwartz inequality that

$$J_z(p_o) + \nabla J_z(p_o)^T \left( \frac{-\nabla J_z(p_o)}{\|\nabla J_z(p_o)\|} \right) \leq J_z(p_o) + \nabla J_z(p_o)^T \left( \frac{d}{\|d\|} \right) \quad (3.17)$$

---

<sup>3</sup>Note: ‘‘n’’ is chosen to indicate the ‘‘normal’’ vector and ‘‘t’’ is chosen to designate the ‘‘tangential’’ vector. This is not the same ‘‘t’’ used to parameterize isoperformance contours presented in Subsection 3.2.3

for any  $d \neq 0$ . Thus, the steepest descent vector (step direction) at  $p_o$  is obtained as

$$n_o = \left( \frac{-\nabla J_z(p_o)}{\|\nabla J_z(p_o)\|} \right) \quad (3.18)$$

The step size,  $\alpha$ , is found by assuming linearity from the initial guess,  $p_o$ , to the first point on the isoperformance contour  $p_{iso,1}$ . Traditionally a line search is performed on a function  $h(\alpha)$  for steepest descent in order to find  $\alpha$ . This, however, is only useful in an optimization problem setting [33]. Here, we are merely trying to intercept the desired isoperformance contour  $J_{z,req}$  in a minimum number of iterations. From the expression

$$J_z(p_o + \alpha_o \cdot d_o) \cong J_z(p_o) + \nabla J_z(p_o)^T \cdot \underbrace{\alpha_o \left( \frac{-\nabla J_z(p_o)}{\|\nabla J_z(p_o)\|} \right)}_{n_o} \equiv J_{z,req} \quad (3.19)$$

one can solve for  $\alpha_o$ , such that

$$\alpha_o = \left( \frac{-\nabla J_z(p_o)^T \nabla J_z(p_o)}{\|\nabla J_z(p_o)\|^2} \right)^{-1} \cdot (J_{z,req} - J_z(p_o)) \quad (3.20)$$

This assumes that  $p_o$  is *not* an extremum or a saddle point of  $J_z(p_1, p_2)$ , where  $\|\nabla J_z(p_o)\| = 0$  would be true. Using the above equations, the algorithm generally intercepts an isoperformance contour, if it exists within  $\mathbf{B}$ , within a few iterations. In practice an upper limit is imposed on the step size to avoid “overshooting”, when going from a small gradient to a large gradient area of the parameter space  $\mathbf{B}$ .

If one assumes that the point  $p_{iso,k}$  lies on the isoperformance contour, one can find a neighboring point  $p_{iso,k+1} = p_{iso,k} + \Delta p_k$  such that  $J_z(p_{iso,k} + \Delta p_k) = J_z(p_{iso,k}) = J_{z,req}$  by recalling the Taylor series expansion in Equation (3.12). One then neglects second-order and higher terms and sets the first order term (perturbation) to zero. Specifically, if

$$J_z(p_{iso,k+1}) = J_z(p_{iso,k} + \Delta p_k) \cong J_z(p_{iso,k}) + (\nabla J_z)^T \Big|_{p_{iso,k}} \Delta p_k \equiv J_{z,req} \quad (3.21)$$

is to be true, then

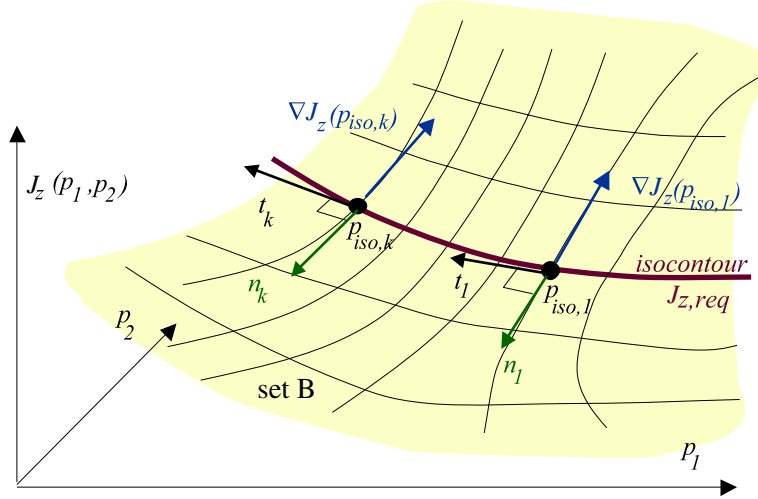
$$\Delta J_{z,k} = (\nabla J_z)^T \Big|_{p_{iso,k}} \Delta p_k \equiv 0 \quad (3.22)$$

*In other words, one must choose the vector  $\Delta p_k$ , such that it is in the nullspace of the transposed gradient vector  $(\nabla J_z)^T$ .* This condition can be written out componentwise as

$$\Delta J_{z,k} = \frac{\partial J_z}{\partial p_1} \Big|_{p_{1,k}} \Delta p_{1,k} + \frac{\partial J_z}{\partial p_2} \Big|_{p_{2,k}} \Delta p_{2,k} \equiv 0 \quad (3.23)$$



Geometrically this condition corresponds to following the tangential vector  $t$  along the isocontour. Figure 3.4 shows that  $t_k$  can be considered the tangential vector at point  $p_{iso,k}$ , which is orthogonal to the normal vector  $n_k$ . It is true that the vector  $n_k$  is the direction of steepest descent and is obtained according to equation (3.18). There are two ways in which



**Figure 3.4:** Algorithm II: Depiction of gradient vector  $\nabla J_z$ , normal vector  $n$  and tangential vector  $t$  along the isoperformance contour.

$t_k$  can be obtained from  $\nabla J_z(p_k)$ . First one can obtain the normal vector  $n_k$  from equation (3.18) and then rotate it by 90 degrees to obtain the tangential vector  $t_k$ .

$$t_k = \mathcal{R} \cdot n_k = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \cdot n_k \quad (3.24)$$

The second method is more general, since it is also applicable to the case of  $n_z > 1$  performances and  $n_p > n_z$  parameters. A singular value decomposition (SVD) [118] is performed on the transpose of the gradient vector.

$$U_k S_k V_k^T = \nabla J_k^T \quad (3.25)$$

In the bivariate case two singular values are obtained. The non-zero singular value,  $s_{1,k} \neq 0$ , corresponds to the direction of steepest descent  $n_k$  and the zero singular value,  $s_{2,k} = 0$ , corresponds to the tangential direction  $t_k$ . In the extension of this framework to the multivariable case, see Chapter 4, the power of the SVD to map out this nullspace will be discussed in detail. The vectors  $n_k$  and  $t_k$  are the columns of the unitary  $V_k$ -matrix, as in the

following example:

$$V_k = \begin{bmatrix} n_k & t_k \end{bmatrix} = \underbrace{\begin{bmatrix} 0.0369 \\ -0.9993 \end{bmatrix}}_{\text{normal vector}} \quad \underbrace{\begin{bmatrix} 0.9993 \\ 0.0369 \end{bmatrix}}_{\text{tangential vector}} \quad (3.26)$$

Note that  $n_k$  and  $t_k$  are orthogonal to each other, i.e.  $n_k^T \cdot t_k = 0$ . The vector  $t_k$  is used as the direction, when stepping along the isoperformance contour.

Now that the step direction  $t_k$  has been found an appropriate step size,  $\alpha_k$ , needs to be chosen. The step size should be variable and take into account errors incurred due to the linearization in equation (3.16). An estimate of the linearization error incurred due to a step of size  $\Delta p_k$  can be written as:

$$\epsilon_k = \frac{1}{2} \Delta p_k^T H|_{p_k} \Delta p_k + \text{H.O.T.} \quad (3.27)$$

Neglecting higher order terms, one solves for the step size  $\alpha_k$ , by substituting  $\Delta p_k = \alpha_k \cdot t_k$  in the above equation and setting  $\epsilon_k = \tau J_{z,req}/100$ .

$$\alpha_k = \left[ \frac{2\tau J_{z,req}}{100} (t_k^T \cdot H|_{p_k} \cdot t_k)^{-1} \right]^{1/2} \quad (3.28)$$

In practice it is expected that the exact computation of the Hessian matrix  $H|_{p_k}$  will be quite expensive. Gutierrez shows in Appendix C of Reference [45] that solving for the second order derivatives, i.e. a single entry in the Hessian,  $\partial^2 J_z / \partial p_j^2$ , requires the solution of four Lyapunov equations for  $\mathcal{H}_2$  performance metrics. However, assuming that the exact first partial derivatives  $\partial J_z / \partial p_j$ , where  $j = 1, 2$ , are known, one can approximate the Hessian matrix  $H$ <sup>4</sup> for purposes of finding the step size  $\alpha_k$  by finite differences as follows:

$$H|_{p_k} = \begin{bmatrix} \frac{\partial^2 J_z}{\partial p_1^2} & \frac{\partial^2 J_z}{\partial p_1 \partial p_2} \\ \frac{\partial^2 J_z}{\partial p_2 \partial p_1} & \frac{\partial^2 J_z}{\partial p_2^2} \end{bmatrix}_{p_k} \cong \begin{bmatrix} \frac{\frac{\partial J_z}{\partial p_1} \Big|_k - \frac{\partial J_z}{\partial p_1} \Big|_{k-1}}{p_{1,k} - p_{1,k-1}} & \frac{\frac{\partial J_z}{\partial p_2} \Big|_k - \frac{\partial J_z}{\partial p_2} \Big|_{k-1}}{p_{1,k} - p_{1,k-1}} \\ \frac{\frac{\partial J_z}{\partial p_1} \Big|_k - \frac{\partial J_z}{\partial p_1} \Big|_{k-1}}{p_{2,k} - p_{2,k-1}} & \frac{\frac{\partial J_z}{\partial p_2} \Big|_k - \frac{\partial J_z}{\partial p_2} \Big|_{k-1}}{p_{2,k} - p_{2,k-1}} \end{bmatrix} \quad (3.29)$$

With equations (3.26) and (3.28) the step direction  $t_k$  and the step size  $\alpha_k$  have been determined and one can find the next point on the isoperformance contour  $p_{iso,k+1} = p_{iso,k} +$

<sup>4</sup>Note that the step size  $\alpha_k$  has to be artificially limited in a region of small curvature, i.e. when the norm of  $t_k^T H^{-1} t_k$  becomes very small.

$\alpha_k t_k$ . At this new point the performance  $J_z(p_{iso,k+1})$  is recomputed along with the gradient vector  $\nabla J_z(p_{iso,k+1})$ . The process is repeated until the parameter boundaries of  $\mathbf{B}$  are reached, the solution reaches the unstable subspace  $\mathbf{U}$  or the isoperformance contour closes on itself.

In practice a “front” is created, where new points on the isocontour are found in the positive and negative step directions. The new points represent the “front”,  $p_{front}$ , which is used to find the next generation of front points. Also a “control” algorithm was implemented to check whether the isoperformance solution points are drifting away from the  $J_{z,req}$  “centerline”. If the inequality

$$\left| \frac{J_z(p_{iso,k+1}) - J_{z,req}}{J_{z,req}} \right| > \frac{\tau}{100} \quad (3.30)$$

is true, a correction term in the normal direction,  $n_k$ , is applied to bring the solution back on the centerline of the isoperformance contour. In that instance the parameter step  $\Delta p_k$  is augmented by a normal correction term as:

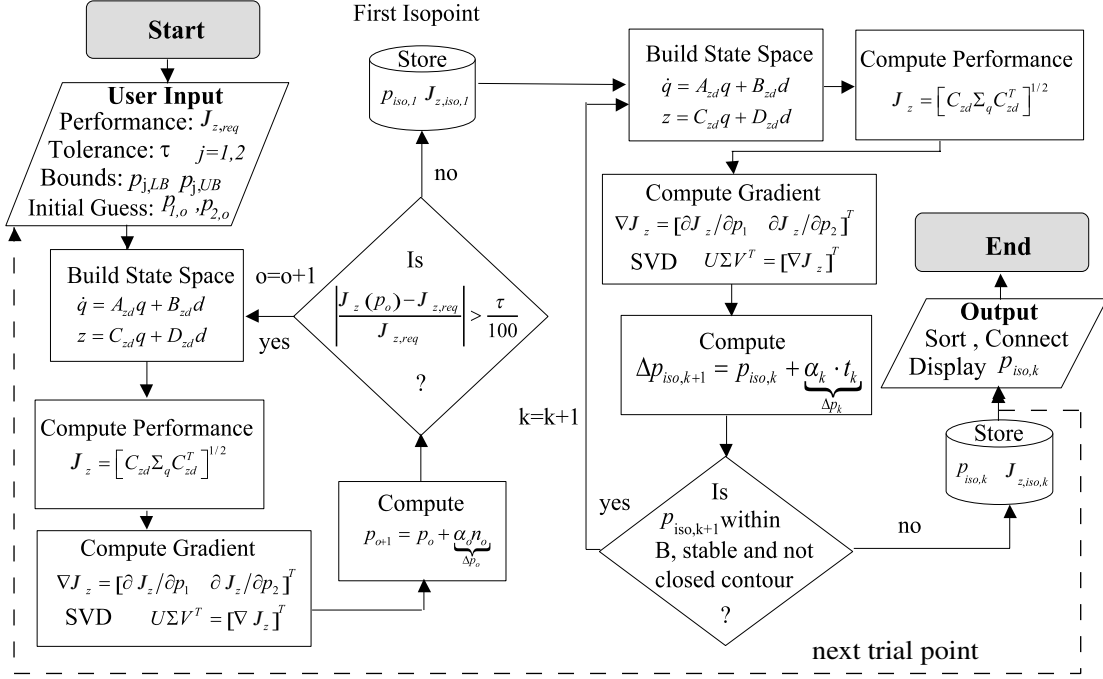
$$\Delta p_k = \alpha_k \cdot t_k + \underbrace{\beta_k \cdot n_k}_{\text{correction term}} \quad (3.31)$$

where  $\beta_k$  is computed from equation (3.20).

An overview of Algorithm II is shown in the flow diagram of Figure 3.5. The first iteration loop on the left brings the solution from the initial guess  $p_o$  to the first point on the isoperformance contour  $p_{iso,1}$ . The second loop on the right hand side traces along the isoperformance contour until the parameter space boundaries defined by the upper and lower bounds  $p_{j,LB}$  and  $p_{j,UB}$ , where  $j = 1, 2$ , are intercepted, the solution becomes unstable or the contour closes on itself. The outer loop indicated by a dashed line is relevant if several (random) initial guesses are used to detect multiple segments within  $\mathbf{B}$ . Finally the points,  $p_{iso}$ , on the isoperformance contours are sorted and stored for further processing.

### 3.2.3 Algorithm III: Progressive Spline Approximation

The previous algorithm is an improvement over exhaustive search, since it uses information about previously computed points along the isoperformance contours. Also the algorithm automatically adjusts the step size  $\Delta p$  in “shallow” gradient areas, where  $\|\nabla J_z\|$  is small. A further efficiency increase can be obtained by exploiting a progressive spline approximation scheme. This assumes that the isoperformance contour intersects the boundary  $\mathbf{B}$ , i.e. that



**Figure 3.5:** Flow Diagram for Algorithm II: Gradient-based Contour Following. Note: The correction step from equation (3.31) is omitted for simplicity.

no closed loops are present. This is most often the case, when the performance function  $J_z(p_1, p_2)$  is monotonic in at least one of the two parameters. This condition will be referred to as “quasi-monotonic” throughout this thesis.

The basic idea of this algorithm is to approximate the isoperformance contour with a piecewise polynomial (pp) function. The spline mathematics and tools developed by Carl de Boor [22] as well as the resulting MATLAB spline toolbox are leveraged for this algorithm. A function  $f(x)$  can be approximated by a truncated Taylor series as follows:

$$f(x) = \sum_{i=0}^n \frac{(x-a)^i}{i!} \left. \frac{df^i}{dx^i} \right|_a = \sum_{i=0}^n \frac{(x-a)^i}{i!} D^i f(a) \quad (3.32)$$

This provides a satisfactory approximation to  $f(x)$  if  $f$  is sufficiently smooth and  $x$  is sufficiently close to  $a$ . The approximation interval  $[a, b]$  is then subdivided into sufficiently small intervals  $[\zeta_l, \zeta_{l+1}, \dots]$  with

$$a = \zeta_1 < \dots < \zeta_{l+1} = b \quad (3.33)$$

On each such interval a polynomial  $P_l$  of relatively low degree can provide a good approxima-

tion to  $f$ . The composite function (piecewise polynomial) is then:

$$s(x) = P_l(x) \text{ for } x \in [\zeta_l, \zeta_{l+1}] \quad \forall l \quad (3.34)$$

A smooth blending of the polynomial pieces can be achieved such that  $s(x)$  has several continuous derivatives. We will call  $s(x)$  a spline, an expression that was originally created by I.J. Schoenberg. Splines have been extensively treated by de Boor [22]. This work also resulted in the MATLAB `spline` toolbox, which is used extensively throughout this thesis.

Of particular interest is the `ppform` of a spline, which provides a mathematical description in terms of its break points (`breaks`)  $\zeta_1, \dots, \zeta_{l+1}$  and the local polynomial coefficients  $c_{l,i}$  of its `pieces`.

$$P_l(x) = \sum_{i=1}^k \frac{(x - \zeta_j)^{k-i}}{(k-i)!} c_{l,i} \quad (3.35)$$

This form is especially convenient for evaluation, while the **B**-form is often used for construction of a spline approximation. The `order` is chosen as  $k = 4$ , which leads to cubic splines and two continuous derivatives across the break points.

The progressive spline approximation algorithm assumes that the two endpoints  $a, b$  are on the parameter space boundary. The knowledge of the first derivatives at these endpoints facilitates the formulation of correct end conditions. Thus, the initial estimate of the isoperformance contour consists of a single piece. Instead of using the formulation in (3.34), the isoperformance contours are parameterized with parameter  $t$  from endpoint to endpoint. Thus at endpoint  $a$  we have  $t = 0$  and at endpoint  $b$  we set  $t = 1.0$ . Instead of the coordinates  $x$  and  $y = f(x)$  as in equation (3.35) the algorithm works with *vector* splines such that

$$t \mapsto P_l(t) = \begin{bmatrix} p_{iso,1}(t) \\ p_{iso,2}(t) \end{bmatrix} = \begin{bmatrix} s_1(t) \\ s_2(t) \end{bmatrix} = p_{iso} \quad (3.36)$$

where

$$t \in [0, 1] \mapsto P_l(t) \in [a, b] \quad (3.37)$$

the vector components of each spline `piece` are approximated as piecewise polynomials in `ppform`, where

$$s_j(t) = f_{j,l}(t) \text{ for } j = 1, 2 \text{ and } \forall l \quad (3.38)$$

The functional approximation for each piece is then given as

$$f_{j,l}(t) = \sum_{i=1}^k \frac{(t - \zeta_l)^{k-i}}{(k-i)!} \cdot c_{j,l,i} \text{ where } t \in [\zeta_l \dots \zeta_{l+1}] \quad (3.39)$$

Note that *all* relevant information is contained in the break point sequence,  $\zeta_1 \dots \zeta_{l+1}$  and in the polynomial coefficient array  $c_{j,l,i}$ . The subscript  $j$  refers to the vector component of  $p_{iso}$ ,  $l$  refers to the piece number of the **pp** approximation and  $i$  is the index of the polynomial degree. In practice the coefficient array  $c_{j,l,i}$  is stored as a 2-dimensional matrix by stacking the coefficient matrices of the vector components  $j$  on top of each other, along the first non-singleton dimension. As a practical example the isoperformance contour shown in Figure 3.6 is saved in MATLAB with the following variables:

- **form:** 'pp'
- **breaks:**  $t \in \zeta_1 \dots \zeta_{l+1} = [0 \ 0.1667 \ 0.3333 \ 0.5000 \ 0.6667 \ 0.8333 \ 1]^T$
- **coefs:**  $c_{j,l,i}$ : 12x4 double precision matrix
- **pieces:**  $l = 6$
- **order:**  $k = 4$
- **dim:**  $max(j) = 2$

Next a bisection is performed at the mid-point of the first piece, ( $t = 0.5$ ), resulting in the point  $p_{mid,1}$ . If the true isoperformance contour is close to the cubic spline approximation, then  $p_{mid,1}$  will lie on the contour. Generally this will not be the case and  $p_{mid,1}$  is then used as the starting point for a steepest gradient search to find the closest point on the contour. This point  $p_{iso,1}$  represents a new break  $\zeta_2$  and splits the original interval  $[a, b]$  into two pieces. The MATLAB function `csape.m` is used to compute the spline coefficient matrix  $c$  for the pieces  $[a = \zeta_1, \zeta_2]$  and  $[\zeta_2, b = \zeta_3]$ . This bisection procedure is repeated until the midpoints of all pieces lie on the contour, subject to a tolerance  $\tau$ . This is graphically shown in Figure 3.6.

A flow diagram of the progressive spline algorithm is shown in Figure 3.7. The algorithm first finds the isoperformance points,  $p_{iso}$ , on the boundary as well as their first derivatives. Next a cubic spline (**cs**) interpolation is performed in order to obtain an initial estimate of the shape of the isoperformance contour. This initial estimate is refined via a bisection at the mid-point of each piece ( $t=0.5$ ). Pieces, where the mid-point is not within the tolerance  $\tau$  are bisected further until all pieces meet the tolerance. Thus, computational resources are focused on areas where the shape of the contours varies rapidly. This algorithm requires access to the MATLAB **spline** toolbox (e.g. function `csape.m`).

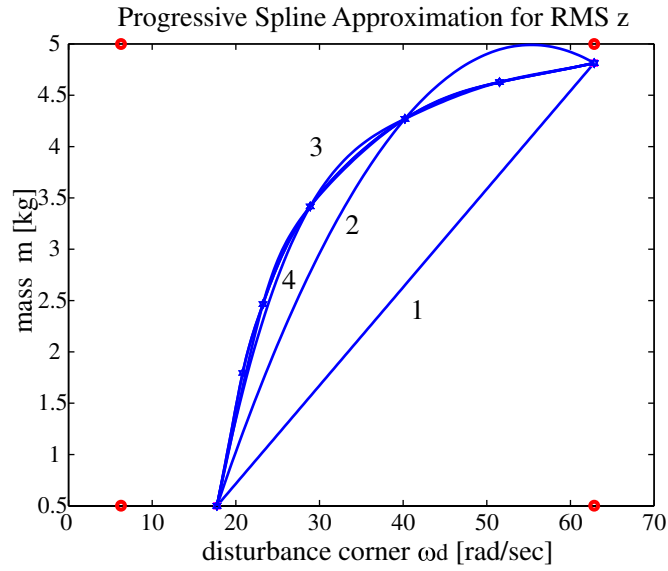


Figure 3.6: Progressive (cubic) spline approximation.

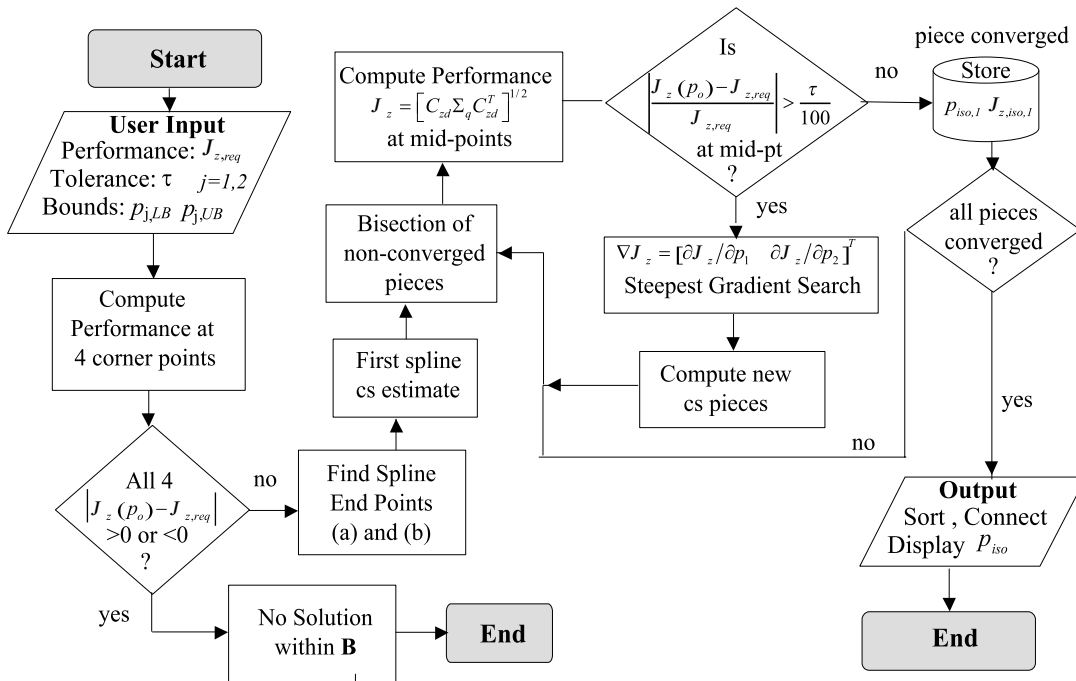


Figure 3.7: Flow Diagram for algorithm III: Progressive (cubic) spline approximation

The algorithm is expected to be the most efficient, but also the most restrictive, since it is only able to find open segments (no closed segments), which intercept the boundary of  $\mathbf{B}$ , as is the case for (quasi-)monotonic functions.

### 3.3 Algorithm Evaluation

This section applies the three algorithms, which have been implemented in MATLAB code, to the single DOF sample problem and quantitatively as well as qualitatively compares the answers. The conclusions will provide guidance for applications to larger problems and the multivariable case. The single DOF problem from Section 2.1 is used to compute and display isoperformance contours. We choose the disturbance corner frequency,  $\omega_d$ , and oscillator mass,  $m$ , as the variable parameters with the parameter bounds given in Figure 2.1.

#### 3.3.1 Quality of Isoperformance Solution

In order to assess how well the resulting isoperformance points,  $p_{iso}$ , actually meet the isoperformance condition (3.3) it is necessary to define a solution “quality” metric. The “quality” of the isoperformance solution can be quantified as follows. Let

$$\Upsilon_{iso} = \frac{100}{J_{z,req}} \cdot \left[ \frac{\sum_{k=1}^{n_{iso}} [J_z(p_{iso,k}) - J_{z,req}]^2}{n_{iso}} \right]^{1/2} \quad (3.40)$$

be a quality metric expressing the relative % error with respect to  $J_{z,req}$ . In the above equation  $n_{iso}$  is the total number of isopoints computed,  $J_z(p_{iso,k})$  is the performance of the  $k$ -th isopoint and  $J_{z,req}$  is the performance requirement, i.e. the desired performance “level”. This number can then be directly compared to the desired isoperformance contour tolerance,  $\tau$ , and should always be smaller than it. Note that this definition of solution quality does not prevent individual solutions,  $p_{iso}$ , from falling outside the tolerance band  $[(1 - \tau/100) \cdot J_{z,req}, (1 + \tau/100) \cdot J_{z,req}]$ .

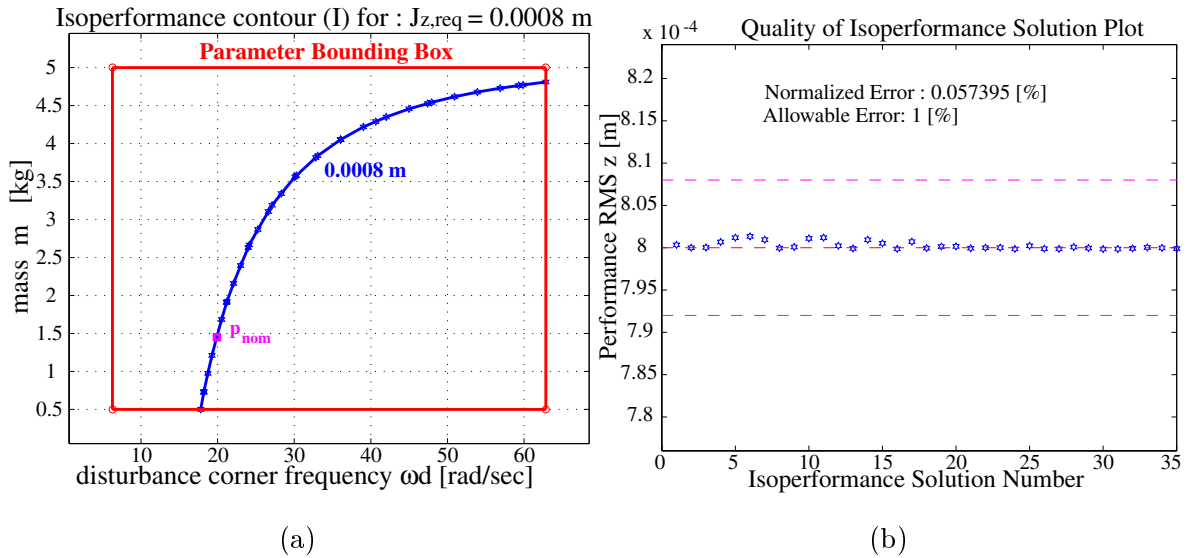
#### 3.3.2 Results for Exhaustive Search (Algorithm I)

This subsection shows the results obtained for Algorithm I (exhaustive search). Subsection 3.2.1 specified that it is necessary to define a discretization step size  $\Delta p$ . This was chosen as  $1/20$  of the interval  $p_{UB} - p_{LB}$ . The results for exhaustive search are shown in Figure 3.8(a).



The isoperformance curve shows that a small increase in the disturbance filter corner frequency,  $\omega_d$ , below about 30 radians per second (roughly 5 Hz), which is the natural undamped frequency of the oscillator, requires a large increase in mass  $m$  in order to maintain the same RMS level. Thus, as the disturbance energy increases the disturbability of the oscillator must be decreased by increasing the mass. Once the disturbance corner frequency,  $\omega_d$ , exceeds the natural undamped frequency of the oscillator any further increases in  $\omega_d$  have only a small effect on the RMS of  $z$  and the isoperformance curve flattens out.

The quality of the isoperformance contour is very dependent on the discretization level. The smaller  $\Delta p$ , the better the contour will be interpolated but the more computation time is required. For the exhaustive search algorithm the solution quality is shown in Figure 3.8(b).

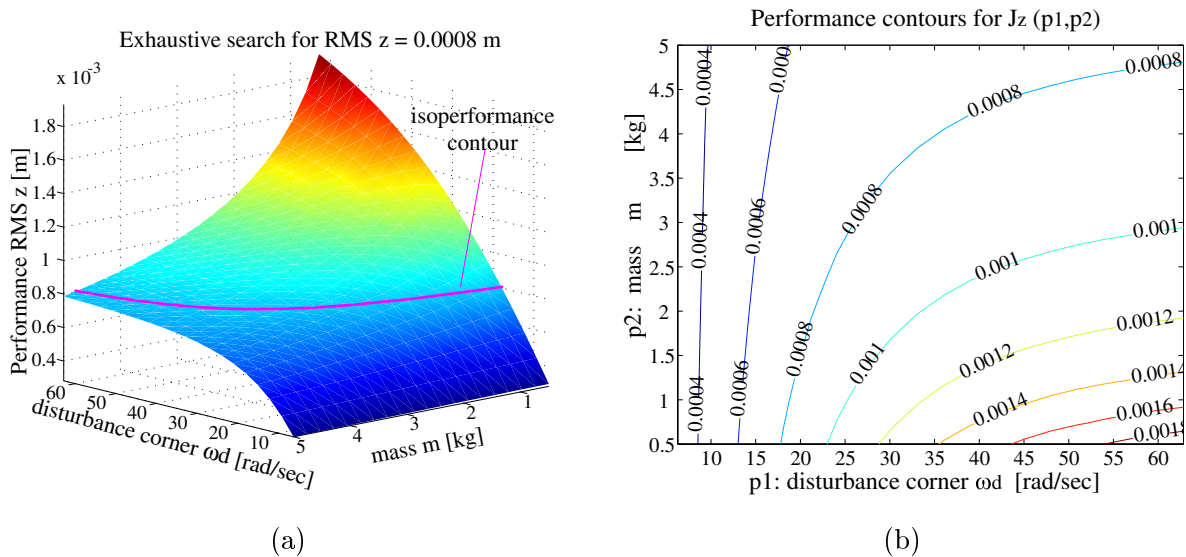


**Figure 3.8:** (a) Algorithm I (Exhaustive Search): Isoperformance contour for single DOF problem  $(\omega_d, m)$  with discretization  $\Delta p = (1/20)[p_{UB} - p_{LB}]$  and a tolerance of  $\tau = 1\%$ . (b) Quality: Contour quality according to (3.40).

The points on the isoperformance contour in Figure 3.8(a) were obtained by using the MATLAB function `contourc.m` and exhaustive search. The tolerance for the isoperformance contour was set to  $\tau = 1.0\%$ . In order to obtain this plot a total of  $21 \times 21 = 441$  plant models in the form of equation (1.1) had to be evaluated. This suggests that this algorithm will likely be computationally expensive for large order models with  $n_p > 2$  variable parameters. A comparison of the computational cost with the other algorithms is shown in Table 3.1.

Since for Algorithm I (exhaustive search), solutions are computed over the entire parameter space  $\mathbf{B}$ , it is possible to graphically represent the bivariate function  $J_z(p_1, p_2)$  as shown in

Figure 3.9(a). The best performance (lowest RMS value  $\sigma_z$ ) in the system is achieved for a large mass  $m$  (small disturbability) and low disturbance corner frequency  $\omega_d$  (low disturbance power). Another advantage is that a family of contours can be displayed, see Figure 3.9(b), with little additional computational expense.



**Figure 3.9:** (a) Algorithm I (Exhaustive Search): Surface plot of  $J_z(p_1, p_2)$  for SDOF oscillator problem. (b) Family of isoperformance contours in 0.2 mm RMS increments.

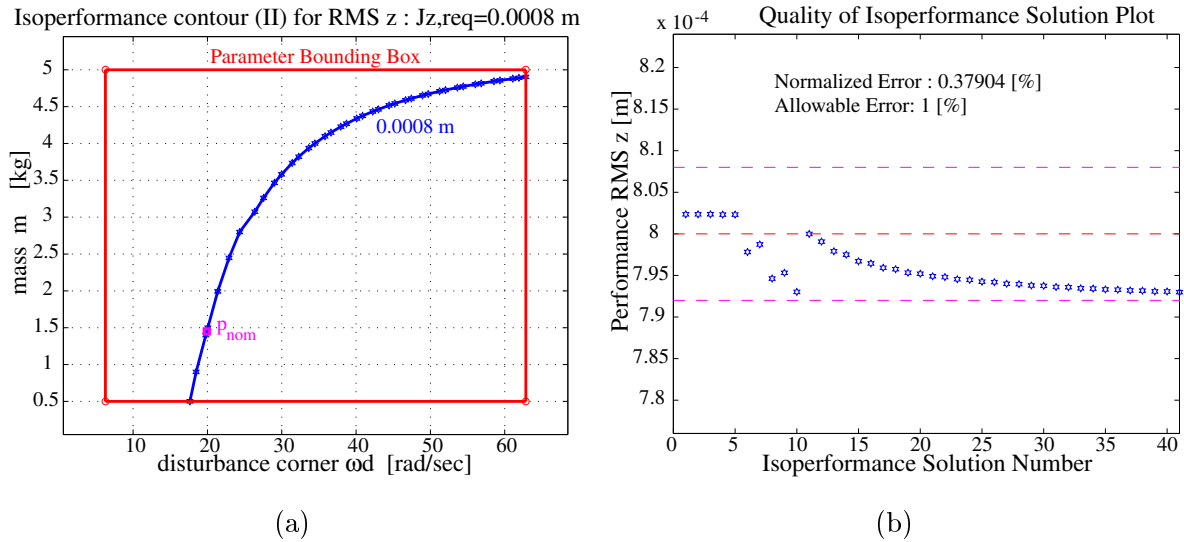
The answers to the questions posed in Subsection 1.3.1 for the single DOF problem are: (1) Yes, there are solutions within  $p_{j, LB}$  and  $p_{j, UB}$ , where  $j = 1, 2$ , which yield the desired performance  $J_{z, req} = 0.8$  [mm]. Since  $p_1$  and  $p_2$  are considered to be continuous, there is an infinite number of combinations, which yield  $J_{z, req}$ . (2) The locus of points that yield  $J_{z, req}$  within a  $\tau = 1.0$  % tolerance is shown in Figure 3.8(a).

At this point one might be tempted to ask: “Why do we compute the isoperformance contours at all, when we can achieve much better performance within the parameter bounds  $\mathbf{B}$  as shown in Figure 3.9 for small  $\omega_d$  and large  $m$ ?”. In general it is expected that parameter values that yield good performance (i.e. low  $J_z$ ) will also have a higher “cost” associated with them. In this example it means that we cannot tolerate a large amount of external disturbance (large  $\omega_d$ ) and that a lot of mass  $m$  has to be used to reduce oscillator disturbability. This notion of cost will be discussed in more detail in Chapters 4 and 7. Examples of other system parameters that will improve dynamic performance at a higher “cost” are small reaction wheel imbalance, stiffer secondary tower, lower detector noise, higher control bandwidth

etc. Thus, instead of finding the best performance achievable within the parameter bounds (=performance optimization) and perhaps over-designing the system, we want the system to perform just “well enough”, as expressed by the  $J_{z,req}$  curve (which includes the necessary safety margins). Once that family of designs is found, one can determine the lowest cost solution within this isoperformance set **I**.

### 3.3.3 Results for Gradient-Based Contour Following (Algorithm II)

Figure 3.10(a) shows the isoperformance contour obtained with Algorithm II (gradient-based contour following) and the associated error,  $\Upsilon_{iso}$ , in Figure 3.10(b) for the single DOF sample problem. By comparing the isoperformance contours in Figure 3.8 and 3.10, one sees that both algorithms yield similar results. A detailed comparison of the computational effort required for each method is shown in Table 3.1.

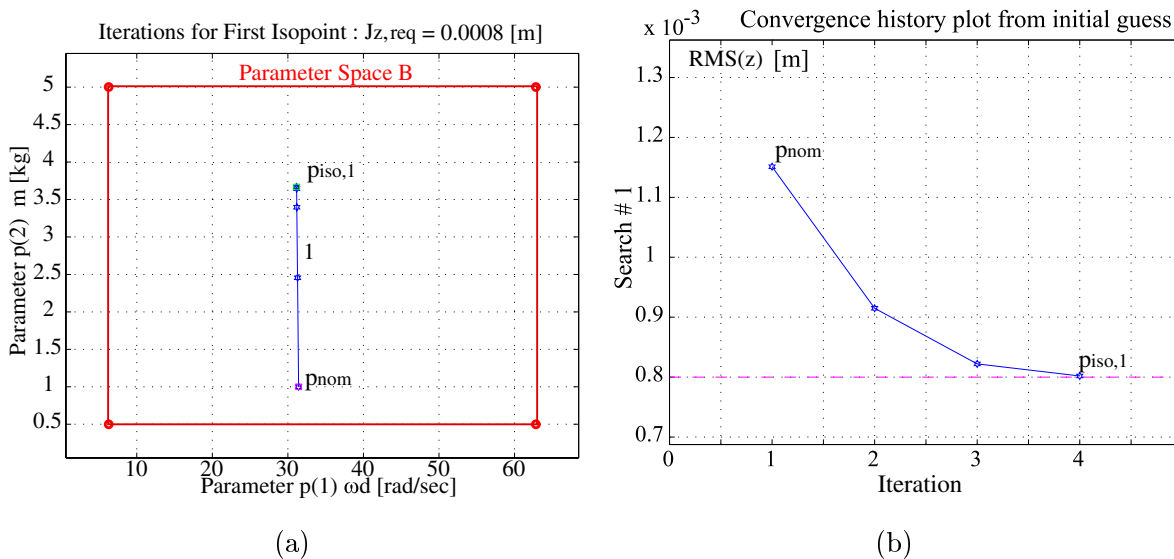


**Figure 3.10:** (a) Algorithm II (Gradient-Based Contour Following): Isoperformance contour for single DOF problem  $(\omega_d, m)$  with tolerance  $\tau = 1\%$ . (b) Quality: Contour solution quality according to (3.40).

The quality of the solution in algorithm II is governed by the tolerance  $\tau$ , which leads to a variable step size. The step size also depends on the curvature of the underlying bivariate function,  $J_z(p_1, p_2)$ . This is a very desirable property of the algorithm, since it will use many points to represent areas with a large change in sensitivity and few points in areas of small sensitivity change. Note that the quality of the solution, as measured by the  $\Upsilon_{iso}$  metric, see equation (3.40), is well within the tolerance mandated by  $\tau$ . This precise tracking of

the isoperformance contour was only achievable once the “control” mechanism introduced in Subsection 3.2.2 was implemented. Previously the isoperformance solution would gradually wander away from the “centerline” of the isoperformance contour, due to the cumulative effect of linearization errors.

The algorithm starts by evaluating the performance  $J_z(p_{nom})$  and the gradient vector  $\nabla J_z(p_{nom})$  for the nominal solution  $p_{nom}$ , which was found via the performance enhancement step in Section 2.1. By design this point is already on the isoperformance contour. In general, however, the initial guess,  $p_{nom}$ , is close to, but not exactly on the isoperformance contour. The algorithm then tries to find an initial point on the contour,  $p_{iso,1}$ , using the steepest descent algorithm from Subsection 3.2.2. This initial convergence to intercept the desired isoperformance contour is shown for a different initial guess  $p_{nom} = [\omega_d, m, k, c, \omega_o] = [31.416, 1, 1449.7, 0.72486, 455.44]^T$  in Figure 3.11. The isoperformance contour is intercepted after three iterations.



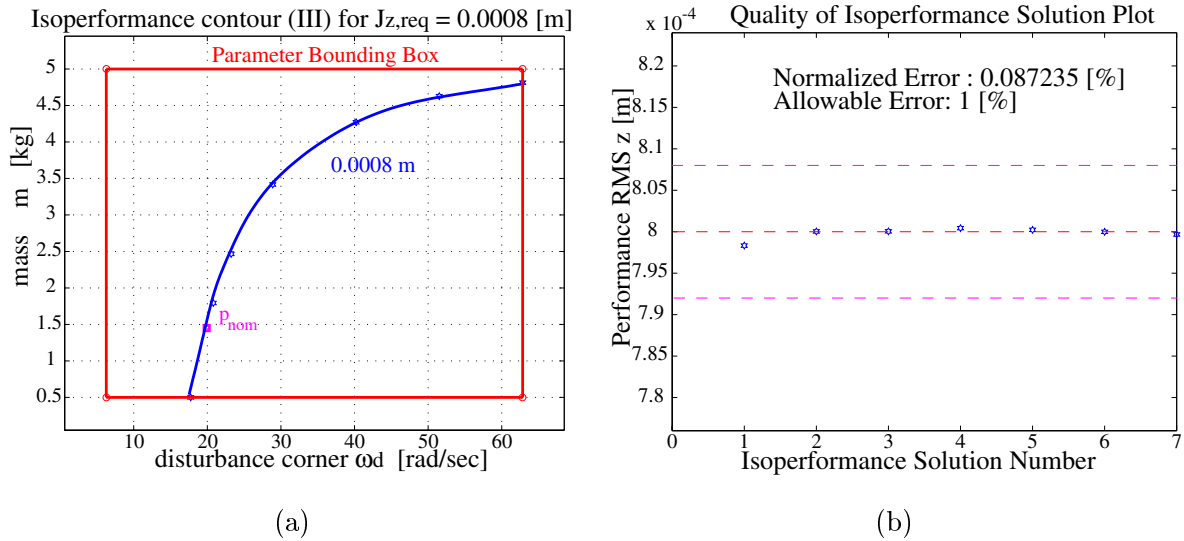
**Figure 3.11:** Intercept of isoperformance contour: (a) Iterates from initial point ( $p_{nom}$ ) to isoperformance contour ( $p_{iso,1}$ ). (b) Convergence history.

It noteworthy that the steepest descent algorithm can overshoot the isoperformance contour, when the bivariate function  $J_z(p_1, p_2)$  transitions from an area of low curvature to an area of higher curvature. The algorithm then gradually “climbs” back to the isoperformance contour  $J_{z,req}$ . Once the first isopoint,  $p_{iso,1}$ , is found, the algorithm switches to using the tangential vector  $t$  introduced in Subsection 3.2.2 to trace along the isocontour until a boundary

of  $\mathbf{B}$  or the unstable set  $\mathbf{U}$  is intercepted or the contour closes on itself.

### 3.3.4 Results for Progressive Spline Approximation (Algorithm III)

The progressive spline approximation algorithm produces the results shown in Figure 3.12. The predicted isoperformance contour resembles the results obtained with the previous two methods. One difference is that far fewer isopoints (**break points**) need to be computed than with the previous methods, since the `ppform` approximation is used.



**Figure 3.12:** (a) Algorithm III Progressive Spline Approximation: Isoperformance contour for single DOF problem  $(\omega_d, m)$  with tolerance  $\tau = 1\%$ . (b) Quality: Contour solution quality according to (3.40).

The progress in approximating the isocontour with cubic splines can be graphically recorded. Figure 3.6 shows the successive iterations, which are computed until all mid-points lie on the isoperformance contour within tolerance  $\tau$ . This algorithm performs very well for open contours and for performance functions  $J_z(p)$ , which are smooth and (quasi-)monotonic over the interval  $[p_{LB}, p_{UB}]$ .

### 3.3.5 Algorithm comparison

The purpose of this subsection is to compare the three isoperformance algorithms and to derive recommendations for their use in bivariate problems. In order to achieve a fair comparison it was deemed necessary that all three methods yield isoperformance solutions of nearly equal quality as expressed by the  $\Upsilon_{iso}$  metric. A comparison of the three methods in terms of

computational cost and quality is followed by a more qualitative comparison of advantages and disadvantages.

From Table 3.1 and by comparing Figures 3.8, 3.10 and 3.12, one sees that all three algorithms have the ability of accurately finding an isoperformance contour, if it exists within **B**. There are, however, significant differences in the amount of computational resources needed and in the scalability to larger problems. Algorithm I is by far the most computationally expensive<sup>5</sup>. This is due to the fact that in the SDOF case 441 points  $(J_z)_{k,l}$  had to be evaluated, but only 35 points form the isoperformance contour. If a multivariable scenario is considered, say four variable parameters and one performance  $J_z$ , and a discretization of  $n_i = 50$  increments is assumed, where  $i = 1, 2, 3, 4$ , in each parameter axis, one will have to evaluate  $50^4 = 6.25 \cdot 10^6$  points. If one solves for the output RMS of a state space system of a realistic size, say 400 states, and if it takes 90 seconds to solve the Lyapunov equation for the state covariance matrix,  $\Sigma_q$ , it would take 6510.4 days of CPU time to evaluate all combinations  $(J_z)_{k,l}$ , which is clearly prohibitive.

**Table 3.1:** Comparison of algorithms I-III for isoperformance analysis of single degree-of-freedom problem.

Result	I: Ex. Search	II: Co. Follow	III: Sp. Approx.
FLOPS	2,140,897	783,761	377,196
Tolerance: $\tau$	1.0 %	1.0 %	1.0 %
Actual Error: $\Upsilon_{iso}$	0.057 %	0.379 %	0.087 %
number of isopoints	35	41	7

The third approach (progressive spline approximation) is the fastest, however it only works for open segments and assumes that there is only a single isoperformance contour which intersects the boundary **B**. Thus, it is the most restrictive (least general) of the three algorithms. The first algorithm (exhaustive search) is the most general. Also the quality of the solution is strongly depended on the assumed discretization  $\Delta p$ .

The second algorithm (gradient-based contour following) has a computational cost which is in between the other two methods. Multiple open or closed segments can be detected, but

<sup>5</sup>This can be different when multiple contours are desired.

several random trial points  $p_{nom,i}$ , where  $i = 1, 2, \dots, \#$ of trial points, are required to detect multiple contours. The advantage of this method is that it uses knowledge about the previous points  $p_{iso,k}$  obtained in order to compute the next isoperformance solution  $p_{iso,k+1}$ . Another advantage is that the step size  $\alpha_k$  automatically adjusts according to the local curvature of  $J_z(p_{iso,k})$  by means of a finite difference approximation of the Hessian matrix, see Equation (3.29). The disadvantage of algorithm II is that one must recompute the gradient  $\nabla J_z(p_{iso,k})$  at each new isopoint. Methods to accelerate the speed of gradient calculations are presented in Chapter 5. The scalability of these algorithms to the multivariable case is investigated in Chapter 4.

A fourth algorithm uses Newton's method, but is based on a closed form solution of the  $J_z(p_1, p_2)$  function and is therefore not applicable to general large order problems. Unfortunately the expressions for systems with more than just a few states become intractable and after about 8 states, closed form, published expressions for the integral of  $S_{zz}(\omega)$  no longer exist [141]. Even symbolic mathematical engines such as `Maple` would not be able to deal with a state space model on the order of  $10^2$  to  $10^3$  states. For these reasons this algorithm was abandoned early in this thesis research.

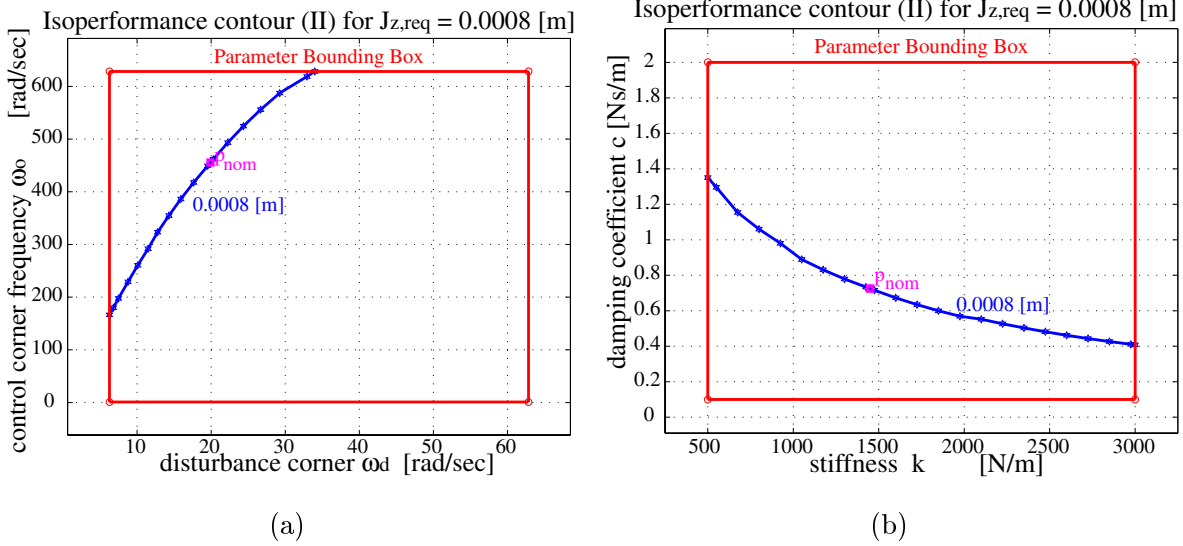
## 3.4 Application Examples

The purpose of this section is to demonstrate the kind of answers and insights that can be gained by solving bivariate isoperformance problems. The problems are essentially a subset of the sample problems introduced in Chapter 2 as well as a more realistic SIM Classic example.

### 3.4.1 Single DOF oscillator

Figure 3.13 shows isoperformance contours for the SDOF problem and the variable parameter combinations  $[\omega_d, \omega_o]$  and  $[k, c]$ . The two plots show a tradeoff between disturbance corner frequency  $\omega_d$  and control "bandwidth"  $\omega_o$ , see Figure 3.13(a). The other values  $m, k, c$  are fixed at the nominal values for  $p_{nom}$  specified in Section 2.1.

As expected the control bandwidth,  $\omega_o$ , needs to be increased as  $\omega_d$  increases, since more disturbance energy enters the system at higher frequencies. With the other parameters fixed at the values of  $p_{nom}$  we cannot tolerate a disturbance corner frequency of more than roughly 35 [rad/sec]. The results also suggest that disturbance isolation (reducing  $\omega_d$ ) is more effective



**Figure 3.13:** (a) SDOF isoperformance for  $\omega_d$  versus  $\omega_o$  with  $J_{z,req} = 0.0008$  [m] and  $\tau = 1$  %. (b) SDOF isoperformance for  $k$  versus  $c$ .

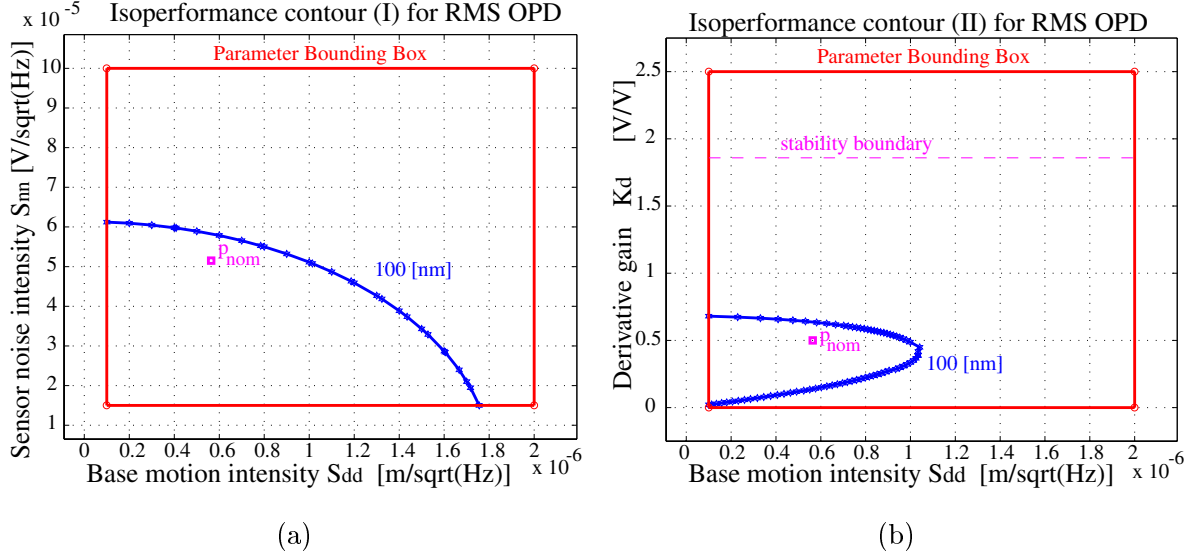
than output isolation (increasing  $\omega_o$ ) on a relative (per Hz) basis. The contour in Figure 3.13(b) trades the spring stiffness  $k$  with the damping coefficient  $c$ . In order to maintain a constant displacement RMS, the restoring force in the oscillator comes from a combination of a displacement dependent force  $kx$  and velocity dependent force  $c\dot{x}$ . As the spring is made softer more damping needs to be added in order to maintain the same level of (RMS) performance.

### 3.4.2 ODL Design Problem

An application of the bivariate isoperformance methodology to the ODL problem is demonstrated in this subsection. In Section 2.3 an initial design,  $p_o$ , was developed, which resulted in a RMS OPD performance of 89.6 [nm], which meets the requirement of 100 [nm]. There was a suspicion that sensor noise of intensity  $S_{nn}$  was a major contributor to this error and that better performance might be achieved with a smaller derivative gain  $K_d$ . In this context two important isoperformance contours were computed at the 100 [nm] level, see Figure 3.14.

The left isoperformance contour (a) shows the trade between the process noise and sensor noise intensity. As the process noise increases the ODL can tolerate less sensor noise (at the same control gain  $K_d$ ) in order to maintain constant performance. The relative contribution of each disturbance source to the total performance  $J_z$  is the subject of error budgeting, which is discussed in Chapter 7.

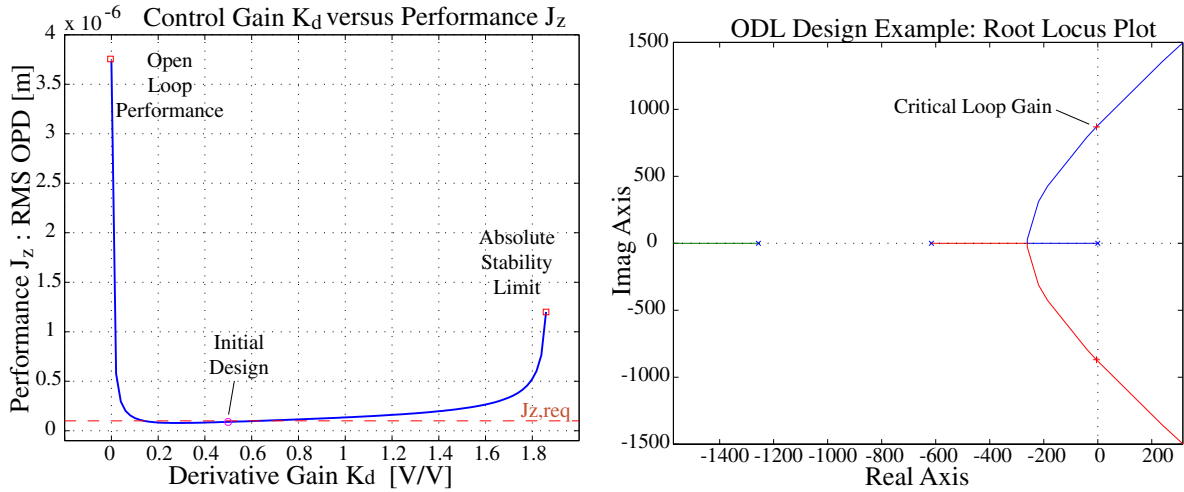




**Figure 3.14:** (a) ODL isoperformance for  $S_{dd}$  versus  $S_{nn}$  at  $J_{z,req} = 100$  [nm] and  $\tau = 1$  %. (b) ODL isoperformance for  $S_{dd}$  versus  $K_d$ .

The isoperformance contour in Figure 3.14(b) shows the relationship between process noise intensity  $S_{dd}$  and (derivative) control gain  $K_d$ . Without sensor noise and stability considerations one expects the control gain to increase monotonically as the process noise increases. In the presence of sensor noise, however, the relationship is more complex. The 100 [nm] contour shows that, all other parameters being constant as shown by  $p_o$  in Table 2.3, one cannot tolerate a process noise intensity  $S_{dd}$  larger than  $1.05 \cdot 10^{-6}$  [ $m/\sqrt{Hz}$ ]. For smaller  $S_{dd}$  there are two control gain settings  $K_d$  that will yield a 100 [nm] RMS OPD performance. At the higher setting residual sensor noise is dominant, while at the lower setting residual process noise dominates. The stability boundary is shown as a dashed horizontal line within the parameter bounding box **B**. Another way to visualize this relationship is by computing the performance  $J_z$  (RMS OPD) as a function of control gain  $K_d$  from zero to the absolute stability limit, see Figure 3.15(a).

One sees that the open loop performance ( $K_d = 0$ ) does not meet the requirements. The initial design ( $K_d = 0.5$ ) is satisfactory but not optimal and the absolute stability limit is at  $K_d = 1.87$ . It is interesting to note that the performance improves very quickly as the gain is increased from 0 to 0.2. After roughly  $K_d = 0.4$  the performance worsens with increasing control gain due to the increased contribution of sensor noise. The optimum appears to be where sensor noise and process noise have equal contribution around  $K_d \approx 0.4$ . The absolute



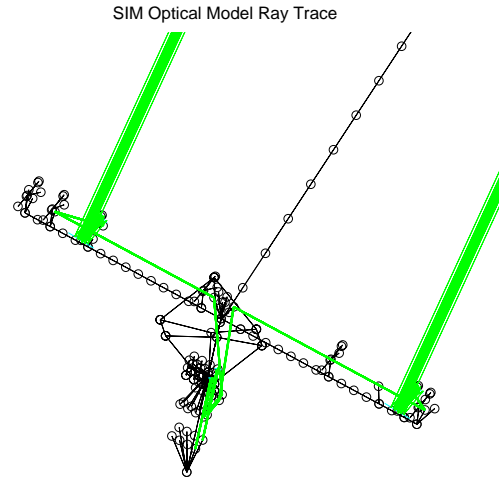
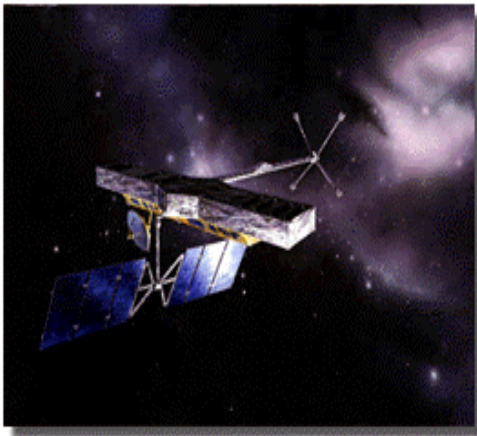
**Figure 3.15:** Left(a): Performance  $J_z$  (RMS OPD) as a function of  $K_d$  alone. Right(b): Root Locus for ODL sample problem (parameter:  $K_d$ ).

stability limit is confirmed via root locus, see Figure 2.15(b). The isoperformance contours provide engineering insight into the relationship between control gain  $K_d$  as well as the process and sensor noise intensities  $S_{dd}$  and  $S_{nn}$ , respectively.

### 3.4.3 SIM Classic (V1.0)

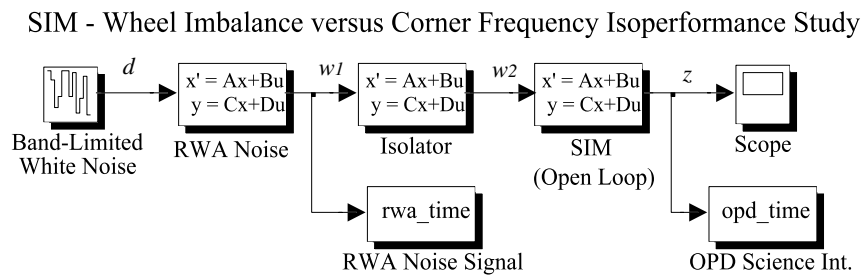
As an application of the bivariate isoperformance technique to a realistic spacecraft problem the Space Interferometry Mission (SIM - Version 1.0) was chosen. This is the same model used by Gutierrez in Reference [45] and in Chapter 5. The system considered consists of a state space representation of reaction wheel noise (assumptions: single wheel, fundamental harmonic only, only radial force  $F_y$ ), a mechanical RWA isolator and the open-loop SIM plant dynamics. The goal is to find the combination of isolator corner frequency  $p_1 = f_i$  in [Hz] and static wheel imbalance  $p_2 = U_s$  in units of [gcm] that will yield a specified RMS OPD performance  $J_z = \sigma_{z,req}$ .

The Space Interferometry Mission (SIM) is a cornerstone of the ORIGINS program and will rely on a fine-phasing capability in order to achieve 10 nm fringe-tracking on the science interferometer (closed loop) in the astrometry mode. This requirement has to be achieved in the presence of dynamic disturbances, such as reaction wheel noise. A picture of SIM Classic is shown in Figure 3.16(a), along with a ray-trace/FEM of SIM in Figure 3.16(b). A block diagram of the state space model for this isoperformance analysis is shown in Figure 3.17.



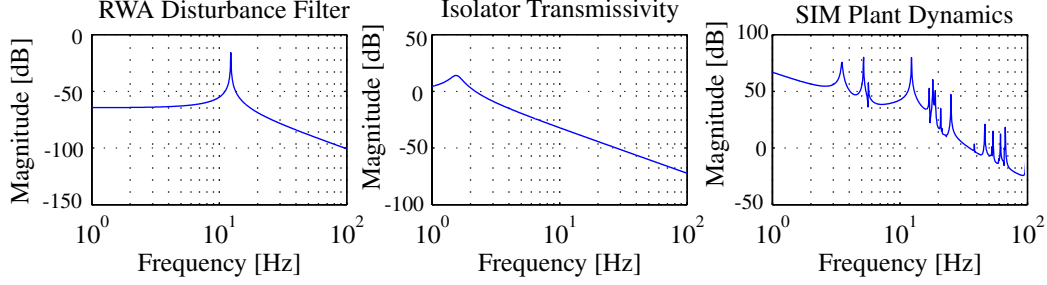
**Figure 3.16:** Left: (a) Artistic rendering of SIM Classic (V1.0). Right: (b) Corresponding FEM with ray trace for interferometer #1 (version 1.0).

The three blocks contain the dynamics of (from left to right) the RWA noise, the isolator transmissivity and the open loop plant dynamics. A realization of these dynamics (magnitude only) is shown in Figure 3.18. Note that the plant dynamics up to 100 Hz have been taken into account.



**Figure 3.17:** SIM Classic (version 1.0) Simulink block diagram for bivariate isoperformance problem

In the block diagram the first block represents the magnitude of the radial force ( $F_y$ ) generated by a single reaction wheel due to a static imbalance  $U_s$ , which expresses itself as the fundamental harmonic ( $h = 1.0$ ) at the frequency of the wheel rotation [89, 45]. Note that the radial force generated is related to the square of the wheel speed as  $F_y(t) = U_s \omega^2 \cdot \sin(\omega t + \phi)$  according to Bialke [7]. Only the fundamental harmonic is taken into account and a state



**Figure 3.18:** SIM Classic frequency response functions: (a) wheel disturbance, (b) mechanical isolator and (c) open-loop plant dynamics.

space representation of the wheel disturbance, in accordance with (A.11) is written as:

$$\begin{aligned}
 \begin{bmatrix} \dot{\eta} \\ \ddot{\eta} \end{bmatrix} &= \underbrace{\begin{bmatrix} 0 & 1 \\ -\omega_d^2 & -2\zeta_d\omega_d \end{bmatrix}}_{A_d} \begin{bmatrix} \eta \\ \dot{\eta} \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}_{B_d} d \\
 w_1 &= \underbrace{\begin{bmatrix} 2\zeta_d\omega_d^4 U_s & 0 \end{bmatrix}}_{C_d} \begin{bmatrix} \eta \\ \dot{\eta} \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \end{bmatrix}}_{D_d} d
 \end{aligned} \tag{3.41}$$

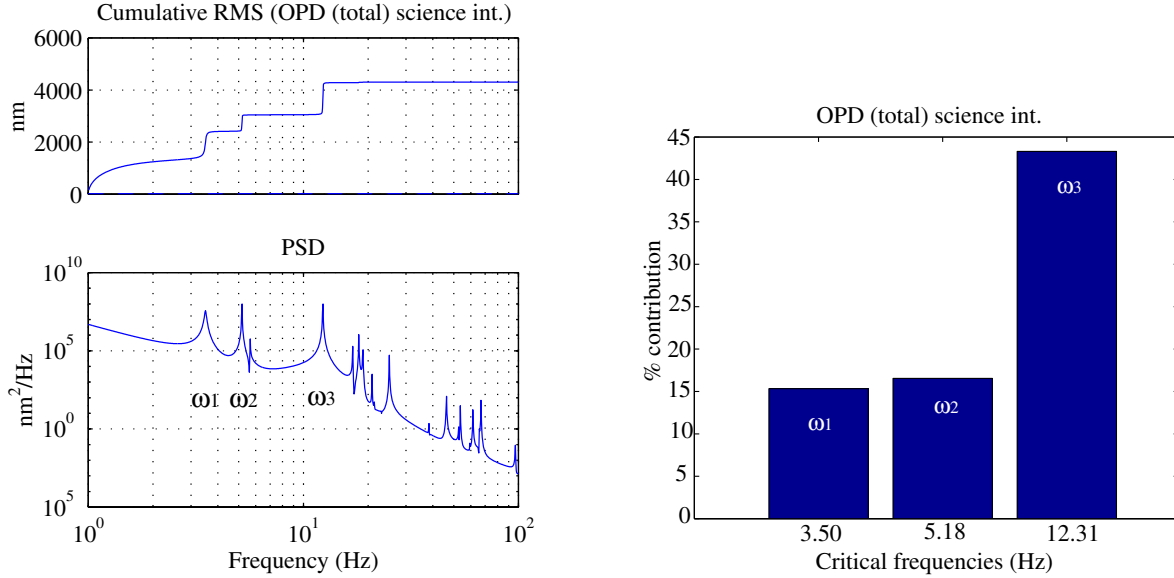
This representation approximates the fundamental harmonic of radial force as a lightly damped second order system with  $\zeta_d = 0.001$  and a disturbance magnitude, which is proportional to the second parameter  $p_2 = U_s$ . See the work by Uebelhart [136] for further details on state-space modeling of discrete harmonic disturbances. The wheel static imbalance is bounded below as  $U_{s, LB} = 0.01$  [gcm] and above as  $U_{s, UB} = 5$  [gcm]. These values are reasonable within current state-of-the-art wheel manufacturing practices [7].

The second block represents the isolator transmissivity. It is approximated by a second order low-pass-filter (LPF) and obeys the following state space representation:

$$\begin{aligned}
 \begin{bmatrix} \dot{\xi} \\ \ddot{\xi} \end{bmatrix} &= \underbrace{\begin{bmatrix} 0 & 1 \\ -\omega_i^2 & -2\zeta_i\omega_i \end{bmatrix}}_{A_i} \begin{bmatrix} \xi \\ \dot{\xi} \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}_{B_i} w_1 \\
 w_2 &= \underbrace{\begin{bmatrix} \omega_i^2 & 0 \end{bmatrix}}_{C_i} \begin{bmatrix} \xi \\ \dot{\xi} \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \end{bmatrix}}_{D_i} w_1
 \end{aligned} \tag{3.42}$$

Note that for the isolator a damping ratio  $\zeta_i$  of 5 % was assumed. The first variable parameter  $p_1$  is equal to  $f_i = \omega_i/2\pi$ . The isolator corner frequency is bounded below as  $f_{i,LB} = 0.5$  [Hz] and above as  $f_{i,UB} = 20$  [Hz].

The open-loop plant dynamics shown in the third block are kept constant throughout this analysis. The next step consists in deciding at which wheel speed [RPM] to run the reaction



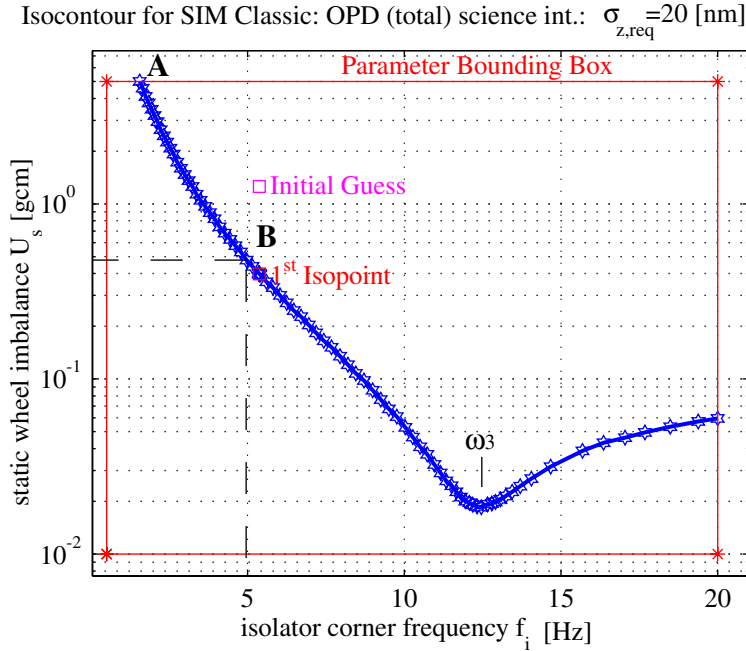
**Figure 3.19:** Left:(a) PSD-based disturbance analysis for SIM Classic (1.0). (bottom) PSD of OPD (total) int. # 1. (top) Cumulative RMS curve. Right: (b) Bar chart of relative % contribution of the three most critical modes to the RMS.

wheel. It was decided to run the wheel at a speed, which corresponds to most critical mode of the open loop plant, when driven by unit-intensity white noise. A PSD disturbance analysis, as discussed in Appendix A, was conducted on the open loop SIM Classic plant (version 1.0) and the result is shown in Figure 3.19(a). The bottom plot shows the output PSD (total OPD int #1) and the upper plot shows the cumulative RMS curve. The disturbance analysis assumes unit intensity white noise at the  $F_y$  RWA input channel. From the cumulative RMS curve one can extract the three most critical frequencies. This leads to plot 3.19(b).

Based on the relative percentage contribution of these modes it was decided to run the reaction wheel at 738.9 RPM, which provides a worst case amplification of the 12.31 Hz mode. The actual isoperformance analysis is entered with the wheel speed fixed at the above speed. A required performance of  $J_{z,req} = \sigma_{z,req} = 20$  [nm] was specified<sup>6</sup>. The variable

<sup>6</sup>Recall that the SIM model (V1.0) is open-loop, thus we don't expect the 10 nm phasing requirement to be

system parameters are  $p_1 = f_i$  the isolator corner frequency [Hz] and  $p_2 = U_s$ , the static wheel imbalance [gcm]. The isoperformance analysis was conducted using Algorithm II with a tolerance of  $\tau = 1.0\%$ . The resulting isoperformance contour is shown in Figure 3.20.



**Figure 3.20:** SIM Classic (version 1.0): Isoperformance analysis for OPD (total) interferometer # 1 fixed at 20 nm. Trading isolator corner frequency  $f_i$  vs. reaction wheel static imbalance  $U_s$ .

It is interesting to interpret the isoperformance contour somewhat closer. Three important insights can be gained:

- As isolator corner frequency  $f_i$  decreases we can tolerate larger values of wheel imbalance  $U_s$  below the critical 12.31 Hz mode.
- It is not advisable to choose the isolator corner frequency  $f_i$  to coincide with the critical system modes. The “dip” in the isoperformance contour results from dynamic amplification of the isolator around the critical 12.31 Hz mode.
- The isoperformance contour is useful for subsystem requirements definition purposes. If

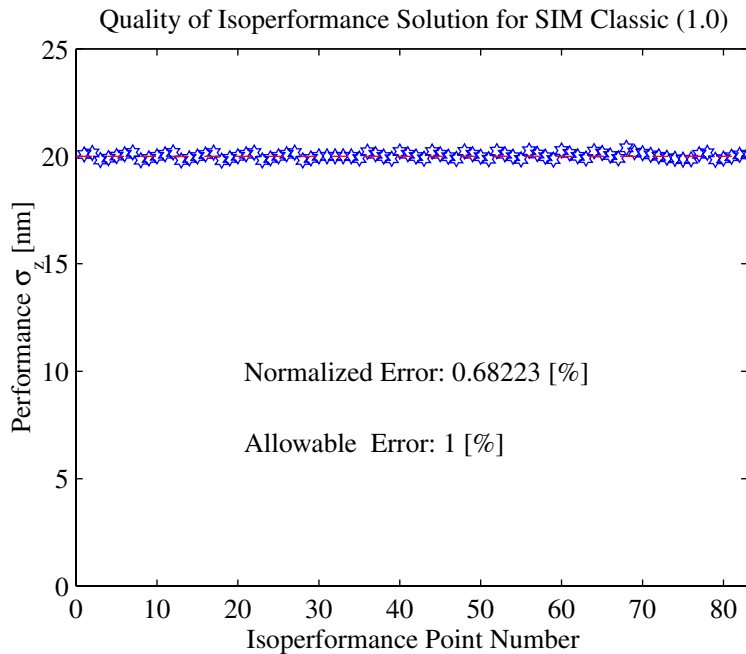
for example a 5 [Hz] corner frequency is specified for the isolator (achievable passively),  


---

 achievable in the absence of optical control.

one will levy a wheel imbalance specification of  $U_s < 0.4$  [gcm] to the reaction wheel supplier, see point B in Figure 3.20.<sup>7</sup>

The quality of the isoperformance solution in Figure 3.20 is within the specified  $\tau = 1.0$  % tolerance. Figure 3.21 shows the isoperformance quality plot, whereby the isoperformance points  $(\sigma_{z,iso})_k$  have been ordered in increasing value of the isolator corner frequency  $f_i$ .



**Figure 3.21:** SIM Classic (version 1.0): Isoperformance quality plot for  $\tau = 1.0$  % and the isoperformance solution shown in Figure 3.20.

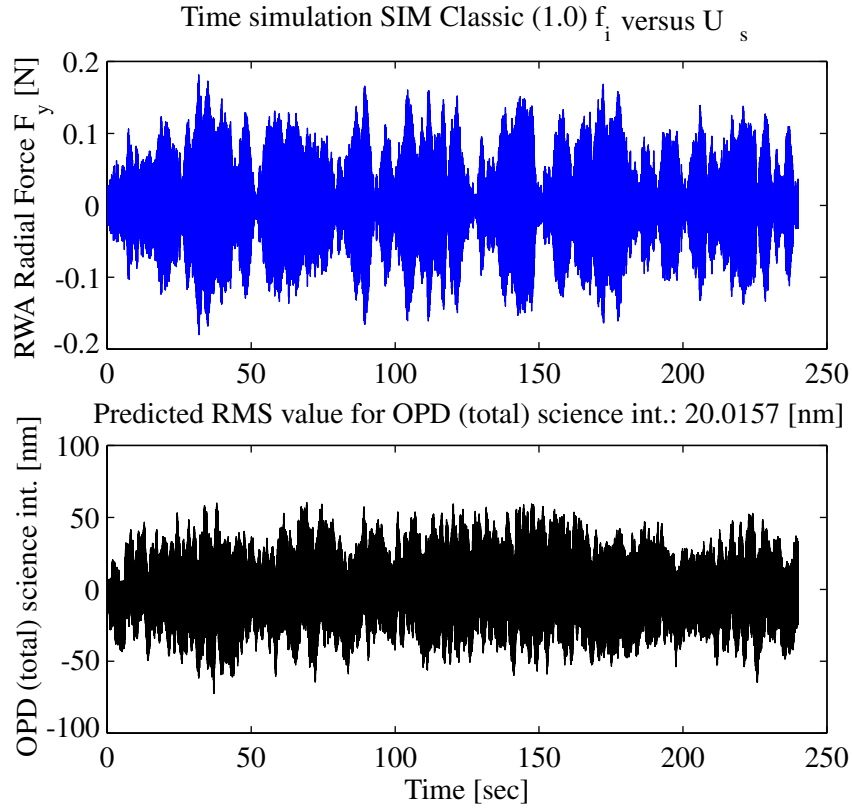
Upon closer inspection of this figure one notices the points at which the “control” algorithm, which keeps the isoperformance solution close to the  $\sigma_{z,req}$  centerline, has applied correction steps according to equation (3.31).

The computational expense of the isoperformance solution was  $6.185 \cdot 10^9$  floating point operations, which required a total of 121.34 seconds of CPU-time<sup>8</sup>. In order to verify the results obtained from the isoperformance analysis a particular solution (Point A in Figure 3.20:  $f_i = 1.5571$  [Hz] and  $U_s = 5.0$  [gcm]) was chosen and a Simulink time simulation was con-

<sup>7</sup>This number is not unreasonable, since e.g. the current ITHACO E-Wheel TW-50E300 specification calls for a static imbalance  $< 1.8$  [gcm] and Bialke [7] mentions that a factor 2-5 improvement of this figure is feasible with careful balancing.

<sup>8</sup>PC with Pentium III 500 MHz processor

ducted. The results are shown in Figure 3.22.



**Figure 3.22:** SIM Classic (v1.0): Time simulation (240 seconds) for  $f_i = 1.5571$  [Hz] and  $U_s = 5.0$  [gcm]. RMS value of  $z$  is 20.0157 [nm], which is  $\approx \sigma_{z,req}$ .

The top plot shows the sample realization of the radial reaction wheel force  $F_y$  [N] (this signal is denoted as “rwa\_time” in the block diagram of Figure 3.17), whereas the bottom plot shows the resulting OPD (total) for the science interferometer in [nm] (this signal is denoted as “opd\_time” in the block diagram of Figure 3.17). The RMS predicted for the OPD is obtained by taking the standard deviation of the “opd\_time” signal. The result is a value of 20.01 nm, which is very close to the requirement  $J_{z,req} = \sigma_{z,req}$ . This validates the accuracy of the isoperformance analysis. Note that the RWA force sample realization is dominated by the 12.3 Hz signal. There are, however, other frequency contributions, since the PSD of the radial force was modeled as a “lightly damped 2nd order system” and not as a discrete delta-Dirac function. This is why the sample realization “rwa\_time” does not appear to be a pure harmonic.



### 3.5 Summary

This chapter attempts to find solutions to the bivariate isoperformance problem. In other words, given a scalar performance function  $J_z(p_1, p_2)$ , we wanted to find contours of equal performance  $J_z = J_{z,req}$  in the trade space,  $\mathbf{B}$ . Three algorithms are developed for this purpose. The exhaustive search algorithm attempts to find contours of equal performance by laying a fine grid over the trade space  $\mathbf{B}$  in  $\mathbb{R}^2$  and evaluating all grid points. It obtains an estimate of the isoperformance contours via linear interpolation. The contour following algorithm begins with an initial guess,  $p_o$ , and intercepts an iso-contour via gradient search. Once a point,  $p_{iso}$ , on a contour is found, it is used as a starting point for a tangential stepping scheme along the open or closed contour. The most efficient, but also the most restrictive algorithm, appears to be vector spline approximation. Here an initial cubic spline approximation to the contour is refined by doing successive bisections until a tolerance (exit criterion) is met. The algorithms are applied to sample problem and their computational expense is compared. For a model of SIM Classic the isolator corner frequency,  $f_i$ , and the static wheel imbalance,  $U_s$ , are traded against each other and engineering insights are gained from the results.



## Chapter 4

# Multivariable Isoperformance

## Methodology

This chapter generalizes the algorithms developed in the previous chapter to the multivariable case. Specifically, there can be more than two variable parameters and multiple performances, i.e.  $n_p > 2$  and  $n_z > 1$ . The condition that the number of variable parameters always exceeds the number of performances,  $n_p - n_z > 1$ , has to be maintained in order for there to be a non-zero isoperformance set,  $\mathbf{I}$ . This condition is necessary but not sufficient. In addition the rank of the Jacobian  $\nabla J_z$  (containing the partial derivatives  $\partial J_{z,i}/\partial p_j$ ) must be such that the largest dimension of  $\nabla J_z$  minus its rank is greater than 1. In other words, two parameters that affect the performance in the same way effectively correspond to one parameter.

The generalization to the multivariable case is essential in order to render isoperformance a useful technique for realistic problems. There are two main challenges in the multivariable case:

- Computational complexity as a function of  $n_p$  (# of parameters),  $n_z$  (# of performances) and  $n_s$  (# of states)
- Visualization of isoperformance set  $\mathbf{I}$  in  $\mathbb{R}^{n_p}$ -space

The extensions of the three core isoperformance algorithms (exhaustive search, gradient-based contour following, progressive spline approximation) are developed in  $\mathbb{R}^{n_p}$ -space. The single degree-of-freedom (SDOF) problem is used for a quantitative comparison. The chap-

ter then introduces some basic notions in complexity theory and their relationship with the isoperformance algorithms. Issues of graphical representation of the isoperformance set,  $\mathbf{I}$ , for  $n_p > 3$  are discussed. Examples of multivariable isoperformance analyses for the sample problems introduced in Chapter 2 conclude the chapter. The MATLAB module, which embodies the multivariable algorithms is designated as `isoperf_mvar.m`.

## 4.1 Branch and Bound Search Algorithm

The exhaustive search algorithm (Ia) in the multivariable case ( $n_p > 2$ ) discretizes the parameters,  $p_j$ , where  $j = 1, 2, \dots, n_p$ , in set  $\mathbf{B}$ , defined by the lower and upper bounds  $p_{LB,j}$  and  $p_{UB,j}$ , with a fine grid and evaluates all grid points. This was presented for the case when  $n_p = 2$  in Subsection 3.2.1. Subsequently, each grid point is tested, and if the isoperformance condition (1.6) is met, the grid point is retained in the isoperformance set  $\mathbf{I}$ . The exhaustive search algorithm for the multivariable problem can be implemented as  $n_p$ -nested loops as follows:

```

for  $i_1 = 1 : n_1$ 
    :
    for  $i_{n_p} = 1 : n_{n_p}$ 
        evaluate performance:  $(J_z)_{i_1, \dots, i_{n_p}} = J_z(p_{1,i_1}, \dots, p_{n_p, i_{n_p}})$ 
    end
    :
end

```

where

$$n_j = \left\lceil \frac{p_{j,UB} - p_{j,LB}}{\Delta p_j} \right\rceil \quad (4.1)$$

is the number of grid points along the  $j$ -th parameter axis.

The code recognizes the number of variable parameters,  $n_p$ , and automatically writes and executes the above nested loops if exhaustive search is chosen. Note that the value of the  $j$ -th parameter in the above loops is given as

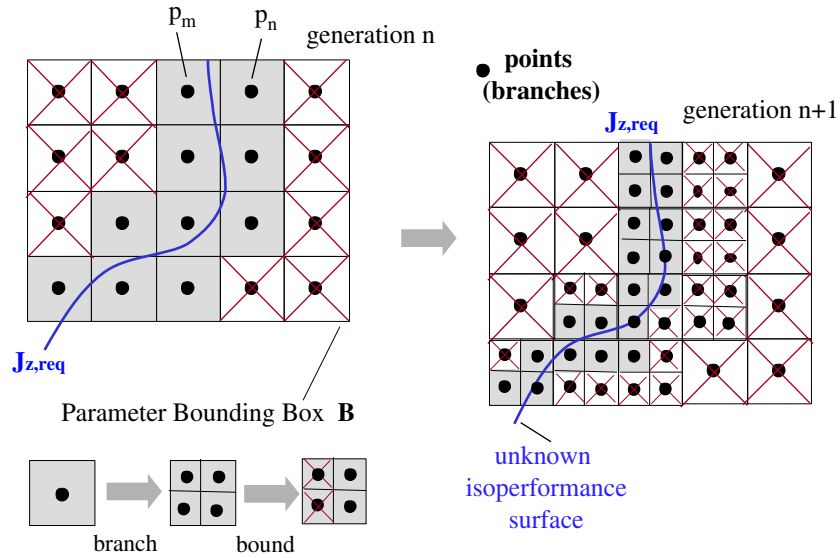
$$p_{j,i_j} = p_{j,LB} + (i_j - 1) \cdot \Delta p_j \quad \text{where} \quad j = 1, 2, \dots, n_p \quad (4.2)$$

Clearly, this is not practical even for relatively modest problems. Assume for example that  $n_p = 6$  and that  $n_1 = n_2 = \dots = n_{n_p} = 50$ , then the performance evaluation  $p_j \mapsto J_z$  has to be carried out  $50^6 = 1.56 \cdot 10^{10}$  times. If it took one second of CPU time per performance evaluation it would take 495.5 years to evaluate the entire trade space,  $\mathbf{B}$ , on a single computer.

A remedy is found by modifying exhaustive search as a branch-and-bound algorithm. The branch-and-bound algorithm starts with an initial population of points (branches), which are evenly distributed in  $\mathbf{B}$ . It then tests if the performance at neighboring points (branches),  $p_m$  and  $p_n$ , is such that the isoperformance surface passes in between them. This test is

$$[J_z(p_m) \geq J_{z,req} \geq J_z(p_n)] \cup [J_z(p_m) \leq J_{z,req} \leq J_z(p_n)] \quad (4.3)$$

where  $p_m, p_n$  are  $n_p \times 1$  vectors and  $J_{z,req}$  is a  $n_z \times 1$  vector. If the answer is **true**, both branches are retained and further refined in the next generation. If the answer is **false** the point (branch)  $p_m$  is eliminated. This is graphically shown in Figure 4.1 for two dimensions. In the multivariable case the squares shown in Figure 4.1 are actually hyper-rectangles. The



**Figure 4.1:** Multivariable Isoperformance (Ib): Branch-and-Bound graphic representation. Crossed out points are dropped in the next generation.

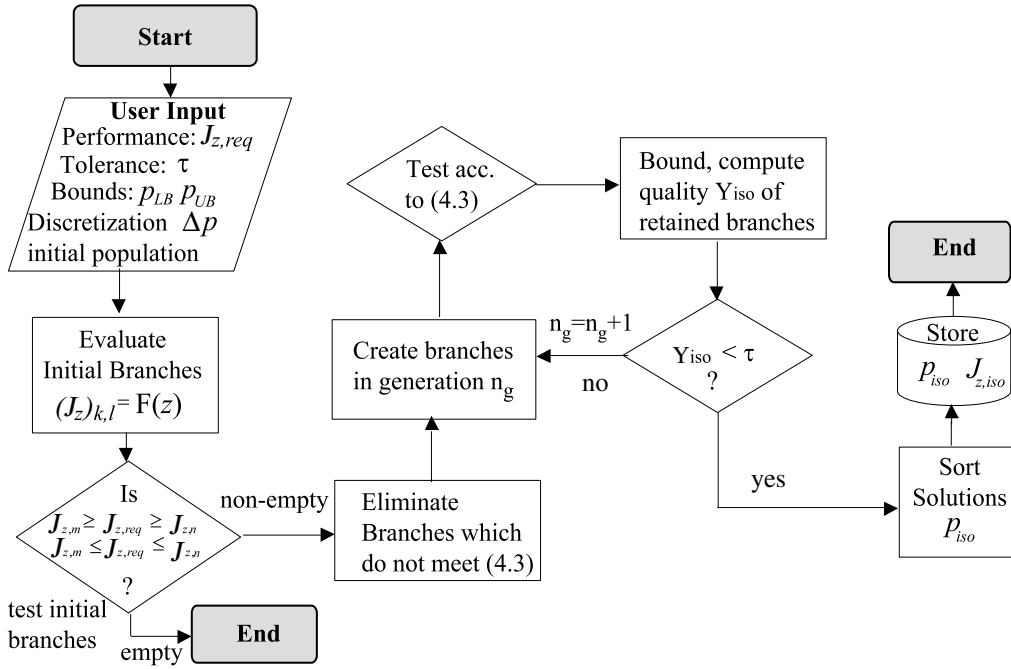
size of the hyper-rectangles is reduced by a factor of two (edge length) with each generation.

This refinement continues with each generation, until the exit criterion

$$\Upsilon_{iso,n_g} < \tau \quad (4.4)$$

is met, where  $\Upsilon_{iso}$  is the isoperformance error according to Equation (3.40),  $n_g$  is the generation number or level in the branch-and-bound tree and  $\tau$  is a numerical tolerance as introduced in Section 3.3.

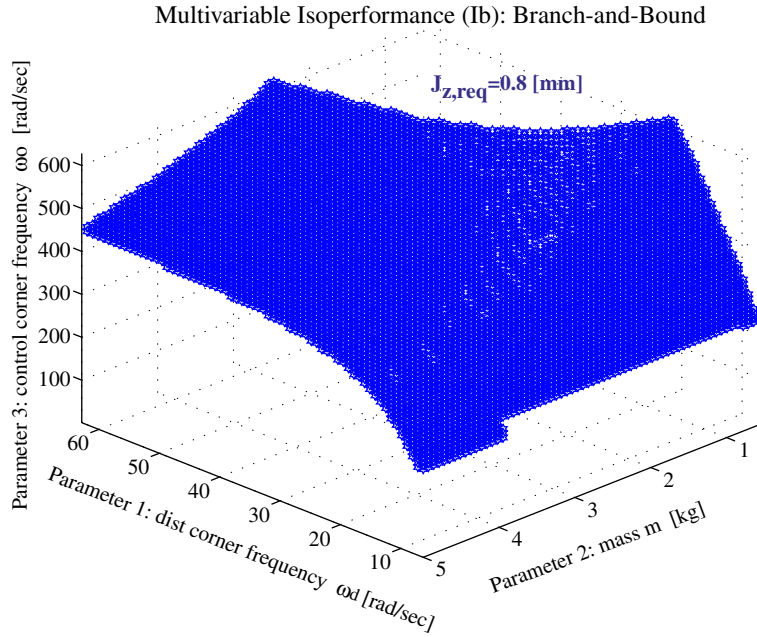
A flow diagram of the branch-and-bound algorithm is shown in Figure 4.2. An initial population of points is defined in terms of a coarse discretization  $n_j = \lceil \frac{p_{j,UB} - p_{j,LB}}{\Delta p_j} \rceil$ , where  $n_j$  is empirically chosen as a number in the range 5-20. Next the performance,  $J_z$ , at these initial points is computed and subjected to the test in Equation (4.3). The branches (points), which do not pass the test are bounded and the algorithm enters an iterative loop. With each subsequent generation,  $n_g$ , the remaining points better approximate the isoperformance surface in  $\mathbb{R}^{n_p}$  space. The branches are tested with the quality criterion (4.4), and once the exit criterion is met the remaining branches are considered to be the solutions,  $p_{iso}$ , of the isoperformance problem (1.3).



**Figure 4.2:** Multivariable Isoperformance (Ib): Flow Diagram for Branch-and-Bound Algorithm.

To demonstrate this algorithm the single DOF oscillator problem from Subsection 3.3.2 is augmented by adding a third variable parameter,  $\omega_o$ . The variable parameters are now the disturbance corner frequency,  $\omega_d$ , the mass,  $m$ , and the optical control corner frequency,  $\omega_o$ . The performance requirement is that the RMS of  $z$  be equal to  $J_z = 0.0008$  [m], corresponding

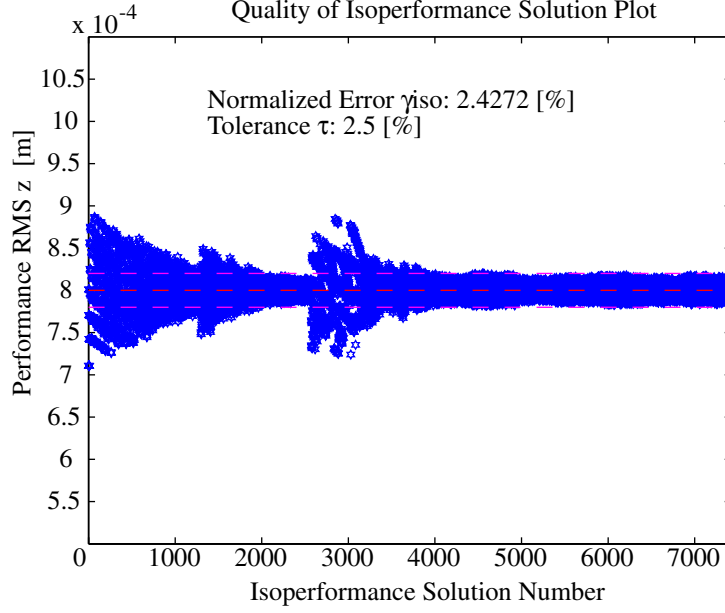
to 1 [mm] of filtered RMS motion with 20% performance margin, as before. The lower and upper bounds for the parameters are given in Figure 2.1. The results using the (multivariable) branch-and-bound algorithm are graphically shown in Figure 4.3 and the isoperformance set,  $p_{iso} \in \mathbf{I}$ , in this instance is a 2-dimensional surface in 3-dimensional space. The quality of this solution is computed with Equation (3.40) and shown in Figure 4.4.



**Figure 4.3:** Multivariable Isoperformance (Ib): Branch-and-Bound results for SDOF problem with three variable parameters:  $\omega_d$ ,  $m$  and  $\omega_o$  and required performance  $J_{z,req} = 0.8$  [mm].

From the results one can see that for a large disturbance corner frequency,  $\omega_d$ , a high control bandwidth,  $\omega_o$ , and large mass,  $m$ , are required to meet the requirement,  $J_{z,req}$ . Also it is noteworthy that if  $\omega_d$  is low, the isoperformance surface is not very dependent on the mass  $m$ , but strongly dependent on the control bandwidth  $\omega_o$ . The branch-and-bound algorithm finds a large number of isopoints  $p_{iso}$  ( $> 7000$ ) and approximates the continuous isoperformance surface quite well. The quality metric,  $\Upsilon_{iso}$ , remains under the numerical tolerance,  $\tau = 2.5\%$ .

It was noted that setting a tolerance tighter than 2% becomes very expensive, since in the branch and bound approach each generation is roughly  $2^{n_p}$  times larger than the previous generation. An advantage of the branch-and-bound algorithm, however, is that it does not require any sensitivity (gradient) information.



**Figure 4.4:** Multivariable Isoperformance (Ib): Quality of solution for SDOF example. Each one of the  $\sim 7000$  points corresponds to a point on the performance-invariant surface in Figure 4.3.

## 4.2 Tangential Front Following Algorithm

The tangential front following algorithm is a generalization of the gradient-based contour following algorithm, which was developed for the case when  $n_p - n_z = 1$ , see Subsection 3.2.2. The idea is to gradually explore the isoperformance set  $\mathbf{I}$ , starting from a random initial point,  $p_{nom}$ , and subsequently stepping in tangential, orthogonal directions,  $v_j$ , where  $j = n_z + 1, \dots, n_p$ , which lie in the null space of the Jacobian. Thus, instead of following a single curved line, the locus of performance invariant points,  $p_{iso}$ , will be an  $(n_p - n_z)$ -dimensional surface in  $\mathbb{R}^{n_p}$ -space as was shown in Figure 4.3.

A first order Taylor approximation of the vector performance function  $J_z$  at a point  $p^k = [p_1^k \ p_2^k \ \dots \ p_{n_p}^k]^T \in \mathbf{B}$  can be written as:

$$J_z(p^{k+1}) = J_z(p^k + \Delta p) = J_z(p^k) + \nabla J_z^T|_{p^k} \Delta p + \text{H.O.T.} \quad (4.5)$$

The Jacobian,  $\nabla J_z$ , is the matrix of first order partial derivatives of  $J_z$  with respect to  $p$  and



is written as follows:

$$\nabla J_z = \begin{bmatrix} \frac{\partial J_{z,1}}{\partial p_1} & \frac{\partial J_{z,2}}{\partial p_1} & \dots & \frac{\partial J_{z,n_z}}{\partial p_1} \\ \frac{\partial J_{z,1}}{\partial p_2} & \frac{\partial J_{z,2}}{\partial p_2} & \dots & \frac{\partial J_{z,n_z}}{\partial p_2} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial J_{z,1}}{\partial p_{n_p}} & \frac{\partial J_{z,2}}{\partial p_{n_p}} & \dots & \frac{\partial J_{z,n_z}}{\partial p_{n_p}} \end{bmatrix} \quad (4.6)$$

A singular value decomposition (SVD) of the Jacobian is a key step. It provides a set of orthogonal unit-length vectors,  $v_j$ , as the columns of matrix,  $V$ , thus forming the column space and null space of the Jacobian. See Reference [132] by Strang for a good introduction to vector spaces and the SVD.

$$U\Sigma V^T = \nabla J_z^T \quad (4.7)$$

The individual matrices are structured as follows:

$$\begin{aligned} U &= \underbrace{\begin{bmatrix} u_1 & \dots & u_{n_z} \end{bmatrix}}_{n_z \times n_z} \\ \Sigma &= \underbrace{\begin{bmatrix} \text{diag}(\sigma_1 \dots \sigma_{n_z}) & \mathbf{0}_{n_z \times (n_p - n_z)} \end{bmatrix}}_{n_z \times n_p} \\ V &= \underbrace{\begin{bmatrix} v_1 & \dots & v_{n_z} \end{bmatrix}}_{\text{column space}} \underbrace{\begin{bmatrix} v_{n_z+1} & \dots & v_{n_p} \end{bmatrix}}_{\text{null space}} \end{aligned} \quad (4.8)$$

Thus, at each point there are  $n_p - n_z$  directions in the null space. This can be illustrated with the following matrices from the SDOF example from Section 2.1, evaluated at  $p_{nom} = [\omega_d \ m \ k \ c \ \omega_o]^T = [19.92 \ , \ 1.45 \ , \ 1450 \ , \ 0.725 \ , \ 455.4]^T$ .

$$\begin{aligned} U &= [1] \\ \Sigma &= \begin{bmatrix} 0.56 \cdot 10^{-3} & 0 & 0 & 0 & 0 \end{bmatrix} \\ V &= \begin{bmatrix} 0.0511 & -0.9987 & 0 & 0 & 0 \\ -0.1361 & -0.0070 & -0.0004 & -0.9907 & -0.0031 \\ -0.0004 & -0.0000 & 1.0000 & -0.0003 & -0.0000 \\ -0.9894 & -0.0506 & -0.0003 & 0.1363 & -0.0027 \\ -0.0031 & -0.0002 & -0.0000 & -0.0027 & 1.0000 \end{bmatrix} \end{aligned} \quad (4.9)$$

$\underbrace{\hspace{10em}}_{V_n \text{ column space}} \quad \underbrace{\hspace{10em}}_{V_t \text{ null space}}$

Since there is only one performance metric,  $J_z = \text{RMS of } z(t)$ , there will only be one non-zero singular value,  $\sigma_1 = 0.56 \cdot 10^{-3}$ . The corresponding vector  $V_n$  is normal to the isoperformance surface and the vectors in  $V_t$  are tangential to it and form the nullspace of the Jacobian. A linear combination of the vectors,  $V_t$ , in the nullspace is used to determine a tangential step,  $\Delta p$ , in a performance invariant direction starting from an initial point such as  $p_{nom}$ .

$$\Delta p = \alpha \cdot (\beta_1 v_{n_z+1} + \dots + \beta_{n_p-n_z} v_{n_p}) = \alpha V_t \beta \quad (4.10)$$

where  $\Delta p$  is the performance invariant step increment in  $\mathbb{R}^{n_p}$ ,  $\beta = [\beta_1, \dots, \beta_{n_p}]^T$  is a vector of coefficients, which determines the linear combination of directions in the nullspace,  $V_t$ , and  $\alpha$  is a step size. Recall that in the bivariate case the step size,  $\alpha$ , was found using a finite difference approximation of the Hessian, see Equation (3.28). Currently, in the multivariable case, the step size,  $\alpha$ , is arbitrarily set by the designer. An automatic step size determination could be added as a refinement in the future. The coefficient vector,  $\beta$ , is determined as follows

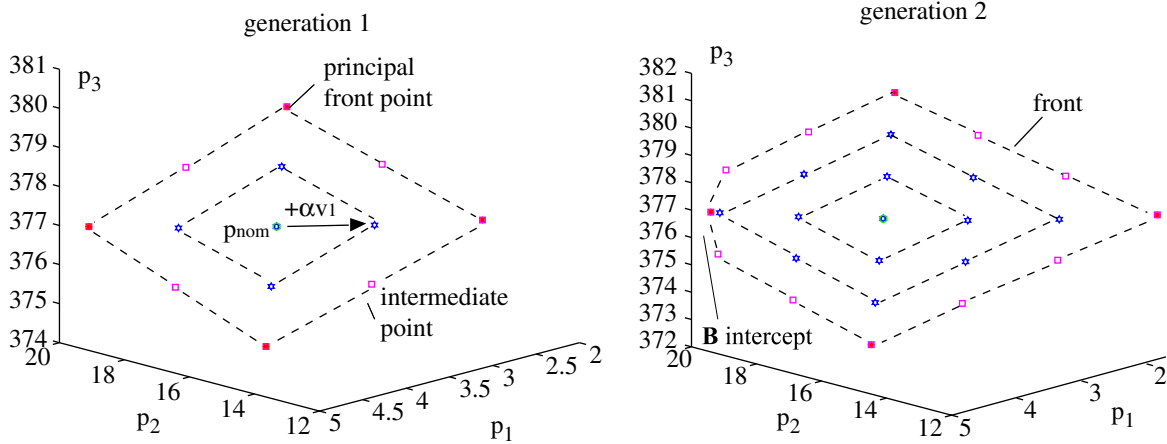
$$\beta = \begin{cases} \beta_i = \pm 1, \beta_j = 0 \text{ for } j \neq i & \text{(principal front points)} \\ \beta_i = \pm \frac{1}{\sqrt{n_p - n_z}} \quad \forall \quad i = 1, \dots, n_p - n_z & \text{(intermediate front points)} \end{cases} \quad (4.11)$$

The principal front points, as shown in Figure 4.5, propagate in one of the positive or negative directions given by the principal vectors,  $v_i$ , in the null space. The intermediate front points on the other hand propagate in directions which have equal contributions from all vectors in  $v_t$ . The  $\pm$  sign for each  $\beta_i$  determines in which “quadrant” the front point propagates.

The active points form a “front”, when connected to each other. The front grows gradually outwards from the initial point until the boundary is intercepted. This is analogous to “moss”, which grows from an initial seed to gradually cover the entire exposed surface of an imaginary  $\mathbb{R}^{n_p}$ -dimensional rock. This is shown graphically in Figure 4.5.

The main advantage of this algorithm, is that it converts the computational complexity from a  $n_p$  to a  $(n_p - n_z)$ -dimensional problem. This is further explored in Section 4.5. The disadvantage of the algorithm is that a non-uniform distribution of isoperformance points can result from the behavior of the Jacobian in different regions of the set  $\mathbf{B}$  or at the boundary of  $\mathbf{B}$ . Additionally if there are several separate regions of  $\mathbf{I}$  in  $\mathbf{B}$ , they can only be found via separate, random initial guesses. Nevertheless, tangential front following is a very reliable

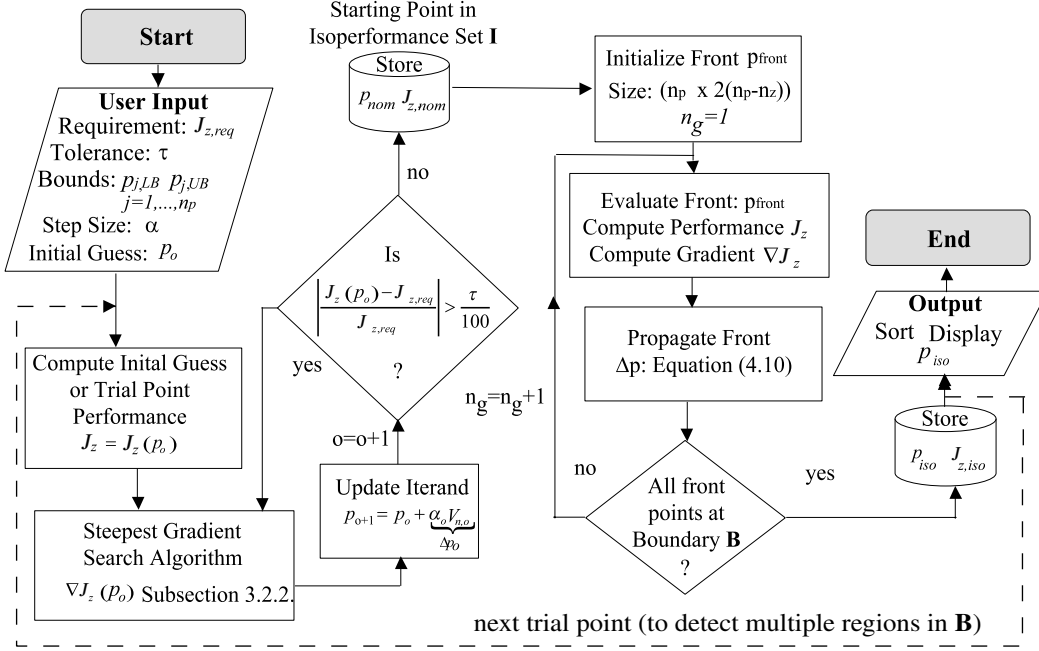
### Tangential Front Following Principle



**Figure 4.5:** Multivariable Isoperformance (II): Tangential Front Following principle. Note the difference between principal and intermediate front points. Axes  $p_1, p_2$  and  $p_3$  are values of some generic parameters.

algorithm, which can also detect closed regions. Note that the performance vector,  $J_z$ , and Jacobian,  $\nabla J_z$ , have to be recomputed at each front point. The underlying performance function,  $J_z(p_j)$ , has to be continuous and differentiable over the entire set  $\mathbf{B}$ . A flow diagram of the multivariable tangential front following algorithm is shown in Figure 4.6.

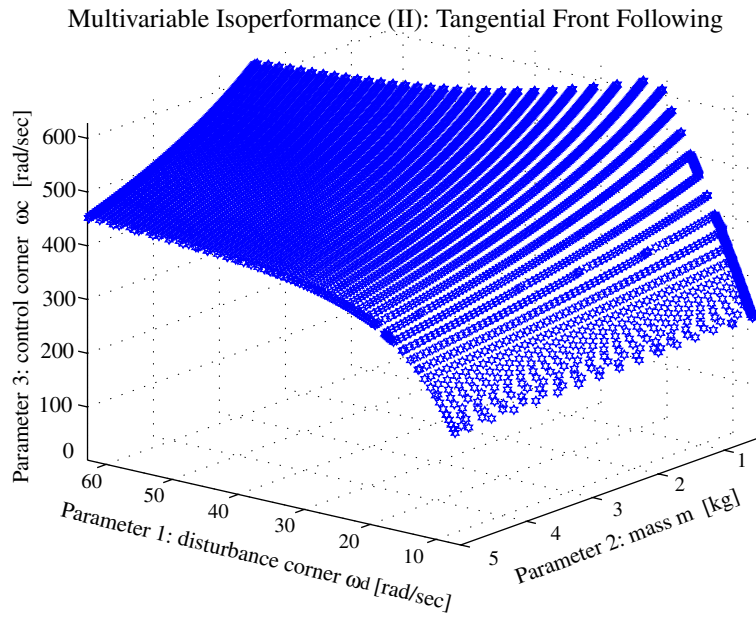
The results for the SDOF problem obtained with the tangential front following Algorithm are shown in Figure 4.7. The same three variable (design) parameters,  $\omega_d$ ,  $m$  and  $\omega_o$  as before are considered. By comparing Figures 4.3 and 4.7 one can ascertain that both algorithms produce the same isoperformance surface, apart from slight numerical and algorithmic differences. Tangential front following features characteristic “bands” of isoperformance points,  $p_{iso}$ , which are the result of the front points propagating along the principal directions,  $v_j$ , of the null space. The density of points is mainly a function of the user-defined step size,  $\alpha$ .



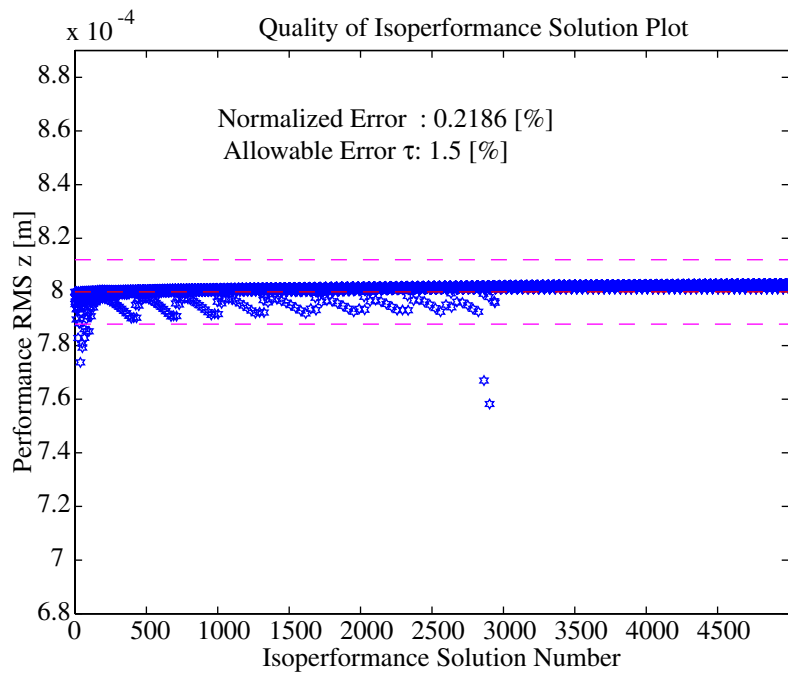
**Figure 4.6:** Tangential Front Following (II) Flow Diagram.

The quality of this solution is computed with Equation (3.40) and is shown in Figure 4.8. The ratio of the normalized error over the tolerance,  $\Upsilon_{iso}/\tau$ , is smaller for this algorithm than for branch-and-bound. This suggests that tangential front following delivers a numerically superior solution due to incorporation of gradient (sensitivity) information. Also the same “control” mechanism was implemented as in the bivariate case, where a normal correction increment is applied whenever a front point drifts outside the tolerance band defined by  $\tau$ . This does not exclude outlier solutions such as the pair seen in Figure 4.8 around solution number 2800.

Even though the tangential front following algorithm is more efficient than branch-and-bound, it will still be computationally expensive if the dimensionality of the isoperformance set  $\mathbf{I}$ , i.e.  $n_p - n_z$ , is large. An estimate of the computational expense of each algorithm is given in Section 4.5. Hence, it is desirable to find an algorithm with a further increase in efficiency.



**Figure 4.7:** Multivariable Isoperformance (II): Tangential Front Following results for SDOF problem with three variable parameters:  $\omega_d$ ,  $m$  and  $\omega_o$ .



**Figure 4.8:** Multivariable Isoperformance (II): Quality of solution

### 4.3 Vector Spline Approximation

An efficient algorithm is constructed by generalizing the bivariate progressive spline approximation from Subsection 3.2.3. The basic idea of vector spline approximation is to only capture important border and interior points of the isoperformance set,  $\mathbf{I}$ , including the cubic splines, which connect them. A  $t$ -parameterized vector spline in  $\mathbb{R}^{n_p}$ -dimensional space connecting two points A and B can be written as

$$p(t) = \begin{bmatrix} p_1(t) \\ p_j(t) \\ \vdots \\ p_{n_p}(t) \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^k \frac{(t-t_A)^{k-i}}{(k-i)!} \cdot c_{1,i} \\ \sum_{i=1}^k \frac{(t-t_A)^{k-i}}{(k-i)!} \cdot c_{j,i} \\ \vdots \\ \sum_{i=1}^k \frac{(t-t_A)^{k-i}}{(k-i)!} \cdot c_{n_p,i} \end{bmatrix} = C \cdot \hat{t} \quad (4.12)$$

where  $C$  is the vector spline coefficient matrix

$$C = \begin{bmatrix} c_{1,1} & \cdots & c_{1,i} & \cdots & c_{1,k} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ c_{j,1} & \cdots & c_{j,i} & \cdots & c_{j,k} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ c_{n_p,1} & \cdots & c_{n_p,i} & \cdots & c_{n_p,k} \end{bmatrix} \quad (4.13)$$

and  $\hat{t}$  is a vector, which depends on the parameter  $t$

$$\hat{t} = \left[ 1 \quad \cdots \quad \frac{(t-t_A)^{k-i}}{(k-i)!} \quad \cdots \quad \frac{(t-t_A)^{k-1}}{(k-1)!} \right]^T \quad (4.14)$$

whereby  $t \in [t_A, t_B]$  if the spline connects the points A and B in  $\mathbb{R}^{n_p}$ -space. The vector spline approximation algorithm presented here uses cubic splines of order,  $k = 4$ . One can then write:

$$\hat{t}(t) = \left[ 1 \quad t - t_A \quad \frac{(t-t_A)^2}{2} \quad \frac{(t-t_A)^3}{6} \right]^T \quad (4.15)$$

and the cubic spline coefficient matrix,  $C$ , simplifies to

$$C = \begin{bmatrix} c_{1,1} & c_{1,2} & c_{1,3} & c_{1,4} \\ \vdots & \vdots & \vdots & \vdots \\ c_{j,1} & c_{j,2} & c_{j,3} & c_{j,4} \\ \vdots & \vdots & \vdots & \vdots \\ c_{n_p,1} & c_{n_p,2} & c_{n_p,3} & c_{n_p,4} \end{bmatrix} \quad (4.16)$$

The first step of the vector spline approximation algorithm is to find the border points,  $p_{iso,border}$ , which meet the isoperformance condition (1.6) and lie on an edge of the parameter bounding box  $\mathbf{B}$ . These points are found by first computing the performance vector,  $J_z$ , at all  $2^{n_p}$  corner points and searching for boundary points,  $p_{iso,border}$ , which lie on an edge connecting two corner points, which meet the condition

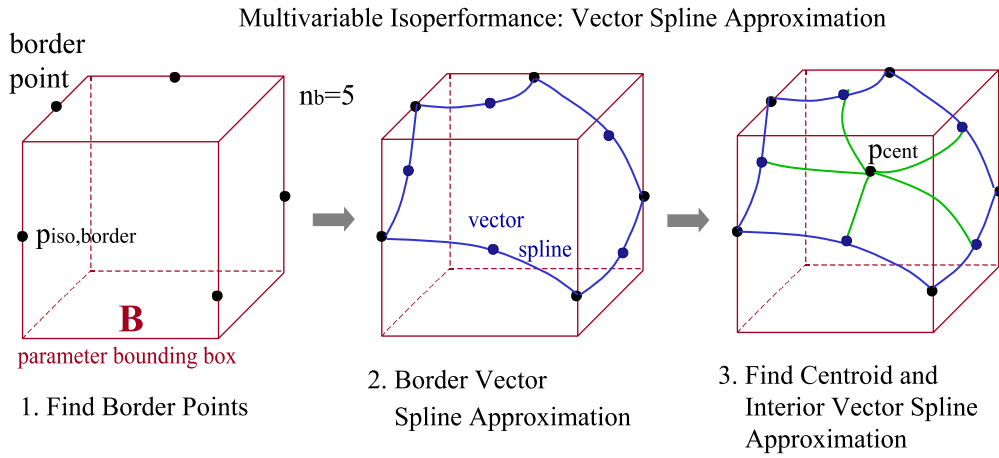
$$J_z(p_{corner,i}) \leq J_{z,req} \leq J_z(p_{corner,j}) \cup J_z(p_{corner,i}) \geq J_{z,req} \geq J_z(p_{corner,j}) \quad (4.17)$$

This is the first step in the vector spline approximation algorithm as shown on the left side of Figure 4.9. The second step is to connect the isoperformance border points with cubic splines along the boundary of  $\mathbf{B}$ . This is shown in the middle of Figure 4.9 and utilizes the progressive spline approximation method presented in Subsection 3.2.3. In this step the mid-points of the border splines are also determined. Finally interior points of the isoperformance set  $\mathbf{I}$  are obtained by computing the *centroid*. This can be considered to be the center point of  $\mathbf{I}$ . An initial guess for the centroid is:

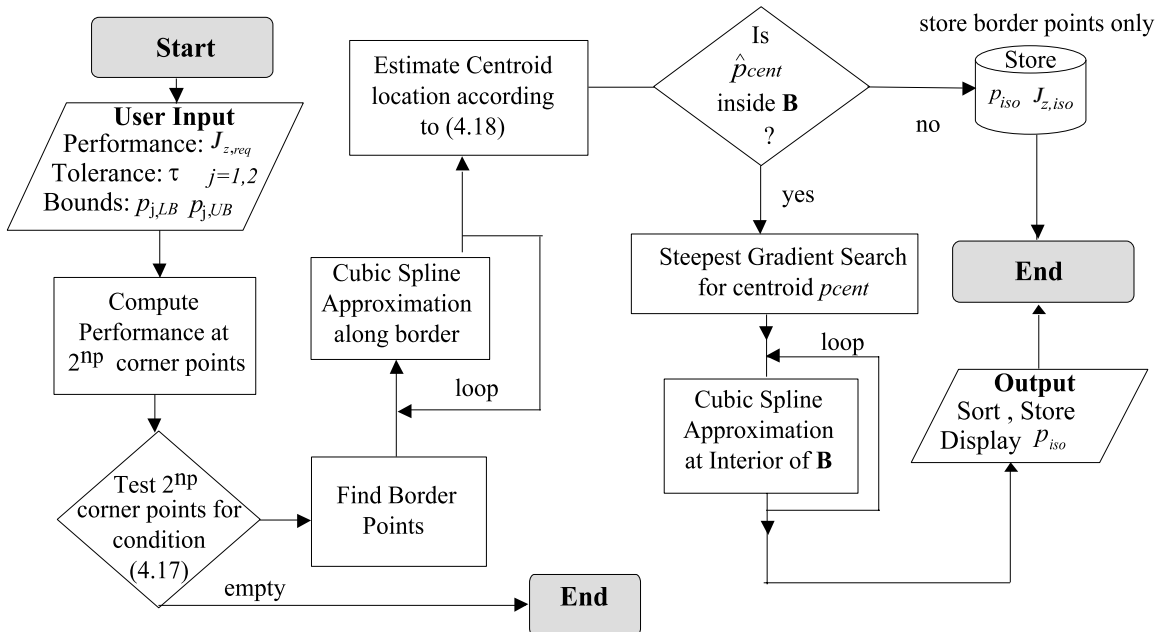
$$\hat{p}_{cent} = \begin{bmatrix} \hat{p}_{c,1} & \cdots & \hat{p}_{c,j} & \cdots & \hat{p}_{c,n_p} \end{bmatrix}^T \quad (4.18)$$

where  $\hat{p}_{c,j} = \frac{1}{n_b} \sum_{i=1}^{n_b} p_{iso,border,i,j}$

and  $n_b$  is the number of border points. The actual centroid,  $p_{cent}$ , is found by steepest gradient search, see Subsection 3.2.2. Finally the cubic splines connecting the centroid and the mid-points of the border splines are found, subject to tolerance,  $\tau$ . This third step is depicted in Figure 4.9 on the right side. The functions `csape.m` and `fnval.m` from the MATLAB spline toolbox are used extensively for computing the coefficient matrix,  $C$ , and for spline evaluation, respectively. Figure 4.9 shows the elements of this approach for an imaginary isoperformance surface. The vector spline approximation algorithm is summarized in the flow diagram of Figure 4.10.



**Figure 4.9:** Multivariable Isoperformance (III): Vector Spline Approximation Principle. Boundary splines in dark (blue) and interior splines in light (green). Centroid is  $p_{cent}$ .

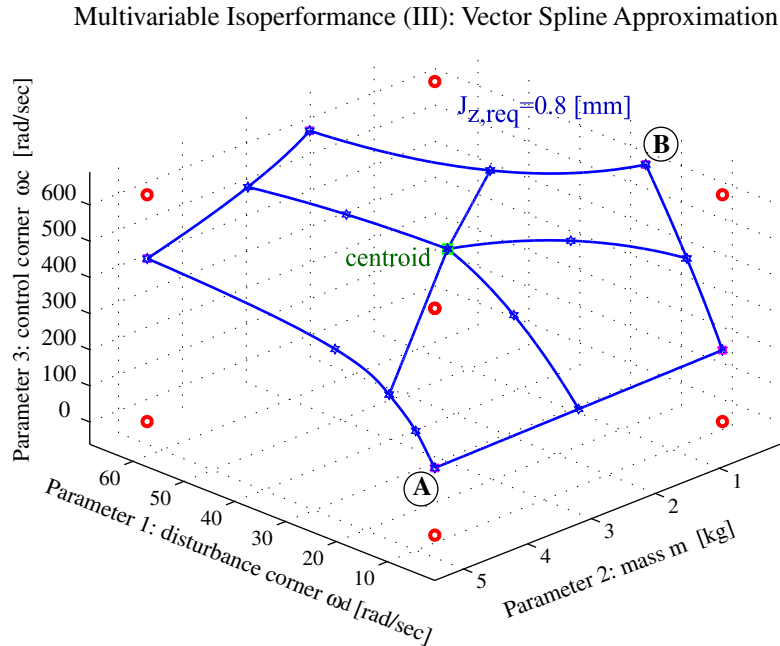


**Figure 4.10:** Multivariable Isoperformance (III): Flow Diagram



The vector spline approximation algorithm does not provide the same large number of isoperformance points,  $p_{iso}$ , and “continuous” approximation to  $\mathbf{I}$  as branch-and-bound or tangential front following. Rather, it only computes some key points and their connecting splines. If the isoperformance set  $\mathbf{I}$  were an imaginary “house” this algorithm provides the frame, but not the panels in between. This might be acceptable, since one of the goals of the isoperformance methodology is to find solutions which are very “different” in a design vector sense, while still yielding the same performance vector,  $J_z$ . Also it is true that, for the front following algorithm, with a small tangential step size  $\alpha$ , neighboring points,  $p_{iso}$ , will not be dramatically different from each other. An early idea was to use Product Tensor Splines [22]. This was ultimately abandoned, since product tensor splines require gridded data.

The same three parameters,  $\omega_d$ ,  $m$  and  $\omega_o$  are considered for the SDOF problem. The desired performance is  $J_{z,req} = 0.8$  [mm] RMS. Results for the single DOF oscillator problem using vector spline approximation are shown in Figure 4.11.

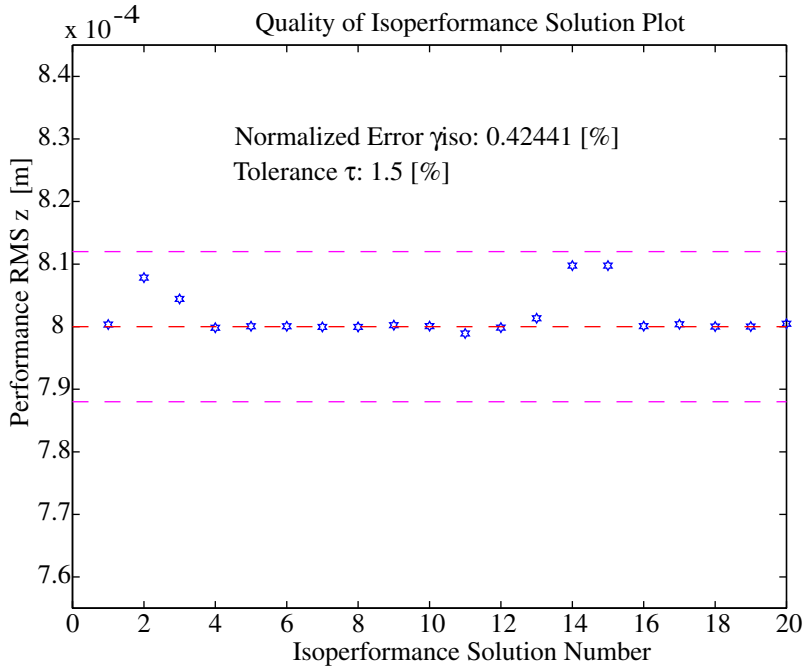


**Figure 4.11:** Multivariable Isoperformance (III): Vector Spline Approximation for SDOF Problem and performance level  $J_{z,req} = 0.8$  [mm].

There are a total of five border points, which are connected via cubic splines along the perimeter of  $\mathbf{B}$ . The centroid connects to the mid-points of the border spline approximations. This looks similar to Figures 4.3 and 4.7, since the outline of the isoperformance surface can

clearly be seen.

The solution quality is plotted in Figure 4.12 for the isoperformance points. Note that additional information is obtained in the form of a spline coefficient matrix,  $C$ , for each segment. It is expected that this algorithm will be the “workhorse” of the isoperformance methodology for large order problems.



**Figure 4.12:** Multivariable Isoperformance (III): Quality of solution

## 4.4 Algorithm Comparison

A comparison of the multivariable algorithms using the single degree-of-freedom (SDOF) problem is presented in Table 4.1. The algorithms are compared based on the CPU run time, the number of floating-point operations required, the solution quality expressed as  $\Upsilon_{iso}$ , see Equation (3.40), and the number (quantity) of isoperformance points,  $p_{iso}$ , found.

Even though the above numbers are obtained for a specific low-order example, the relative trends between algorithms are likely to apply to large-order problems as well. As expected the exhaustive search is the most expensive algorithm and requires almost 1.5 hours to run. The vector spline approximation on the other hand completes in merely 5 seconds.

Branch-and-bound improves over exhaustive search by a factor of roughly 10 and tangen-

**Table 4.1:** Comparison of multivariable algorithms for SDOF problem: (Ia) Exhaustive Search, (Ib) Branch-and-Bound, (II) Tangential Front Following and (III) Vector Spline Approximation.

Metric	Algorithm Ia	Algorithm Ib	Algorithm II	Algorithm III
MFLOPS	6,163.72	891.35	106.04	1.49
CPU time [sec]	5078.19	498.56	69.59	4.45
Tolerance $\tau$	1.5 %	2.5 %	1.5 %	1.5%
Actual Error $\Upsilon_{iso}$	0.87 %	2.43 %	0.22 %	0.42 %
# of isopoints	2073	7421	4999	20
Results	N/A	Fig. 4.3	Fig. 4.7	Fig. 4.11

tial front following in turn improves over branch-and-bound by a factor of roughly 7. The tangential front following algorithm results in the best numerical solution quality as measured by,  $\Upsilon_{iso}$ . Branch-and-bound provides the largest number of isopoints ( $\sim 7500$ ), whereas vector spline approximation yields “only” 20 such points. Recall, however, that the spline approximation also provides the spline coefficient matrices, such that additional points could be easily generated along the connecting splines. These results suggest that vector spline approximation is the most advantageous algorithm, provided the “sparseness” of the results is acceptable.

Vector spline approximation is the most restrictive algorithm in the sense that it requires the underlying performance vector function,  $p_j \mapsto J_z(p_j)$ , where  $p_j = 1, \dots, n_p$ , to be continuous, smooth, differentiable and quasi-monotonic in  $\mathbf{B}$ . Thus, if  $\mathbf{I}$  were a closed region with no boundary points on  $\mathbf{B}$ , the vector spline approximation would fail. Tangential front following does not require quasi-monotony and can deal with closed regions. Here the problem is that if  $\mathbf{I}$  consists of several, distinct regions in  $\mathbf{B}$  the algorithm requires several random initial guesses,  $p_o$ , in order to find all regions. There is no guarantee of completeness with a finite number of trial points. Distinct regions are rarely observed in practice. Finally branch-and-bound is the most general algorithm and is very robust, as long as the initial grid is chosen reasonably fine. Another advantage of branch-and-bound is that it does not require gradient (sensitivity)

information. The general strategy is to first attempt an isoperformance solution with vector spline approximation and move to the other, more expensive algorithms if a solution in  $\mathbf{B}$  is expected to exist but cannot be found. This algorithm switching strategy was suggested in the thesis roadmap, see Figure 1.6.

## 4.5 Complexity Theory

The previous section revealed that there are large differences in the computational expense of an isoperformance problem as a function of the selected algorithm. This has implications for the approach to large order problems. Hence, some basic notions of complexity theory and how it relates to isoperformance will be introduced in this section. Complexity theory, according to Goldreich, is concerned with the study of the *intrinsic* complexity of computational tasks and the relationship with limited resources [37]. One of the aims of complexity theory is to establish concrete lower bounds on the complexity of various kinds of problems, via an analysis of the evolution of the process of computation. This section will attempt to quantify the complexity of the isoperformance problem. This is done by looking at the asymptotic growth of the number of floating point operations required as a function of the number of performances,  $n_z$ , the number of disturbances,  $n_d$ , the number of parameters,  $n_p$ , and the number of states,  $n_s$ .

The different types of problems in complexity theory are: the *search* problem, i.e., finding a feasible solution, the *decision* problem, i.e., determining whether a feasible solution exists, and the *verification* problem, i.e. deciding whether a given solution is correct. The isoperformance problem is essentially a *search* problem, where one attempts to find solutions  $p_j$  to the problem  $J_z(p_j) - J_{z,req} = 0$ , where  $j = 1, 2, \dots, n_p$ .

### 4.5.1 Exhaustive Search

The exhaustive search approach requires the following number of performance function  $J_z(p_j)$  evaluations:

$$N_{exs} = \prod_{j=1}^{n_p} \left\lceil \frac{p_{UB,j} - p_{LB,j}}{\Delta p_j} \right\rceil \quad (4.19)$$

where  $N_{exs}$  is the number of performance function evaluations,  $n_p$  is the number of variable parameters,  $p_{LB,j}$  and  $p_{UB,j}$  are the lower and upper bounds of the  $j$ -th parameter and  $\Delta p_j$  is the discretization step size of the  $j$ -th parameter. If all parameters are discretized with an

equal number of steps, Equation (4.19) simplifies to:

$$N_{exs} = \underbrace{\left[ \frac{p_{UB} - p_{LB}}{\Delta p} \right]}_{\delta}^{n_p} \quad (4.20)$$

As an approximation we can assume that the main computational cost for computing  $J_z$  comes from solving the Lyapunov equation (A.25) for the state covariance  $\Sigma_q$ . Chapter 5 empirically derives that this cost is roughly  $50 \cdot n_s^3$  floating point operations. Thus the expected number of floating point operations (FLOPS) for exhaustive search is

$$J_{exs} = \delta^{n_p} \cdot 50n_s^3 \quad (4.21)$$

Looking at the asymptotic growth of the algorithm can be accomplished by taking the log of (4.21) such that

$$\log(J_{exs}) = n_p \log(\delta) + 3 \log(n_s) + \text{const.} \quad (4.22)$$

Thus, exhaustive search is solvable in polynomial time as a function of  $n_s$  (“size of the model”), but it is non-polynomial (NP) in  $n_p$  (“number of variable parameters”).

## 4.5.2 Branch-and-Bound

The complexity of branch-and-bound, as developed in Section 4.1, is more difficult to assess than exhaustive search, since the number of branches kept at each generation is problem dependent. One could, however, assume that the ratio,  $\beta$ , of kept to discarded branches is roughly constant at each generation. If the the initial generation has  $n_1$  branches, then the second generation will have  $2^{n_p} \cdot n_1 \cdot \beta$  branches after bounding, the third will have  $2^{2n_p} n_1 \beta^2$  branches and so forth. The computational cost for branch-and-bound is then approximated as

$$J_{bab} = 2^{n_g n_p} \cdot n_1 \cdot \beta^{n_g} \cdot 50n_s^3 \quad (4.23)$$

Again, taking the logarithm (base 10) provides insight into the asymptotic behavior.

$$\log(J_{bab}) = n_g (n_p \log 2 + \log \beta) + 3 \log(n_s) + \text{const.} \quad (4.24)$$

Note that the number of generations,  $n_g$ , is difficult to predict apriori but is strongly dependent on the isoperformance tolerance,  $\tau$ . The smaller the tolerance,  $\tau$ , the more generations will be required to meet the exit criterion (4.4). A small value for  $\tau$ , say smaller than 1%, can actually render the branch-and-bound algorithm more expensive than exhaustive search, since each generation is roughly  $2^{n_p}$  times more expensive than the previous one.

### 4.5.3 Tangential Front Following

The tangential front following computational cost can be estimated by considering that at each point we must compute the performance,  $J_z$ , and the Jacobian,  $\nabla J_z$ . This requires  $(1 + n_z)$  Lyapunov equations each costing  $50n_s^3$  floating point operations. The number of performances,  $n_z$ , appears in this expression, since a separate Lagrange multiplier matrix,  $L_i$ , must be solved for each performance metric according to (A.31). The number of directions in the nullspace is  $n_p - n_z$ . The distance before reaching the parameter boundary  $\mathbf{B}$  from a starting point such as,  $p_{nom}$ , can be approximated as some constant,  $\gamma$ , which depends on the parameter bounds,  $p_{LB}, p_{UB}$  and the step size,  $\alpha$ . The cost of tangential front following is thus approximated as

$$J_{tff} = \gamma^{n_p - n_z} \cdot (1 + n_z) \cdot 50n_s^3 \quad (4.25)$$

The logarithmic cost is

$$\log(J_{tff}) = (n_p - n_z) \log \gamma + \log(1 + n_z) + 3 \log(n_s) + \text{const.} \quad (4.26)$$

### 4.5.4 Vector Spline Approximation

The first step in the vector spline approximation algorithm is to compute the performance at all  $2^{n_p}$  corner points, see Section 4.3. These boundary points have to be connected with vector splines at an additional cost. An approximate expression for the cost of vector spline approximation, using constants where appropriate, is

$$J_{vsa} = 2^{n_p} \cdot 50n_s^3 + (1 + n_z) \cdot 50n_s^3 \quad (4.27)$$

The logarithmic cost is

$$\log(J_{vsa}) = n_p \log 2 + \log(1 + n_z) + 3 \log(n_s) + \text{const.} \quad (4.28)$$

From Equations (4.22), (4.24), (4.26) and (4.28) one can see that the isoperformance problem is intrinsically non-polynomial (NP) in  $n_p$ . There are, however, some noticeable differences between algorithms. The benefit of tangential front following is that it reduces the logarithmic asymptote from  $n_p$  to  $n_p - n_z$ . This means that solving a problem with 105 parameters and 101 performance metrics has the “same” theoretical cost as solving a problem with 5 parameters and a single performance. The actual number of floating point operations (FLOPS) required is problem dependent. There is no doubt that isoperformance problems

with more than  $\sim 10$  parameters are still quite expensive to solve. This is true despite the contributions of Chapter 5, which deals mainly with large  $n_s$ , i.e. a large number of states in the model.

## 4.6 Graphical Representation of Results

The second challenge in the multivariable case is the graphical portrayal of isoperformance results. This is an important issue, since one use of isoperformance “charts” is as a tradeoff and negotiating tool. Graphs showing trades between “key” design parameters help to decide design questions involving multiple design parameters, possibly “owned” by different subsystem teams. The two fundamental ways of showing isoperformance results in this thesis are contour plots and radar plots.

### 4.6.1 Contour Plots

Contour plots are the most common way of showing isoperformance results. These plots can be constructed whenever  $n_p \leq 3$ . Possible scenarios are one-dimensional contours in 2-dimensional space, see Figure 3.10, one-dimensional contours in 3-dimensional space, or 2-dimensional contours in 3-dimensional space, see Figure 4.11. A potential technique for showing multivariable relationships is nomography, as discussed in Reference [5]. In nomography, contour charts in multiple dimensions can be constructed by using auxiliary variables, thus linking 2-dimensional charts together.

### 4.6.2 Radar Plots

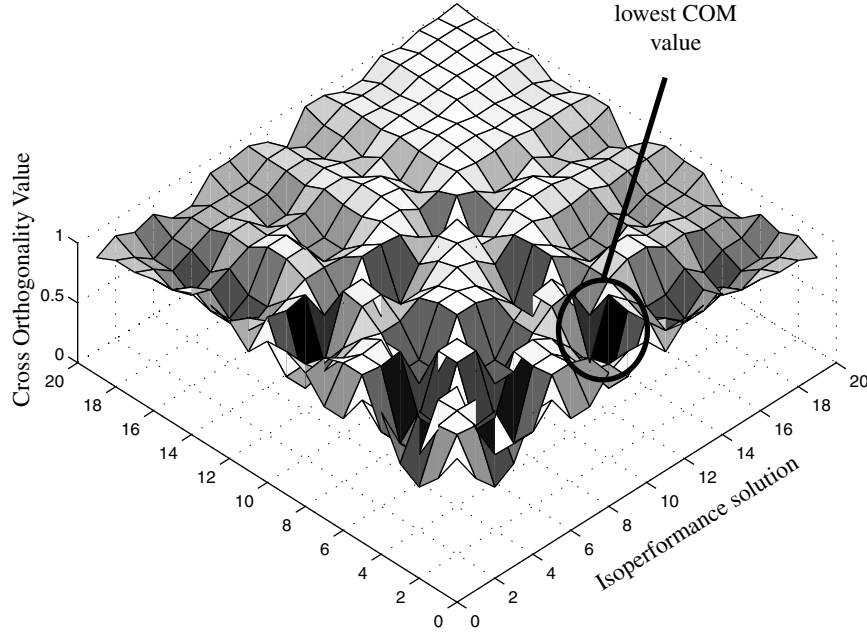
Once there are more than 5-6 variable parameters,  $p_j$ , where  $j = 1, 2, \dots, n_p$ , even nomograms become impractical and it is no longer feasible to effectively graphically represent all vector solutions,  $p_{iso}$ , which are members of the isoperformance set. An alternative is to select a subset of “interesting” solutions for display to the designer. An interesting subset could be a set of vectors,  $p_{iso}$ , which are most different from each other. “Different” in a vector sense is related to the notion of orthogonality. This leads naturally to the concept of a cross-orthogonality matrix<sup>1</sup>. The cross-orthogonality matrix (COM) is defined as:

---

<sup>1</sup>This is similar to the matrix used for the modal assurance criterion (MAC) in structural dynamics

$$COM(i, j) = \frac{\tilde{p}_{iso,i} \cdot \tilde{p}_{iso,j}}{|\tilde{p}_{iso,i}| \cdot |\tilde{p}_{iso,j}|} \quad (4.29)$$

Note that the vectors  $\tilde{p}_{iso,i}$  and  $\tilde{p}_{iso,j}$  have been normalized to their upper bound,  $p_{UB}$ . A pair of performance invariant solutions  $i, j$  which has a low COM value, possibly close to zero, represents a pair of nearly orthogonal design vectors. Note that the COM  $i, j$  values will always be between 0 and 1. The COM matrix for the single DOF problem solution using vector spline approximation (Section 4.3) is shown graphically in Figure 4.13.



**Figure 4.13:** Cross-orthogonality matrix (COM) for solution to SDOF problem with Algorithm III (Tangential Front Following).

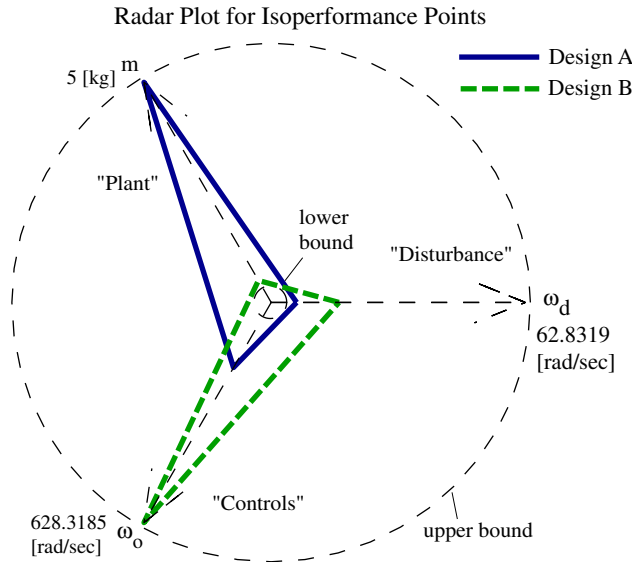
From the COM an arbitrary number of combinations,  $p_{iso}$ , corresponding to low cross-orthogonality can be extracted. In the above example the minimum COM value is 0.4325 and is found for  $i = 3$  and  $j = 11$ . The two vectors,  $p_{iso,3}$  and  $p_{iso,11}$ , are both members of the isopformance set. The corresponding values of the design parameters are:

$$p_{iso,3} = \begin{bmatrix} 6.2832 \\ 5.0000 \\ 186.5751 \end{bmatrix} \quad \text{and} \quad p_{iso,11} = \begin{bmatrix} 21.3705 \\ 0.5000 \\ 628.3185 \end{bmatrix} \quad \begin{matrix} \omega_d [\text{rad/sec}] \\ m [\text{kg}] \\ \omega_o [\text{rad/sec}] \end{matrix} \quad (4.30)$$



These two solutions are marked as “A” ( $p_{iso,3}$ ) and “B” ( $p_{iso,11}$ ) in the isoperformance chart on Figure 4.11. As can be seen, both are on the boundary of the trade space  $\mathbf{B}$ . This is not always the case. In Figure 3.14(b) the design solutions which are most different from each other include the design point in the interior of  $\mathbf{B}$  with  $S_{bb} = 10^{-6}$  [m/ $\sqrt{Hz}$ ] and  $K_d = 0.4$  [V/V]. The two solutions “A” and “B” are interesting, since the first one,  $p_{iso,3}$ , achieves the required RMS performance mainly passively, by specifying a low disturbance environment (small  $\omega_d$ ), large mass  $m$  and low optical control bandwidth  $\omega_o$ . The second solution, however, could be termed the “active” solution since it features a larger disturbance input, is light-weight and relies mainly on a large optical control bandwidth,  $\omega_o$ . Both solutions nominally produce the same RMS performance of 0.008 [m]. In order to distinguish between these options further additional objective functions have to be introduced. This is the subject of Section 4.7.

The two solutions above can be graphically shown in a so called radar plot, see Figure 4.14. The radial spokes on the radar plot correspond to the  $j = 1, 2, \dots, n_p$  variable parameters of



**Figure 4.14:** Radar plot for SDOF problem solutions

the problem. Each axis has been normalized by the upper and lower bounds  $p_{UB}$  and  $p_{LB}$  respectively, such that a parameter value at the upper bound always corresponds to a value of 1, and a lower bound value always corresponds to a value of 0.1 on the radar plot. The vector entries for a solution are then connected to form a polygon. Thus, multiple  $n_p$ -dimensional vectors can be shown in the same graph and differences between them become apparent to

the designer. The display of more than 10 or so design vectors at the same time renders the radar plot cluttered and somewhat less effective. The next section applies the multivariable theory and plotting techniques to multivariable isoperformance cases, which are based on the sample problems introduced in Chapter 2.

## 4.7 Examples

This section develops and discusses results from exercising the multivariable isoperformance methodology on the following sample problems: SDOF, 2DOF and ODL. Aside from computing the performance invariant set,  $\mathbf{I}$ , the topic of multiobjective optimization is introduced. This is essentially the process of obtaining a pareto set,  $p_{iso}^*$  from  $p_{iso}$ .

### 4.7.1 Single DOF oscillator

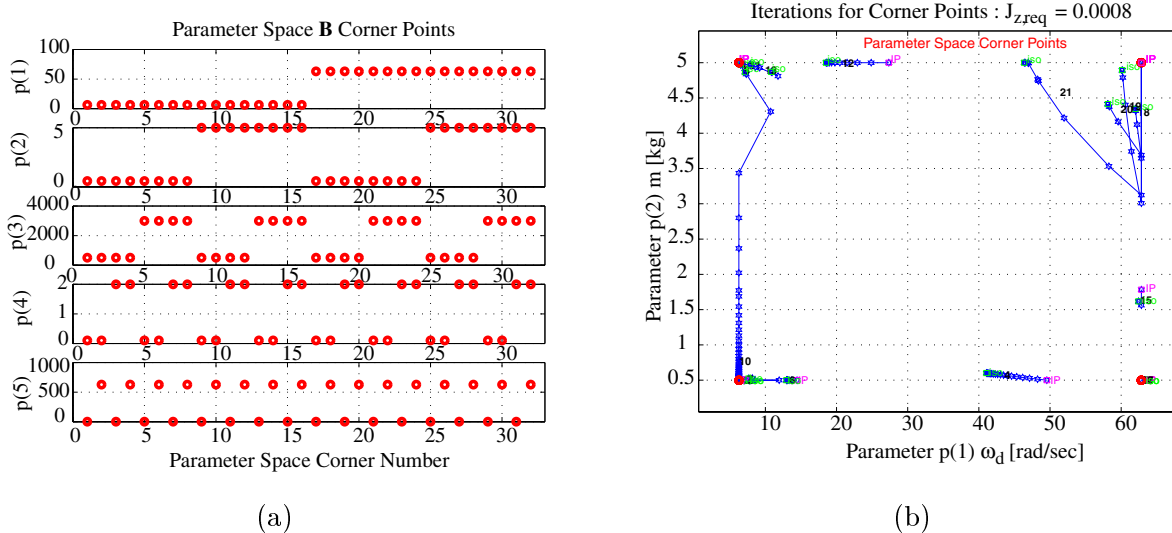
The SDOF oscillator problem was first introduced in Section 2.1. A graphical representation of the oscillator and a block diagram are shown in Figures 2.1 and 2.2, respectively. The five variable parameters are the disturbance corner frequency  $\omega_d$  [rad/sec], the mass  $m$  [kg], the linear stiffness  $k$  [N/m], the linear damper (dashpot) with coefficient  $c$  [Ns/m] and the optical control corner frequency  $\omega_o$  [rad/sec]. The performance requirement was set to  $J_{z,req} = 0.8$  [mm], which includes a 20% margin. In Section 2.1 a disturbance, sensitivity and uncertainty analysis was carried out for an initial design,  $p_o$ , and an improved, nominal design,  $p_{nom}$ . Bivariate isoperformance trades for the SDOF problem were carried out in Section 3.4.1, trading  $[\omega_d, \omega_o]$  and  $[k, c]$ . In this chapter the SDOF problem was used to demonstrate the multivariable isoperformance algorithms using the parameter combination  $[\omega_d, m, \omega_c]$ .

The final step is to include *all five parameters* in a multivariable isoperformance analysis simultaneously<sup>2</sup>. This has been done with the parameter bounds given in the Table of Figure 2.1 and the vector spline approximation algorithm presented in Section 4.3. The results are shown in Figures 4.15 and 4.16.

Figure 4.15(a) shows the 32 corner points, which are first computed for the SDOF problem with all five parameters, whereas Figure 4.15(b) shows the gradient search for the border points,  $p_{iso,border}$ , projected onto the plane of the first two parameters.

---

<sup>2</sup>For realistic opto-mechanical systems there can be hundreds of design parameters, such that a relevant subset has to be selected.

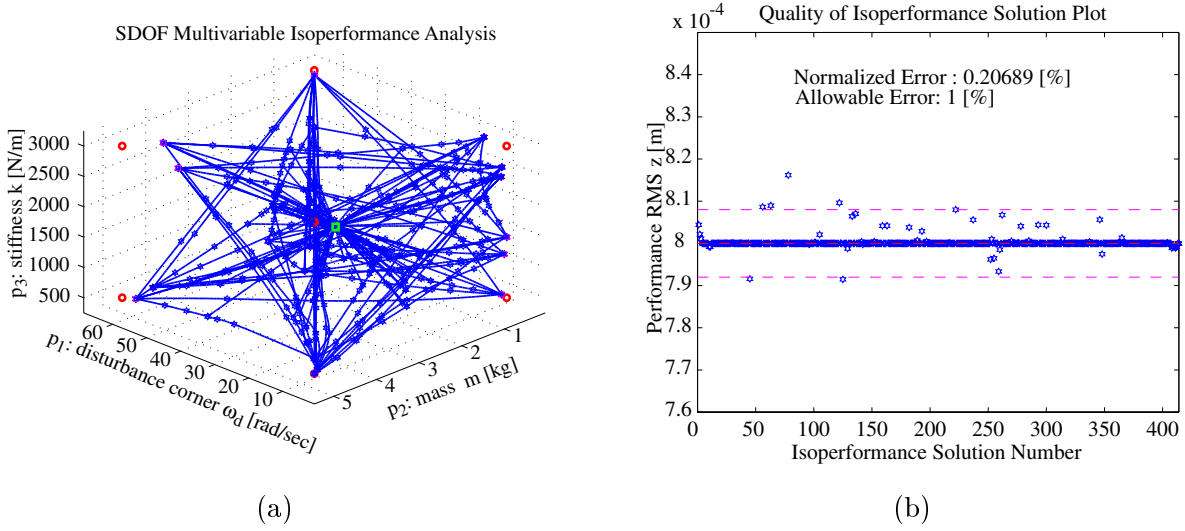


**Figure 4.15:** (a) Representation of the 32 corners ( $2^5$ ) of the trade space **B** for the SDOF problem. (b) Border point search ( $p_{iso,border}$ ) for boundary points of SDOF isoperformance set.

The isoperformance set, **I**, computed for the SDOF problem with five parameters (Algorithm III: Vector Spline Approximation) is shown in Figure 4.16(a). Note that the performance invariant points,  $p_{iso}$ , are  $5 \times 1$  vectors and have been projected to three dimensions. The quality of the solution, according to metric,  $\Upsilon_{iso}$ , is 0.2069% and is shown in Figure 4.16(b). The computation with function `isoperf_mvar.m` yields 414 isoperformance points, required  $29.2 \cdot 10^6$  floating point operations and took 241.78 seconds of CPU time (Pentium III, 650 MHz processor).

Figure 4.16(a) shows that isoperformance solutions appear to exist over a large portion of trade space **B**, at least for the displayed parameters  $p_j$ ,  $j = 1, 2, 3$ . Otherwise, it is difficult to gain any insights from the plot, since trends between parameters are not easily detected. Thus, a small subset of the 414 isoperformance solutions will subsequently be extracted from set **I**. One way to do this was presented earlier, using cross-orthogonality as defined in Equation 4.29. Another way is to introduce additional objective criteria (multiobjective optimization).

The use of a scalar or vector cost function,  $J_c$ , and risk function,  $J_r$ , was suggested in the thesis roadmap, see Figure 1.6. We will consider “closeness to mid-range” as a measure of implementation cost of the system. The underlying notion is that for each variable parameter there is a “cheap” bound and an “expensive” bound. For example for a parameter such as static wheel imbalance,  $U_s$ , as introduced in Subsection 3.4.3, the upper bound,  $p_{UB}$ , is the



**Figure 4.16:** (a) Vector Spline Approximation of SDOF isoperformance set,  $\mathbf{I}$ , projected to the first three parameters,  $[\omega_d, m, k]$ . (b) Quality of solution plot for SDOF isoperformance set.

cheap bound and  $p_{LB}$  is the expensive bound. According to Bialke more and more expense is incurred as one tries to improve the balancing of typical ball bearing reaction wheels [7]. For other parameters such as detector quantum efficiency,  $QE$ , (see Chapter 7), the upper bound will be the expensive one.

The reasoning is then that a design that meets the performance requirements,  $J_{z,req}$ , while holding each design parameter close to its mid-range (on average), can be implemented at reasonable cost, since the “burden” is evenly distributed in the system. Mathematically, such a cost metric can be expressed as:

$$J_{c,i} = \sum_{j=1}^{n_p} \left[ \frac{(p_{iso})_{j,i} - p_{LB,j}}{p_{UB,j} - p_{LB,j}} - 0.5 \right]^2 \quad (4.31)$$

where  $J_{c,i}$  is the “implementation cost” of the  $i$ -th isoperformance solution,  $p_{iso,i}$ .

The risk metric,  $J_r$ , used here is the “performance robustness to parametric uncertainty” introduced in Equation (2.7) and repeated here for convenience.

$$J_{r,i} = \frac{100}{J_{z,req}} \cdot \Delta J_z^v(p_{iso,i}) = \frac{100}{J_{z,req}} \cdot \sum_{j=1}^{n_p} \left| \frac{\partial J_z}{\partial p_j} \cdot (v_j p_j) \right|_{p_{iso,i}}$$

This metric corresponds to the “first order” approach to uncertainty analysis as discussed by Gutierrez [45]. In effect it represents the expected  $\pm\%$  uncertainty around the predicted performance level,  $J_z(p_j)$ ,  $j = 1, 2, \dots, n_p$ , given the sensitivities,  $\partial J_z / \partial p_j$  and assumed  $\pm\%$

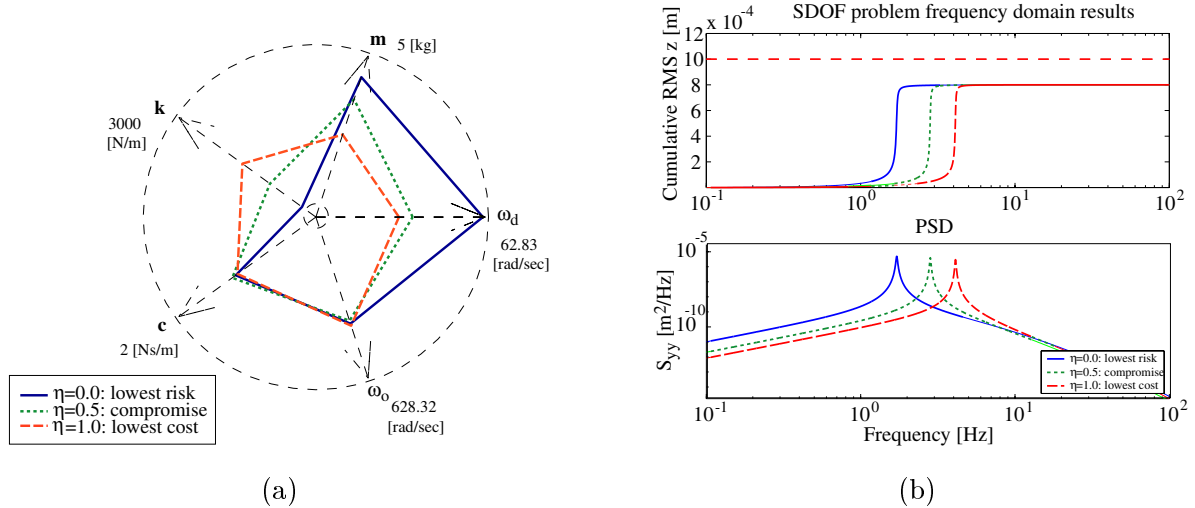
uncertainties in the values of each parameter,  $p_j$ . The assumed uncertainties,  $v_i$ , for the SDOF problem are given in Figure 2.1. The mass,  $m$ , is the least uncertain ( $\pm 1\%$ ), while the damping,  $c$ , and disturbance corner frequency,  $\omega_d$ , are most uncertain ( $\pm 10\%$ ).

A metric,  $J_{cr}$ , which combines cost and risk can be constructed as suggested in Equation (4.32).

$$J_{cr} = \eta J_c^T Q_{cc} J_c + (1 - \eta) J_r^T Q_{rr} J_r \quad (4.32)$$

where  $Q_{cc} = 1/\max(J_c)^2$  and  $Q_{rr} = 1/\max(J_r)^2$  are normalization factors and  $\eta$  is a weighting factor between cost and risk. Thus, by sliding  $\eta$  from 0 to 1 the combined metric,  $J_{cr}$ , goes from weighting exclusively risk to weighting exclusively cost.

For the 414 SDOF isoperformance solutions shown in Figure 4.16 we evaluate the cost function  $J_{cr}$  at  $\eta=0, 0.5$  and  $1.0$ . The three designs which minimize,  $J_{cr}$  at these values of  $\eta$  are shown in the radar plot of Figure 4.17(a). The corresponding power spectral densities (PSD) and cumulative RMS plots are contained in Figure 4.17(b), thus confirming that all three solutions are performance-invariant.



**Figure 4.17:** (a) Radar plot for SDOF problem pareto optimal designs. (b) PSD and cumulative RMS plot for SDOF pareto optimal designs.

At this point we want to critically analyze the difference between the three pareto optimal SDOF designs. The lowest cost design ( $\eta = 1.0$ ), which minimizes  $J_c$ , shows all 5 parameters near the mid-range as expected. The lowest risk design ( $\eta = 0.0$ ), which minimizes  $J_r$ , is significantly different in terms of disturbance corner frequency,  $\omega_d$ , mass  $m$  and stiffness,  $k$ . The low risk design relies on a large mass (well known) and small stiffness to achieve

the desired performance, which results in a low natural frequency. The large disturbance corner frequency,  $\omega_d$ , for the low risk design appears surprising at first, since the parametric uncertainty is large, i.e.  $v_5 = \pm 10\%$ . The fact that  $\omega_d \sim 10$  [Hz], however, shows that the disturbance corner frequency is roughly a decade above the natural frequency of the oscillator and does not significantly affect the performance,  $J_z$ . The compromise design ( $\eta = 0.5$ ) lies between the other two, as expected. Table 4.2 shows a comparison of performance, cost and risk for the three pareto optimal SDOF designs.

**Table 4.2:** Comparison of pareto optimal designs for SDOF sample problem.

Design	$\eta$	$J_z$ [m]	$J_c$ [-]	$J_r$ [%]	$\omega_d$ [rad/s]	$m$ [kg]	$k$ [N/m]	$c$ [Ns/m]	$\omega_o$ [rad/s]
Lowest Risk	0.0	0.8000039	0.79189	10.8853	62.0	4.35	503.3	1.12	393.07
Compromise	0.5	0.8000027	0.33688	12.8982	35.95	3.65	1153.0	1.18	380.2
Lowest Cost	1.0	0.8001169	0.16175	15.4779	30.86	2.56	1712.5	1.10	401.8

As mandated by the isoperformance condition, all three designs are equivalent in terms of  $J_z$ . This is confirmed by looking at the cumulative RMS plot in Figure 4.17(b), see upper subplot. All three designs asymptote to the same RMS value within the numerical tolerance,  $\tau$ . What is different between the designs is the frequency region over which the RMS is accumulated. Since the SDOF problem represents a single mode system, this is only dependent on the natural frequency of the oscillator. The “low risk” design has its natural frequency at  $\sim 1.5$  [Hz], whereas the “low cost” design oscillates around 4 [Hz]. For more realistic problems such as NEXUS (see Chapter 7) it is still true that all designs,  $p_{iso} \in \mathbf{I}$ , asymptote to the same performance values. The cumulative RMS curves are significantly more complicated, however, due to the dynamics of the disturbances, the opto-mechanical plant and the controller(s).

Which pareto optimal design,  $p_{iso}^*$ , from Table 4.2 should be implemented is not necessarily given by the mathematics of the problem or the preference order. At this point the final design,  $p_{iso}^{**}$ , has to be chosen from the pareto set via engineering judgement and negotiation. Therefore the isoperformance methodology is not designed to be a fully automated process, rather it requires the designer to remain “in the decision loop”.

Once a final design,  $p_{iso}^{**}$ , is chosen the roadmap (Figure 1.6) calls for an error budgeting

analysis. An error budgeting analysis is unnecessary for the SDOF problem, since only a single disturbance source (random force  $F$ ) is present, thus contributing 100% to the RMS of  $z(t)$ . This concludes the investigation of the SDOF problem in the context of this thesis.

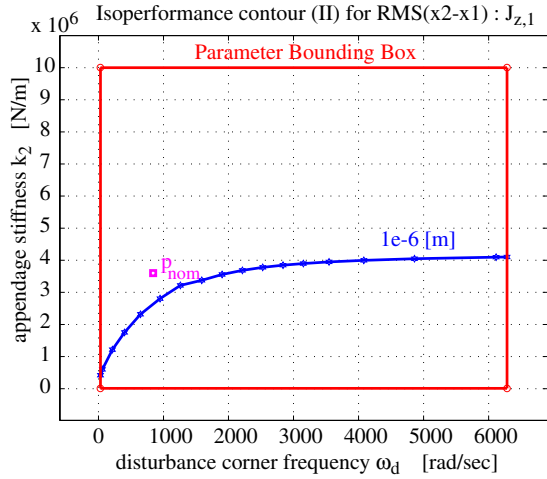
#### 4.7.2 Two DOF oscillator

The two degree-of-freedom problem (2DOF) was introduced in Section 2.2. It adds a second performance metric, i.e.  $n_z = 2$  and additional parameters, i.e.  $n_p = 8$ , compared to the SDOF problem. Section 2.2 contains an initial performance assessment, sensitivity and uncertainty analyses, as well as performance enhancement iterations. It was found that the first performance,  $J_{z,1} = \text{RMS}(x_2 - x_1)$ , is dominated by the second mode, while the second performance,  $J_{z,2} = \text{RMS}(x_1)$ , is mainly a function of the first mode. This subsection demonstrates a multivariable isoperformance analysis for the 2DOF problem.

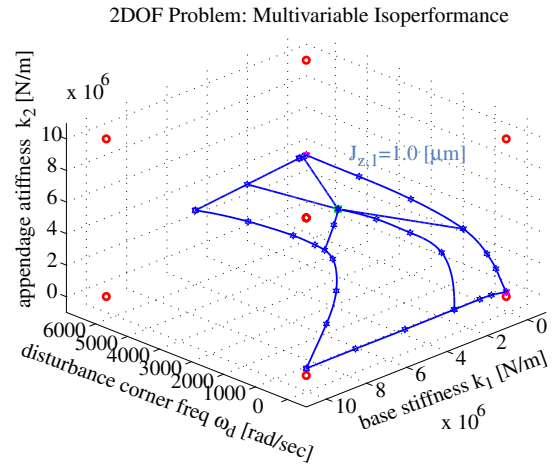
Figure 4.18(a) shows the result of a bivariate isoperformance analysis with the disturbance corner frequency,  $\omega_d$ , and appendage stiffness,  $k_2$  as the variable parameters. The constrained performance is  $J_{z,1} = \text{RMS}(x_2 - x_1)$ . One can see that the appendage stiffness,  $k_2$ , needs to be increased as the disturbance corner frequency,  $\omega_d$ , increases starting from its lower bound in order to maintain constant performance. Above  $\omega_d = 3000$  [rad/sec], however, the value of  $k_2$  can remain constant, since the rolloff region of the disturbance spectrum is well above the flexible modes of the system. This is an important finding, since in this region  $\omega_d$  can be changed without significant effect on  $J_{z,1}$ . The algorithm used here is gradient-based contour following.

When the base stiffness,  $k_1$ , is added as a parameter a multivariable analysis is conducted with three parameters. The results, using progressive spline approximation, are shown in Figure 4.18(b). This two-dimensional isoperformance surface shows that performance  $J_{z,1}$  is not very sensitive to  $k_1$ , but very sensitive to  $k_2$  at low values of the disturbance corner frequency,  $\omega_d$ .

A full isoperformance analysis using all 8 parameters is shown in Figure 4.19. Using vector spline approximation and constraining only  $J_{z,1}$  produces the results in Figure 4.19(a) with the corresponding quality plotted in (c). The most noticeable trend is the tradeoff between disturbance corner frequency,  $\omega_d$ , and disturbance intensity,  $S_{dd}$ . Introducing and constraining the second performance,  $J_{z,2}$ , see Figures 4.19(b) and (d), drops the number of isoperformance solutions,  $p_{iso}$ , from  $\sim 1200$  to 37. Thus, the isoperformance subset,  $\mathbf{I} \in \mathbf{B}$ , gets substantially



(a)

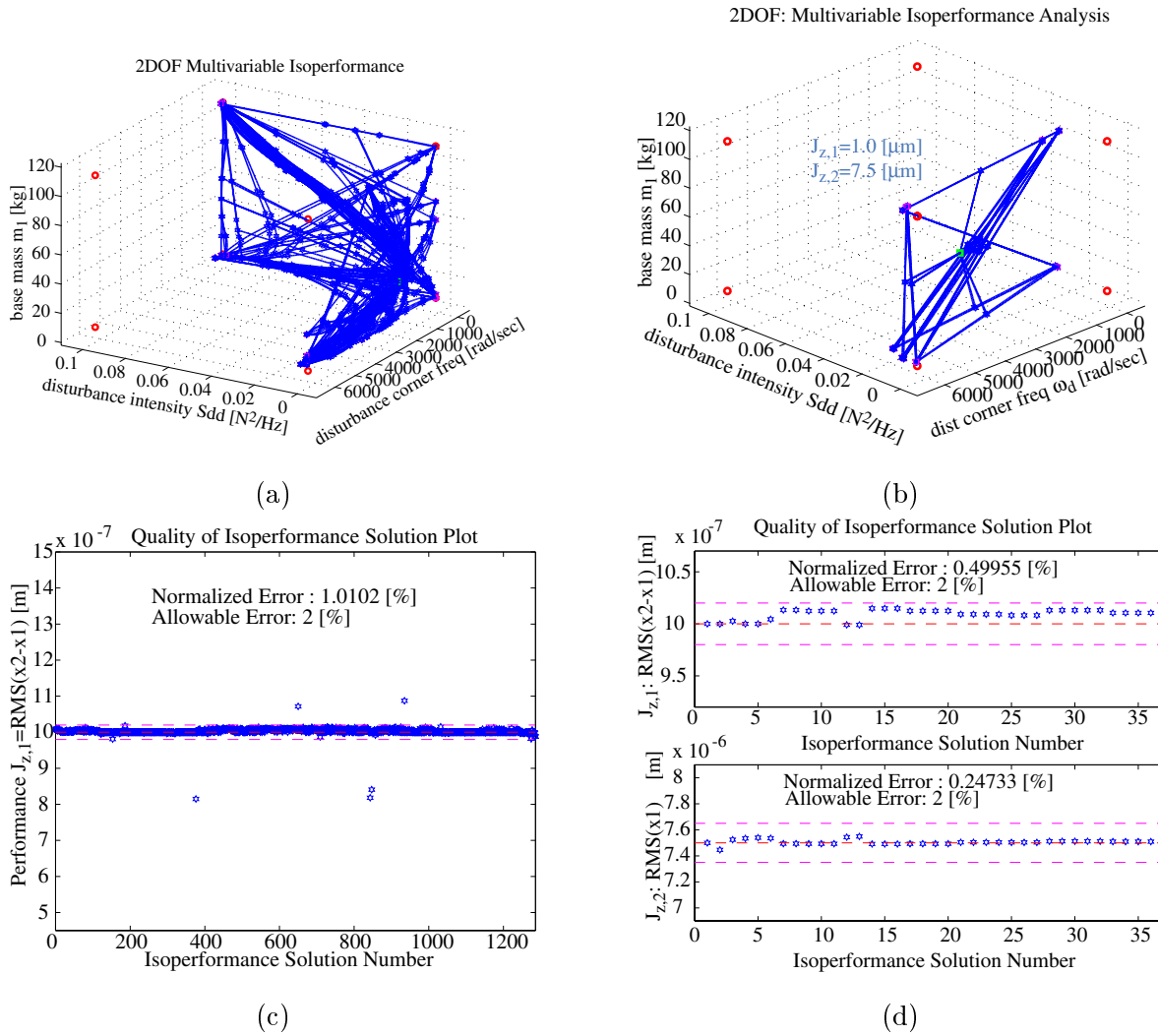


(b)

**Figure 4.18:** (a) Bivariate Isoperformance analysis for 2DOF problem with  $\omega_d$  and  $k_2$ . (b) Multivariable Isoperformance analysis for 2DOF problem with three parameters:  $\omega_d$ ,  $k_1$  and  $k_2$ .

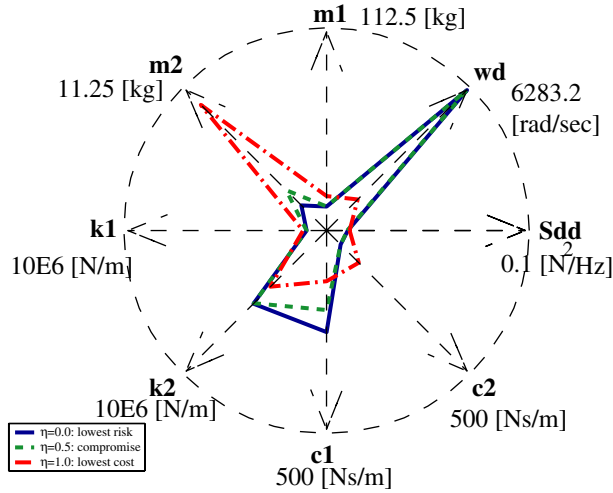
smaller the more performance metrics,  $n_z$ , are introduced.





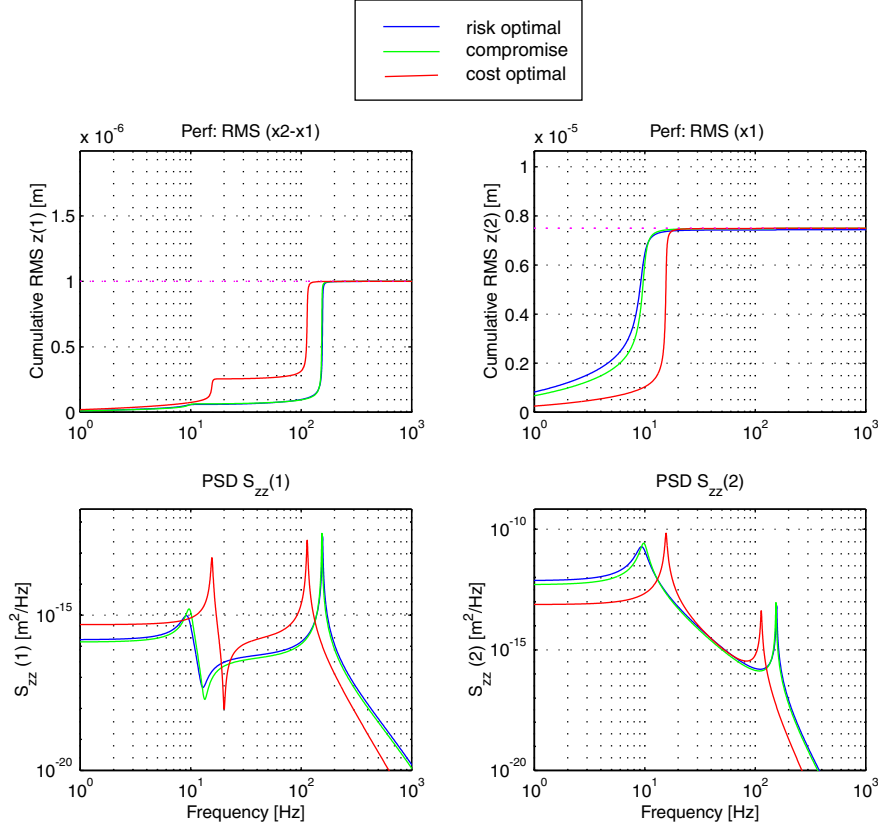
**Figure 4.19:** (a) 2DOF Isoperformance analysis with all parameters ( $n_p = 8$ ) constraining a single performance,  $J_{z,1}$ . (b) 2DOF Isoperformance analysis with all parameters ( $n_p = 8$ ) constraining both performances ( $n_z = 2$ ). (c) Quality plot for 2DOF and  $n_z = 1$  corresponding to (a). (d) Quality plot for 2DOF and  $n_z = 2$  corresponding to (b).

The same cost metric,  $J_c$ , according to Equation (4.31), risk metric  $J_r$  and combined metric,  $J_{cr}$ , see Equation (4.32), as in the SDOF example are used here. This allows extracting a pareto optimal subset of solutions from the isoperformance set. For the same settings of the weighting factor,  $\eta = 0.0, 0.5, 1.0$ , as in the previous example we obtain a low risk, low cost and a compromise design for the 2DOF problem. These three designs are shown in the radar plot of Figure 4.20. The uncertainties,  $v_j$ , of the parameters  $p_j$  were specified in Figure 2.6. Again masses are assumed to be most certain, while damping and disturbance magnitude are least certain.



**Figure 4.20:** Two DOF Problem: Radar plot for pareto optimal designs.

The radar plot shows that the low cost design (“closeness to mid-range”) is not, on average, nearly as close to mid-range as the low cost solution for the SDOF problem. This is mainly due to the fact that the 2DOF problem is more constrained. The low cost design is close to the nominal design,  $p_{nom}$ , from Section 2.2 with the appendage mass,  $m_2$ , pegged at its upper limit. Note that both performances,  $J_{z,1}$  and  $J_{z,2}$ , simultaneously asymptote the desired performance levels,  $J_{z,req}$  for all solutions,  $p_{iso}$ . This is graphically shown in the cumulative RMS plots of Figure 4.21. It is noteworthy that the low risk design has also the lowest natural frequencies for both modes, as in the SDOF example. This is mainly due to the fact that the massive designs lead to lower frequency modes and are assumed to be less uncertain. It is not immediately clear whether this insight can be generalized to more complex flexible structures.



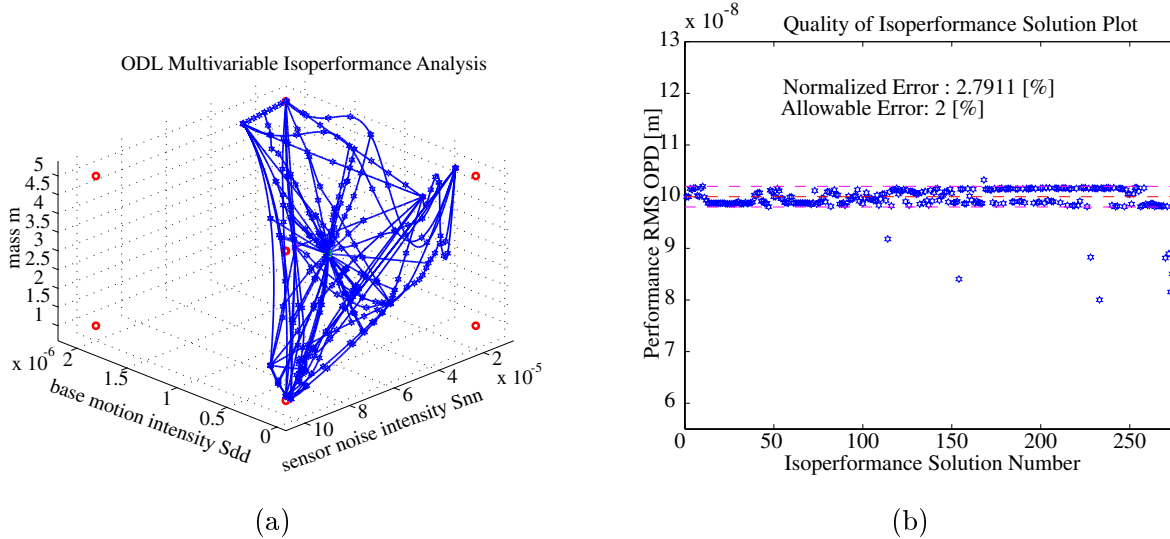
**Figure 4.21:** Two DOF Problem: PSD and cumulative RMS comparison of pareto optimal designs.

### 4.7.3 ODL Design Problem

The optical delay line (ODL) design problem was first introduced in Section 2.3. The driving requirements are summarized in Figure 2.10. The problem deals with a total of eight independently variable parameters, see Table 2.3. The steady-state performance requirement is that  $J_{z,1} = \text{RMS OPD} = 100$  [nm] and the settling time of the ODL after a step command should be  $J_{z,2} = T_{s,z} < 0.05$  [sec]. A bivariate isoperformance analysis (only constraining  $J_{z,1}$ ) was discussed in Subsection 3.4.2. The present subsection presents insights gained from a multivariable isoperformance analysis of the ODL design problem.

Figure 4.22 shows the isoperformance results for an analysis with the four variable parameters: base motion intensity  $S_{dd}$ , sensor noise intensity  $S_{nn}$ , mass  $m$  and derivative control gain,  $K_d$ . The other values are constant at the values given for  $p_{j,o}$ , see Table 2.3.

The isoperformance set,  $\mathbf{I}$ , as approximated in Figure 4.22(a) is a 3-dimensional volume



**Figure 4.22:** (a) ODL Problem: multivariable isoperformance with four parameters:  $S_{dd}$ ,  $S_{nn}$ ,  $m$  and  $K_d$ . Performance Requirement:  $J_{z,1,req} = 100$  [nm]. Isoperformance set,  $\mathbf{I}$  projected onto the first three parameters. (b) Quality of solution plot corresponding to (a) with tolerance  $\tau = 2.0$  %.

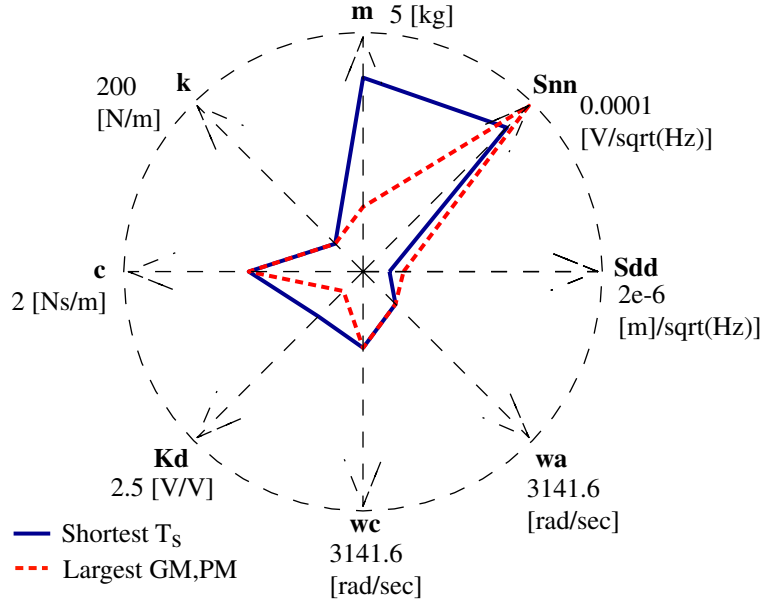
in 4-dimensional space. One can discern the tradeoff between the noise intensities of process noise,  $S_{dd}$ , and sensor noise,  $S_{nn}$ . This means that no solution which has both of these disturbance intensities pegged at their upper limit, see lower left corner of Figure 4.22(a), will meet the requirements. The quality plot Figure 4.22(b) shows the existence of a few outlier solutions which exist when the gradient search does not fully converge in the multivariable case (`isoperf_mvar.m`). In total there are 276 isoperformance solutions,  $p_{iso} \in \mathbf{I}$ , for the ODL design problem which took 1.25 GFLOPS and 556 [sec] to solve<sup>3</sup>.

As in the previous examples we introduce additional objectives to select a small subset of pareto-optimal designs from the 276 performance invariant solutions. We will consider the settling time after a step input command,  $J_{z,2} = T_{s,z}$ , as an additional criterion according to Equation (2.21). The “risk” will be embodied by the gain (GM) and phase (PM) stability margins, normalized by the required stability margins:

$$J_r = \frac{1}{GM_{req}} \cdot GM(p_{iso,i}) + \frac{1}{PM_{req}} \cdot PM(p_{iso,i}) \quad (4.33)$$

Also, of the 276 performance-invariant solutions, only the ones with  $T_{s,z} < 0.05$  [sec] will be considered. Figure 4.23 shows the two pareto-optimal designs, which optimize settling time and stability margins, respectively.

<sup>3</sup>Pentium III at 650 MHz clock speed.



**Figure 4.23:** ODL Problem: Radar plot for pareto optimal designs.

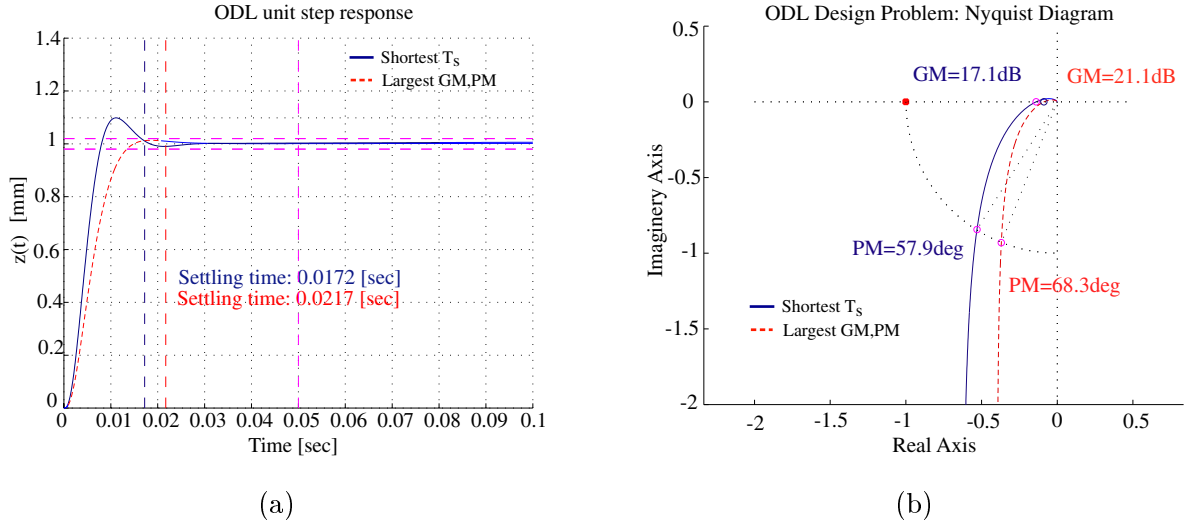
All parameters, except the four variable ones, are the same for both designs. The shortest settling time design (blue, solid line) relies on a large mass,  $m$ , and control gain,  $K_d$ , to achieve the performance. The design with the largest stability margins (red, dashed line) in terms of gain and phase margin on the other hand uses a smaller derivative gain,  $K_d$ , and smaller mass,  $m$ . Figure 4.24 compares the stability margins (Nyquist diagram) and settling time for both designs. Both of these designs are superior to the original design in Section 2.3 and would likely not have been found without the multivariable isoperformance technique presented here.

**Table 4.3:** Comparison of original design with pareto optimal designs for ODL sample problem.

Design	$J_{z,1} = \text{RMS OPD}$	$J_{z,2} = T_{s,z}$	PM	GM
Requirements	100 [nm]	< 0.05 [sec]	30 [deg]	6.0 [dB]
Original Design $p_o$	89.6 [nm]	0.0236 [sec]	39.4 [deg]	11.5 [dB]
Shortest Step Response $p_{iso,47}$	101.24 [nm]	0.0172 [sec]	57.9 [deg]	17.1 [dB]
Largest Stab. Margins $p_{iso,86}$	98.56 [nm]	0.0217 [sec]	68.3 [deg]	21.1 [dB]

Finally both designs are contrasted to the original design,  $p_o$ , in Table 4.3. This concludes

the discussion of the ODL sample problem.



**Figure 4.24:** (a) ODL Problem: Comparison of settling times for a  $r = 1.0$  [mm] step input command for both pareto optimal designs. (b) Nyquist diagram showing phase margin (PM) and gain margin (GM) for both designs.

## 4.8 Summary

This chapter discusses solutions to the multivariable isoperformance problem. The generalization of the bivariate search algorithms results in three multivariable techniques: Branch-and-Bound, Tangential Front Following and Vector Spline Approximation. A comparison for the SDOF problem shows that Vector Spline Approximation is most efficient for multivariable problems, since it only approximates the performance invariant set with border points and some interior points. Complexity theory leads to the conclusion that the isoperformance problem is a search problem, whose computational cost is non-polynomial in the number of variable parameters,  $n_p$ . Thus, problems with more than  $\sim 10$  variable parameters are still expensive to solve without additional advancements (e.g. parallel computing). The application of the multivariable isoperformance algorithms to the sample problems (SDOF, 2DOF and ODL design) demonstrates that superior solutions can be obtained compared to performance enhanced designs, which are based on a random initial guess.

## Chapter 5

# Challenges of Large Order Systems

A key element of the isoperformance methodology is the ability to compute performances,  $J_z$ , and analytical sensitivities,  $\nabla J_z$ , for disturbance, plant and control parameters. This has to be done in a computationally efficient manner for large order, typically numerically ill-conditioned systems. As was seen in the previous chapter it is likely that hundreds or even thousands of Lyapunov equations have to be solved during a comprehensive isoperformance analysis. This is true even when the efficient algorithms (Tangential Front Following and Vector Spline Approximation) are selected. The solutions to the Lyapunov equation provide either the state covariance matrix,  $\Sigma_q$ , see Equation (5.1), or the Lagrange multiplier matrices,  $L_i$ , where  $i = 1, 2, \dots, n_z$ , according to Equation (5.2).

Empirical considerations in Section 5.1 show that the computational cost of solving a Lyapunov equation is roughly  $50 \cdot n_s^3$  floating point operations (FLOPS), where  $n_s$  is the number of states of the state space system  $S_{zd}$ . Thus, for a realistic model with say, 500 states, the cost would be  $6.25 \cdot 10^9$  FLOPS, which would have taken upwards of an hour to solve on 1990 vintage computers. Even taking into account the increase in computational capability in the last decade it is desirable to improve the computational efficiency in solving for  $\Sigma_q$  and  $L_i$ , respectively. The first strategy is to attempt a reduction in the “3” exponent of the cost expression with a fast Lyapunov solver for diagonalizable systems, presented in Section 5.1. The second approach is to reduce the number of states,  $n_s$ , while preserving the essential information in the model. Performance prediction for balanced and reduced systems in this context is the topic of Section 5.2. Finally, the sensitivity analysis has to be generalized such that it can handle similarity transformed and reduced systems, see Section 5.3.

## 5.1 Efficient Solution of Lyapunov Equation

Lyapunov equations in isoperformance must be solved frequently for the state covariance matrix,  $\Sigma_q$ , and the Lagrange multiplier matrices,  $L_i$ , where  $i = 1, 2, \dots, n_z$ . For convenience we repeat the expressions for the Lyapunov equation of the state covariance matrix

$$A_{zd}\Sigma_q + \Sigma_q A_{zd}^T + B_{zd}B_{zd}^T = 0 \quad (5.1)$$

and the Lagrange multiplier matrix

$$L_i A_{zd} + A_{zd}^T L_i + C_{zd,i}^T C_{zd,i} = 0 \quad (5.2)$$

for the  $i$ -th performance, respectively. Recall that  $A_{zd}$ ,  $B_{zd}$  and  $C_{zd}$  represent the (closed loop) assembled state space matrices containing the disturbance, plant and compensator dynamics. Note that matrix  $A_{zd}$  is of size  $n_s \times n_s$ , where  $n_s$  is a measure of the model order (size). This section presents an algorithm which significantly accelerates these computations, provided that the state space system can be transformed into the real modal (diagonal) form. The algorithm was applied to the solution of large order gramians in the context of model conditioning by Uebelhart [136]. The following description contains results by Uebelhart with respect to optimizing the solution block size,  $m$ . The author's contribution lies in recognizing the key role the algorithm plays in isoperformance computations and in integrating the solver into the DOCS framework via the new function `newlyap.m`.

### 5.1.1 Computational Cost of Lyapunov Solution and Balancing

In order to assess the benefits of a more efficient Lyapunov solver one first needs to obtain an estimate of the computational cost of solving Equation (5.1). Additionally an estimate is obtained for the cost of balancing the state space system, which is a prerequisite for model reduction, see Section 5.2. There are computational savings that can be realized by balancing and reducing the size of the model prior to conducting isoperformance analyses. Clearly, some of this savings will be consumed by the balancing process. This section derives the amount of model reduction, as a function of the original model's size, beyond which computational savings are realized.

The computational effort is assessed by considering the number of floating-point operations (FLOPS) rather than CPU time, since FLOPS are machine independent. The operations required for diagonalizing and balancing the system initially are taken into account. The basic

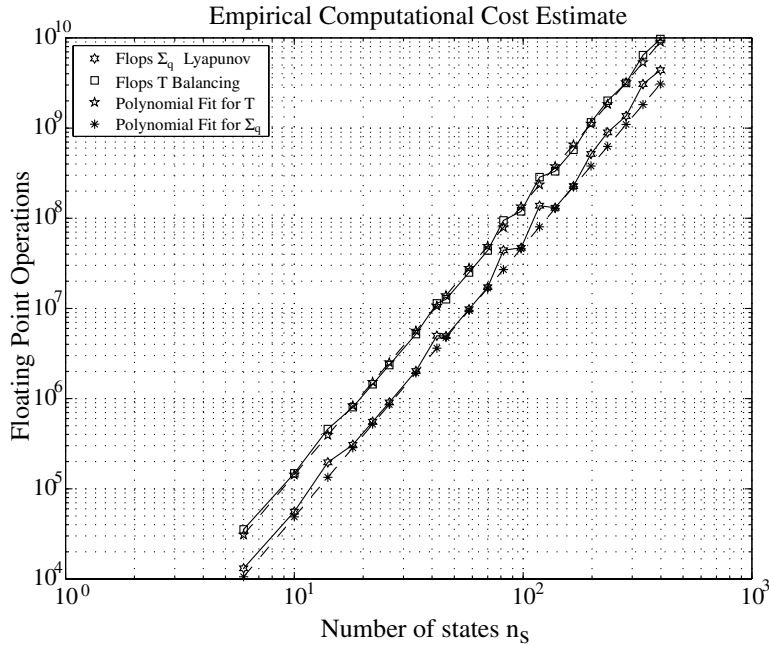


assumption is that the cost of computing  $J_z = f(\Sigma_q)$  is dominated by the computational cost of solving Lyapunov Equation (5.1). For balanced (and reduced systems) the solution cost for the balancing matrix,  $T$ , needs to be considered as well.

Assume that the cost of solving a Lyapunov equation,  $J_{lyap}$ , and for balancing the system,  $J_T$ , can be cast as a polynomial expression of the number of states  $n_s$  with constant coefficients, such that

$$\begin{aligned} J_{lyap} &\cong \alpha \cdot (n_s)^p \\ J_T &\cong \beta \cdot (n_s)^q \end{aligned} \tag{5.3}$$

Since the exact operations count for solving a Lyapunov equation cannot be determined a priori<sup>1</sup>, we will empirically determine the required FLOPS for solving the Lyapunov equation (for  $\Sigma_q$ ) and the balancing of the system (for  $T$ ), as a function of the number of states,  $n_s$ . A multiple degree-of-freedom spring mass system, see Subsection 5.4.2, provides the state space system,  $S_{zd}$ , for this analysis. The results of recording the FLOPS count as a function of  $n_s$ , while solving Equation (5.1) and solving for  $T$  are shown in Figure 5.1 on a loglog scale.



**Figure 5.1:** Computational cost of balancing ( $T$ ) and solving a Lyapunov equation for  $\Sigma_q$  as a function of the number of states  $n_s$ .

---

<sup>1</sup>This is similar to LU-factorization, which goes as roughly  $(2/3)n_s^3$  FLOPS, but depends on the intricacies of the Gaussian elimination algorithm, e.g. row swapping if a zero-pivot is encountered.

Curve fitting with the least-squares method leads to the following results:

$$\begin{aligned}\alpha &= 50.42 \approx 50 ; p = 3.0412 \approx 3 \\ \beta &= 146.7 \approx 150 ; q = 2.998 \approx 3\end{aligned}\tag{5.4}$$

The results suggest that there is a linear relationship on the loglog-scale. Both operations depend on the number of states,  $n_s$ , to the *third power*. Balancing a system is roughly *three times* more expensive than solving a Lyapunov equation of the same dimensions.

With this information one can compare the cost of performing a sensitivity analysis (only taking into account balancing and the solution of Lyapunov equations) for the original system,  $S_{zd}$ , and a balanced-reduced system,  $\bar{S}_{zd}$ . The computational cost of the disturbance and sensitivity analysis for the original system is the cost of solving for the state covariance matrix,  $\Sigma_q$ , and for the Lagrange multiplier matrices,  $L_i$ . Each equation has dimension  $n_s$ . The equivalent cost for the balanced-reduced system is the upfront cost of balancing the system (order  $n_s$ ) - this can be regarded as a “fixed” cost component - and the subsequent Lyapunov solutions for the reduced  $\bar{\Sigma}_q$  and  $\bar{L}_i$ 's of smaller order  $n_k$ . The computational cost in terms of floating-point operations can be expressed as:

$$J_o = \underbrace{\alpha (n_s)^p}_{\Sigma_q} + n_z \cdot \underbrace{\alpha (n_s)^p}_{L_i}\tag{5.5}$$

$$J_r = \underbrace{\beta (n_s)^q}_T + 0 \cdot \underbrace{\alpha (n_k)^p}_{\bar{\Sigma}_q} + n_z \cdot \underbrace{\alpha (n_k)^p}_{\bar{L}_i}\tag{5.6}$$

where  $n_s$  is the number of original states,  $n_k$  is the number of states kept in the system and  $n_z$  is the number of performances. In Equation (5.6) we can set the cost of computing the reduced state covariance matrix to zero, since the Hankel singular values, which are kept in the system, are already known. Thus, one can establish an inequality such that  $J_r < J_o$  :

$$\beta \cdot n_s^q + n_z \cdot \alpha \cdot n_k^p < (n_z + 1) \cdot \alpha \cdot n_s^p\tag{5.7}$$

After substituting the coefficients from Equation (5.4) in Equations (5.5) and (5.6) and rearranging we obtain:

$$\left(\frac{n_k}{n_s}\right)^3 < \frac{n_z - 2}{n_z}\tag{5.8}$$

The above inequality is found for the crossover point, i.e. when  $J_r$  (balanced-reduced system cost)  $<$   $J_o$  (original system cost). It is concluded that CPU time savings are only possible once there are 3 or more performances, i.e.  $n_z > 2$ . The above inequality is useful

since it can provide the required state reduction ratio  $n_k/n_s$  that will lead to a time savings, given the number of performances  $n_z$ . For  $n_z = 4$  a reduction of slightly more than 21 % of the number of states of the original system is sufficient to speed up computations. This percentage becomes lower as more performance metrics of interest are present in the system. The SIM Classic model, presented in Subsection 3.4.3, for example has 15 performances of interest. The reason for the high “upfront” cost is that the system balancing operation requires the solution of two Lyapunov equations for the gramians, two Cholesky factorizations of the gramians, a singular value decomposition (SVD) of the product of the Cholesky factors<sup>2</sup> and matrix multiplications to obtain  $T$ ,  $T^{-1}$  and the balanced system matrices. Inequality (5.8) together with (5.65) allows a quantitative assessment of the tradeoff between computational speed and accuracy as a function of the truncation threshold  $n_k$ .

### 5.1.2 Fast Lyapunov Solver for Diagonalizable Systems

This subsection presents a fast Lyapunov solver for diagonalizable state space systems. The goal is to improve on the  $50n_s^3$  Lyapunov computational cost estimate when  $n_s$  is large (e.g.  $n_s > 100$ ). This subsection is essentially identical to Section 3.3 of Reference [136]. The general form of the Lyapunov Equations (5.1) and (5.2) can be written as

$$AX + XA^T + Q = \dot{X} \quad (5.9)$$

Assuming that only the values of the states, but not their statistics, change over time (steady-state analysis), the temporal derivative of the state covariance matrix,  $X$ , on the right side can be set to zero.

$$AX + XA^T + Q = 0 \quad (5.10)$$

Diagonalize  $A$  using its eigenvectors<sup>3</sup>,  $A = SAS^{-1}$ . Substitute this matrix product for  $A$  in Equation (5.10), pre-multiply by  $S^{-1}$  and post-multiply by  $S^{-T}$ .

$$\begin{aligned} S^{-1} ( (SAS^{-1})X + X(S^{-T}AS^T) + Q ) S^{-T} &= 0 \\ \underbrace{S^{-1}S}_{=I} \Lambda S^{-1} X S^{-T} + S^{-1} X S^{-T} \Lambda \underbrace{S^T S^{-T}}_{=I} + S^{-1} Q S^{-T} &= 0 \end{aligned}$$

---

<sup>2</sup>According to Laub's method [78]

<sup>3</sup>Note that MATLAB uses a Schur decomposition. In practice, the eigenvalue method described may be unstable if the matrices are ill-conditioned.

This can be rewritten as

$$\Lambda \underbrace{S^{-1}XS^{-T}}_{\equiv \tilde{X}} + \underbrace{S^{-1}XS^{-T}}_{\equiv \tilde{X}} \Lambda + \underbrace{S^{-1}QS^{-T}}_{\equiv \tilde{Q}} = 0 \quad (5.11)$$

This is a new Lyapunov equation with the matrix definitions given above. The important property is that  $\Lambda$  is diagonal. For a simple  $2 \times 2$  case we obtain

$$\begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \begin{bmatrix} \tilde{x}_{11} & \tilde{x}_{12} \\ \tilde{x}_{21} & \tilde{x}_{22} \end{bmatrix} + \begin{bmatrix} \tilde{x}_{11} & \tilde{x}_{12} \\ \tilde{x}_{21} & \tilde{x}_{22} \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} + \begin{bmatrix} \tilde{q}_{11} & \tilde{q}_{12} \\ \tilde{q}_{21} & \tilde{q}_{22} \end{bmatrix} = 0 \quad (5.12)$$

$\lambda_i$  are the diagonal values of  $\Lambda$ , and the  $\tilde{x}_{ij}$  are the scalar entries in each  $(i, j)$  element of  $\tilde{X}$ . We also know that  $\tilde{x}_{21} = \tilde{x}_{12}$  and that  $\tilde{q}_{21} = \tilde{q}_{12}$ . By multiplying the matrices out, the diagonal nature of  $\Lambda$  succeeds in decoupling the entire set of equations.

$$\begin{bmatrix} \lambda_1 \tilde{x}_{11} + \tilde{x}_{11} \lambda_1 + \tilde{x}_{11} = 0 & \lambda_1 \tilde{x}_{12} + \tilde{x}_{12} \lambda_2 + \tilde{q}_{12} = 0 & \cdots \\ \lambda_2 \tilde{x}_{21} + \tilde{x}_{21} \lambda_1 + \tilde{q}_{21} = 0 & & \\ \vdots & & \end{bmatrix}$$

The resulting equations are decoupled, so that solving for each  $\tilde{x}_{ij}$  requires solving a simple algebraic equation  $\lambda_i \tilde{x}_{ij} + \tilde{x}_{ij} \lambda_j + \tilde{q}_{ij} = 0$ .

$$\tilde{x}_{ij} = -\frac{\tilde{q}_{ij}}{(\lambda_i + \lambda_j)}. \quad (5.13)$$

After solving for all the  $\tilde{x}_{ij}$ , they can be incorporated into  $\tilde{X}$ , which in turn can be substituted into the original matrix product  $X = S\tilde{X}S^T$ . This produces the Lyapunov solution  $X$ .

For systems where  $n_s > 2000$ , this one computation may take upwards of an hour to complete and involves over  $10^{11}$  floating point operations. Clearly this would preclude meaningful isoperformance analyses for large order models. A potentially faster algorithm can be constructed, which exploits the diagonal properties described above. Consider again the Lyapunov form outlined in Equation (5.12). The diagonal nature of  $\Lambda$  permits the individual equations to decouple for each  $\tilde{x}_{ij}$ , thus allowing independent solutions to build the entire  $n_s \times n_s$  matrix  $X$ . This same idea can be applied when  $\Lambda$  is block diagonal instead of diagonal. Using the variable  $A$  in place of  $\Lambda$ , consider the general Lyapunov form described by Equation (5.10), but for the case where  $A$  is  $2 \times 2$  matrix-block diagonal.

$$\begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix} \begin{bmatrix} X_{11} & X_{12} \\ X_{12}^T & X_{22} \end{bmatrix} + \begin{bmatrix} X_{11} & X_{12} \\ X_{12}^T & X_{22} \end{bmatrix} \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix}^T + \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{12}^T & Q_{22} \end{bmatrix} = 0 \quad (5.14)$$

Four matrix equations result.

$$\begin{aligned} 1) & A_1 X_{11} + X_{11} A_1^T + Q_{11} = 0 & 2) & A_1 X_{12} + X_{12} A_2^T + Q_{12} = 0 \\ 3) & A_2 X_{12}^T + X_{12}^T A_1^T + Q_{12}^T = 0 & 4) & A_2 X_{22} + X_{22} A_2^T + Q_{22} = 0 \end{aligned} \quad (5.15)$$

Notice that equations 1) and 4) in Equation (5.15) are just new Lyapunov equations. Equations 2) and 3) are also Lyapunov equations, though in a more general form  $AX + XB + C = 0$ . This is sometimes called the *Sylvester* equation. Each of these can be solved using whichever technique was formerly applied for the full matrix  $X$ , but because of their smaller size they can individually take far less time than the full  $n_s \times n_s$  system. This is due to the fact that the Lyapunov computational cost goes with  $n_s^3$ . Also note that Equation 3) is just the transpose of Equation 2), so of the four equations only three must be solved.

The requirement on  $A$  is that it can be made into a block diagonal form. Luckily the decoupled modal form resulting from a normal modes analysis can be easily written in such a form. If the system is no longer in a modal form, the eigenvalues of most  $A$  matrices can be written in a diagonal *Jordan* form [143]. This is also sometimes referred to as the real modal form.

$$A_i = \begin{bmatrix} -\zeta_i \omega_i & \omega_i \sqrt{1 - \zeta_i^2} \\ -\omega_i \sqrt{1 - \zeta_i^2} & -\zeta_i \omega_i \end{bmatrix} \quad (5.16)$$

Using the  $2 \times 2$  modal system, there are now  $n_s/2$  blocks along the diagonal of  $A$ . Keeping in mind the symmetry of  $X$ , this means

$$N_{lyap} = \frac{n_s}{2} \left( \frac{n_s}{2} + 1 \right) \quad (5.17)$$

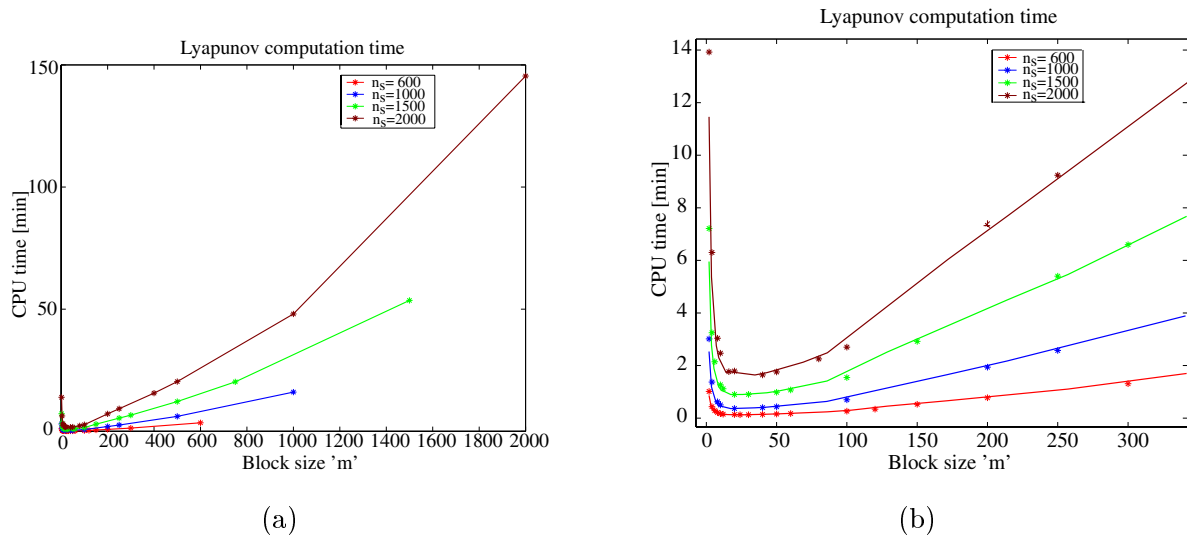
separate  $2 \times 2$  Lyapunov solutions  $X_{ij}$  must be solved. One concern may be that although the computational time for each solution is fast, the number of computations and the inefficiencies of the required *for*-loops may not yield an overall time savings. As will be seen, the time to solve all the  $2 \times 2$  equations is less than the time required to solve the entire  $n_s \times n_s$  problem; however this is not the most efficient block size,  $m$ , to use.

There is no reason that larger blocks can not be selected, so long as the size is an even factor of  $n_s$ . Using a block size of  $m$ , a general relation for the number of Lyapunov equations can be written.

$$N_{lyap} = \frac{n_s}{m} \left( \frac{n_s}{m} + 1 \right) \quad (5.18)$$

As an example of the efficiency that can be gained, this method was run on sample problems with  $n_s = 600, 1000, 1500,$  and  $2000$  states by Uebelhart [136]. A modal system was created using  $n_s/2$  logarithmically even-spaced frequencies between  $10^{-3}$  and  $10^3$  Hertz. A damping ratio of 0.001 was used, and an input  $B_{zd}$  matrix was chosen randomly. All even factors of  $n_s$  are chosen as possible block sizes  $m$ . For  $n = 1000$ , this allows  $m = [2 \ 4 \ 8 \ 10 \ 20 \ 40 \ 50 \ 100 \ 200 \ 250 \ 500 \ 1000]$ . Even numbers are required since the  $A_{zd}$  matrix is still  $2 \times 2$  block diagonal.

Figure 5.2(a) shows the variation of CPU time in minutes versus block size,  $m$ , for each problem of size  $n_s$ . The last point on each curve represents the full  $n_s \times n_s$  Lyapunov solution. For  $m = n_s = 2000$ , nearly 2.5 hours are required to obtain an answer (on a 850 MHz Pentium III computer). This compares to 14 minutes for  $m = 2$ . This considerable time savings is improved further with slightly larger block sizes. A minimum in all four curves, shown clearer in Figure 5.2(b), indicates an optimum  $m$  between  $m = 20$  and  $40$ . For  $m = 40$ , the 2000 state model takes only 1.6 minutes. A summary of the results from Figure 5.2 is included in Table 5.1.



**Figure 5.2:** (a) CPU Time in minutes to solve a Lyapunov equation of size  $n_s$  using blocks of size  $m$ . (b) Zoom in on optimal region between  $m = 20 - 40$ .

**Table 5.1:** Time improvements using diagonal Lyapunov solver

$n_s$	time to solve full $n_s \times n_s$ system (minutes)	fastest block size $m$	time to solve using $m \times m$ blocks (minutes)	time savings ratio $t_{max}/t_{min}$
600	3.5	24	0.13	26.9
1000	16.1	20	0.37	43.5
1500	53.7	20	0.90	59.7
2000	145.5	40	1.65	88.2

The quality of the solution is checked by placing the answer back into the Lyapunov Equation (5.10). This resultant matrix should equal zero. Due to numerical inaccuracies, the maximum value of the resultant is actually on the order of  $10^{-13}$  for all  $n_s$ . What is important is that the resultant for each block solution is identical to the resultant for the full  $n_s \times n_s$  solution. The solutions suffer from no additional inaccuracies.

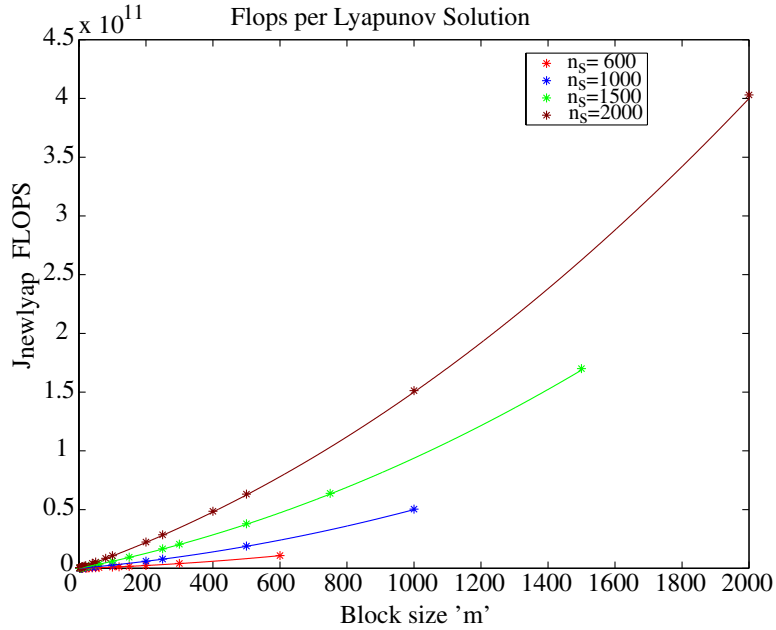
Another metric with which to judge the routines is the number of floating point operations, or FLOPS. The advantage of measuring FLOPS over CPU time is that FLOPS are machine independent. The nature of MATLAB's `lyap.m` function is such that an exact FLOPS count based on the number of required operations can not be specified. The Schur decomposition used by MATLAB requires a Gram-Schmidt factorization  $A = QR$ . This may require a different number of operations depending on intricacies of the  $QR$  factorization (e.g. row swapping). Reference [2], used for the `lyap.m` function, suggests that the number of required operations is probably overestimated by

$$N_{FLOPS} < (5.5 + 4\sigma)n_s^3$$

where  $\sigma$  is an average number of steps in the factorization, dependent on the matrix  $A$ . Earlier it was empirically found that the number of FLOPS goes by  $50n_s^3$ , which would give a value of  $\sigma = 11.125$ . Using the fast Lyapunov method, a single  $50n_s^3$  operation is replaced by many  $50m^3$  operations, with  $n_s \gg m$ . If the number of Lyapunov equations is described by Equation (5.18), then the estimated number of FLOPS is

$$J_{newlyap} = 25(mn_s^2 + m^2n_s) \tag{5.19}$$

The total number of FLOPS to solve every problem of size  $n_s$  with block sizes  $m$  is plotted as points in Figure 5.3. The curves plotted over the points are the approximations given in Equation (5.19) above. The approximation is shown to work very well. The fast Lyapunov method succeeds in reducing the FLOPS count by several orders of magnitude. For the 2000 state model, the full  $n_s \times n_s$  system requires over  $4 \times 10^{11}$  FLOPS compared to  $4.9 \times 10^8$  FLOPS when a block size of  $m = 2$  is used. The number of FLOPS for those block sizes with the fastest times (as given in Table 5.1) are provided in Table 5.2.



**Figure 5.3:** Number of FLOPS for the diagonal Lyapunov solver, along with curves showing the flop count approximation.

**Table 5.2:** Improvements in FLOPS using fast Lyapunov solver

$n_s$ states	# flops for full $n_s \times n_s$	fastest block size $m$	# flops using $m \times m$ blocks
600	$1.1 \times 10^{10}$	24	$2.4 \times 10^8$
1000	$5.0 \times 10^{10}$	20	$5.6 \times 10^8$
1500	$1.7 \times 10^{11}$	20	$1.3 \times 10^9$
2000	$4.0 \times 10^{11}$	40	$4.3 \times 10^9$



The diagonal Lyapunov solver was initially written for gramian computations as `newgram.m` and has been incorporated in DOCS as a general Lyapunov solver, `newlyap.m`. Note that function `ss2mod7.m` (available from MIT Space Systems Laboratory) must be used to diagonalize the state space system prior to using the fast Lyapunov solver. The fast diagonal Lyapunov solver drops the highest exponent in the Lyapunov computational cost expression from “3” to “2”, which results in a significant benefit for large order systems. This approach can be combined with the balanced reduction discussed below.

## 5.2 Analyses for Similarity Transformed Systems

The second strategy for reducing the burden of computing the performance,  $J_z$ , is to reduce the number of states from  $n_s$  to  $n_k$ , while preserving the important dynamics in the model. The first step in model reduction is typically a similarity transformation, such as internal balancing. Therefore in this section the computation of the performance and sensitivity, assuming root-mean-square (RMS) metrics, for open or closed-loop LTI systems driven by white noise is derived for *internally balanced (transformed) models*.

The results are first validated with single and multi-degree-of-freedom examples. The extensions to the framework have then been incorporated in revisions of the DOCS functions `dist_analysis.m` and `sens_analysis.m`. Finally a validation is carried out by applying the method to internally balanced and subsequently reduced models of the Space Interferometry Mission (SIM Classic). A number of numerical issues such as minimality have been successfully addressed, which allow fast and numerically robust sensitivity computations for large order, closed-loop systems that are inherently ill-conditioned. This work contributes to higher computational efficiency and robustness for trade and optimization studies such as the one presented in Chapter 7 for the NEXUS spacecraft. It is assumed that the systems are all linear and time-invariant.

### 5.2.1 Motivation and Scope

For realistic MIMO systems such as SIM, NGST, Nexus or TPF the models can contain several hundred or even thousands of states,  $n_s$ . Factors that tend to increase the size of the overall state vector are a complicated disturbance spectrum and many different disturbance sources, important structural modes over a large bandwidth and large order (= modern) compensators.

A significant impediment for the practical use of isoperformance for these large order models has been the fact that the state-space realization of the system<sup>4</sup> representing the appended overall dynamics (open or closed loop) had to be preserved. This was necessary in order for the system parameters of interest (e.g. modal parameters  $\omega, \zeta, \hat{m}, \phi_{ij}$ )<sup>5</sup> to appear explicitly in the system matrices  $A_{zd}, B_{zd}, C_{zd}$ . The assumed structure of the appended  $A_{zd}$  matrix, which can span on the order of 100-10000 states, as stipulated by Gutierrez [45], is given as follows:

$$A_{zd} = \left[ \begin{array}{c} A_d \\ \left[ \begin{array}{c} B_w \\ B_c D_{yw} \end{array} \right] C_d \left[ \begin{array}{cc} A_p & B_u C_c \\ B_c C_y & A_c + B_c D_{yu} C_c \end{array} \right] \end{array} \right] \quad (5.20)$$

Any kind of similarity transformation [102],[77, 78] removes the explicit dependence of the state space matrices on the parameters of interest. In that case the matrix partial derivatives  $\partial A_{zd}/\partial p_j, \partial B_{zd}/\partial p_j, \partial C_{zd}/\partial p_j$  needed for determining the sensitivity of the root-mean-square (RMS) of the i-th performance metric  $J_{z,i}$ , with respect to the j-th parameter  $p_j$ , i.e.  $\partial J_{z,i}/\partial p_j = \partial \sigma_{z_i}/\partial p_j$  cannot be easily computed. Additionally when model reduction is performed on the balanced system, states which correspond to small Hankel singular values are removed, which affects the resulting performance RMS and sensitivity values. Gutierrez [45, p.268] recognized the need for an extension of the existing framework in his recommendations for future work (third item):

“The numerical conditioning problem discovered during the finite-difference computations in the SIM Classic example needs to be resolved. Balancing and model reduction techniques might alleviate this problem; however, the sensitivity framework currently does not support alternative system realizations. Efforts to accommodate other state-space realizations should be made.”

This recommendation gave the impetus for the work presented in this section. The current capabilities are extended such that RMS values and sensitivities (for modal or physical parameters) can be computed for internally balanced and reduced systems. The equations are validated with simple and large-order examples and useful error bounds are found.

<sup>4</sup>Defined in Equations (4.58)-(4.60) of Reference [45]

<sup>5</sup>The physical parameters appear implicit and are recovered from the modal parameters via the chain rule.

## 5.2.2 Internal Balancing Transformation

The internal balancing transformation  $\tilde{q} = Tq$  of a state space system was first described by Moore [102] as well as Laub [77],[78] and co-workers. Gregory [39] successfully applied internal balancing to the model reduction of lightly damped space structures. Essentially the internal balancing operation is a similarity transformation, which does not affect the input-output relationship of the state space system (i.e. the transfer function matrix  $G_{zd}(j\omega)$ ). It is true that for a similarity transformation the transfer function matrix  $G_{zd}(s)$  of the original system and the transfer matrix of the balanced system  $\tilde{G}_{zd}(s)$  are equal. This can be proven by showing that the difference between the transfer function matrices is zero, see Appendix B.1.

The goal of the operation is to obtain a (unique) state space realization in which the transformed controllability and observability gramians  $\tilde{W}_c$  and  $\tilde{W}_o$  are equal to each other, thus the term “balanced”.

$$\tilde{W}_c = \tilde{W}_o = \Sigma_H \quad (5.21)$$

This allows ranking the states in decreasing order of observability/controllability. In the context of the disturbance analysis one could also say that the states are ranked in decreasing order of disturbability/performability. The diagonal elements of the transformed gramians are the Hankel singular values of the system.

$$\tilde{W}_c = TW_cT^T = \tilde{W}_o = (T^{-1})^T W_oT^{-1} = \Sigma_H = \text{diag} \{ \sigma_1^H, \dots, \sigma_{n_s}^H \} \quad (5.22)$$

The key question is how to compute the balancing matrix  $T$ . The first step is always to compute the controllability and observability gramians of the original system  $W_c, W_o$ . First we solve the Lyapunov equation for the controllability gramian

$$A_{zd}W_c + W_cA_{zd}^T + B_{zd}B_{zd}^T = 0 \quad (5.23)$$

and then for the observability gramian

$$A_{zd}^T W_o + W_o A_{zd} + C_{zd}^T C_{zd} = 0 \quad (5.24)$$

A number of different algorithms for finding  $T$ , starting with Moore’s method [102] in 1981, have been suggested over the years. The second method uses a singular value decomposition (SVD) and was developed by Laub [78]. Both algorithms are presented in more detail in Appendix B.2.

The resulting properties of  $T$  are such that it is a real, square and non-singular similarity transformation matrix. The matrix  $T$  does indeed depend on the physical and modal parameters of the system. The assumption by Gutierrez that " ...  $T$  is not a function of the parameter  $p$ " [45, p.177] is not necessary as will be shown in the next section. Once  $T$  and its inverse are known, the internal balancing operation transforms the states of the original system  $q$  into the states of the internally balanced system  $\tilde{q}$ .

$$\tilde{q} = Tq \quad (5.25)$$

We recover the original state vector by pre-multiplying the above equation with the inverse of the transformation matrix  $T$ . (Note: Caution is advised, since sometimes in practice this definition is reversed). The original state vector is recovered as:

$$q = T^{-1}\tilde{q} \quad (5.26)$$

Substituting the above equation into the original state space system (A.13), noting that  $T$  is independent of time and pre-multiplying with the transformation matrix  $T$  we obtain the internally balanced state space system as

$$\begin{aligned} \dot{\tilde{q}} &= TA_{zd}T^{-1}\tilde{q} + TB_{zd}d = \tilde{A}_{zd}\tilde{q} + \tilde{B}_{zd}d \\ z &= C_{zd}T^{-1}\tilde{q} = \tilde{C}_{zd}\tilde{q} \end{aligned} \quad (5.27)$$

The tilde superscript will denote the balanced realization in the subsequent development. We can use the S-matrix notation and write:

$$S_{zd} = \begin{bmatrix} A_{zd} & B_{zd} \\ C_{zd} & D_{zd} \end{bmatrix} \Rightarrow \tilde{S}_{zd} = \begin{bmatrix} TA_{zd}T^{-1} & TB_{zd} \\ C_{zd}T^{-1} & D_{zd} \end{bmatrix} \quad (5.28)$$

Looking at the example of a single degree-of-freedom (SDOF) oscillator (2 states), which is subjected to a white-noise random disturbance force, we see that the S-matrix of the original system, see left side of Equation (5.29), is sparse. The S-matrix of the internally balanced system,  $\tilde{S}_{zd}$ , is fully populated with the exception of the zero, which corresponds to the feedthrough term  $\tilde{D}_{zd}$ .

$$S = \begin{bmatrix} 0 & 1 & 0 \\ -1714.3 & -0.42857 & 1.1952 \\ 1.1952 & 0 & 0 \end{bmatrix} \Rightarrow \tilde{S} = \begin{bmatrix} -0.21318 & 41.403 & 0.13134 \\ -41.403 & -0.21539 & 0.13134 \\ 0.13134 & -0.13134 & 0 \end{bmatrix} \quad (5.29)$$

Also we note that the entries of the balanced matrix are closer to each other in magnitude and that symmetries in the entries can be seen. However, neither the  $\tilde{A}_{zd}$ , nor the  $\tilde{S}$  matrix are truly symmetric or skew-symmetric. The next subsection discusses the implications of balancing (similarity transformations) on disturbance and sensitivity analyses.

### 5.2.3 Disturbance and Sensitivity Analysis for Similarity Transformed State Space Systems

This subsection discusses performance and sensitivity analysis for balanced systems. The results are applicable to any similarity transformation. One assumption is that  $J_{z,i} = \sigma_{z,i}$ . The performance (RMS) is computed exactly as for the original system, except that the transformed system matrices (superscript tilde) are used. Based on substitution of the matrices from Equation (5.27) into Equation (A.26), the RMS of the  $i$ -th performance metric of the internally balanced system is obtained as

$$\tilde{\sigma}_{z_i} = \left( \tilde{C}_{zd,i} \Sigma_{\tilde{q}} \tilde{C}_{zd,i}^T \right)^{1/2} = \left( C_{zd,i} T^{-1} \Sigma_{\tilde{q}} (T^{-1})^T C_{zd,i}^T \right)^{1/2} \quad (5.30)$$

Since the internal balancing operation does not change the input-output relationship, but only transforms the internal states of the system, the RMS of the  $i$ -th performance of the original and the transformed system are identical in theory. In practice it was observed that the answers did not match exactly, if  $T$  was computed for a large-order nearly unobservable/uncontrollable system.

We can write:

$$\sigma_{z_i} = \tilde{\sigma}_{z_i} \quad (5.31)$$

Thus by comparing the inner terms of Equation (5.30) with (A.26), we see that the following identities are true:

$$\Sigma_q = T^{-1} \Sigma_{\tilde{q}} (T^{-1})^T, \quad \Sigma_{\tilde{q}} = T \Sigma_q T^T \quad (5.32)$$

Here  $\Sigma_{\tilde{q}}$  is the state covariance matrix of the internally balanced system and is obtained as a solution to the steady-state Lyapunov equation:

$$\tilde{A}_{zd} \Sigma_{\tilde{q}} + \Sigma_{\tilde{q}} \tilde{A}_{zd}^T + \tilde{B}_{zd} \tilde{B}_{zd}^T = 0 \quad (5.33)$$

This can be written as a function of the original system matrices as:

$$T A_{zd} T^{-1} \Sigma_{\tilde{q}} + \Sigma_{\tilde{q}} (T^{-1})^T A_{zd}^T T^T + T B_{zd} B_{zd}^T T^T = 0 \quad (5.34)$$

At this point we recognize that the *state covariance matrix* of the internally balanced system  $\Sigma_{\tilde{q}}$  has a very important and useful property. It is diagonal and contains the Hankel singular values as diagonal elements, thus  $\Sigma_{\tilde{q}} = \Sigma_H$ . Here the subscript “H” stands for “Hankel” singular value matrix and not the Hermitian operator. The proof of this statement is straightforward, since for the internally balanced system the observability and controllability gramians  $\tilde{W}_o$  and  $\tilde{W}_c$  are equal to each other and equal to the Hankel singular value matrix<sup>6</sup>, see [102],[77]. We write

$$\tilde{W}_o = \tilde{W}_c = \Sigma_H \quad (5.35)$$

Furthermore the controllability gramian of the internally balanced system obeys the same Lyapunov equation (5.33) as the state covariance matrix  $\Sigma_{\tilde{q}}$ .

$$\tilde{A}_{zd}\tilde{W}_c + \tilde{W}_c\tilde{A}_{zd}^T + \tilde{B}_{zd}\tilde{B}_{zd}^T = 0 \quad (5.36)$$

From (5.36), (5.35) and (5.33) we conclude that

$$\tilde{W}_c = \Sigma_{\tilde{q}} = \Sigma_H \quad (5.37)$$

This is a very useful property, since it means that the Lyapunov equation for the balanced state covariance matrix does not have to be solved, since the entries (same as  $\Sigma_H$ ) are already known. The sensitivity of the i-th performance RMS with respect to the j-th parameter for an internally balanced system is then computed as follows:

$$\frac{\partial \tilde{\sigma}_{z_i}}{\partial p_j} = \frac{1}{2\tilde{\sigma}_{z_i}} \cdot \frac{\partial \tilde{\sigma}_{z_i}^2}{\partial p_j} \quad (5.38)$$

The RMS in the denominator of the first term is directly substituted from Equation (5.30). The second term is obtained by solving the *transformed governing sensitivity equation* (TGSE). The governing sensitivity equation for a similarity transformed system (e.g. internally balanced) is the most important contribution in this section and was determined to be:

$$\frac{\partial \tilde{\sigma}_{z_i}^2}{\partial p_j} = \text{trace} \left[ T^{-1}\Sigma_{\tilde{q}} (T^{-1})^T \frac{\partial (C_{zd,i}^T C_{zd,i})}{\partial p_j} \right] + \text{trace} \left[ \tilde{L}_i \left\{ T \frac{\partial A_{zd}}{\partial p_j} T^{-1}\Sigma_{\tilde{q}} + \Sigma_{\tilde{q}} (T^{-1})^T \frac{\partial A_{zd}^T}{\partial p_j} T^T + T \frac{\partial (B_{zd} B_{zd}^T)}{\partial p_j} T^T \right\} \right] \quad (5.39)$$

---

<sup>6</sup>The squares of the Hankel singular values are sometimes referred to as the “second order modes” of the system [102].

The TGSE (5.39) allows computing the partial derivative of the variance of the  $i$ -th performance  $J_{z,i} = \sigma_{z,i}$  with respect to the  $j$ -th parameter  $p_j$  using the transformed quantities, including the Lagrange multiplier matrices for the balanced system. The Lagrange multiplier matrix  $\tilde{L}_i$  for the  $i$ -th performance of the internally balanced system obeys the Lyapunov equation

$$\tilde{A}_{zd}^T \tilde{L}_i + \tilde{L}_i \tilde{A}_{zd} + \tilde{C}_{zd,i}^T \tilde{C}_{zd,i} = 0 \quad (5.40)$$

This can be written explicitly in terms of the original system matrices as:

$$(T^{-1})^T A_{zd}^T T^T \tilde{L}_i + \tilde{L}_i T A_{zd} T^{-1} + (T^{-1})^T C_{zd,i}^T C_{zd,i} T^{-1} = 0 \quad (5.41)$$

The matrix derivatives in (5.39) may still be computed using the original (non-transformed) system matrices, where the parameters appear in known locations. At first the simplicity of the Governing Sensitivity Equation (5.39) is surprising, since we expect to find derivative terms of the transformation matrix  $T$  in this equation. If we substitute the transformed matrices  $\tilde{A}_{zd}$ ,  $\tilde{B}_{zd}$ ,  $\tilde{C}_{zd}$  into (A.32), we will obtain non-zero derivative terms of the transformation matrix  $T$  and in general we have:

$$\frac{\partial T}{\partial p_j} \neq 0 \quad (5.42)$$

The mathematical derivation of the TGSE, however, shows that all derivative terms of the transformation matrix,  $T$ , are multiplied by a term which is equal to zero, which leads to Equation (5.39). The proof for the correctness of the TGSE is contained in Appendix B.3. The transformed governing sensitivity equation is correct regardless of whether  $p_j$  is a modal or physical parameter of the system. Finally it shall be noted that balancing does not introduce any approximation and that the values obtained for  $\tilde{\sigma}_{z_i}$  and  $\partial \tilde{\sigma}_{z_i} / \partial p_j$  are identical to the ones for the original system, subject to good numerical conditioning.

### 5.3 Disturbance and Sensitivity Analysis for Reduced Systems

This section presents the extension of the framework to reduced systems. First the process of model reduction by truncation is explained in detail. Next the computation of the RMS and its error bound for reduced systems is derived. Finally the sensitivity and its error bounds are established for reduced LTI systems.

As shown above, the balanced state covariance matrix  $\Sigma_{\tilde{q}}$  is identical to the Hankel singular value matrix  $\Sigma_H$  for the internally balanced system. Thus states that are both highly disturbable (= controllable) and contribute significantly to the performance (= observable) correspond to large Hankel singular values  $\sigma_i^H$ . This property can be invoked for model reduction in the sense that only states with large Hankel singular values are kept in the model. Once the system has been internally balanced we can partition the state space system from Equation (5.27) into states that are going to be “kept” and states that are going to be “removed” as follows:

$$\begin{aligned}\dot{\tilde{q}} &= \tilde{A}_{zd}\tilde{q} + \tilde{B}_{zd}d = \begin{bmatrix} \tilde{A}_{kk} & \tilde{A}_{kr} \\ \tilde{A}_{rk} & \tilde{A}_{rr} \end{bmatrix} \begin{bmatrix} \tilde{q}_k \\ \tilde{q}_r \end{bmatrix} + \begin{bmatrix} \tilde{B}_k \\ \tilde{B}_r \end{bmatrix} d \\ z &= \tilde{C}_{zd}\tilde{q} = \begin{bmatrix} \tilde{C}_k & \tilde{C}_r \end{bmatrix} \begin{bmatrix} \tilde{q}_k \\ \tilde{q}_r \end{bmatrix}\end{aligned}\quad (5.43)$$

Here the subscript “k” indicates states that are “kept” and subscript “r” refers to states that are “removed”. The number of states kept,  $n_k$ , and the number of states removed,  $n_r$ , add up to the original number of states,  $n_s$ , in the state vector  $q$  or  $\tilde{q}$ .

The reduced state space system can then be written as:

$$\begin{aligned}\dot{\tilde{q}} &= \bar{A}_{zd}\tilde{q} + \bar{B}_{zd}d = \bar{A}_{kk}\tilde{q}_k + \bar{B}_k d \\ z &= \bar{C}_{zd}\tilde{q} = \bar{C}_k\tilde{q}_k\end{aligned}\quad (5.44)$$

Here the superscript bar refers to the reduced (= truncated and previously internally balanced) system. The reduction operation to go from the balanced system (5.43) to the reduced system (5.44) can also be represented by a multiplication with a selection matrix  $P$ , defined as:

$$P = \begin{bmatrix} I_{n_k \times n_k} & 0_{n_k \times n_r} \end{bmatrix}\quad (5.45)$$

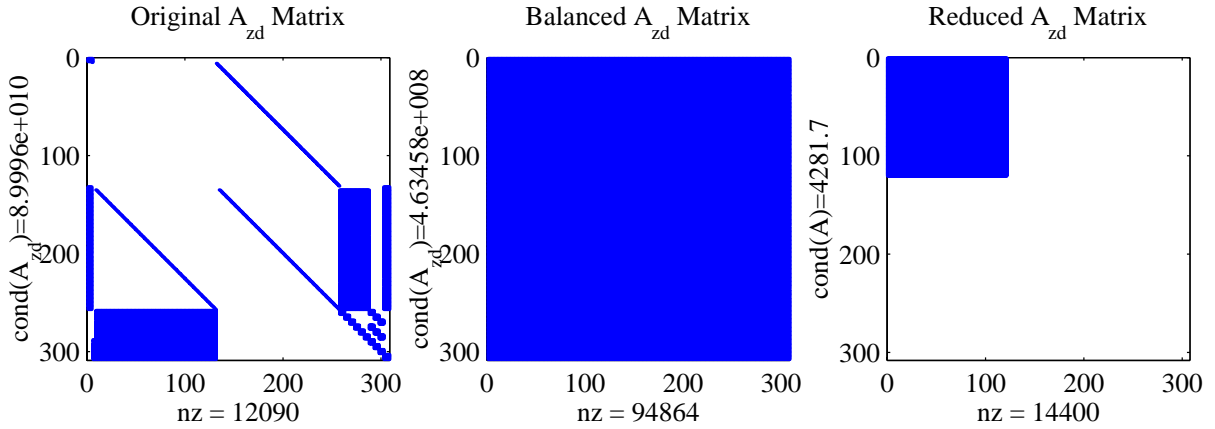
Then the reduced system can be written as:

$$\begin{aligned}\dot{\tilde{q}} &= \bar{A}_{zd}\tilde{q} + \bar{B}_{zd}d = P\tilde{A}_{zd}P^T\tilde{q} + P\tilde{B}_{zd}d = PT A_{zd}T^{-1}P^T\tilde{q} + PT B_{zd}d \\ z &= \bar{C}_{zd}\tilde{q} = \tilde{C}_{zd}P^T\tilde{q} = C_{zd}T^{-1}P^T\tilde{q}\end{aligned}\quad (5.46)$$

where the feedthrough matrix is  $\bar{D}_{zd} = 0$ . The operator  $P$  is a rectangular matrix, which selects the states to be kept. This truncation method is sometimes casually referred to as “brutal” truncation. The reason for this designation is that the effect of the removed states on the DC-gain is lost and the reduced and original system do - in general - not have matching DC-gains.



Static condensation is a more sophisticated reduction method, which is not treated here, but which can achieve matching DC-gains followed by removal of the resulting D-term as shown by Gutierrez [45, p.171]. In general, however, the disturbance and sensitivity analysis results obtained for a truncated model are very good, since “DC” contributes only negligibly to the RMS values and the sensitivities for zero-mean processes. By reducing the system, the number of states is reduced from  $n_s$  to  $n_k$ , while keeping the essential dynamics of the system. The effect of balancing the original system on the state transition matrix,  $A_{zd}$ , and the subsequent truncation operation are illustrated for the SIM Classic model (version 1.0, closed ACS and ODL loops, open FSM loops) in Figure 5.4.



**Figure 5.4:** Sparsity structure of (a) the original SIM-Classic A-matrix (308 states) according to equation (5.20); note cross-coupling terms. (b) Balanced A-matrix (308 states, fully populated) and (c) the reduced A-matrix (120 states, fully populated).

### 5.3.1 RMS and Error Bound for Reduced Systems

First the RMS can be computed using the reduced system matrices directly by substituting Equation (5.44) into Equation (A.26). The RMS performance of the  $i$ -th performance metric is computed as:

$$\bar{\sigma}_{z_i} = (\bar{C}_{zd,i} \Sigma_{\bar{q}} \bar{C}_{zd,i}^T)^{1/2} \quad (5.47)$$

Now we have introduced an approximation, since the matrix  $P$  selects only the  $n_k$  states that have Hankel singular values  $\sigma_i^H$  above a threshold value that we have defined before the reduction step. Thus, it is evident that  $\bar{\sigma}_{z_i} \neq \sigma_{z_i} = \tilde{\sigma}_{z_i}$ , since states have been removed and

so has their contribution to the resulting RMS. We are first interested in finding the error in the RMS value,  $\bar{\sigma}_{z_i}$ , introduced due to the model reduction. The absolute error for the RMS is the difference between the RMS of the original (balanced) and the reduced system. This difference is always a positive quantity, since the variance (RMS squared) is by definition always a positive quantity. This error term, designated as  $\varepsilon_{RMS}$ , is a function of the number of states removed during model reduction,  $n_r$ . Additionally we will develop a simple error bound, which allows prediction of the maximum expected relative error on the RMS based only on the singular values of the states that have been (or will be = a priori estimate) removed.

First the error on the performance RMS can be computed as the difference in the variance of the performance as predicted by the original and reduced models :

$$\begin{aligned}\Delta\sigma_{z_i}^2 &= \sigma_{z_i}^2 - \bar{\sigma}_{z_i}^2 = \tilde{\sigma}_{z_i}^2 - \bar{\sigma}_{z_i}^2 = \tilde{C}_{z_d,i}\Sigma_{\tilde{q}}\tilde{C}_{z_d,i}^T - \bar{C}_{z_d,i}\Sigma_{\bar{q}}\bar{C}_{z_d,i}^T \\ &= \tilde{C}_{z_d,i}\Sigma_{\tilde{q}}\tilde{C}_{z_d,i}^T - \tilde{C}_{z_d,i}P^T\Sigma_{\bar{q}}P\tilde{C}_{z_d,i}^T = \tilde{C}_{z_d,i}(\Sigma_{\tilde{q}} - P^T\Sigma_{\bar{q}}P)\tilde{C}_{z_d,i}^T\end{aligned}\quad (5.48)$$

The expression in round brackets is interesting enough to be analyzed in more depth. We have previously established that the balanced state covariance matrix  $\Sigma_{\tilde{q}}$  is equal to the Hankel singular value matrix  $\Sigma_H$ . We use this fact to rewrite (5.48) as

$$\begin{aligned}\Delta\sigma_{z_i}^2 &= \tilde{\sigma}_{z_i}^2 - \bar{\sigma}_{z_i}^2 = \tilde{C}_{z_d,i}(\Sigma_{\tilde{q}} - P^T\Sigma_{\bar{q}}P)\tilde{C}_{z_d,i}^T = \tilde{C}_{z_d,i}\left(\Sigma_{\tilde{q}} - \begin{bmatrix} \Sigma_{\tilde{q}} & 0^{n_k \times n_r} \\ 0^{n_r \times n_k} & 0^{n_r \times n_r} \end{bmatrix}\right)\tilde{C}_{z_d,i}^T \\ &= \tilde{C}_{z_d,i}\left(\Sigma_H - \begin{bmatrix} \Sigma_k & 0^{n_k \times n_r} \\ 0^{n_r \times n_k} & 0^{n_r \times n_r} \end{bmatrix}\right)\tilde{C}_{z_d,i}^T = \tilde{C}_{z_d,i}diag\left(0^{1 \times n_k}, \underbrace{\sigma_{n_k+1}^H, \dots, \sigma_{n_s}^H}_{\text{removed singular values}}\right)\tilde{C}_{z_d,i}^T \\ &= \tilde{C}_{z_d,i}\Sigma_R\tilde{C}_{z_d,i}^T\end{aligned}\quad (5.49)$$

The matrix  $\Sigma_R$  is thus a diagonal matrix of the same size as the Hankel singular value matrix  $\Sigma_H$ , but with only the removed singular values retained and zeroes everywhere else as defined in Equation (5.63). The original RMS performance can then be written as:

$$\sigma_{z_i} = \tilde{\sigma}_{z_i} = (\bar{\sigma}_{z_i}^2 + (\tilde{\sigma}_{z_i}^2 - \bar{\sigma}_{z_i}^2))^{1/2} = \left(\bar{\sigma}_{z_i}^2 + \tilde{C}_{z_d,i}\Sigma_R\tilde{C}_{z_d,i}^T\right)^{1/2}\quad (5.50)$$

This can be used to obtain a closed form expression for the RMS prediction error due to balanced model reduction.

$$\varepsilon_{RMS} = \Delta\sigma_{z_i} = \tilde{\sigma}_{z_i} - \bar{\sigma}_{z_i} = \left(\bar{\sigma}_{z_i}^2 + \tilde{C}_{z_d,i}\Sigma_R\tilde{C}_{z_d,i}^T\right)^{1/2} - \bar{\sigma}_{z_i}\quad (5.51)$$

The relative error for the RMS  $\sigma_{z_i}$  due to model reduction is obtained by dividing equation (5.51) by the nominal RMS value for the  $i$ -th performance metric  $\sigma_{z_i}$ :

$$\frac{\Delta\sigma_{z_i}}{\sigma_{z_i}} = \frac{\Delta\sigma_{z_i}}{\bar{\sigma}_{z_i}} = 1 - \frac{\bar{\sigma}_{z_i}}{\left(\bar{\sigma}_{z_i}^2 + \tilde{C}_{zd,i}\Sigma_R\tilde{C}_{zd,i}^T\right)^{1/2}} \quad (5.52)$$

It is noteworthy that the reduced model performance  $\bar{\sigma}_{z_i}$  must first be computed before the relative RMS error according to (5.52) can be obtained. It would be useful to have an upper bound for the relative RMS error  $\Delta\sigma_{z_i}/\sigma_{z_i}$  before the model reduction is actually performed. In order to develop this error bound we first consider the algebraic quantities  $a$  and  $b$ , where we assume that  $a > b > 0$ . If we know the difference of the squares of  $a$  and  $b$  we can write

$$a^2 - b^2 = (a + b)(a - b) \quad (5.53)$$

Then the following inequality is true, since  $a > b > 0$ :

$$2a(a - b) > \underbrace{(a + b)(a - b)}_{a^2 - b^2} > 2b(a - b) \quad (5.54)$$

Dividing inequality (5.54) by  $2b$  and only focusing on the two last terms we obtain

$$\frac{a^2 - b^2}{2b} > a - b \quad (5.55)$$

Now we substitute for  $a = \tilde{\sigma}_{z_i}$  and  $b = \bar{\sigma}_{z_i}$  and knowing that the inequality  $\tilde{\sigma}_{z_i} > \bar{\sigma}_{z_i} > 0$  holds, we get an error bound on the RMS for  $\Delta\sigma_{z_i} = \tilde{\sigma}_{z_i} - \bar{\sigma}_{z_i}$  such that

$$\frac{\tilde{\sigma}_{z_i}^2 - \bar{\sigma}_{z_i}^2}{2\bar{\sigma}_{z_i}} > \Delta\sigma_{z_i} \quad (5.56)$$

Dividing by the reduced model performance RMS  $\bar{\sigma}_{z_i}$  yields the desired relative error bound

$$\frac{\Delta\sigma_{z_i}}{\bar{\sigma}_{z_i}} < \frac{\tilde{\sigma}_{z_i}^2 - \bar{\sigma}_{z_i}^2}{2\bar{\sigma}_{z_i}^2} \quad (5.57)$$

At this point we will attempt to express the inequality in terms of the singular values of the system. Substituting (5.49) in the numerator and (5.47) in the denominator of (5.57) we can write

$$\frac{\Delta\sigma_{z_i}}{\bar{\sigma}_{z_i}} < \frac{\tilde{C}_{zd,i}\Sigma_R\tilde{C}_{zd,i}^T}{2 \cdot \tilde{C}_{zd,i}\Sigma_{\bar{q}}\tilde{C}_{zd,i}^T} = \frac{1}{2} \frac{\tilde{C}_{zd,i}\Sigma_R\tilde{C}_{zd,i}^T}{\tilde{C}_{zd,i}P^T\Sigma_{\bar{q}}P\tilde{C}_{zd,i}^T} = \frac{1}{2} \frac{\tilde{C}_{zd,i}\text{diag}(0^{1 \times n_k}, \sigma_{n_k+1}^H, \dots, \sigma_{n_s}^H)\tilde{C}_{zd,i}^T}{\tilde{C}_{zd,i}\text{diag}(\sigma_1^H, \dots, \sigma_{n_k}^H, 0^{1 \times n_r})\tilde{C}_{zd,i}^T} \quad (5.58)$$

Taking the trace operator of both sides of (5.58) is allowed, since they represent scalars. The  $C$  matrices cannot be eliminated inside the trace operator, since they are not square matrices. Thus a tight error bound on the relative RMS error is given by:

$$\frac{\Delta\sigma_{z_i}}{\bar{\sigma}_{z_i}} < \frac{1}{2} \frac{\text{trace} \left[ \tilde{C}_{zd,i} \Sigma_R \tilde{C}_{zd,i}^T \right]}{\text{trace} \left[ \tilde{C}_{zd,i} \Sigma_K \tilde{C}_{zd,i}^T \right]} \quad (5.59)$$

The computation of this tight error bound involves taking the trace of the expressions in the numerator and denominator of (5.59). Note that all the information in (5.59) is available before the RMS of the reduced system  $\bar{\sigma}_{z_i}$  is actually computed. A more elegant, but also more *conservative error bound* is achieved, when we realize that the magnitude of the entries in the  $\tilde{C}_{zd,i}$ -matrix (it actually is a vector, since it only comprises the  $i$ -th row of  $\tilde{C}_{zd}$ ) is decreasing in a way such that the following inequality generally holds true:

$$\frac{\text{trace} \left[ \tilde{C}_{zd,i} \Sigma_R \tilde{C}_{zd,i}^T \right]}{\text{trace} \left[ \tilde{C}_{zd,i} \Sigma_K \tilde{C}_{zd,i}^T \right]} < \frac{\text{trace} \left[ \Sigma_R \right]}{\text{trace} \left[ \Sigma_K \right]} \quad (5.60)$$

A more compact error bound for the relative RMS error due to model reduction is:

$$\frac{\Delta\sigma_{z_i}}{\bar{\sigma}_{z_i}} < \frac{1}{2} \cdot \frac{\sum_{i=k+1}^n \sigma_i^H}{\sum_{i=1}^k \sigma_i^H} \quad (5.61)$$

This error bound is useful for determining the number of states  $n_k$  to be kept during model reduction in order to achieve a desired accuracy of the RMS prediction. The error term  $\varepsilon_{RMS}$  is computed as follows:

$$\varepsilon_{RMS} = \Delta\sigma_{z_i} = \tilde{\sigma}_{z_i} - \bar{\sigma}_{z_i} = \left( \bar{\sigma}_{z_i}^2 + \tilde{C}_{zd,i} \Sigma_R \tilde{C}_{zd,i}^T \right)^{1/2} - \bar{\sigma}_{z_i} \quad (5.62)$$

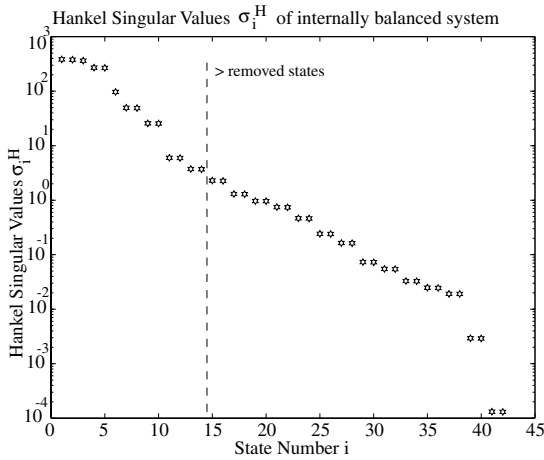
We see that we must save the unreduced, balanced matrix  $\tilde{C}_{zd}$  if we want to obtain an exact value for  $\varepsilon_{RMS}$ . Here the diagonal matrix  $\Sigma_R \in \mathbb{R}^{n_s \times n_s}$  contains only the removed Hankel



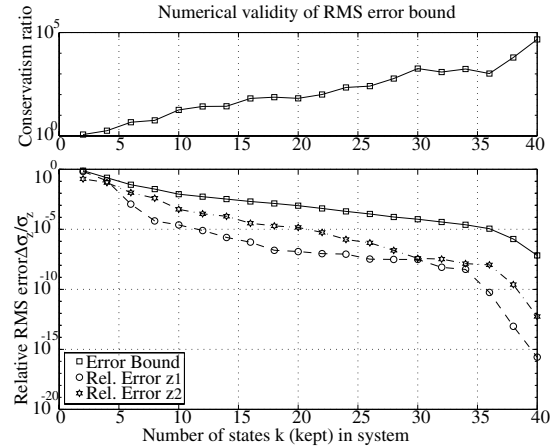
for all values of  $n_k$  between 1 (all states except one truncated) and  $n_s - 1$  (only one state truncated), since the Hankel singular values  $\sigma_i^H$  are known<sup>7</sup>. The correct value for  $n_k$  is then the smallest number of states, which still meets the above inequality. Also the error bound is less conservative than the model reduction bound typically used in the literature, see Equation (5.66) from Reference [143, p. 159].

$$\|G(s) - \bar{G}(s)\|_\infty \leq 2 \sum_{i=n_k+1}^{n_s} \sigma_i^H \quad (5.66)$$

To illustrate this, the Hankel singular values, i.e. the diagonal elements of  $\Sigma_H$ , for a multi-DOF system, as presented in Subsection 5.4.2, are shown graphically in Figure 5.5(a). For the MDOF example an allowable RMS error  $< 0.5\%$  leads to a truncation level  $n_k = 14$  (dashed vertical line), where the original system had 42 states. Further details for this example are shown in Subsection 5.4.2.



(a)



(b)

**Figure 5.5:** (a) Hankel singular values and model truncation threshold level  $n_k$  for RMS error  $< 0.5\%$ . The original MDOF model has 42 states, of which 14 must be kept in the reduced model. (b) Validity of RMS error bound for various truncation levels according to Inequality (5.65)

The relative RMS error (MDOF example) as a function of the truncation level,  $n_k$ , is shown in Figure 5.5(b). It can be seen that the conservative error bound from Inequality (5.65) is valid. The conservatism ratio, defined as the ratio of error bound (5.65) over the actual relative RMS error, was observed to decrease from 46794 with only 2 states removed to

<sup>7</sup>In practice we remove 2 states at a time, since 2 states are required to represent an open-loop structural mode or a complex conjugate pair of poles.

a ratio of 1.169 with only 2 states kept in the system. This means that the RMS error bound (5.65) becomes increasingly accurate the more states are truncated.

### 5.3.2 Sensitivity and Error Bound for Reduced System

Having computed the RMS performance  $J_{z,i} \approx \bar{\sigma}_{z_i}$  and associated error for the balanced reduced system, we want to obtain the sensitivity  $\partial \bar{\sigma}_{z_i} / \partial p_j$  and its error. The derivation of the sensitivity for the reduced, balanced system is straightforward with Equation (5.39) as a starting point. This is due to the fact that the selection matrix  $P$  is not a function of any system parameters  $p_j$ , since it contains only ones and zeros. The sensitivity of a reduced, internally balanced state space system is given as:

$$\frac{\partial \bar{\sigma}_{z_i}}{\partial p_j} = \frac{1}{2\bar{\sigma}_{z_i}} \cdot \frac{\partial \bar{\sigma}_{z_i}^2}{\partial p_j} \quad (5.67)$$

Here the partial derivative with respect to the variance is obtained from the transformed governing sensitivity equation (TGSE) with the reduced system matrices from (5.44) substituted in.

$$\frac{\partial \bar{\sigma}_{z_i}^2}{\partial p_j} = \text{trace} \left[ \Sigma_{\bar{q}} \frac{\partial \left( \bar{C}_{zd,i}^T \bar{C}_{zd,i} \right)}{\partial p_j} \right] + \text{trace} \left[ \bar{L}_i \left\{ \frac{\partial \bar{A}_{zd}}{\partial p_j} \Sigma_{\bar{q}} + \Sigma_{\bar{q}} \frac{\partial \bar{A}_{zd}^T}{\partial p_j} + \frac{\partial \left( \bar{B}_{zd} \bar{B}_{zd}^T \right)}{\partial p_j} \right\} \right] \quad (5.68)$$

After taking into account (5.39) and some matrix algebra we obtain

$$\begin{aligned} \frac{\partial \bar{\sigma}_{z_i}^2}{\partial p_j} = & \text{trace} \left[ T^{-1} P^T \Sigma_{\bar{q}} P (T^{-1})^T \frac{\partial \left( C_{zd,i}^T C_{zd,i} \right)}{\partial p_j} \right] + \\ & \text{trace} \left[ \bar{L}_i \left\{ P T \frac{\partial A_{zd}}{\partial p_j} T^{-1} P^T \Sigma_{\bar{q}} + \Sigma_{\bar{q}} P (T^{-1})^T \frac{\partial A_{zd}^T}{\partial p_j} T^T P^T + P T \frac{\partial \left( B_{zd} B_{zd}^T \right)}{\partial p_j} T^T P^T \right\} \right] \end{aligned} \quad (5.69)$$

The last Equation (5.69) shows that the sensitivity of a reduced, internally balanced system can be calculated, if the matrix partial derivatives  $\partial A_{zd} / \partial p_j$ ,  $\partial B_{zd} / \partial p_j$ ,  $\partial C_{zd} / \partial p_j$  of the original system<sup>8</sup>, the balancing transformation matrix  $T$  and the selection matrix  $P$  are known. This is an important result, since it was previously not possible to compute this quantity due to the fact that the balancing transformation and model reduction removes the explicit dependency of the parameters  $p_j$  in the system matrices  $\bar{A}_{zd}$ ,  $\bar{B}_{zd}$  and  $\bar{C}_{zd}$ , as mentioned by Gutierrez [45, p.176].

<sup>8</sup>These derivatives are identical to the ones computed in the previous framework [45].

The Lagrange multiplier matrix for the reduced system obeys:

$$\bar{A}_{zd}^T \bar{L}_i + \bar{L}_i \bar{A}_{zd} + \bar{C}_{zd,i}^T \bar{C}_{zd,i} = 0 \quad (5.70)$$

In the case where there is only one performance metric  $i = 1$ , i.e.  $\bar{C}_{zd} \in \mathbb{R}^{1 \times n_k}$ , the Lagrange multiplier matrix  $\bar{L}_i$ , is equal to the diagonal matrix of Hankel singular values  $\Sigma_H^{n_k \times n_k}$  that are kept in the system,  $\bar{L}_i = \Sigma_H^{n_k \times n_k}$ , where

$$\Sigma_H^{n_k \times n_k} = \begin{bmatrix} \sigma_1^H & & \\ & \ddots & \\ & & \sigma_{n_k}^H \end{bmatrix} \quad (5.71)$$

Consider that in this particular case the solution to the Lyapunov equation (5.70) is already known, since there is only one performance metric. In this instance it is actually computationally cheaper to compute the sensitivity for a balanced system (reduced or unreduced) than for an unbalanced system, provided the transformation matrix  $T$  is already known. A computational benefit also exists in the general case (of several performances), since we can now solve for the Lagrange multiplier matrices for the reduced system according to (5.70). The more performance metrics we have, the larger we expect that benefit to be. This benefit was quantified in Section 5.1, see Equation (5.8).

As was the case for the computation of the reduced system RMS performance  $\bar{\sigma}_{z_i}$ , we can expect that the reduced system sensitivity  $\partial \bar{\sigma}_{z_i} / \partial p_j$  will be in error due to the removal of system states. A sensitivity error bound similar to the error bound for the performance RMS (5.65) has been derived in Appendix B.4.

$$\left| \frac{\Delta \frac{\partial \sigma_{z_i}}{\partial p_j}}{\frac{\partial \bar{\sigma}_{z_i}}{\partial p_j}} \right| < \frac{\sum_{i=1}^{n_s} (\sigma_i^H)^2}{\sum_{i=1}^{n_k} (\sigma_i^H)^2} - 1 + \frac{1}{2} \cdot \frac{\sum_{i=n_k+1}^{n_s} \sigma_i^H}{\sum_{i=1}^{n_k} \sigma_i^H} \quad (5.72)$$

The ratio involving the squared singular values on the right side is generally very close to 1, since the Hankel singular values that are kept in the system are often larger by several orders of magnitude than the singular values that have been removed. In practice it is then observed that the relative error bound on the sensitivity (5.72) is nearly identical to the relative error bound on the RMS itself according to (5.65). This bound involves the same ratio of Hankel singular values as for the relative RMS error in addition to a ratio of the Hankel singular values squared.



Again the sensitivity error bound was shown to be valid for the MDOF example for all truncation levels in Figure 5.10(b), see Subsection 5.4.2. The sensitivity error bound is significantly less conservative than the RMS error bound and the conservatism ratio (ratio of error bound to actual maximum sensitivity error) is independent of the truncation level. It shall be noted that in this case the error bound is on the absolute value of the relative sensitivity error, since the sensitivity  $\partial\bar{\sigma}_{z_i}/\partial p_j$  could be positive or negative.

## 5.4 Verification and Examples

In order to validate the equations for the performance (RMS) and sensitivity of internally balanced and reduced systems from the previous sections, a number of problems were computed numerically. These examples include the single degree-of-freedom oscillator, a multi-degree-of-freedom (MDOF) system and a large-order SIM closed-loop model.

### 5.4.1 Single Degree of Freedom Oscillator

The single-degree-of-freedom (SDOF) oscillator is the simplest dynamic mechanical system. It was introduced in Section 2.1. Ignoring the disturbance and control filters here, the system consists of a discrete mass  $m$ , which is attached to a fixed boundary condition via a linear spring of stiffness  $k$  and a dashpot with damping constant  $c$ . The system is driven with a zero-mean unit-intensity white noise disturbance force  $d$ . We want to determine the root-mean-square (RMS) value of the displacement response  $x$  of the system and the sensitivity of this RMS with respect to the system parameters.

We have arbitrarily chosen the nominal physical parameters  $m = 0.7$  [kg],  $k = 1200$  [N/m] and  $c = 0.3$  [Ns/m], which results in the following modal parameters:

$$\begin{aligned}
 \text{Modal frequency : } \omega &= \sqrt{k/m} = 41.4 \text{ [rad/sec]} = 2\pi f \Rightarrow f = 6.6 \text{ [Hz]} \\
 \text{Modal damping : } \zeta &= \frac{c}{2\sqrt{km}} = 0.0052 \text{ [-]} \\
 \text{Modal mass : } \hat{m} &= {}^o\phi^T m {}^o\phi = \frac{1}{\sqrt{m}} \cdot m \cdot \frac{1}{\sqrt{m}} = 1 \text{ [-]} \\
 \text{Mode shape : } {}^o\phi &= \frac{1}{\sqrt{m}} = 1.195 \text{ [kg}^{-1/2}]
 \end{aligned} \tag{5.73}$$

The closed form expression for the RMS was found to be equal to:

$$\sigma_z = \left( \frac{1}{2\pi} \int_{-\infty}^{+\infty} |G_{zd}(j\omega)|^2 d\omega \right)^{1/2} = (C_{zd}\Sigma_q C_{zd}^T)^{1/2} = \sqrt{\frac{{}^o\phi^4}{4\hat{m}^2\zeta\omega^3}} = \frac{1}{\sqrt{2ck}} \quad (5.74)$$

The physical mass,  $m$ , drops out of the expression for the RMS due to the assumption of proportional damping. Substituting the numerical values  $c = 0.3$  and  $k = 1200$  in (5.74) we obtain a RMS value of  $\sigma_z = 0.037268$  [m]. This result is used as a reference for the numerical answers. Now that we have an expression for  $\sigma_z$ , we want to compute the sensitivity with respect to the different parameters of the system. For large order systems a closed form expression for  $\sigma_{z_i}$ , such as (5.74), cannot be obtained and the numerical method, which employs Equation (5.39) has to be used. Next the RMS was computed for the internally balanced SDOF system using (5.30). As expected we obtain the same answer:  $\tilde{\sigma}_z = \sigma_z = 0.037268$  [m]. Internal balancing does not affect the performance for the SDOF oscillator model. Subsequently the sensitivities of the RMS with respect to the physical parameters  $m$ ,  $c$  and  $k$  and the modal parameters  $\omega$ ,  $\zeta$ ,  $\hat{m}$ ,  ${}^o\phi$  were computed for the balanced system using the *Transformed Governing Sensitivity Equation* (5.39). The results, including a finite difference approximation, are shown and compared to the original system in Table 5.3.

**Table 5.3:** Comparison of RMS sensitivities for single degree-of-freedom (SDOF) physical parameters (top) and modal parameters (bottom), including a 1% finite difference (fd) approximation. Normalized Sensitivities are  $s_{norm}$  and  $\tilde{s}_{norm}$  for the balanced system.

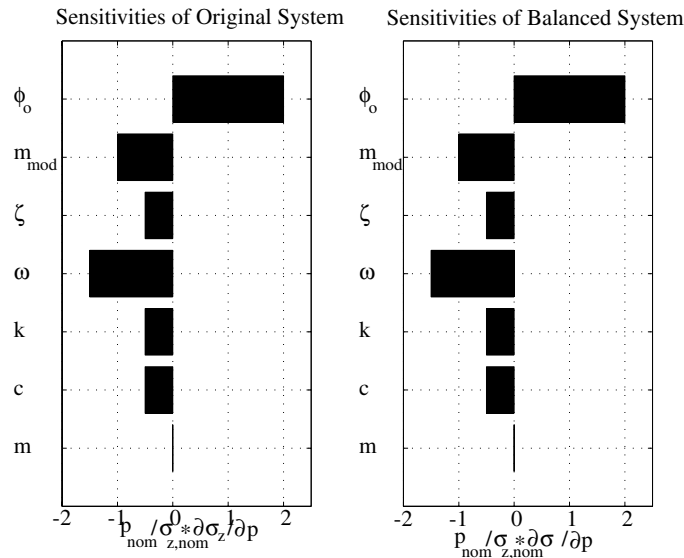
$p$	$\frac{\Delta\sigma_z}{\Delta p}$ (fd)	$\frac{\partial\sigma_z}{\partial p}$	$s_{norm}$	$\frac{\partial\tilde{\sigma}_z}{\partial p}$	$\tilde{s}_{norm}$
$m$	0	0	0	$-5.7 \cdot 10^{-16}$	0
$c$	-0.061651	-0.062113	-0.5	-0.062113	-0.5
$k$	$-1.54 \cdot 10^{-5}$	$-1.55 \cdot 10^{-5}$	-0.5	$-1.55 \cdot 10^{-5}$	-0.5
$\omega$	-0.001334	-0.001350	-1.5	-0.001350	-1.5
$\zeta$	-3.5736	-3.6004	-0.5	-3.6004	-0.5
$\hat{m}$	-0.036899	-0.037268	-1.0	-0.037268	-1.0
${}^o\phi$	0.062673	0.062361	2	0.062361	2

It is difficult to directly compare the sensitivities of various parameters with each other,

since they can vary by several orders of magnitude and are generally in different units. Therefore normalization has been proposed by Gutierrez [45, p.112] according to:

$$s_{norm} = \frac{p_{nom}}{(\sigma_z)_{nom}} \cdot \frac{\partial \sigma_z}{\partial p} = \frac{\frac{\partial \sigma_z}{(\sigma_z)_{nom}}}{\frac{\partial p}{p_{nom}}} \approx \frac{\% \text{ change in } \sigma_z}{\% \text{ change in } p} \quad (5.75)$$

This quantity can be interpreted as the percent change in the performance RMS for a given percent change in the value of the parameter  $p$  in the vicinity of the nominal parameter value  $p_{nom}$ . This is used in Figure 5.6 to compare the normalized sensitivities of all SDOF oscillator parameters with each other. As expected, the normalized sensitivity with respect to the (physical) mass  $m$  is zero. This is consistent with (5.74). The normalized sensitivities for all other parameters are negative with exception of the mode shape ("displacement")  $\phi_o$ . Intuitively this is satisfying. If we increase the damping ratio,  $\zeta$ , for example, we will expect a decrease in the displacement RMS of the mass, therefore the associated sensitivity has to be negative. Also an increase in the spring stiffness  $k$  or the damping coefficient  $c$  will lead to a smaller RMS value.



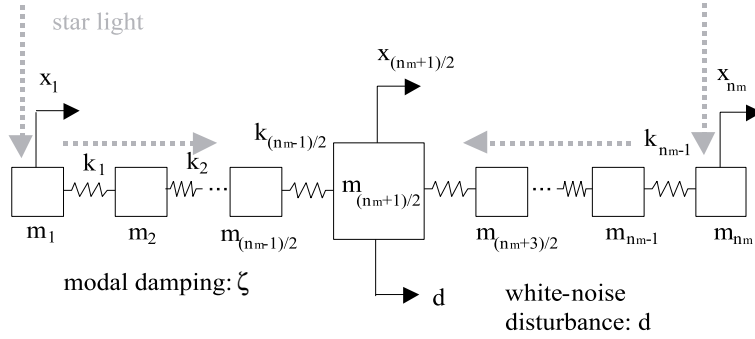
**Figure 5.6:** Comparison of normalized sensitivities for SDOF system parameters (from top to bottom). Modal parameters:  $\phi_o$  mode shape,  $m_{mod} = \hat{m}$  modal mass,  $\zeta$  modal damping ratio,  $\omega$  modal frequency. Physical Parameters:  $k$  spring stiffness,  $c$  damping coefficient and  $m$  (physical) mass.

Figure 5.6 also shows that the normalized sensitivities in the unbalanced (left subplot) and

balanced case (right subplot) are equal. We can conclude that the extension of the performance prediction and sensitivity analysis framework to balanced systems as presented in Section 5.2 has been validated for single degree-of-freedom, single mode systems. The next section will “inductively” extend the proof to multi-degree-of-freedom systems. Model reduction cannot be investigated for the SDOF case, since both states are needed to represent the resonance.

### 5.4.2 Multiple Degree-of-Freedom Problem

A multiple degree-of-freedom (MDOF) problem extends the verification of the balanced and reduced sensitivity analysis. Figure 5.7 shows a structure consisting of  $n_m$  masses that are interconnected by linear springs of stiffness  $k_i$ . The stiffness of the first spring is computed from the relationship  $k_1 = EA/l_o$  for the axial stiffness of a rod. Each following spring constant is a multiple of the first one such that  $k_i = i \cdot k_1$ , where  $i \in [1, n_m - 1]$ . Both ends of the chain have a free boundary condition. The central mass is  $m_{hub} = m_{(n_m+1)/2}$  and is significantly larger than the other masses,  $m_{arm}$ . It is assumed that a random white-noise disturbance force  $d$  of unit-intensity  $S_{dd} = 1$  acts on the hub mass. This arrangement could represent a simplified interferometer model.



**Figure 5.7:** Mass-spring model of a multi-degree-of-freedom (MDOF) structure representing an interferometer with free-free boundary conditions.

The first performance metric,  $J_{z,1} = RMS(z_1)$ , is the RMS of the displacement of the hub mass itself, whereas the second performance,  $J_{z,2} = RMS(z_2)$ , is the optical pathlength difference (OPD) between the left and right arm. The performance matrix,  $C_{z,x}$ , relates the

physical degrees of freedom  $x$  to the performances  $z$ .

$$z = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & \cdots & 0 & \cdots & 1 & \cdots & 0 & \cdots & 0 \\ -1 & \cdots & 0 & \cdots & 2 & \cdots & 0 & \cdots & -1 \end{bmatrix}}_{C_{zx}} \begin{bmatrix} x_1 \\ \vdots \\ x_{(n_m+1)/2} \\ \vdots \\ x_{n_m} \end{bmatrix} \quad (5.76)$$

Note that the number of masses,  $n_m$ , is always an odd number. We arbitrarily choose  $n_m = 21$ . The stiffness  $k_1$  is computed as  $k_1 = 3.55 \cdot 10^4$  [N/m], the hub mass is chosen to be  $m_{hub} = 120$  [kg], the appendage mass is  $m_{arm} = 10$  [kg] and the modal damping ratio is set to  $\zeta = 0.005$  for all modes. The equations of motion are then written as:

$$\underbrace{\begin{bmatrix} m_{arm} & & & & \\ & \ddots & & & \\ & & m_{hub} & & \\ & & & \ddots & \\ & & & & m_{arm} \end{bmatrix}}_M \ddot{x} + \underbrace{\begin{bmatrix} 1 & -1 & & & \\ -1 & 3 & -2 & & \\ & -2 & \ddots & \ddots & \\ & & \ddots & (n-1) + (n-2) & -(n-1) \\ & & & -(n-1) & (n-1) \end{bmatrix}}_K x = \underbrace{\begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix}}_{\beta_d} d \quad (5.77)$$

After solving the generalized eigenvalue problem  $[K - \omega_j^2 M] \phi_j = 0$  the system is written in orthonormal second order modal form as follows:

$$\dot{q}_p = \underbrace{\begin{bmatrix} O & I \\ -\Omega^2 & -2Z\Omega \end{bmatrix}}_{A_{zd}} q_p + \underbrace{\begin{bmatrix} 0 \\ \hat{M}^{-1} {}^o\Phi^T \beta_d \end{bmatrix}}_{B_{zd}} d \quad (5.78)$$

$$z = \underbrace{\begin{bmatrix} C_{zx} {}^o\Phi & 0 \end{bmatrix}}_{C_{zd}} q_p$$

where  $\Omega$ ,  $Z$ ,  ${}^o\Phi$ ,  $\hat{M}$ ,  $\beta_d$ ,  $C_{zx}$  are the modal frequency matrix, the modal damping matrix, the mass-normalized modeshape matrix, the mass-normalized modal mass matrix, the disturbance influence coefficient matrix and the performance influence coefficient matrix respectively.

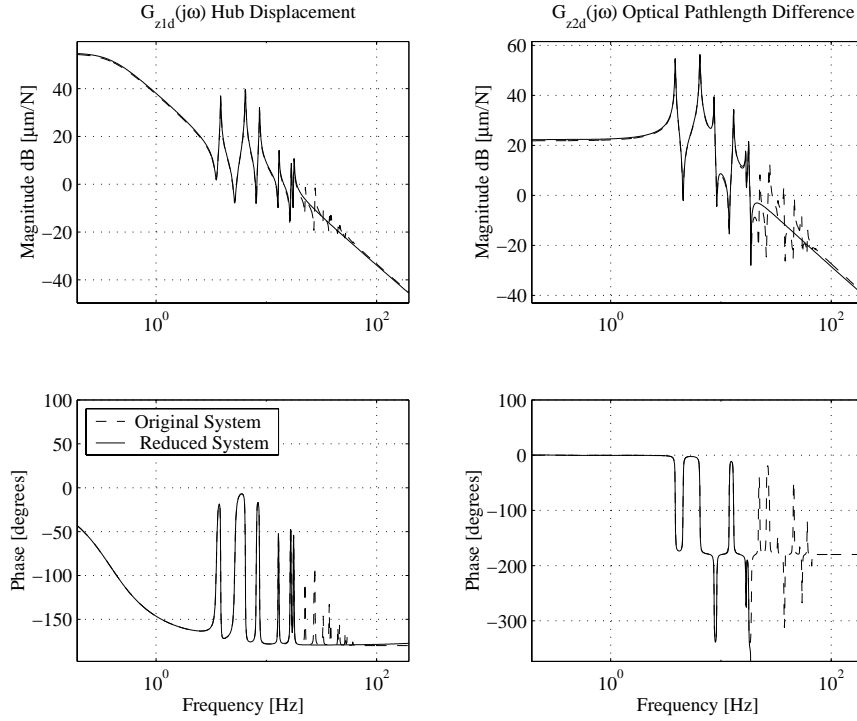
The RMS performances for the system in (5.78) are computed according to (A.25). Next the system is internally balanced according to the procedure in Section 5.2. Again the performances are computed according to Equation (5.30). Next model reduction is performed using the procedure outlined in Section 5.3. The error bound (5.65) is used with a tolerance of 0.5% on the RMS for both performances. The RMS values for the reduced system are computed according to (5.47). The resulting truncation threshold,  $n_k$ , says that only 14 out of the 42 states of the original system need to be kept as shown in Figure 5.5(a). The RMS results for the original, the balanced and the reduced system are shown in Table 5.4.

**Table 5.4:** Comparison of RMS results [in  $\mu m$ ] for MDOF example.

model	# of states	cond(A)	$\sigma_{z_1}$	$\Delta\sigma_{z_1}/\sigma_{z_1}$	$\sigma_{z_2}$	$\Delta\sigma_{z_2}/\sigma_{z_2}$
original	42	281848.16	496.4692	0	389.9561	0
balanced	42	303.5706	496.4692	0	389.9561	0
reduced	14	68.9907	496.4682	$2.125 \cdot 10^{-6}$	389.9088	0.0001213

The RMS comparison shows that the RMS for the original and balanced system are equal as expected. Removing (42-14=28 states) causes a relative RMS error of 0.0002125 % for  $z_1$  and 0.01213 % for  $z_2$ , which is below the 0.5% error bound. Also it can be seen that the reduced system is much better conditioned than the original system. Figure 5.8 contains the disturbance to performance transfer functions for both performances. The full order system (42 states) and reduced order model (14 states) are shown. It can be seen that the RMS of the hub displacement is mainly driven by the low-frequency (rigid body) behavior of the system. We expect the sensitivities with respect to modal parameters to be small for  $z_1$ . In the second performance  $z_2$  the rigid body mode is not observable and the flexible dynamics of modes 2, 3 and 4 dominate the RMS response. This could be confirmed with a frequency-domain performance analysis and cumulative RMS plots as presented by Gutierrez [45]. We expect the modal parameter sensitivities for these critical modes to be large.

The normalized modal sensitivities for the MDOF problem were computed using Equations (5.67-5.71) for the modal frequencies  $\omega_j$ , modal damping  $\zeta_j$  and modal mass  $\hat{m}_j$ . The modes  $j = 2, 3, \dots, 10$  were taken into consideration as shown in the horizontal bar charts of Figure 5.9 for performance  $z_1$ .

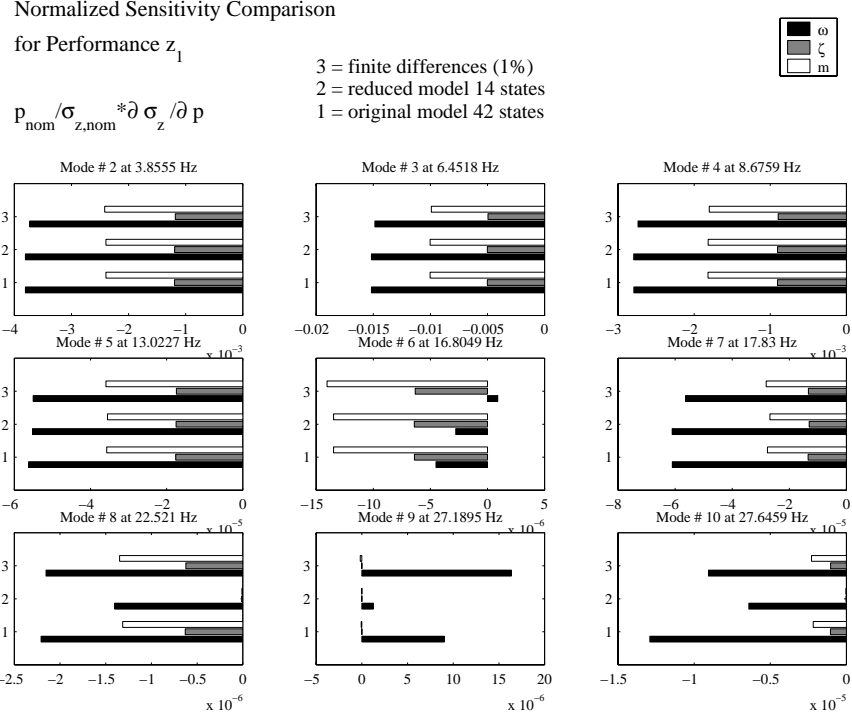


**Figure 5.8:** Disturbance to performance transfer functions for MDOF example. Left column:  $z_1$ . Right column:  $z_2$ . The reduced transfer function (14 states - solid line) is shown along with the original unreduced system (42 states - dashed line).

Three cases are compared: (1) original full order model (42 states), (2) reduced model (14 states) and (3) a 1% finite difference approximation. The results are seen to match well up to Mode 7. For higher modes two important phenomena are observed.

First we see that the modal sensitivities of the reduced system do not match the (correct) sensitivities of the full order system. This is *due to the fact that the states, which correspond to the modes of interest have been truncated* (7 modes = 14 states). It is thus not meaningful to try and compute the modal sensitivities of a mode which is no longer represented in the system dynamics. Secondly the finite difference approximation starts to break down for higher order modes. As indicated by Gutierrez [45], it has been observed empirically that the errors of finite difference approximations of the sensitivity increases with model order due to numerical ill-conditioning. The normalized sensitivity comparison for the second performance  $z_2$  is shown in Figure 5.10(a).

As expected the normalized modal sensitivities (modes  $j=2$  to 5) for the second perfor-



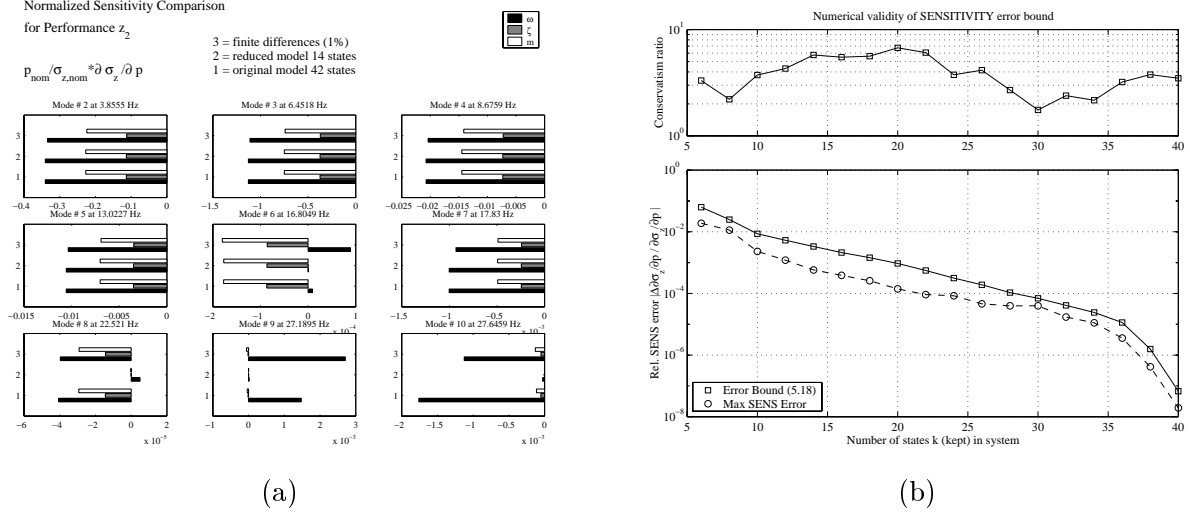
**Figure 5.9:** (MDOF) Normalized modal sensitivity comparison for performance 1

performance  $z_2$  are larger than for  $z_1$ . Subsequently the validity of the sensitivity error bound (5.72) was verified numerically by comparing the absolute value of the relative sensitivity error  $|\Delta(\partial\sigma_{z_i}/\partial p_j)/(\partial\bar{\sigma}_{z_i}/\partial p_j)|$  with the conservative error bound from (5.72). Figure 5.10(b) shows that the conservative error bound on the sensitivity from (5.72) is valid in the MDOF case. The maximum relative sensitivity error for the modal parameters of the three dominant modes 2, 3 and 4 is plotted and compared to the upper error bound from (5.72). It is interesting to observe that the error bound (5.72) is significantly tighter than the RMS error bound (5.65). The conservatism ratio<sup>9</sup> lies between a value of 1.75 and 6.73 and does not show the same dependence on the truncation level,  $n_k$ , as the RMS error bound did.

So far the sensitivity computations for the balanced and reduced MDOF model have only been validated for modal parameter sensitivities. We need to verify that the physical parameter sensitivities for the reduced model would still be accurate within the error bound (5.72). This is presumably true since the contribution to the sensitivity is captured in the critical modes ( $j = 1, 2, \dots, 7$ ) via the chain rule as shown in (5.79-5.81).

<sup>9</sup>Ratio of sensitivity error bound (5.72) over the actual maximum sensitivity error computed with (B.58).





**Figure 5.10:** (a) Normalized Sensitivity comparison for performance 2. (b) Numerical validity of relative sensitivity error bound (5.72) in the MDOF case. The maximum sensitivity error (Max SENS Error) is taken from the set of modal parameter sensitivities for performances  $i = 1, 2$  and modes  $j = 2, 3, 4$

The physical parameter sensitivities in the MDOF example are computed with respect to the mass of the hub,  $m_{hub}$ , the mass parameter of the arms,  $m_{arm}$ , and the spring stiffness parameter,  $k = k_1$ . The physical parameter sensitivities are significantly more expensive to compute than the modal parameter sensitivities. This is due to the fact that the physical parameters  $m_{hub}$ ,  $m_{arm}$  and  $k$  do not explicitly appear in the matrix derivatives shown in Equation (A.32). The derivatives with respect to physical parameters of the system matrices are computed as:

$$A_{zd} \rightarrow A_{zd}(\Omega(p)) \quad \text{and} \quad \frac{\partial A_{zd}}{\partial p} = \sum_{j=1}^N \left( \frac{\partial A_{zd}}{\partial \omega_j} \cdot \frac{\partial \omega_j}{\partial p} \right) \quad (5.79)$$

$$B_{zd} \rightarrow B_{zd}(\hat{M}(p), {}^o\Phi(p)) \quad \text{and} \quad \frac{\partial B_{zd}}{\partial p} = \sum_{j=1}^N \left( \frac{\partial B_{zd}}{\partial \hat{m}_j} \cdot \frac{\partial \hat{m}_j}{\partial p} + \sum_{i=1}^{n_m} \left( \frac{\partial B_{zd}}{\partial {}^o\phi_{ij}} \cdot \frac{\partial {}^o\phi_{ij}}{\partial p} \right) \right) \quad (5.80)$$

$$C_{zd} \rightarrow C_{zd}({}^o\Phi(p)) \quad \text{and} \quad \frac{\partial C_{zd}}{\partial p} = \sum_{j=1}^N \sum_{i=1}^{n_m} \left( \frac{\partial C_{zd}}{\partial {}^o\phi_{ij}} \cdot \frac{\partial {}^o\phi_{ij}}{\partial p} \right) \quad (5.81)$$

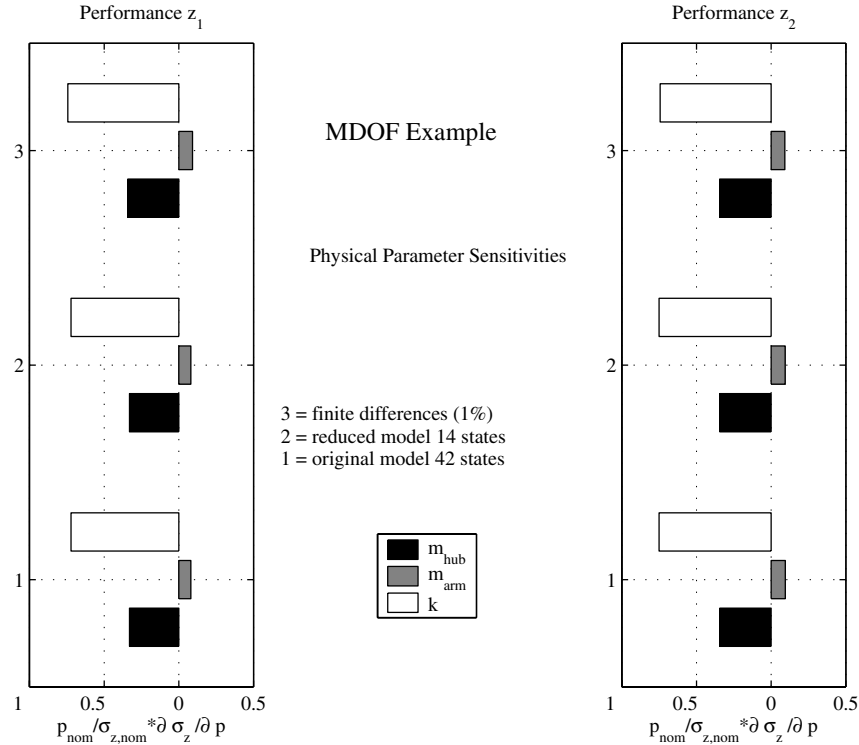
where  $N$  is the number of modes and  $n_m$  is the number of masses (equal to the number of degrees-of freedom).

Inside the expressions (5.79)-(5.81) the derivatives of the modal frequency, mode shapes and modal mass with respect to the physical parameter  $p$  are required. These derivatives have

been computed with the methods suggested by Gutierrez [45, p.126-133], including Nelson's method [107] for the mode shape derivatives  $\partial^o \phi_{ij} / \partial p$ . The derivatives of the global mass matrix  $\partial M / \partial p$  and stiffness matrix  $\partial K / \partial p$  are required and can be easily computed since the parameters  $m_{hub}$ ,  $m_{arm}$  and  $k$  appear explicitly in (5.77). The governing sensitivity equation of the internally balanced and reduced system from (5.69) is implemented as follows:

$$\begin{aligned} \frac{\partial \bar{\sigma}_{z_i}^2}{\partial p_j} = & \text{trace} \left[ T^{-1} P^T \Sigma_{\bar{q}} P (T^{-1})^T \left\{ \left( \frac{\partial C_{zd,i}}{\partial p_j} \right)^T C_{zd,i} + C_{zd,i}^T \frac{\partial C_{zd,i}}{\partial p_j} \right\} \right] + \\ & \text{trace} \left[ \bar{L}_i \left\{ P T \frac{\partial A_{zd}}{\partial p_j} T^{-1} P^T \Sigma_{\bar{q}} + \Sigma_{\bar{q}} P (T^{-1})^T \left( \frac{\partial A_{zd}}{\partial p_j} \right)^T T^T P^T \right\} \right] + \\ & \text{trace} \left[ \bar{L}_i \left\{ P T \frac{\partial B_{zd}}{\partial p_j} B_{zd}^T + B_{zd} \left( \frac{\partial B_{zd}}{\partial p_j} \right)^T T^T P^T \right\} \right] \end{aligned} \quad (5.82)$$

The physical parameter sensitivities for a balanced reduced system can be computed provided we know the matrix derivatives of the original system. The results of the physical parameter sensitivity analysis for the MDOF case are normalized and shown in Figure 5.11.



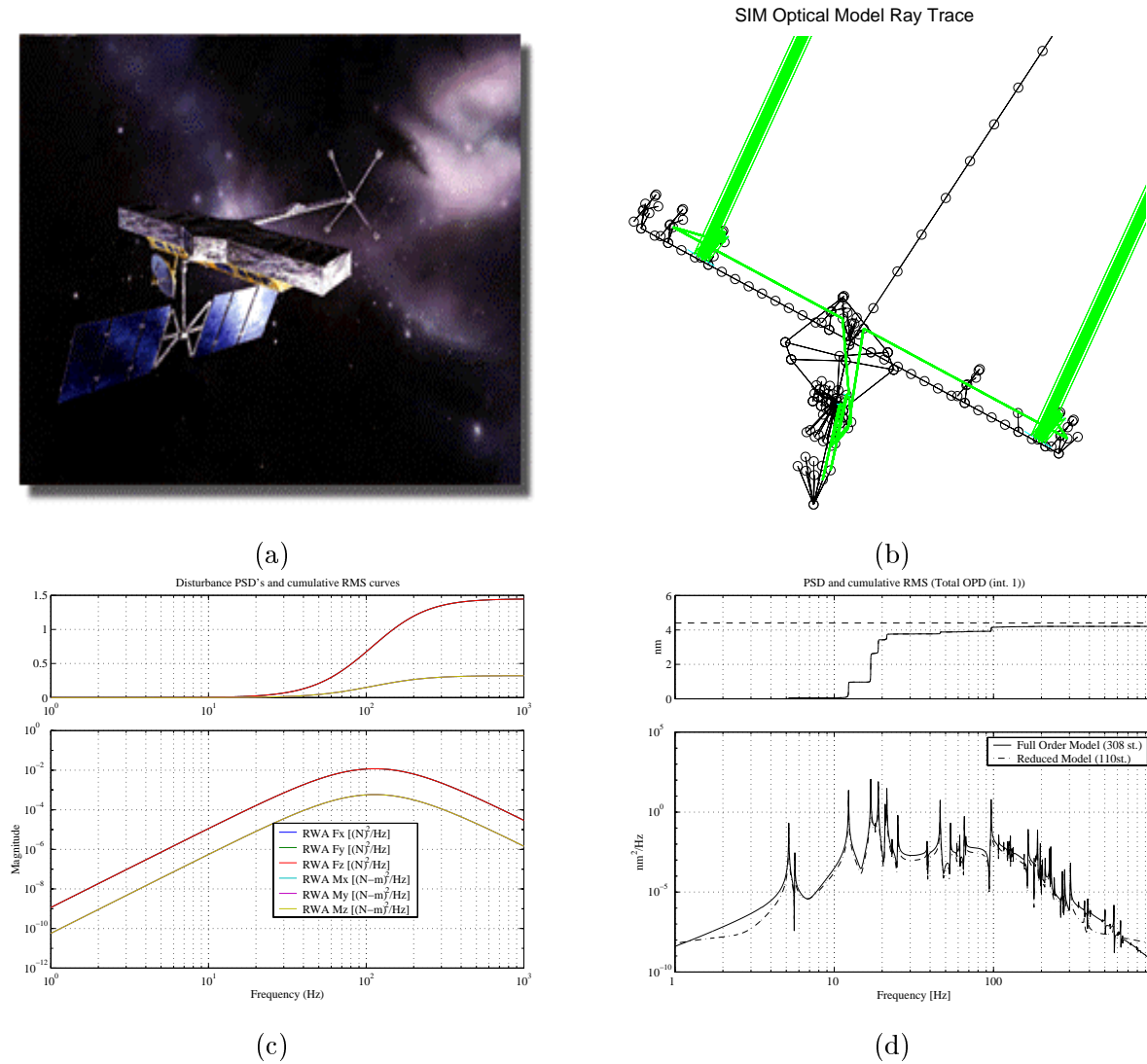
**Figure 5.11:** Normalized Sensitivities for physical parameters (MDOF problem), left shows performance 1 (hub displacement RMS), right shows performance 2 (OPD RMS) sensitivities.

The left subplot contains the result for performance  $z_1$ , whereas the right subplot shows the results for performance  $z_2$ . The results are verified with a 1 % finite difference approximation of the sensitivities. We see that the performances are most sensitive to the spring stiffness parameter  $k$ . Increasing the mass on the extremities, i.e. increasing the parameter  $m_{arm}$  will actually increase the resulting RMS. This is a counterintuitive result at first, since we generally expect that increasing *any* mass parameter will always lower the RMS of a displacement related performance metric. This kind of insight is valuable for system optimization, isoperformance and uncertainty analyses. We see that in order to increase the performance of the "interferometer" (i.e. lower the RMS values of  $z_1$  and  $z_2$ ) in the presence of the white-noise hub disturbance  $d$  we should increase the mass of the hub, decrease the mass in the arms and increase the stiffness of the arms.

The results (1) are for the original system (42 states), (2) for the reduced system (14 states) and (3) for a 1% perturbation size finite difference approximation. The physical parameter sensitivity analysis for the original system cost  $1.28 \cdot 10^7$  floating-point operations, where it cost only  $6.70 \cdot 10^6$  floating-point operations for the reduced system (factor 2 savings). We can also see that the answers for the three systems (1), (2) and (3) are very close. The error bound (5.72) predicts that the absolute value of the relative sensitivity error  $|\Delta(\partial\sigma_{z_i}/\partial p_j)/(\partial\bar{\sigma}_{z_i}/\partial p_j)|$  should be smaller than 0.0033. The largest relative sensitivity error was found to be equal to 0.0024 for the sensitivity  $\partial\sigma_{z_2}/\partial m_{arm}$ . Thus the sensitivity error bound (5.72) is shown to be valid for physical parameter sensitivities as well.

### 5.4.3 Application to SIM Classic (V1.0)

In order to demonstrate the usefulness for realistic models, a disturbance and modal sensitivity analysis was conducted for a closed-loop model of SIM. The model of SIM Classic and the disturbance analysis results for the full-order (308 states) and reduced model (110 states) are shown in Figure 5.12.



**Figure 5.12:** (a) SIM Classic; (b) FEM with ray trace for interferometer #1; (c) Disturbance PSD's for RWA disturbances; (d) Cumulative RMS plot (top) and PSD (bottom) for the total OPD of interferometer #1, where the dashed line is the reduced model (110 states) and the solid line is the full order model (308 states).

The disturbance analysis (PSD-method according to [45]) was performed using a 4-wheel

reaction wheel assembly (RWA) as disturbance input, see Figure 5.12(c). The power spectral density (PSD) of the performance (Total OPD interferometer #1) along with its cumulative RMS curve is shown in Figure 5.12(d). We see that a performance around 4.2 nm RMS is predicted, which meets the 4.4 nm requirement. The cumulative RMS plot shows that 6 “critical” modes are responsible for the RMS. Subsequently a Lyapunov analysis was conducted with the full order (308 states) and a balanced reduced model (110 states). The equations derived above were used to compute the RMS and modal parameter sensitivities after they were incorporated in the functions `dist_analysis.m` and `sens_analysis.m`.

The RMS results for the original and the reduced system match closely, when using the PSD method, where the precision is mainly a function of the resolution of the frequency vector. The RMS predicted by the Lyapunov method differs by 5.2% between the original (308 states) and the reduced model (110 states), which is in excess of the 0.2% allowed by the RMS error bound. This is likely due to numerical ill-conditioning as discussed below. The results are compared in Table 5.5. The reduced system solves at 4% the CPU time of the

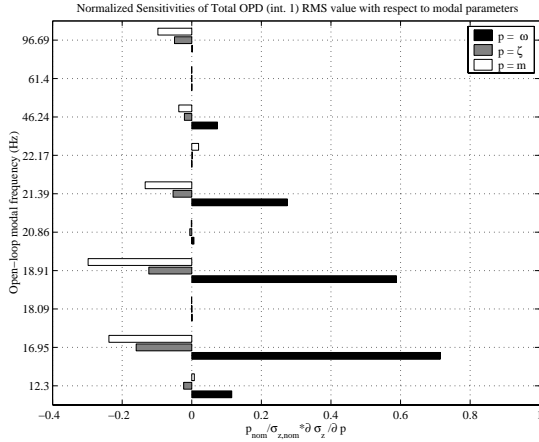
**Table 5.5:** RMS Results comparison for SIM Classic

Results	Full Model	Red Model
# of states	308	110
RMS (PSD)	4.21 nm	4.21 nm
RMS (Lyap)	4.3321 nm	4.1077 nm
CPU (Lyap)	39.567 sec	1.552 sec

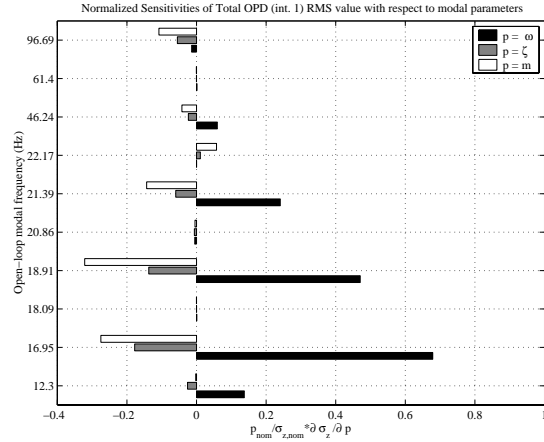
original system for a 65 % reduction in the number of states. (This doesn’t include the CPU time for the balancing and reduction process itself).

The normalized modal parameter sensitivity results for 10 modes of the system are compared for the original system (308 states) and the reduced system (110 states) of SIM Classic on the left and right of Figure 5.13(a) and (b), respectively. The results are in good agreement with the exception of some of the less dominant modes. It is noticeable that the agreement is better for “critical” modes of the system compared to “non-critical” modes (i.e. modes that do not significantly contribute to the performance RMS). It took 78.2 seconds to solve for the original system sensitivities, but only 3.3 seconds to solve for the reduced system.

Significant and persistent numerical problems were initially encountered, when trying to



(a) Full Model



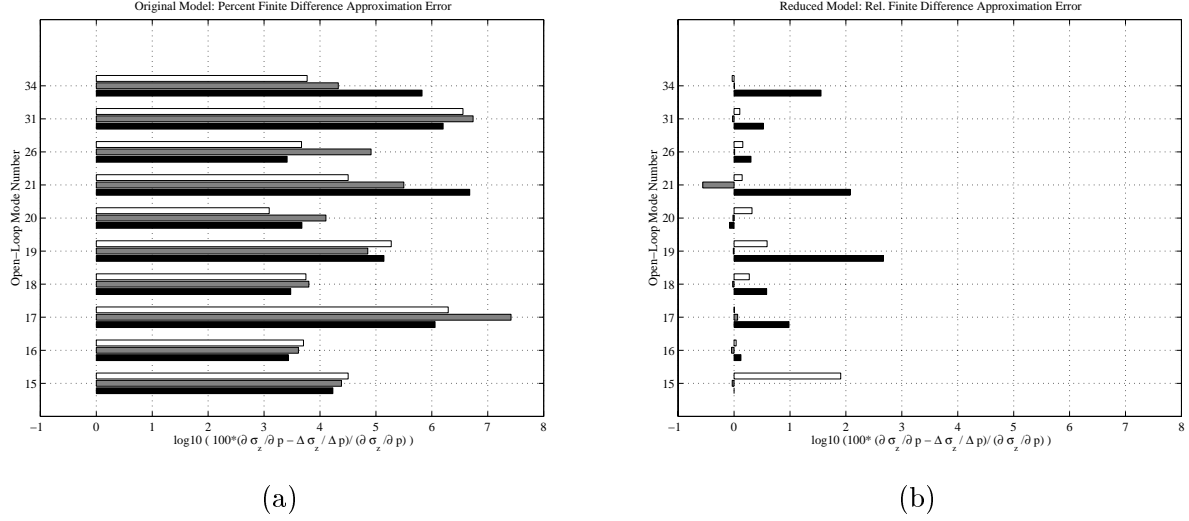
(b) Balanced Reduced Model

**Figure 5.13:** (a) SIM Classic modal parameter sensitivities for 10 modes and full-order system (308 states, 78.2 seconds CPU time), (b) Modal sensitivity results for balanced and reduced model (110 states, 3.3 seconds CPU time).

produce these results. The reason, discussed below, has to do with observability/controllability of the system. The results in Figure 5.13 were obtained by successfully applying Mallory’s pre-balancing method for model reduction [85]. The key to the pre-balancing method is that the system is brought into 2x2 block-diagonal form and each of these 2x2 blocks is balanced individually. Then nearly uncontrollable states are removed based on a user-defined tolerance before the overall  $T$ -matrix is computed.

Another noteworthy result is that *the finite difference validation of the modal sensitivities became possible with the balanced reduced system*. Gutierrez has previously reported the inability of the finite difference approximation to validate the analytical sensitivities for large order ill-conditioned systems [45, p.124]. This is due to the fact that the state covariance matrix solution does not vary smoothly with small changes in a system parameter due to ill-conditioning. This ill-conditioning of the system is removed with the method presented in this thesis. The resulting finite differences for the balanced reduced system closely match the analytical answers as shown by the bar chart in Figure 5.14(b). The figure shows the logarithm of the relative percent difference between the analytical sensitivities and the finite difference approximations for the original system (left) and the balanced, reduced system (right). This quantity was computed as  $\log \{100 \cdot (\partial\sigma_z/\partial p - \Delta\sigma_z/\Delta p) / (\partial\sigma_z/\partial p)\}$ , where the term with the partials comes from the analytical sensitivity calculations and the  $\Delta$ -term comes from

the finite difference approximation. Thus, a value of one for a horizontal bar indicates a 10% difference between the analytical sensitivity and the finite difference approximation. A value of two indicates a 100% difference.



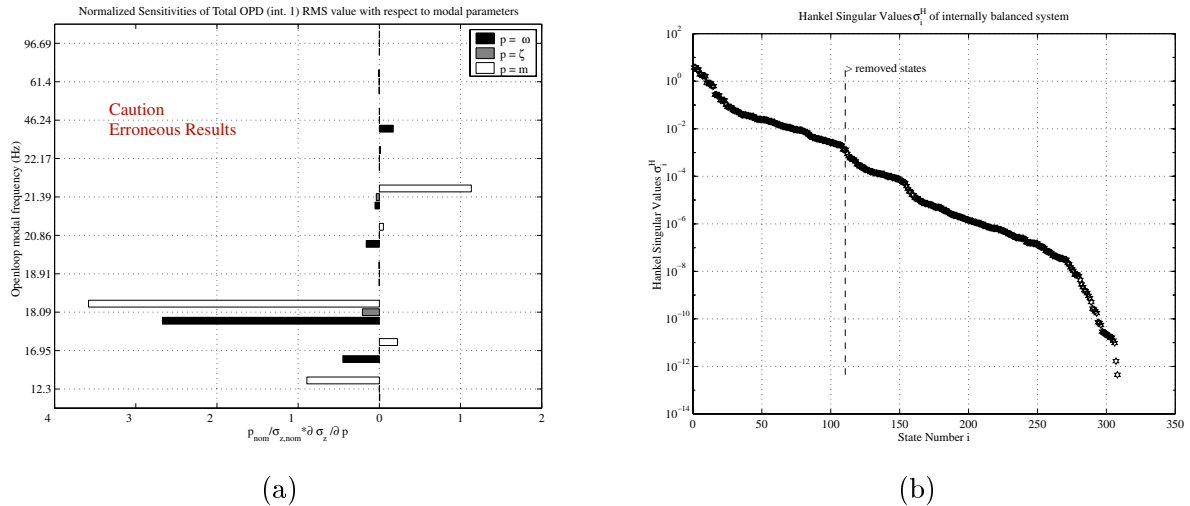
**Figure 5.14:** (a) Very large errors between finite difference approximation (1%) and the analytical modal parameter sensitivities for the full order model; (b) Good approximation with finite differences (1%) in the reduced model case. Mode numbers correspond to modes in Figure 5.13

Some numerical conditions must be met in order to successfully apply the balanced and reduced LTI system extensions to the DOCS framework that are presented in this chapter. First the system must be asymptotically stable. This means that no poles (of the appended, closed-loop system) are allowed on the  $j\omega$ -axis or in the right half-plane (RHP). Physically this would mean that the system response diverges for a finite input disturbance and that the performance RMS is not bounded. Secondly the system must be in a minimal realization. This requirement is harder to meet since larger order models (SIM, NGST, TPF and Nexus) invariably all feature pole-zero cancellations after initial model assembly, which are indicative of uncontrollable or unobservable states. Even though these states do not prevent a solution for  $\Sigma_q$  to be found (by definition they don't contribute to the RMS), they cause the system to be ill-conditioned. This causes problems when we try to compute the balancing matrix  $T$ .

Finally we want the system (and the  $A_{zd}$ - matrix in particular) to be numerically well conditioned from the start. Certain canonical realizations of the disturbances or the compensator and the traditional 2nd-order modal form of the structural dynamics are counterproductive in this sense. Alternate realizations such as the real modal form, see Equation (5.16), are

recommended. All these conditions point to the need for a reliable, flexible and robust model reduction and conditioning module in DOCS. This need has been addressed by Uebelhart [136].

An interesting observation was made when the RMS for OPD #1 for SIM Classic (308 states) was initially computed for the original and the balanced (unreduced) system and the answers were compared. Even though the RMS resulting from the Lyapunov approach should be identical for both cases (similarity transformation), the answers differed by roughly 10% (4.33 nm versus 3.85 nm) even though the system was merely balanced and no states were removed. The cause of this disturbing result was linked to the presence of unobservable/uncontrollable states in the original system. The erroneous results for the modal parameter sensitivities in this case are shown in Figure 5.15(a). The correct answers are shown in Figure 5.13. The erroneous answers in Figure 5.15(a) are produced without any warning, which suggests that numerical conditioning checks and safeguards as well as engineering judgement are indispensable.



**Figure 5.15:** (a) Erroneous modal parameter sensitivity results for SIM Classic obtained when balancing matrix  $T$  is computed without first removing unobservable and uncontrollable states, e.g. via pre-balancing. (b) Hankel singular values for original closed-loop SIM Classic model.

The numerical reason for the erroneous result is that very small (very small means on the order of  $\epsilon_{ps} = 2.2204 \cdot 10^{-16}$ , the numerical floating-point accuracy of the computer) Hankel singular values are associated with the (nearly) unobservable/uncontrollable states. These very small numbers are inverted in the computation of  $T$  or  $T^{-1}$ , where they become very big



numbers, even though they correspond to insignificant states. Consider the last term in the following equation, which inverts the Hankel singular values in Moore’s method [102]:

$$T^{-1} = \Sigma_c^{-\frac{1}{2}} U_c^T U_b \Sigma_H^{-\frac{1}{2}} \quad (5.83)$$

This leads to ill-conditioning of the  $T$ -matrix itself and to numerical errors in the TGSE. Mallory’s pre-balancing method (`gmxoba1.m`) [85] was successfully applied to solve this problem, since the pre-balancing algorithm removes the unobservable/uncontrollable states before the inversion. This leads to a balanced, reduced, minimal and well-conditioned system and accurate answers for the RMS and sensitivities. This is a key element in the accurate prediction of isoperformance contours.

## 5.5 Summary

The cost of solving a Lyapunov Equation (5.1) is shown to be roughly  $50 \cdot n_s^3$ . Since hundreds or even thousands of Lyapunov equations have to be solved in a multivariable isoperformance problem, efforts were undertaken to find a solution more efficiently. This can be achieved in two different ways. The first way is to diagonalize the integrated state space system, (1.1), and to apply the new, fast Lyapunov solver presented in Section 5.1. This drops the exponent of  $n_s$  in the Lyapunov solution cost from a value of three to two. The second approach is to reduce the number of states,  $n_s$ , while retaining the important information in the model. Section 5.2 discussed improvements in performance prediction and sensitivity analysis for similarity transformed systems, including derivation of the transformed governing sensitivity equation (TGSE). Section 5.3 derived error bounds for performance and sensitivity analysis for reduced systems. Using these methods analytical sensitivities can be accurately computed, even when matrix derivatives, such as  $\partial A_{zd}/\partial p_j$ , are only known with respect to the original matrices in the “assembly” realization. The equations are verified with a single degree-of-freedom example, a MDOF example and an analysis of the SIM Classic model. The sum of these contributions enables a meaningful isoperformance analysis for realistic, large order systems.



# Chapter 6

## Experimental Validation

The goal of the experimental validation is to demonstrate the ability of the isoperformance methodology to accurately predict performance contours for a physical laboratory testbed in a 1-g environment. This chapter provides a testbed description, lays out the experimental approach, presents results from the testbed characterization and discusses a comparison of theoretical predictions with experimentally measured isoperformance contours.

### 6.1 Testbed Description

The DOLCE<sup>1</sup> testbed shown in Figure 6.1 was explicitly designed for this purpose. The main feature of DOLCE is that system parameters can be varied over a large range. This is different from the cantilever truss employed by Gutierrez [45], which was used for physical parameter sensitivity validation via small perturbations of masses and stiffnesses. The four variable parameters on DOLCE are shown in Table 6.1.

**Table 6.1:** Variable parameters of DOLCE testbed.

Symbol	Parameter	Range	Units
$V_s$	excitation RMS voltage	0.1-1.0	[Vrms]
$m_p$	payload mass	0-200	[lbs]
$m_s$	seismic mass	30-850	[grams]
$k_s$	suspension spring stiffness	80-216	[lbs/in]

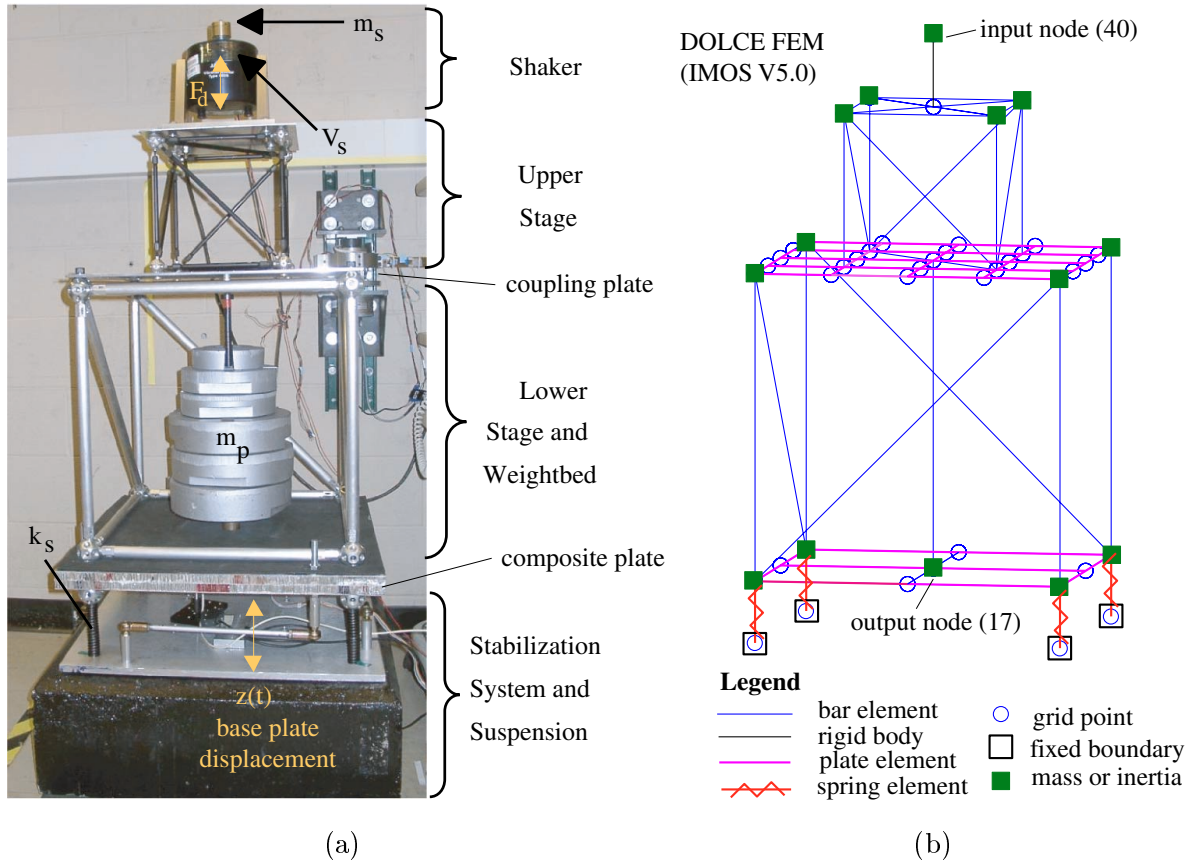
---

<sup>1</sup>The testbed name contains the initials of all students who worked on its design and development: **D**usty DeQuine, **O**livier de Weck, **L**aila Elias and **C**emocan Yesil.

### 6.1.1 Physical Test Setup

Figure 6.1(a) shows the testbed which, starting from the top, is comprised of an uniaxial vibration exciter (shaker), with a seismic mass,  $m_s$ , driven by a band-pass filtered (0-100 Hz), random excitation voltage,  $V_s$ . Next the upper stage contains a single small bay of a square truss and a coupling plate. The lower stage consists of a large square truss, a weight bed holding a payload mass,  $m_p$ , and an aluminum sandwich base plate. Finally an axial stabilization system and four (4) suspension springs of stiffness  $k_s$  complete the arrangement.

The shaker disturbance force,  $F_d$ , excites the testbed mainly in the vertical (axial) direction and simulates an on-board mechanical disturbance source. The induced axial motion is sensed and recorded as described below. This can be thought of as representing jitter of a spacecraft, with the soft suspension springs providing a functions similar to the attitude control system (ACS). The lower stage is then analogous to a spacecraft bus, while the upper stage represents the payload. The testbed is open-loop, i.e. no controller was implemented.



**Figure 6.1:** (a) DOLCE testbed setup in laboratory. (b) Finite element model (FEM) of DOLCE using IMOS version 5.0.

### 6.1.2 Finite-Element Model

A finite element model (FEM) of the testbed was constructed using the MATLAB toolbox IMOS (version 5.0), see Reference [62]. The model consists of 44 nodes with 188 independent degrees-of-freedom (after Guyan reduction). A graphical rendering of the FEM is shown in Figure 6.1(b).

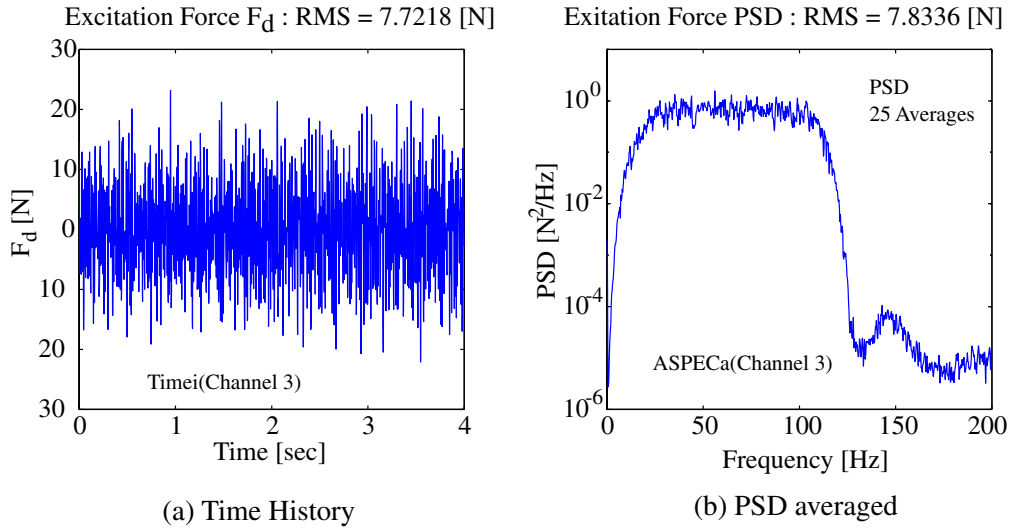
The truss members are represented by beam elements, while quadrilateral plate elements are used for the coupling plate (`uplate`) and the composite plate (`splate`). This latter plate is actually an aluminum sandwich plate with two 1.9 [mm] thick face sheets and a 3.15 [mm] thick core sheet. The core consists of two half inch thick hexagonal honeycomb cores. This sturdy construction was necessary in order to accommodate a payload mass up to 200 [lbs] without significant elastic deformation. The four suspension springs are modeled with uniaxial spring elements (`celas`), while the shaker and its attachment is modeled as a rigid body (`rbe2`). Concentrated masses (`conm`) are placed at the nodes of the truss elements, accounting for the ball masses, as well as at the locations of the seismic mass,  $m_s$ , and the payload mass,  $m_p$ , respectively. The lower grid points of the suspension springs are assigned a fixed boundary condition, which is representative of the attachment of the testbed to the concrete slab in the laboratory.

The input node for the disturbance force,  $F_d$ , is #40, while the z-displacement of node #17 is used to predict the axial motion of the testbed. The assembly of the FEM, including solution of the generalized eigenvalue problem takes 0.82 [sec] (Pentium III, 650 MHz) and requires 13.1 million floating point operations. The first flexible mode is a rocking mode and occurs at a frequency of 6.28 [Hz] with  $m_p = 100$  [lbs] and  $k_s = 168$  [lbs/in] suspension springs. A listing of the FEM definition file is contained in Appendix D.

### 6.1.3 Disturbance Model

The shaker (Model 4809 Brüel & Kjaer) generates a random axial disturbance force,  $F_d$ , whose magnitude and frequency content depend on the excitation voltage,  $V_s$ , and the seismic mass,  $m_s$ . The seismic mass,  $m_s$ , was determined to be 339.75 [grams]; it is held constant in this experiment. This device is meant to simulate the disturbances generated by vibrating on-board machinery on a spacecraft (e.g. reaction wheel, cryocooler), albeit at a significantly higher force level.

A sample time realization of the measured disturbance force is shown in Figure 6.2(a). The sensor consists of a uniaxial load cell (see Table 6.2 for specification), which is attached to the seismic mass and measures the disturbance force,  $F_d$ . The power spectral density (PSD) of the disturbance force,  $S_{dd}$ , can be seen in Figure 6.2(b). The bandpass nature of the disturbance is evident, where the low frequency rolloff below 20 [Hz] is due to the combined amplifier, shaker and load cell (sensor) dynamics. The high frequency rolloff was chosen at 100 [Hz]. This bandpass nature is typical from many disturbance sources, such as reaction wheels [89].



**Figure 6.2:** (a) Shaker disturbance force,  $F_d(t)$ , sample time history for shaker voltage,  $V_s = 0.5$  [Vrms]. (b) Averaged disturbance PSD  $S_{dd}$  - 25 averages.

#### 6.1.4 Performance Model

The performance is the root-mean-square (RMS) of the composite (base) plate displacement,  $z(t)$ :

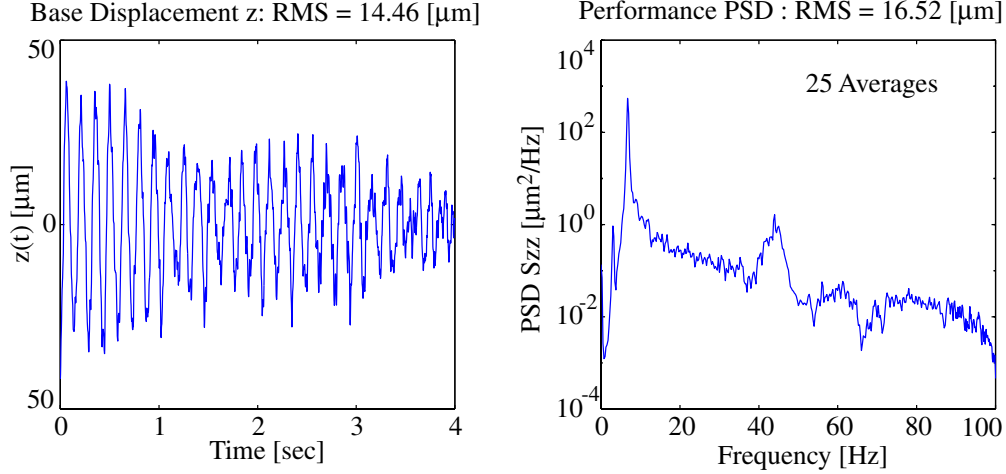
$$J_z = E [z^T z]^{1/2} \quad (6.1)$$

This quantity can be computed from the time history of  $z(t)$  directly or alternatively from the PSD,  $S_{zz}$ . The performance  $J_z$  is computed by integrating under  $S_{zz}$  and taking the square root.

$$J_z = \left[ 2 \int_{f_{\min}}^{f_{\max}} S_{zz}(f) df \right]^{1/2} \quad (6.2)$$

The second approach was chosen here since it allows averaging of the PSD's and a more accurate result. Figure 6.3(a) shows a sample time history of the base plate displacement,

$z(t)$ , as measured with the proximator (gap sensor). The corresponding (averaged) power spectral density function,  $S_{zz}$ , in units of  $[\mu\text{m}^2/\text{Hz}]$  is shown in Figure 6.3(b). The features of this function are discussed below.



**Figure 6.3:** (a) Sample time history of displacement,  $z(t)$ , for  $V_s = 0.5$  [Vrms] and  $m_p = 160$  [lbs]. (b) Averaged performance PSD  $S_{zz}$  - 25 averages.

This performance metric would correspond to jitter of the spacecraft bus in a real space system. The performance signal,  $z(t)$ , is measured via an inductive proximator, which acts as a gap sensor (eddy current gap sensor by Bentley XL 5mm). The gap sensor is very sensitive and was calibrated to 0.425 V/mil of displacement with a LB-11/70 Laser Displacement Sensor. Also a Sunstrand DC accelerometer was installed in order to corroborate the gap sensor results. The sensor specifications are summarized in Table 6.2.

**Table 6.2:** DOLCE Testbed sensor specifications.

Sensor	Model	Range	Calibration
Force Sensor (LC)	208A02	0-889.6 [N]	$94.87 \pm 6.67$ [N/V]
DC Accelerometer	QA-1400	$\pm 15$ [g]	$1.0 \pm 0.01$ [g/V]
Laser Displacement Sensor	LB-11/70	60-140 [mm]	$10.28 \pm 0.3$ [mm/V]
Proximator (Gap Sensor)	3300-XL 5mm	10-100 [mil]	2.35 [mil/V]

The sensor suite below the sandwich plate is shown in Figure 6.4. The load cell is attached below the seismic mass (top of testbed) and cannot be seen on this picture.

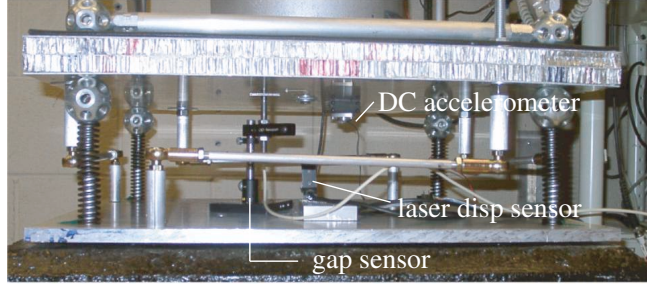


Figure 6.4: DOLCE Testbed Sensors

## 6.2 Experimental Approach

The experimental approach is presented in Figure 6.5(a). First the testbed was assembled, instrumented and calibrated. It was decided to conduct a bivariate isoperformance test, with the performance given by Equation 6.1. The variable parameters were the excitation voltage,  $V_s$ , ranging from 0.1-1.0 [Vrms] as well as the payload mass,  $m_p$ , ranging from 0-200 [lbs] as shown in Figure 6.5(b). A test matrix was run on the testbed and recorded with parameter increments  $\Delta V_s = 0.1$  [Vrms] and  $\Delta m_p = 10$  [lbs], respectively. From this gridded data isoperformance contours were extracted via linear interpolation, see Subsection 3.2.1.

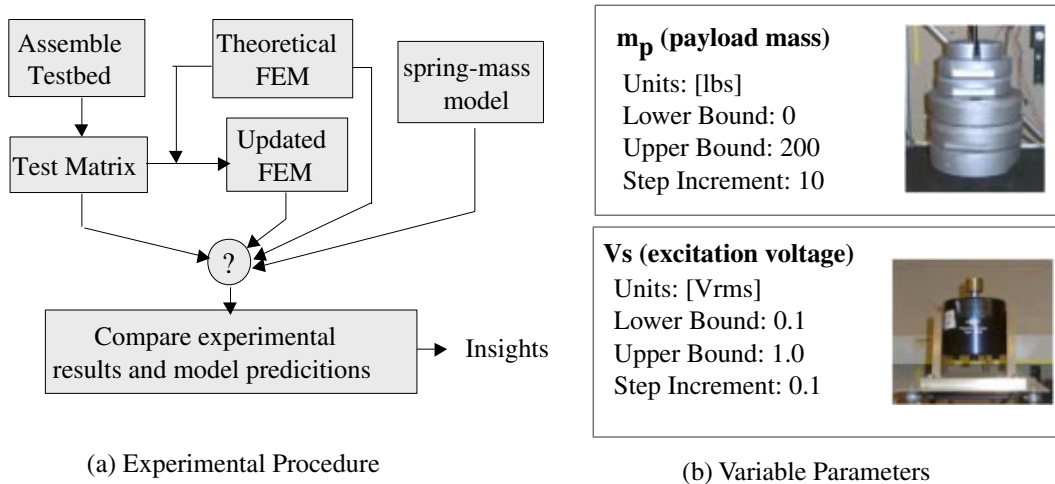


Figure 6.5: (a) DOLCE Experimental Approach. (b) Variable system parameters  $m_p$  and  $V_s$ .

Independently and without knowledge of the experimental results an apriori finite element model (FEM) was constructed (“original FEM”). This FEM was described in the previous subsection, see Figure 6.1(b), and a complete listing of the IMOS model input deck is contained



in Appendix D. Only assembly drawings, masses from scale measurements and catalogue values for material properties and spring stiffnesses were used for this “original” (unupdated) FEM. The predictions from this model would be equivalent to what could be expected from isoperformance analyses for spacecraft in the conceptual and preliminary design phases, such as NEXUS. For these types of early models system-level test data is usually unavailable.

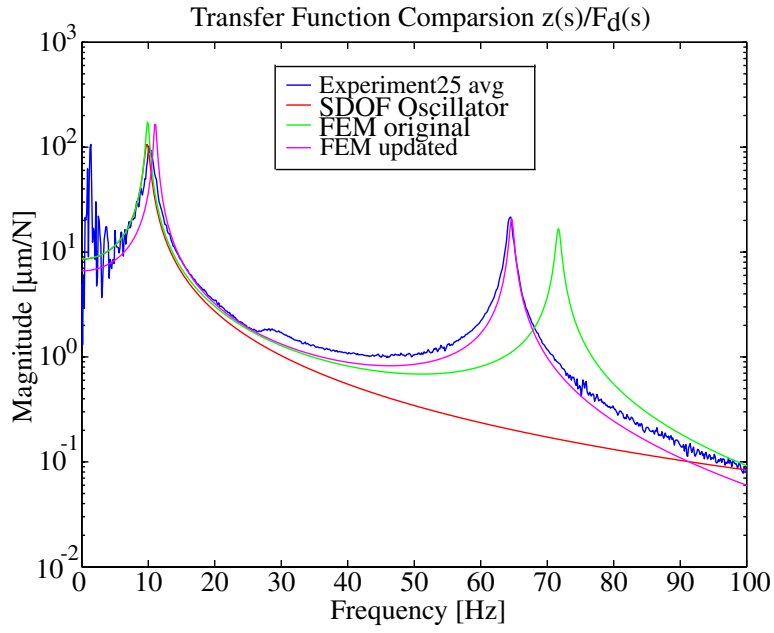
A more accurate prediction is expected from an updated FEM, which has its physical parameters tuned such that the FEM and experimental transfer function (measurement model) from  $F_d$  to  $z$  coincide well. A tuning procedure similar to the one used by Gutierrez [45] was used here. Finally the isoperformance contours for DOLCE are predicted with a single degree-of-freedom (SDOF) model. This model lumps the entire testbed mass (30.585 [kg]) together with the payload mass  $m_p$  over the four suspension springs (in parallel) represented as a single compliance. The damping ratio is assumed to be  $\zeta = 0.02$ . The model of the SDOF oscillator used here was previously introduced in Section 2.1.

The goal of the experiment is to gain insights by comparing different performance contours for the experiment with the ones predicted for the three models. In this way limitations of the isoperformance methodology might appear, which would not be revealed by solving additional theoretical problems.

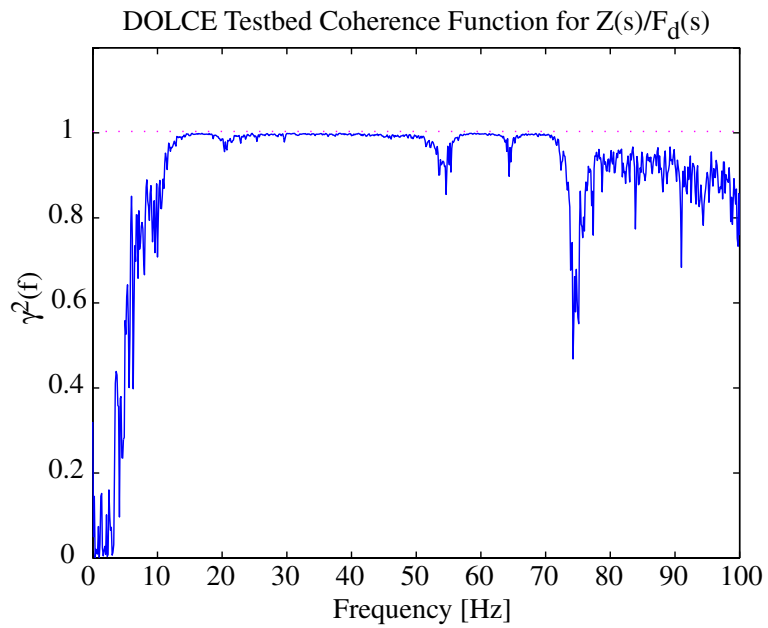
## 6.3 Testbed Characterization

### 6.3.1 System Identification

The transfer function (FRF) from disturbance (shaker) force to base plate displacement,  $G_{zd}(s) = Z(s)/F_d(s)$ , where  $s = j\omega$ , was obtained experimentally and by model prediction, see Figure 6.6. For this system identification the load cell ( $F_d$ ) and the gap sensor ( $z$ ) specified in Table 6.2 were used. The coherence function  $\gamma^2(f)$  corresponding to the experimental transfer function  $Z(s)/F_d(s)$  is shown in Figure 6.7. It can be seen that the coherence is close to one from roughly 15 to 75 [Hz], thus providing a good estimate of the disturbance to performance transfer function in that region. Below 15 [Hz] the coherence drops because the shaker dynamics prevent a lot of low frequency disturbance energy to be generated. Above 75 [Hz] the attenuation of the bandpass filter rolloff starts to take effect.

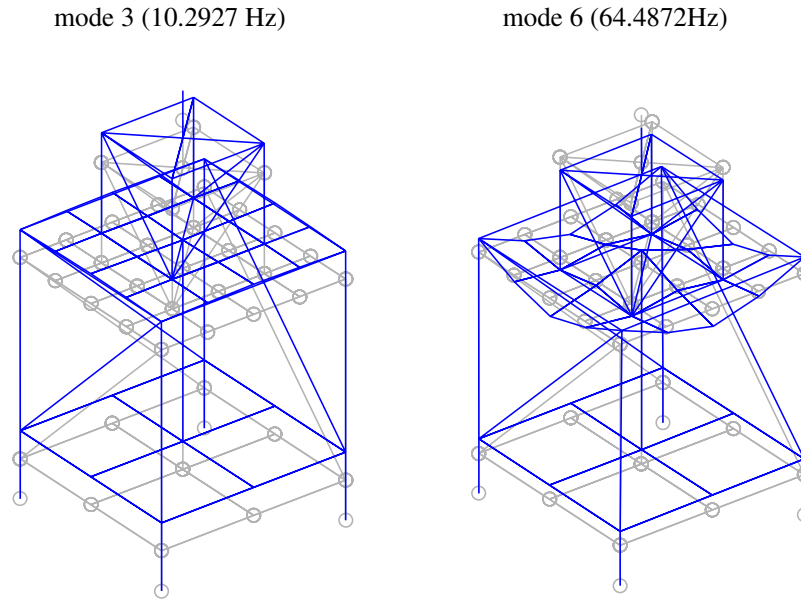


**Figure 6.6:** DOLCE transfer function  $G_{zd} = Z(s)/F_d(s)$  for  $m_p = 0$ ,  $V_s = 1.0$



**Figure 6.7:** DOLCE coherence function  $\gamma^2(f)$  corresponding to the transfer function  $G_{zd} = Z(s)/F_d(s)$ .

As can be seen there are two observable modes in the bandwidth up to 100 Hz. The first mode at 10 Hz is the axial base suspension mode, where the testbed translates vertically up and down on the 4 compression springs. The second mode at 65 Hz is the coupling plate bending mode, which causes a vertical displacement via the center rod. Mode shapes for these two modes are contained in Figure 6.8.



**Figure 6.8:** DOLCE Testbed Observable Modes

As expected the SDOF model can only predict the first resonance. The original FEM overpredicts the upper plate mode by roughly 10 Hz. The agreement between the updated FEM and the experimental transfer function is very good.

### 6.3.2 Performance Testing

Next the testbed response was investigated as a function of the single parameter,  $m_p$ . A waterfall plot showing the power spectral density (PSD) of  $z$  as a function of  $m_p$  is depicted in Figure 6.9.

It can be seen that the axial suspension mode is dominant for all payload masses. As expected the mode softens with increasing mass from about 10 Hz at  $m_p = 0$  [lbs] to 6 Hz at  $m_p = 200$  [lbs]. The resonant plate mode at 65 Hz can also be seen, but it is much less clear for larger  $m_p$ . A higher frequency mode around 40Hz appears  $m_p$ -invariant and we suspect some structural non-linearity.

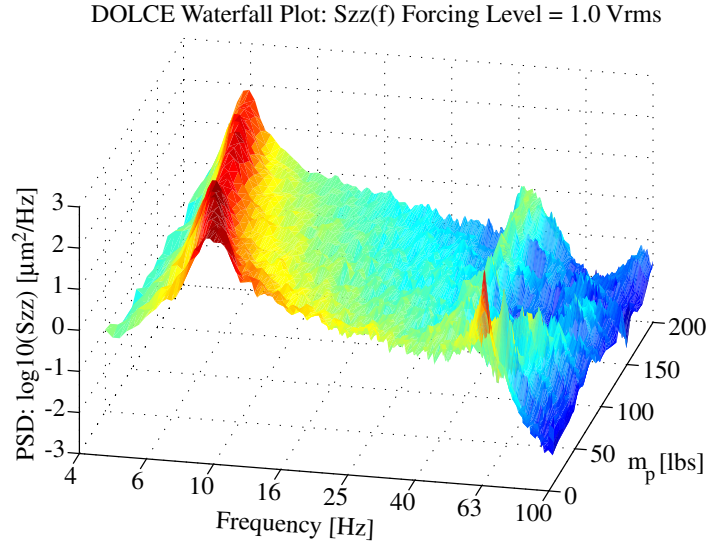


Figure 6.9: Waterfall Plot for 1.0 [Vrms] forcing level

## 6.4 Isoperformance Results and Interpretation

At each parameter combination the time histories of  $F_d(t)$  and  $z(t)$ , were recorded and the performance  $J_z = J_z(V_s, m_p)$  was computed with 25 averages according to Equation (6.2). The results from the test matrix are shown in Figure 6.10.

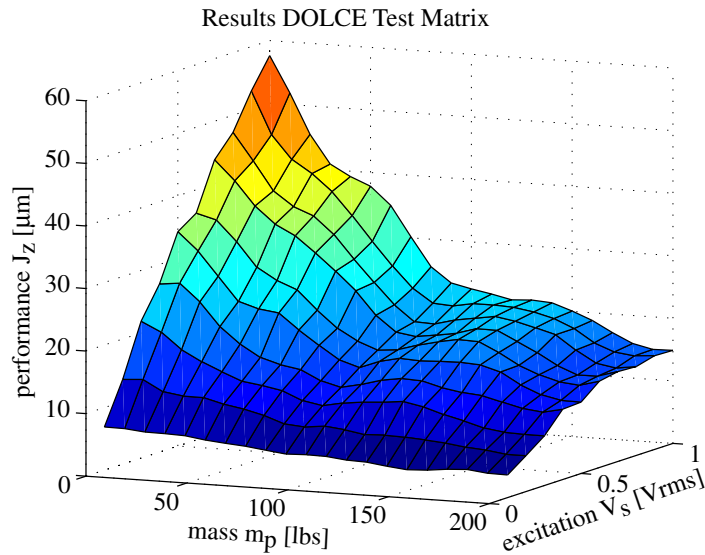
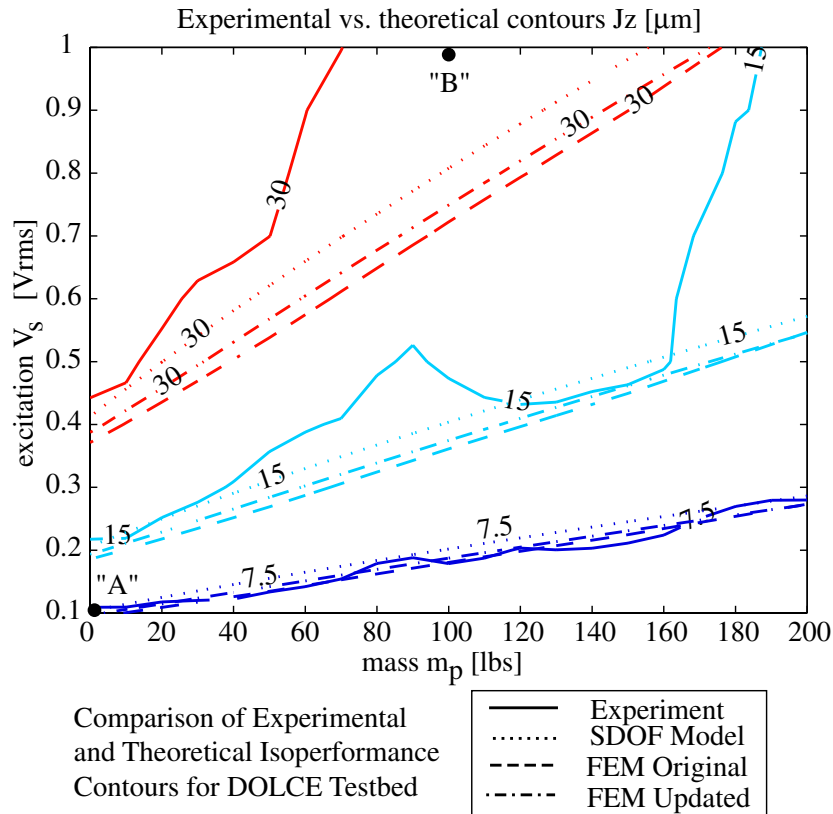


Figure 6.10: DOLCE Experimental Test Matrix ( $V_s, m_p$ ).

The peak displacement RMS value of 57.6 [ $\mu\text{m}$ ] is obtained for the maximum excitation level ( $V_s = 1.0$  [Vrms]) with an empty weight bed ( $m_p = 0$  [lbs]). This is intuitively satisfac-

tory, since at this point the maximum disturbance energy enters the system (about 14.86 [N] of force RMS), while the disturbability of the system is at a maximum. Recall that the plant transfer function for such a system has a  $1/m$  term in the numerator. Conversely the lowest response (“best performance”) is found for  $V_s = 0.1$  and  $m_p = 200$ .

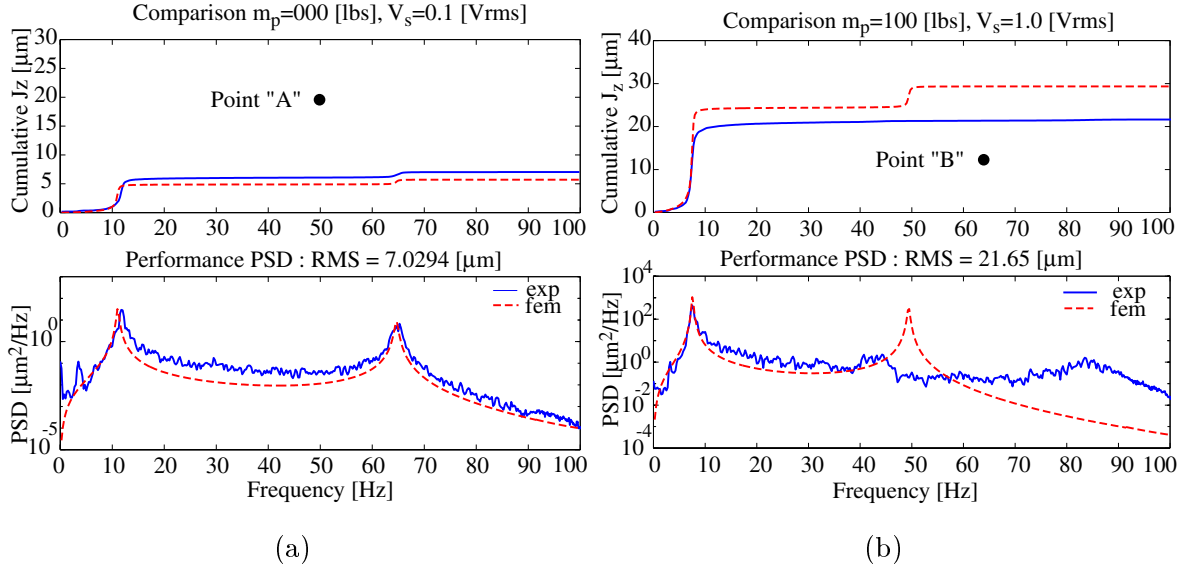
This information is used to obtain isoperformance contours at the 7.5, 15 and 30  $[\mu\text{m}]$  levels (Figure 6.11). The basis for obtaining the experimental isoperformance contours is the test matrix with  $V_s$  and  $m_p$  shown in Figure 6.10 and the linear interpolation scheme from Subsection 3.2.1.



**Figure 6.11:** DOLCE Testbed: Comparison of experimental versus theoretical isoperformance contours at the levels:  $J_z = 7.5, 15$  and  $30$   $[\mu\text{m}]$  with  $p_1 = m_p$  and  $p_2 = V_s$ .

Similar contours are predicted for the SDOF model and the FEM’s (original and updated). This suggests that the axial suspension mode is dominant in most of the trade space. Excellent correlation between experiment and theory is found at low forcing levels, see the 7.5  $\mu\text{m}$  contour. Deviations are found for larger forcing levels (15 and 30  $\mu\text{m}$  contours), even though the general trends are still predicted correctly by the isoperformance models.

The cause for this deviation is difficult to determine, but is likely due to non-linear effects in the structural plant as the shaker amplitude increases. To illustrate this statement the performance PSD's,  $S_{zz}$ , have been plotted for the experimental data (blue - solid line) and the FEM prediction (red - dashed line) at two different points in the trade space, see Figure 6.12. Subplot (a) shows the PSD's for point "A" in Figure 6.11 with  $m_p = 0$  [lbs] and  $V_s = 0.1$  [Vrms]. Here good agreement between theory and experiment is found. Subplot (b) on the other hand represents Point "B" in Figure 6.11 which experiences the maximum disturbance level. Figure 6.12(b) shows that the discrepancy in performance prediction is mainly due to the second mode (coupling plate bending). This mode is not visible in the test data and does not contribute to the experimental cumulative RMS. This is due to a non-linear effect, which has not been investigated in greater detail.



**Figure 6.12:** (a) Comparison of PSD  $S_{zz}$  (bottom) and cumulative RMS plot (top) between experiment and FEM prediction for configuration:  $m_p = 0$  [lbs],  $V_s = 0.1$  [lbs]. Good Agreement. (b) Comparison of PSD  $S_{zz}$  (bottom) and cumulative RMS plot (top) between experiment and FEM prediction for configuration:  $m_p = 100$  [lbs],  $V_s = 1.0$  [lbs]. Poor Agreement.

In conclusion it is found that the isoperformance prediction capability is good at low disturbance levels which are representative of the vibration environment on space based optomechanical systems. Caution must be exercised if non-linearities are suspected in any part of the system and particularly if performance predictions are to be made at high excitation levels. Additional parameters such a  $k_s$  and  $m_s$  could be included in future tests.

## 6.5 Summary

The goal of the experimental validation was to demonstrate the ability of the isoperformance methodology and toolbox to predict isoperformance contours on a real physical system. The 1-g DOLCE testbed was designed to have multiple disturbance and structural parameters, which are variable over a large range. The testbed characterization of the transfer function from disturbance force,  $F_d(t)$  to base displacement,  $z(t)$ , shows that two modes are dominant in the bandwidth of interest up to 100 Hz. These are the fundamental, axial suspension mode and the first coupling plate bending mode. The fundamental mode softens from 10 to 6 [Hz] as more payload mass is added.

A test matrix was experimentally determined with the RMS excitation voltage of the shaker,  $V_s$ , and the payload mass,  $m_p$ , as independent parameters. Theoretical isoperformance contours were predicted with a SDOF spring-mass model, an un-updated FEM and an updated (tuned) FEM. These contours were compared with experimental contours obtained from the test matrix via linear interpolation. Good agreement was shown for the 7.5  $\mu\text{m}$  RMS displacement level, where the excitation amplitude is small. This corresponds to the low vibration levels expected on precision opto-mechanical spacecraft. At higher disturbance levels (15 and 30  $\mu\text{m}$  contours) the agreement is not as good and deviations between the experimental and theoretical performance are likely due to structural non-linearities.





## Chapter 7

# NEXUS Spacecraft Study

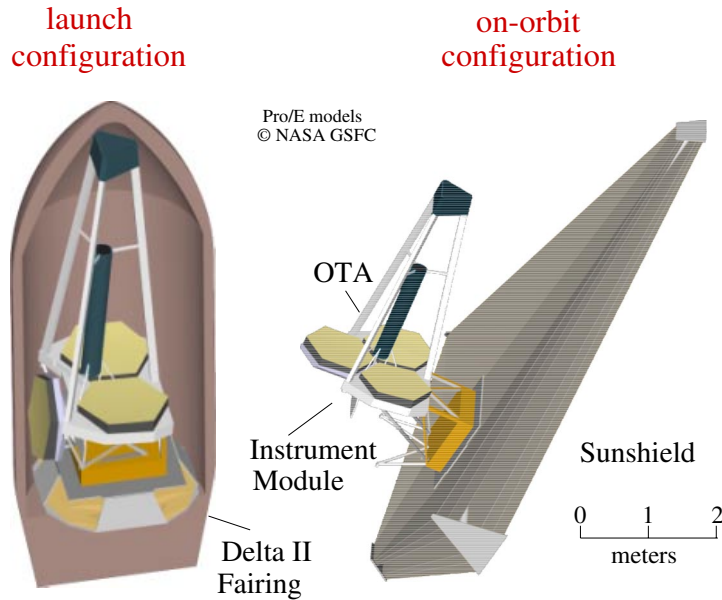
The previous two chapters provide confidence that the isoperformance methodology is applicable to large order, multivariable systems and that isoperformance predictions for real physical systems are possible. The purpose of the NEXUS spacecraft study which is presented in this chapter, is to demonstrate the usefulness of the isoperformance methodology on a realistic conceptual design model of a high-performance spacecraft.

The chapter first provides a description of the NEXUS spacecraft as well as the underlying integrated model. A disturbance analysis (= performance assessment) with time simulation and Lyapunov analysis is carried out for an initial vector,  $p_o$ , of 25 variable system parameters. These variables represent selected disturbance, plant, optics and controls parameters of the system. After establishing that the initial design does not meet the performance requirements for wavefront error (RMMS WFE) and line-of-sight jitter (RSS LOS), a sensitivity analysis is conducted in order to obtain the Jacobian,  $\nabla J_z$ . A bivariate and multivariable isoperformance analysis is presented for NEXUS. The application of isoperformance to multiobjective optimization and error budgeting completes the chapter.

### 7.1 NEXUS Description

NEXUS was planned as a technology risk-reduction experiment in space and as a precursor to NGST with an anticipated launch date of 2004. The innovative technology areas for this project are lightweight optical telescope assembly (OTA) fabrication and verification, cryogenic instrument and actuator development, deployable sunshield technology and image-based wavefront sensing and control. A graphical representation of the launch and on-orbit

configurations of NEXUS is shown in Figure 7.1. The NEXUS project was officially canceled in December 2000 as a part of the NGST rescoping exercise. It was nevertheless decided to use NEXUS for this case study, since the conceptual model is well developed and many lessons learned from previous NGST Yardstick models [24, 103, 23] were incorporated.



**Figure 7.1:** NEXUS Spacecraft Concept

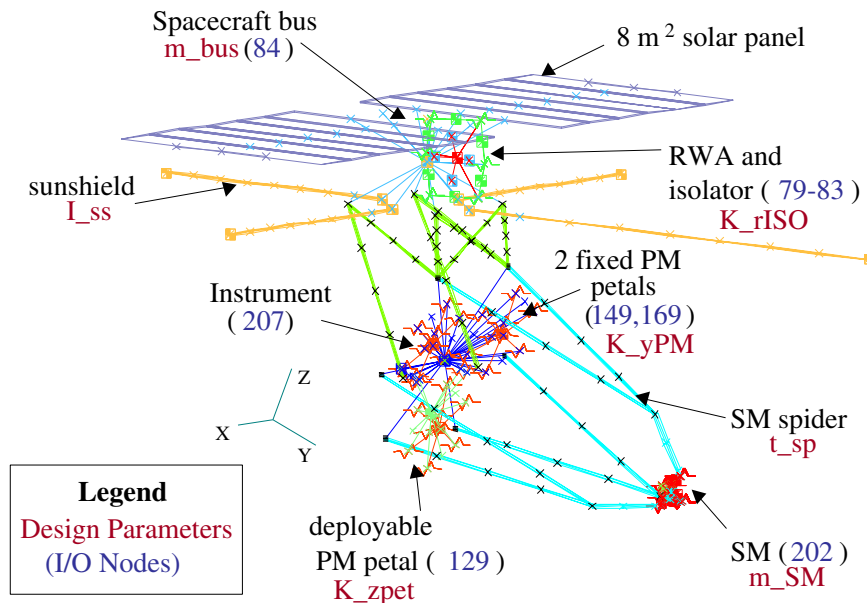
NEXUS features a 2.8 m primary mirror, consisting of three AMSD-sized primary mirror (PM) petals. Two of these are fixed and one is deployable as shown in Figure 7.1 on the left side. The total mass of the spacecraft is nominally 810 [kg] at a cost of \$M 105.88 (FY00), which includes launch and mission operations. The expected power consumption is 225 [W] and the target orbit is the Lagrange point L2 of the Sun/Earth system with a projected launch date of 2004. The optical telescope assembly (OTA) also features a 3-legged spider, which supports the secondary mirror (SM). The instrument module contains the optics downstream of the tertiary mirror and the camera (detector). The sunshield is large, deployable and accounts for the first flexible mode of the spacecraft structure around 0.2 Hz.

The challenge at a systems level is to find a design that will meet optical performance requirements in terms of pointing and phasing of the science light. This has to be done taking into account the flexible dynamics of the system, the control loops for attitude and pointing as well as the on-board mechanical and electronic noise sources. The following analyses are carried out in order to find such a well “balanced” design, using the isoperformance technique.

## 7.2 Integrated Modeling

### 7.2.1 Finite Element Model

The integrated model for NEXUS contains a structural finite element model (FEM) in the deployed configuration, see Figure 7.2. The model was initially created in FEMAP/NASTRAN and subsequently translated to IMOS [62]. The advantage of IMOS is that it can easily manipulate the model for parametric trade studies (such as isoperformance) in MATLAB, whereas NASTRAN is better suited for the analysis of large, high-fidelity point designs. This model features 273 grid points, 678 independent degrees-of-freedom after Guyan reduction and is optimized for use as a dynamics model below  $\sim 100$  [Hz]. Figure 7.2 shows the important locations at which disturbance and control inputs enter as well as important output nodes for the ACS as well as the locations where optical elements are mounted.

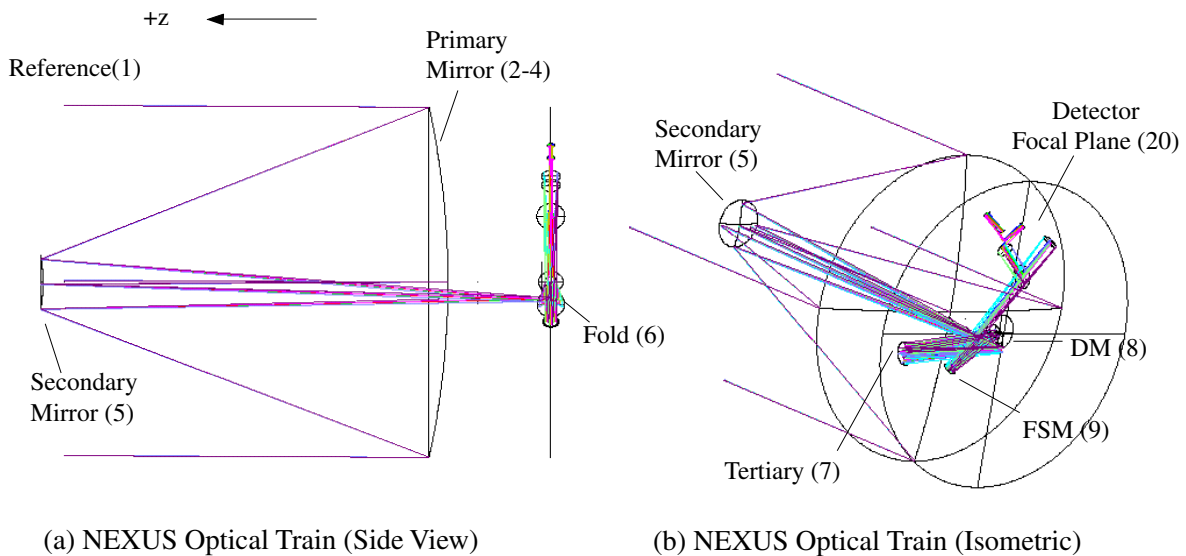


**Figure 7.2:** NEXUS Finite Element Model. Important I/O grid points (nodes) and variable design parameters are shown.

The primary mirror petals are rigid body elements, but the actuators are modeled as spring elements. The sunshield is approximated as four flexible beams with the membrane added as distributed non-structural mass (nsm). The instrument is modeled as a point mass with inertia matrix. The secondary spider support structure is modeled with three flexible bipods. The variable FEM parameters are shown in Table 7.1 under “plant parameters”.

## 7.2.2 Optics Model

The Cassegrain optics of NEXUS consist, among others, of a three petal primary mirror with an equivalent diameter of 2.8 [m]. Two of the petals are fixed (PM segments #2 and 3) to the primary optical bench, while the third petal (PM segment #1) is deployable. The hinge stiffness,  $K_{zpet}$ , is one of the variable design parameters considered in Table 7.1. The light from distant science targets and guide stars is then reflected from the concave primary and directed towards the convex secondary mirror (SM). Note that the optical boresight axis in the optics model (ZEMAX) is in the +z direction. The back end optics consist of a fold mirror, a focal tertiary mirror, a deformable mirror (DM), a flat fast steering mirror (FSM), several dichroics and camera fold mirrors and, finally, the exit pupil and the detector focal plane. A ray tracing diagram of the NEXUS optical train is shown in Figure 7.3. The optical prescription contains a total of 20 optical elements, including the source reference plane (object) and the detector focal plane (image).



**Figure 7.3:** NEXUS Optical Train modeled with ZEMAX. (a) Side View. (b) Isometric View. Selected mirror surfaces are labeled according to their element number (`iEl $t$` ) in the NEXUS OTA prescription. Key optics data: PM  $f/\# = 1.25$ , Magnification  $M = 12$ , back focal length  $BFL = 0.2$  [m], SM diameter 0.27 [m],  $f/15$  beam at Cassegrain focus,  $f/24$  telescope at focal plane, same as NGST. Plate scale = 2.06 [masec/ $\mu$ m].

Ray tracing according to the method developed by Redding and Breckenridge [120] is used to characterize the effect of perturbations in the positions and rotations of the optical

elements. The motion of optical elements affects the image quality of NEXUS. This effect is characterized by the dependence of the image centroid and wavefront error on the translation and rotation of optical components. The two performance metrics of interest are the root-mean-mean-square wavefront error,  $J_{z,1} = \text{RMMS WFE}$ , and the root-sum-square line-of-sight jitter,  $J_{z,2} = \text{RSS LOS}$ . The optical linear sensitivity matrices for these performances with respect to the translations and rotations of the optical elements were computed with MACOS, see Reference [83] for details. The wavefront error and centroid are then computed with the following, linearized relationships:

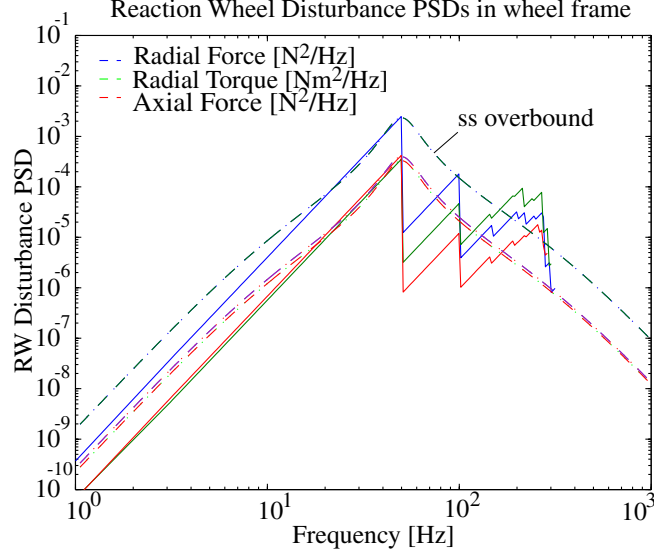
$$\begin{aligned}
 W_i &= W_{o,i} + \sum_{j=1}^{n_{dof}} \frac{\partial W_i}{\partial u_j} \cdot \Delta u_j \quad \text{where } i = 1, 2, \dots, n_{rays} \\
 C_x &= \sum_{j=1}^{n_{dof}} \frac{\partial C_x}{\partial u_j} \cdot \Delta u_j \quad \text{and} \quad C_y = \sum_{j=1}^{n_{dof}} \frac{\partial C_y}{\partial u_j} \cdot \Delta u_j
 \end{aligned}
 \tag{7.1}$$

where  $W_{o,i}$  is the residual design wavefront error of the  $i$ -th ray,  $\partial W/\partial u$ , is the wavefront sensitivity matrix,  $u$  is a vector of displacements and rotations and  $\partial C/\partial u$  is the centroid linear sensitivity matrix. A total of  $n_{rays}=1340$  rays are used for the analysis. The RMMS metric averages the  $W_i$ 's over the entire light bundle, while the LOS jitter metric is the root-sum-squared (RSS) of  $C_x$  and  $C_y$ .

### 7.2.3 Disturbance Sources

There are four expected disturbance sources in the NEXUS integrated model ( $n_d = 4$ ). The first is broadband reaction wheel noise, assuming a 4-wheel pyramid and uniform probability density on the wheel speed distribution, with an upper (operational) wheel speed  $R_u$ . The disturbance forces and torques are caused by static and dynamic imbalances,  $U_s$  and  $U_d$ , as well as higher harmonics. Figure 7.4 shows the typical ‘‘sawtooth’’ pattern of the broadband disturbance PSD’s for a single wheel along with low-order state space overbounds. This allows including the pre-whitening filters in the overall state space system,  $S_{zd}$ .

The second disturbance is due to a linear Sterling cryocooler at drive frequency  $f_c$ . This device is used to cool the IR detector and is installed in the instrument module. The third disturbance is attitude noise, which is based on rate gyro noise and star tracker noise measured on the Cassini mission (JPL). Finally there is guide star noise, which is very sensitive to the guider sampling rate,  $T_{gs}$ , and the guide star brightness,  $M_{gs}$ .

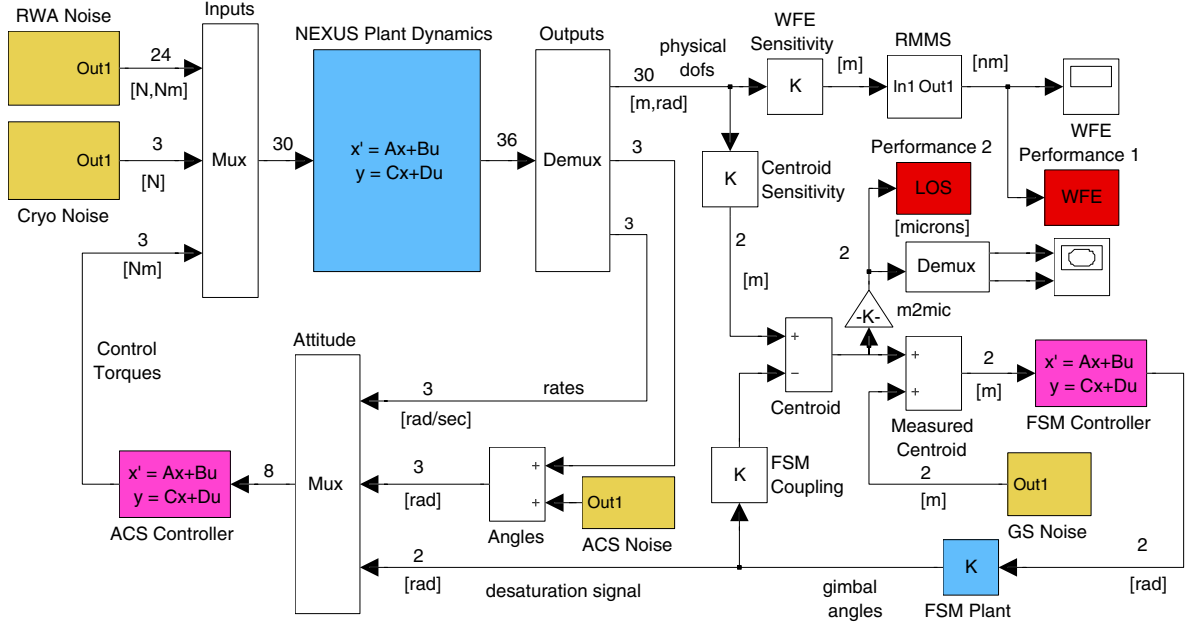


**Figure 7.4:** NEXUS broadband reaction wheel disturbance model. Nominal parameters:  $R_u = 3000$  [RPM],  $U_s = 0.7160$  [gcm] and  $U_d = 29.536$  [gcm<sup>2</sup>].

## 7.2.4 Appended Dynamics and Controls

The appended dynamics of this system are shown in the block diagram of Figure 7.5. These dynamics have also been cast in an equivalent state space form as shown in Equation 7.2. Note that the subscripts refer to the respective subsystem dynamics:  $dw$  reaction wheel disturbance,  $dc$  cryocooler disturbance,  $ds$  ACS sensor noise,  $dg$  guide star noise,  $p$  structural plant,  $ca$  ACS controller and  $cf$  for the FSM controller.

$$\begin{aligned}
 \dot{q}_{zd} = & \begin{bmatrix} A_{dw} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & A_{dc} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & A_{ds} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & A_{dg} & 0 & 0 & 0 & 0 \\ B_{p1}RC_{dw} & B_{p2}C_{dc} & 0 & 0 & A_p & B_{p3}C_{ca} & 0 & 0 \\ 0 & 0 & B_{ca2}C_{ds} & 0 & B_{ca}C_p & A_{ca} & B_{ca3}C_{cf} & 0 \\ 0 & 0 & 0 & B_{cf}C_{dg} & B_{cf}\frac{\partial C}{\partial u}C_{p1} & 0 & A_{cf} - B_{cf}K_{fsm}C_{cf} & 0 \end{bmatrix} q_{zd} \\
 + & \begin{bmatrix} B_{dw} & 0 & 0 & 0 \\ 0 & B_{dc} & 0 & 0 \\ 0 & 0 & B_{ds} & 0 \\ 0 & 0 & 0 & B_{dg} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} d_{RWA} \\ d_{Cryo} \\ d_{ACS} \\ d_{FGS} \end{bmatrix} \\
 z = & \begin{bmatrix} 0 & 0 & 0 & 0 & \frac{\partial W}{\partial u}C_{p1} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{\partial C}{\partial u}C_{p1} & 0 & -K_{fsm}C_{cf} \end{bmatrix} q_{zd} + [0]_{1340 \times 10} d
 \end{aligned} \tag{7.2}$$



**Figure 7.5:** NEXUS block diagram with 4 disturbance sources (RWA, Cryo, ACS noise, GS noise) and 2 performances (RMMS WFE, RSS LOS). Simulation implemented in Simulink as well as state space.

In summary the appended dynamics,  $S_{zd}$ , of this system contain 320 states ( $n_s = 320$ ), two performance metrics ( $n_z = 2$ ), four disturbance sources ( $n_d = 4$ ) and 25 variable design parameters ( $n_p = 25$ ). To the author's knowledge this is the first occurrence in the literature where variable disturbance, structural, optics and control parameters are considered simultaneously. Mostly one finds subsets such as controls/structures, but with the assumption of fixed noise sources. Table 7.1 summarizes the variable design parameters in the NEXUS case study.

### 7.3 Disturbance Analysis

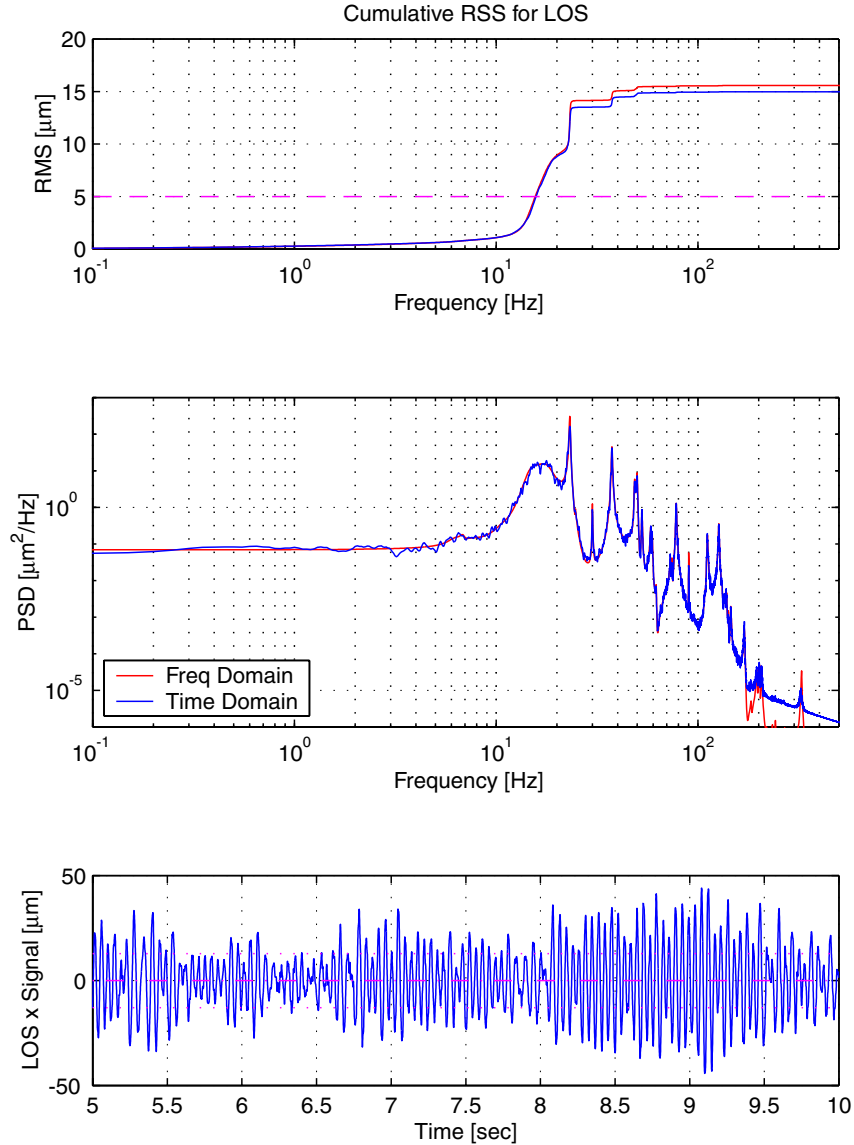
A disturbance analysis of the science target observation mode was carried out with the initial parameters,  $p_o$ , given in Table 7.1. Results for LOS jitter are contained in Figure 7.6. The bottom plot shows a sample time realization for 5 seconds and the centroid X location. The middle plot shows the PSD of LOS jitter (RSS LOS) for a frequency domain and time domain calculation. The top plot is the cumulative RMS of LOS jitter as a function of frequency. One can see that a mode at 23 [Hz] contributes significantly to LOS jitter (secondary tower bending). The group of highly damped modes in the region from 3-10 [Hz] represents the RWA isolator dynamics.

**Table 7.1:** NEXUS Variable Design Parameters  $p_j$ ,  $j = 1, \dots, 25$ .

Number	Symbol	Nominal	Description	Units
disturbance parameters				
1	$Ru$	3000	Upper operational wheel speed	[RPM]
2	$Us$	1.8	Static wheel imbalance	[gcm]
3	$Ud$	60	Dynamic wheel imbalance	[gcm <sup>2</sup> ]
4	$fc$	30	Cryocooler drive frequency	[Hz]
5	$Qc$	0.005	Cryocooler attenuation factor	[-]
6	$Tst$	20	Star tracker update rate	[sec]
7	$Srg$	3e-14	Rate gyro noise intensity	[rad <sup>2</sup> /s]
8	$Sst$	2	Star tracker one sigma	[arcsec]
9	$Tgs$	0.04	Guider integration time	[sec]
plant parameter				
10	$m_{SM}$	2.49	mass of secondary mirror	[kg]
11	$K_y PM$	0.8e6	Primary mirror bipod stiffness	[N/m]
12	$K_r ISO$	3000	RWA Isolator joint stiffness	[Nm/rad]
13	$m_{bus}$	0.3e3	Spacecraft bus mass	[kg]
14	$K_{zpet}$	0.9e8	PM petal hinge stiffness	[N/m]
15	$t_{sp}$	0.003	Spider wall thickness	[m]
16	$I_{ss}$	0.8e-8	Sunshield bending inertia	[m <sup>4</sup> ]
17	$I_{propt}$	5.11	Propulsion system inertia	[kgm <sup>2</sup> ]
18	$\zeta$	0.005	modal damping ratio	[-]
optics parameters				
19	$\lambda$	1e-6	Centerline optical wavelength	[m]
20	$Ro$	0.98	Optical surface reflectivity	[-]
21	$QE$	0.80	CCD quantum efficiency	[-]
22	$Mgs$	15.0	Magnitude of guide star	[mag]
controls parameters				
23	$fca$	0.01	ACS control bandwidth	[Hz]
24	$Kc$	0.0	FSM/ACS coupling gain	[0-1]
25	$Kcf$	2000	FSM controller gain	[-]



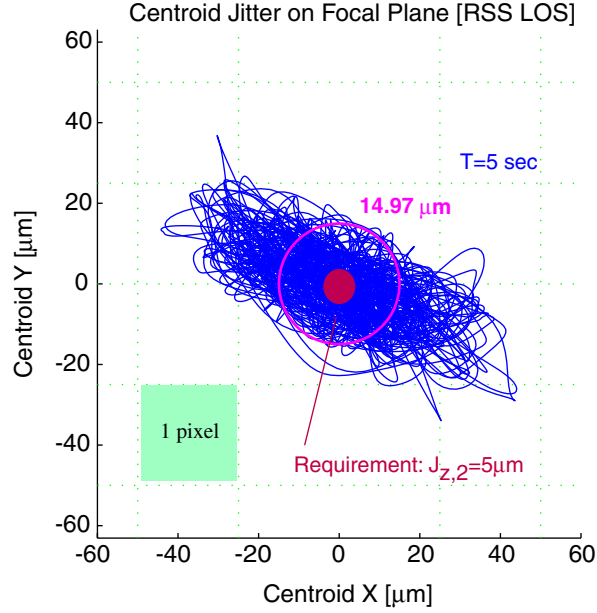
Another way to look at performance  $J_{z,2}$  is to plot the time histories from the motions of centroid X and Y versus each other. This has been done in Figure 7.7. The predicted RSS LOS is  $14.97 \mu m$ , versus a requirement of  $5 \mu m^1$ . Note that the RSS of the centroid jitter is larger than the size of a single pixel ( $25 \times 25 [\mu m]$ ), which is undesirable.



**Figure 7.6:** LOS Jitter initial disturbance analysis

The wavefront error performance is omitted here for simplicity. Table 7.2 shows an overview of the predicted performance, using the initial parameters  $p_o$ . The wavefront error

<sup>1</sup>This requirement comes from the assumption of  $25 \mu m$  pixel pitch and a desire to maintain LOS jitter below  $1/5$  of a pixel.



**Figure 7.7:** RSS LOS Centroid Jitter Plot on Focal Plane (NEXUS).

requirement ( $\lambda/50$ ) is nearly met, but the pointing performance has to improve by a factor of roughly 3. This is not atypical for many isoperformance analyses with multiple performances ( $n_z > 1$ ), where only a subset of performance requirements is initially close to being met.

**Table 7.2:** Initial Performance Analysis Results

Performance	Lyap	Time	Req	Units
$J_{z,1}$ RMMS WFE	25.61	19.51	20	[nm]
$J_{z,2}$ RSS LOS	15.51	14.97	5	[ $\mu$ m]

The next step is to conduct a comprehensive sensitivity analysis using the relationships presented in Appendix A and the enhancements developed in Chapter 5 for large order systems.

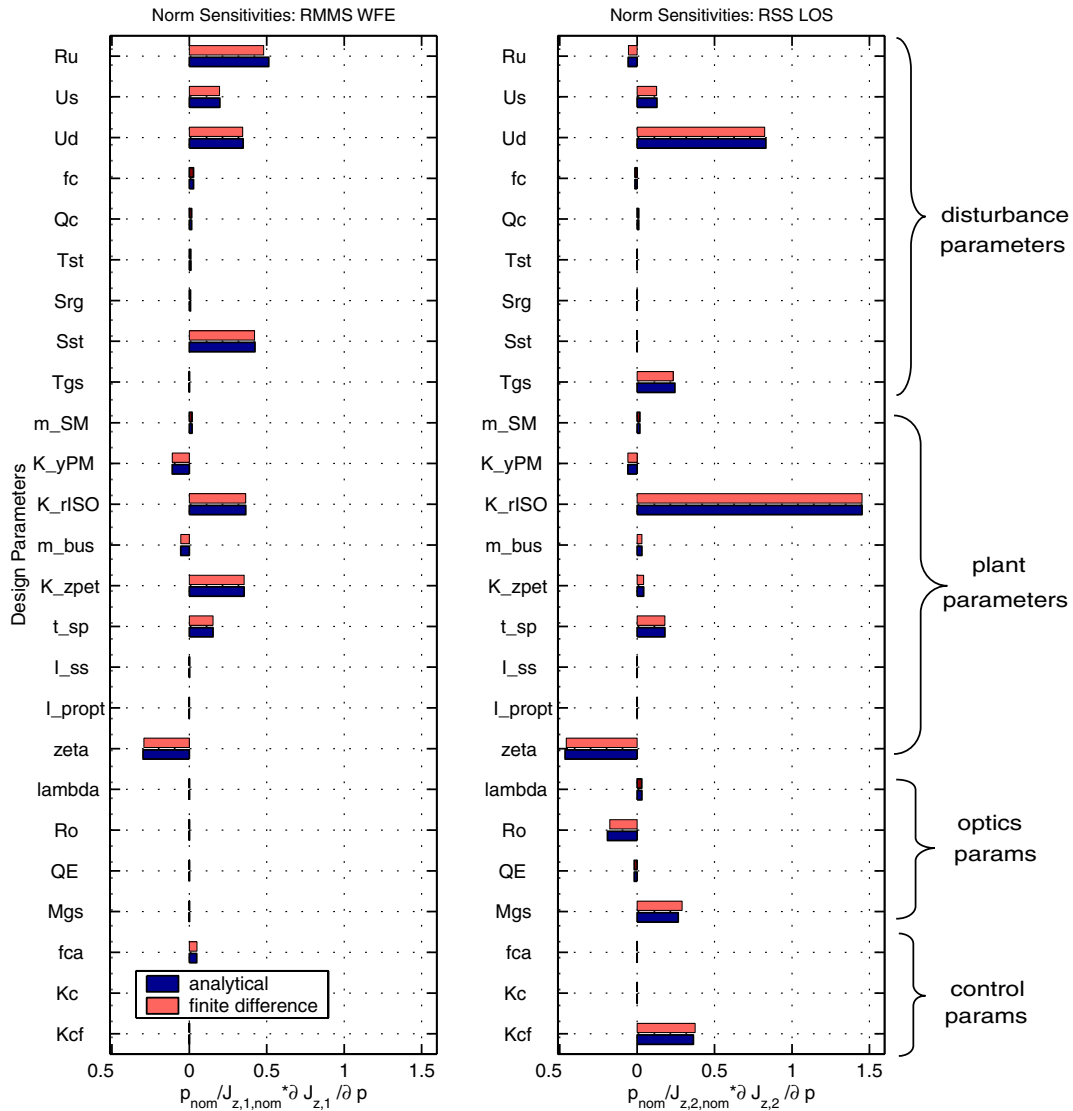
## 7.4 Sensitivity Analysis

This section shows the results of a comprehensive sensitivity analysis for the 25 variable design parameters of NEXUS which are shown in Table 7.1. The sensitivity produces the normalized

Jacobian matrix ( $25 \times 2$  matrix) evaluated at  $p_o$

$$\bar{\nabla} J_z = \frac{p_o}{J_{z,0}} \begin{bmatrix} \frac{\partial J_{z,1}}{\partial R_u} & \frac{\partial J_{z,2}}{\partial R_u} \\ \dots & \dots \\ \frac{\partial J_{z,1}}{\partial K_{cf}} & \frac{\partial J_{z,2}}{\partial K_{cf}} \end{bmatrix} \quad (7.3)$$

which is graphically shown in Figure 7.8. Note that parameters  $Ru$  through  $Tgs$  are disturbance parameters,  $m_{SM}$  through  $zeta$  are structural plant parameters,  $lambda$  through  $Mgs$  are optical parameters and  $fca$  through  $Kcf$  are control parameters.



**Figure 7.8:** NEXUS normalized sensitivity analysis results at  $p^o$ .

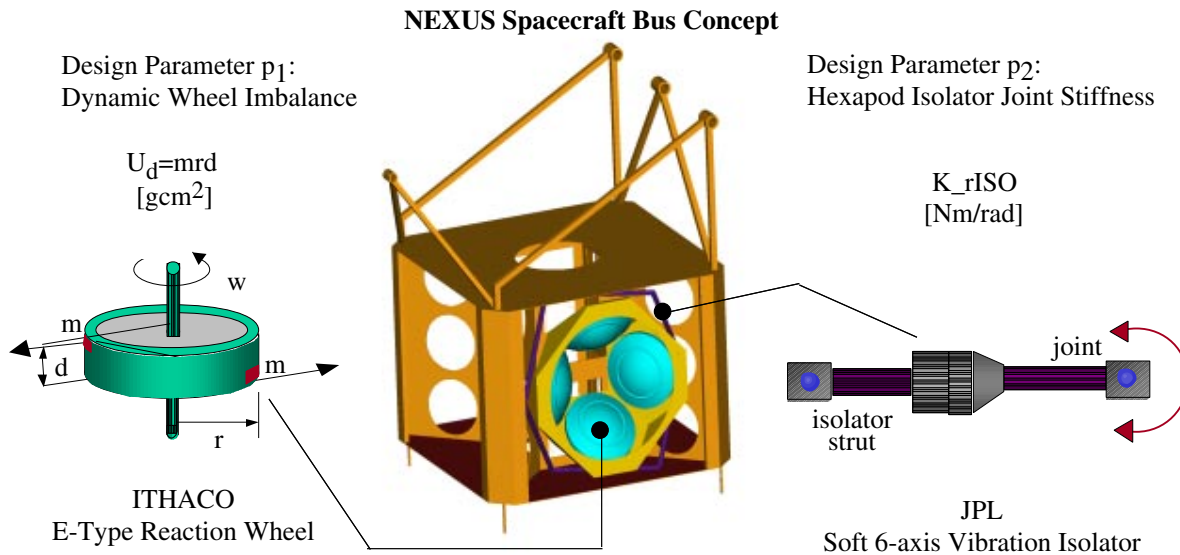
The RMMS WFE is most sensitive to the upper operational wheel speed,  $Ru$ , the RWA

isolator stiffness,  $K_r ISO$ , and the deployable petal hinge stiffness,  $K_z pet$ . The RSS LOS is most sensitive to the dynamic wheel imbalance,  $Ud$ , the RWA isolator stiffness,  $K_r ISO$ , structural damping,  $zeta$ , the guide star magnitude,  $Mgs$  and the FSM (fine pointing loop) control gain,  $Kcf$ . Interpreting these results one would expect for example that a 1.0 % decrease in the isolator stiffness,  $K_r ISO$  should lead to roughly a 1.5 % decrease in LOS jitter. The sensitivity analysis can be used to select a subset of interesting parameters for further analysis.

## 7.5 Bivariate Isoperformance

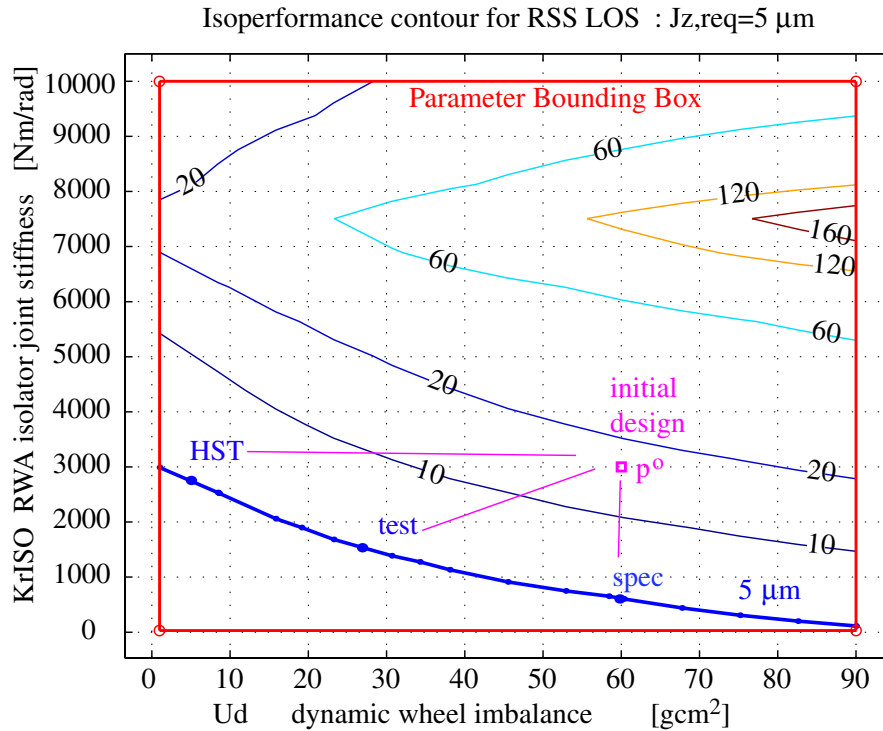
### 7.5.1 Imbalance versus Isolation

A bivariate isoperformance analysis is conducted for NEXUS using  $J_{z,1} = \text{RSS LOS}$  as the performance and the two most sensitive parameters from Figure 7.8, right column, as the parameters. Hence, dynamic wheel imbalance,  $Ud$ , is traded versus RWA isolator joint stiffness,  $K_r ISO$ , while constraining the performance to the requirement,  $J_{z,2,req} = 5[\mu m]$ . A graphical representation of these two variable parameters in the context of the NEXUS spacecraft bus design is shown in Figure 7.9.



**Figure 7.9:** NEXUS Bus design with a 4-wheel symmetric pyramid of ITHACO E-wheels (39.3 [cm] diameter, 16.6 [cm] height, 10.6 [kg] mass each). See Reference [130] for 6-axis Active Vibration Isolator design.

The isoperformance contours (Figure 7.10) were obtained using the exhaustive search algorithm developed in Subsection 3.2.1. This analysis required 1506.2 [sec] of CPU time (Pentium III, 650 MHz processor) and a total of  $2.51 \cdot 10^{11}$  FLOPS. This is significantly more expensive than the analyses for the sample problems presented in Chapter 2. The use of the fast, diagonal Lyapunov solver causes the FEM (mass and stiffness) assembly time to be the most time consuming operation instead of the solution of the Lyapunov equation for the state covariance matrix,  $\Sigma_q$ .



**Figure 7.10:** NEXUS Bivariate Isoperformance analysis with  $p_1 = U_d$ ,  $p_2 = K_r ISO$  and  $J_z = RSS LOS$ .

The isoperformance contour at  $RSS LOS = 5 \mu m$  can be reached from the initial design,  $p^o$ , by keeping the same amount of imbalance in the wheels (specification value of E-wheel:  $U_d = 60 [gcm^2]$ ) and softening the isolator to below 1000 [Nm/rad], thus reducing the isolator corner frequency to roughly 1.2 Hz. Alternatively the isolator can remain the same and the imbalance could be reduced to close to its lower bound,  $U_d=1 [gcm^2]$ . The isoperformance contour passes through these two points, so a combination of the above is likely to result in the desired effect. Note that the performance degrades significantly for stiffer isolator joints and larger imbalances. The region in the upper right of Figure 7.10, where LOS jitter of 160

$\mu m$  is predicted, occurs, when the isolator modes coincide with other flexible modes of the NEXUS structure.

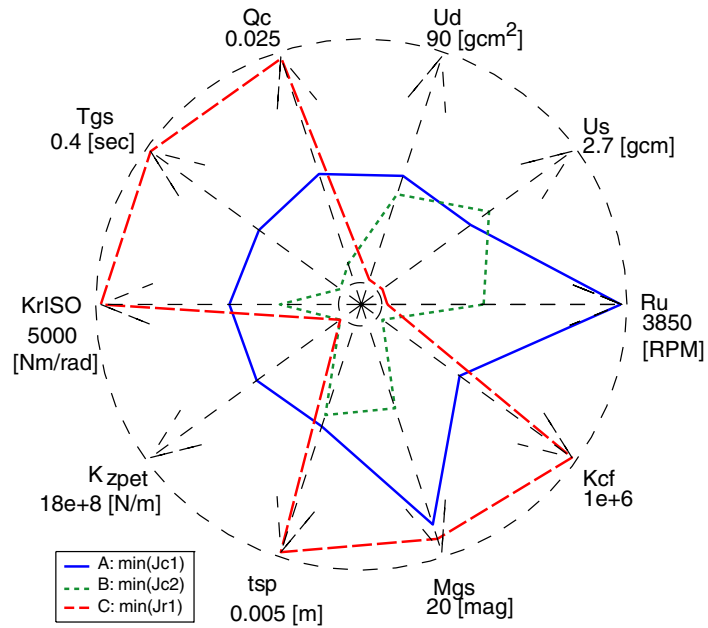
## 7.6 Multivariable Isoperformance

Since solutions,  $p_{iso}$ , in the isoperformance set  $\mathbf{I}$  do not distinguish themselves via their performance, we may satisfy some additional objectives. For the bivariate analysis in Section 7.5 for example it is not immediately clear whether it is more favorable or “expensive” to improve the balancing of the reaction wheels or to build a “softer” hexapod isolator. Once the (iso)performance requirements,  $J_z(p_{iso}) = J_{z,req}$ , are met one may consider competing cost objectives  $J_c$  (control effort, implementation cost, system mass, dissipated power, etc.) or risk objectives  $J_r$  (stability margins, sensitivity of performance to parametric uncertainty etc.). Which combination of  $J_c$  and  $J_r$  to use is application dependent. A non-linear optimization problem, given in (1.7) may be solved, whereby  $Q_{cc}$  and  $Q_{rr}$  are weighting matrices among the cost and risk objectives and  $\eta$  is used to trade between cost and risk. The result is a family of pareto optimal solutions,  $p_{iso}^*$ , which is presented to the designer. This type of multiobjective analysis was demonstrated for the sample problems in Section 4.7.

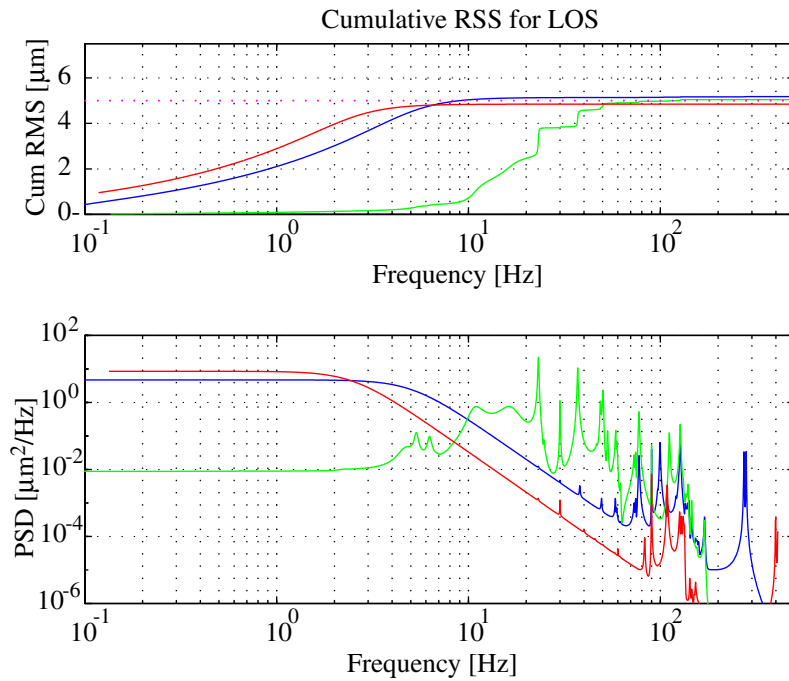
In the NEXUS case a multivariable analysis was conducted for a subset of 10 out of the 25 design parameters from Table 7.1. The two performance objectives RMMS WFE and RSS LOS were defined above. The cost and risk objectives are defined as follows:

- $J_{c,1}$  = Build-to Cost (closeness to “mid-range”), Equation (4.31)
- $J_{c,2}$  = Smallest FSM control gain,  $K_{fsm}$
- $J_{r,1}$  = Percent performance uncertainty, Equation (2.7)

The three pareto optimal solutions, which each individually optimize one of the above objectives, while meeting the isoperformance condition, are shown in the radar plot of Figure 7.11. Specifically, the isoperformance condition leads to the fact that all designs,  $p_{iso}^*$ , asymptote to the same value in the cumulative RMS plot, as shown for RSS LOS in Figure 7.12.



**Figure 7.11:** NEXUS Multivariable Isoperformance. Radar plot of 3 pareto optimal designs.  $J_{c,1}$  is best mid-range design,  $J_{c,2}$  is the design with smallest FSM gain,  $J_{r,1}$  is the design with smallest performance uncertainty.



**Figure 7.12:** NEXUS Pareto Optimal Designs: RSS LOS power spectral densities (bottom) and cumulative RMS curves (top).

The results for the NEXUS pareto optimal designs are summarized in Table 7.3.

**Table 7.3:** NEXUS pareto optimal designs

	$J_{z,1}$	$J_{z,2}$	$J_{c,1}$	$J_{c,2}$	$J_{r,1}$
A	20.0000	5.2013	<b>0.6324</b>	0.4668	$\pm 14.3 \%$
B	20.0012	5.0253	0.8960	<b>0.0017</b>	$\pm 8.8 \%$
C	20.0001	4.8559	1.5627	1.0000	$\pm 5.3 \%$

Even though these designs achieve the same WFE and LOS jitter performance, their dominant contributors in terms of disturbance sources are likely different. This leads naturally to the application of isoperformance for dynamics error budgeting.

## 7.7 Error Budgeting

Error budgeting finds a design, which balances the error contributions from all expected sources (e.g. reaction wheel imbalances, sensor noise) and assesses the feasibility of an apriori allocation. Table 7.4 shows the apriori allocation,  $\Psi$ , and the actual disturbance contributions,  $\Psi^{**}$ , to the variance of RSS LOS for Design “A”, which is chosen as the final design,  $p_{iso}^{**}$ .

**Table 7.4:** NEXUS error budget  $\Psi_2$

Error Source	variance %	Budget [ $\mu\text{m}$ ]	variance %	Capability [ $\mu\text{m}$ ]
RWA	50.00	3.54	0.92	0.499
Cryocooler	25.00	2.50	0.22	0.244
ACS Noise	5.00	1.12	0.00	7E-6
GS Noise	20.00	2.24	98.8	5.172
<b>Total</b>	100	5.00	100	5.2013

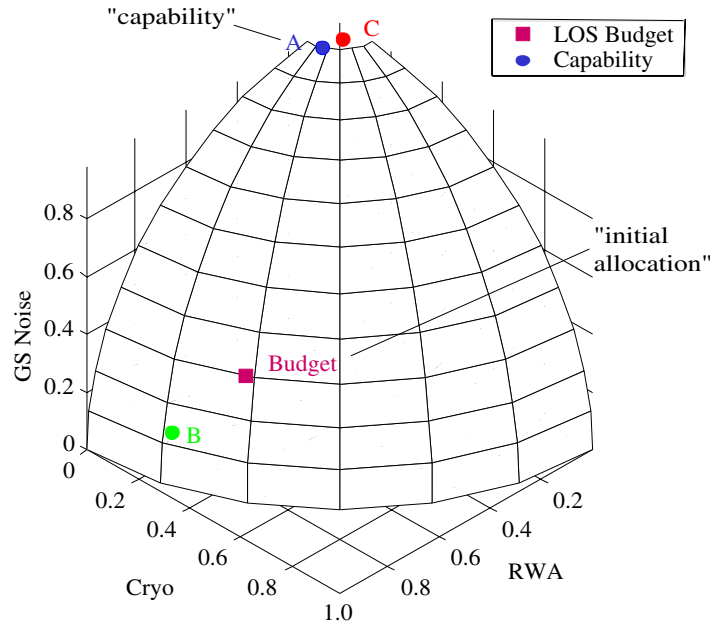
The error budget can be expressed in terms of the fractional contribution of the  $j$ -th disturbance source to the  $i$ -th performance as

$$\Psi_i = \sum_{j=1}^{n_d} \Psi_{i,j} = J_{z,req,i}^2 \quad (7.4)$$

The relative contributions to the performance can be shown by plotting the fractional



contributions of the  $j$ -th error source on a sphere (not showing ACS noise). This sphere is called the Error Sphere, see Figure 7.13.



**Figure 7.13:** NEXUS Error Sphere for RSS LOS. Note: ACS sensor noise contributions not shown.

Error Budgeting is an obvious application of isoperformance, since an a priori error budget will always result in the desired performance level. The advantage of using isoperformance in this context is that a “capability” error budget,  $\Psi^{**}$ , can be found, which is theoretically achievable since it is based on the underlying integrated model.

## 7.8 Summary

This chapter demonstrates the usefulness of isoperformance on a realistic conceptual design of a space telescope. NEXUS is chosen due to its interesting, flexible dynamics and the presence of important disturbance, plant, control and optics parameters. An initial disturbance and sensitivity analysis are conducted for an initial vector,  $p_o$ , of 25 design parameters. A bivariate isoperformance analysis trades dynamic wheel imbalance,  $U_d$ , versus isolator corner frequency,  $K_r ISO$ . A multivariable isoperformance analysis is conducted with 10 parameters. By applying cost and risk objectives, such as implementation cost (closeness to “mid-range”), smallest FSM control gain ( $K_{cf}$ ) and smallest performance uncertainty a set of three pareto

optimal designs was identified. The application of isoperformance to dynamics error budgeting is demonstrated by comparing an apriori allocation (“budget”) with the error source contributions of a pareto optimal design.

## Chapter 8

# Conclusions and Recommendations

### 8.1 Thesis Summary and Conclusions

Precision opto-mechanical systems, such as space telescopes, combine structures, optics and controls in order to meet stringent pointing and phasing requirements. In this context a novel approach to the design of complex, multi-disciplinary systems is presented in this thesis in the form of a multivariable isoperformance methodology. Chapter 1 gives an introduction into the context and previous work and a rigorous mathematical definition of the problem using the state space formulation and set theory. Three sample problems (single degree-of-freedom (SDOF) oscillator, 2-DOF oscillator and ODL design) are developed in Chapter 2. These problems are used throughout the thesis to gain intuitive understanding, to verify the correct implementation of the governing equations and to test the algorithms before they are applied to complex, large and expensive problems.

The isoperformance approach first finds a point design within a topology, which meets the performance requirements with sufficient margins. This is mainly based on the performance assessment and enhancement framework developed by Gutierrez [45], which is summarized in Appendix A. The performance outputs are then treated as equality constraints and the non-uniqueness of the design space is exploited by trading key disturbance, plant, optics and controls parameters with respect to each other. Reasonable upper and lower bounds have to be specified for the design parameters.

The isoperformance problem is initially solved for only two variable parameters ( $n_p = 2$ ) and a single performance ( $n_z = 1$ ), see Chapter 3. Three algorithms (exhaustive search, tangential contour following and cubic spline approximation) are developed and compared

for the SDOF problem. While exhaustive search appears to be the most expensive, it does not require gradient information and provides multiple contours via linear interpolation. The algorithm, which shows most promise in the multivariable case is spline approximation. It, however, is also the most restrictive, since it requires that the underlying performance function  $J_z(p_1, p_2)$  is monotonic in at least one of the two parameters (no closed contours). The bivariate isoperformance methodology is demonstrated for a number of sample problems and for a conceptual model of SIM Classic.

The previous algorithms are generalized for the multivariable case (Chapter 4), when  $n_p > 2$ . These algorithms are branch-and-bound, tangential front following and vector spline approximation. The results are viewed as isoperformance contours, nomograms or radar-plots. The challenges of computational complexity and graphical representation in the multi-dimensional case are also addressed. It turns out that the isoperformance problem cannot be solved in polynomial time as a function of the number of variable parameters,  $n_p$ . However, a meaningful analysis for a limited set of parameters ( $n_p < 20$ ) is reasonably achievable.

The challenges of large order models are addressed in Chapter 5 by presenting a fast diagonal Lyapunov solver, apriori error bounds for model reduction as well as a governing sensitivity equation for similarity transformed state space realizations. These contributions enable sensitivity and isoperformance analyses for large order integrated models, which were previously not possible. Proofs of the governing equations for similarity transformed systems are contained in Appendix B.

The goal of the experimental validation (Chapter 6) is to demonstrate the ability of the isoperformance methodology and toolbox to predict isoperformance contours on a real physical system. The 1-g DOLCE testbed was designed to have multiple disturbance and structural parameters, which are variable over a large range. The testbed characterization of the transfer function from disturbance force,  $F_d(t)$  to base displacement,  $z(t)$ , shows that two modes are dominant in the bandwidth of interest up to 100 Hz. A test matrix is experimentally determined with the RMS excitation voltage of the shaker,  $V_s$ , and the payload mass,  $m_p$ , as independent parameters. Theoretical isoperformance contours are predicted with a SDOF spring-mass model, an un-updated FEM and an updated (tuned) FEM. These contours are compared with experimental contours obtained from the test matrix via linear interpolation. Good agreement is shown for the 7.5  $\mu\text{m}$  RMS displacement level, where the excitation amplitude is small. This corresponds to the low vibration levels expected on precision opto-

mechanical spacecraft. At higher disturbance levels (15 and 30  $\mu\text{m}$  contours) the agreement is not as good and deviations between the experimental and theoretical performance are likely due to structural non-linearities.

Chapter 7 demonstrates the usefulness of isoperformance on a realistic conceptual design of a space telescope. NEXUS is chosen due to its interesting, flexible dynamics and the presence of important disturbance, plant, control and optics parameters. A disturbance and sensitivity analysis is conducted for an initial vector,  $p_o$ , of 25 design parameters. A bivariate isoperformance analysis trades dynamic wheel imbalance,  $U_d$ , versus isolator corner frequency,  $K_r ISO$ . A multivariable isoperformance analysis is conducted with 10 parameters. By applying cost and risk objectives, such as implementation cost (closeness to “mid-range”), smallest FSM control gain,  $K_{cf}$ , and smallest performance uncertainty a set of three pareto optimal designs was identified. The application of isoperformance to dynamics error budgeting is demonstrated by comparing an apriori allocation (“budget”) with the error source contributions of a pareto optimal design.

Isoperformance helps to avoid situations where very costly and hard-to-meet requirements are levied onto one subsystem, while other subsystems hold substantial margins. It is suggested that isoperformance is a useful concept in other fields of engineering science such as crack growth calculations in structures, see Appendix C. The isoperformance approach enhances the understanding of complex opto-mechanical systems by exploiting physical parameter sensitivity and performance information beyond the local neighborhood of a particular point design. All of the thesis research objectives formulated in Subsection 1.3.1 were accomplished.

## 8.2 Contributions

This thesis develops and validates a novel approach to the design of complex multi-disciplinary systems. The following specific thesis contributions can be identified:

1. Developed a methodology for identifying the locus of parameters that yields constant  $\mathcal{H}_2$  performance levels of an LTI system (isoperformance).
2. Applied isoperformance analysis to model-based error budgeting and multiobjective optimization for dynamic systems.
3. Extended disturbance and sensitivity analysis to handle large size models (up to 2200 states demonstrated), transformed state space systems, as well as disturbance, plant, optics and control parameters.
4. Produced and validated a software toolbox for conducting 2D or multivariable isoperformance analyses, compatible with DOCS.
5. Experimental validation of isoperformance technique on a laboratory test article in 1-g with two parameters.
6. Demonstrated applicability of isoperformance technique for a structural fatigue problem, suggesting use in other fields.

## 8.3 Limitations

The limitations of the isoperformance framework are that it assumes Linear-Time-Invariant (LTI) systems and operates on  $H_2$ -performance metrics for zero-mean random processes. At present transient performance metrics, such as the settling time presented in the ODL sample problem (Section 2.3), can only be accommodated as additional objectives during multiobjective optimization but not during an isoperformance analysis. Furthermore the dynamics are treated in continuous time (no z-domain capability). The algorithms (except exhaustive search) require continuous and differentiable parameters and work within a given topology/architecture. Finally it shall be mentioned that thermal dynamics have not been included, since they tend to occur at much slower time scales and don't typically couple strongly to the flexible dynamics of a structure.

## 8.4 Recommendations for Future Work

The recommendations for future work focus on removing some of the current limitations and applying the isoperformance concept on a more holistic level in product design and system architecture. Isoperformance meshes well with a product design philosophy called “satisfying”. In this approach not a system that optimizes the performance is sought, but rather a system that meets identified customer performance requirements, while being designed in a cost effective way. Specific recommendations for future research are:

- Robustify isoperformance toolbox on other projects
- Closed loop experimental validation (ORIGINS testbed)
- Extend methodology for discrete/continuous problems
- Extend methodology for non-steady-state/transients
- Investigate application to FEM updating/tuning
- Analyze effect of non-linearities on isoperformance contour predictions
- Automate step size determination for tangential front following in the multivariable case
- Investigate ways of generating simple functional approximations to the isoperformance surfaces
- Research the relationship between isoperformance, parameter constraints and technology roadmapping
- Link to emerging system architecture, and product design methodologies





# Appendix A

## Summary of Disturbance, Sensitivity and Uncertainty Framework

The kernel of the current disturbance, sensitivity and uncertainty analysis framework was originally developed by Gutierrez [45]. It is a powerful methodology for predicting the root-mean-square (RMS)<sup>1</sup> of the performance outputs,  $z$ , recast as the performance metrics  $J_z$ . This performance prediction can occur using the time domain, PSD or Lyapunov approach. Additionally a Lagrange multiplier approach is used to obtain the (exact) analytical sensitivities. The computation of sensitivities in the time-domain or frequency domain has not yet been explored and remains for future work. The uncertainty analysis approach used in this thesis is the first order approach. The purpose of this appendix is to provide a short summary of the framework and the mathematics presented in [45], thus allowing this thesis to be self-contained and clearly establishing the theoretical foundation upon which the isoperformance methodology builds.

### A.1 Disturbance Modeling

A disturbance analysis is required, when deterministic or stochastic disturbances  $w(t)$  are present and it is unclear whether the performances  $J_z$  will remain within a required value  $J_{z,req}$ .

---

<sup>1</sup>or root-sum-square (RSS), root-mean-mean-square (RMMS)

Note that generally  $w$  and  $J_z$  will be vectors. The disturbances  $w(t)$  can be undesired forces, torques, base motion, sensor and actuator noise, among others. The correlation function of the random process  $w(t)$  is defined as

$$R_{ww}(t_1, t_2) = E[w(t_1)w^T(t_2)] \quad (\text{A.1})$$

where  $E[\ ]$  is the expectation operator and  $w(t)$  is a vector of stochastic random processes. If  $w(t)$  is stationary the values of  $w(t)$  will change over time, but the *statistics* of  $w(t)$  will not and  $R_{ww}$  is a function of the single time-lag  $\tau$ .

$$R_{ww}(\tau) = E[w(t)w^T(t+\tau)] \quad (\text{A.2})$$

Assuming that all  $w(t)$  are zero-mean, the covariance matrix  $\Sigma_w$  of the disturbance signals is the value of the correlation matrix  $R_{ww}$  for  $\tau = 0$ .

$$\begin{aligned} \Sigma_w = R_{ww}(0) &= \begin{bmatrix} E[w_1^2] & E[w_1w_2] & \cdots & E[w_1w_n] \\ E[w_2w_1] & E[w_2^2] & \cdots & E[w_2w_n] \\ \vdots & \vdots & \ddots & \vdots \\ E[w_nw_1] & E[w_nw_2] & \cdots & E[w_n^2] \end{bmatrix} \\ &= \begin{bmatrix} \sigma_{w_1}^2 & \sigma_{w_2w_1} & \cdots & \sigma_{w_1w_n} \\ \sigma_{w_2w_1} & \sigma_{w_2}^2 & \cdots & \sigma_{w_2w_n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{w_nw_1} & \sigma_{w_nw_2} & \cdots & \sigma_{w_n}^2 \end{bmatrix} \end{aligned} \quad (\text{A.3})$$

The simplification in the covariance matrix  $\Sigma_w$  can be made for zero-mean processes, since for a typical term  $\sigma_{w_iw_j}$  in the covariance matrix we can write

$$\begin{aligned} \sigma_{w_iw_j} &= E[(w_i - \mu_{w_i})(w_j - \mu_{w_j})] = \\ &= E[w_iw_j] - \underbrace{E[w_i\mu_{w_j}]}_{=0} - \underbrace{E[\mu_{w_i}w_j]}_{=0} + \underbrace{E[\mu_{w_i}\mu_{w_j}]}_{=0} = E[w_iw_j] = \sigma_{w_iw_j} \end{aligned} \quad (\text{A.4})$$

where  $\mu_{w_i} = E[w_i]$  is the mean (expected value) of the  $i$ -th random process. The mean-square values of the elements of  $w$  are simply the diagonal entries in the covariance matrix.

$$(\Sigma_w)_{i,i} = E[w_i^2(t)] \quad (\text{A.5})$$

where  $w_i(t)$  is the  $i$ -th element in  $w$ . If  $w$  is zero-mean, then the mean-square values and the variances are identical.

$$\sigma_{w_i}^2 = E[w_i^2(t)] - \underbrace{(E[w_i])^2}_{=0} = \text{RMS}^2 \quad (\text{A.6})$$

The (power) spectral density function  $S_{ww}(\omega)$  can be obtained by taking the Fourier transform of equation (A.2)

$$S_{ww}(\omega) = \mathcal{F}[R_{ww}(\tau)] = \int_{-\infty}^{+\infty} R_{ww}(\tau) e^{-j\omega\tau} d\tau \quad (\text{A.7})$$

Note that the  $1/2\pi$  factor is not included in the definition of the Fourier transform. Other authors [141] place it in the Fourier transform formula. Either definition will produce the same result in the end as long as the definition is used consistently. The inverse Fourier transform of  $S_{ww}(\omega)$  recovers the correlation function.

$$R_{ww}(\tau) = \mathcal{F}^{-1}[S_{ww}(\omega)] = \frac{1}{2\pi} \int_{-\infty}^{+\infty} S_{ww}(\omega) e^{+j\omega\tau} d\omega \quad (\text{A.8})$$

Evaluating (A.8) at  $\tau = 0$  will produce the covariance matrix of  $w$ .

$$R_{ww}(0) = \Sigma_w = \frac{1}{2\pi} \int_{-\infty}^{+\infty} S_{ww}(\omega) d\omega \quad (\text{A.9})$$

Equation (A.9) suggests an alternative way of calculating the mean-square values of  $w$  by integrating under the spectral density functions, namely

$$\sigma_{w_i}^2 = [\Sigma_w]_{i,i} = \frac{1}{2\pi} \int_{-\infty}^{+\infty} [S_{ww}(\omega)]_{i,i} d\omega \quad (\text{A.10})$$

The diagonal elements of the spectral density function matrix  $S_{ww}(\omega)$  are usually referred to as power spectral densities (PSD's), whereas the off-diagonal elements are the cross spectral densities. Equation (A.10) states that the variance is equal to the area beneath the PSD scaled by a factor of  $\frac{1}{2\pi}$ . Depending on the shape of the PSD's representing  $w$  it is possible to approximate the shape of the functions  $S_{ww}(\omega)$  by pre-whitening filters in state space form, where the inputs to the state space system are unit intensity white noise processes  $d$ :

$$\begin{aligned} \dot{q}_d &= A_{wd}q_d + B_{wd}d \\ w &= C_{wd}q_d + D_{wd}d \end{aligned} \quad (\text{A.11})$$

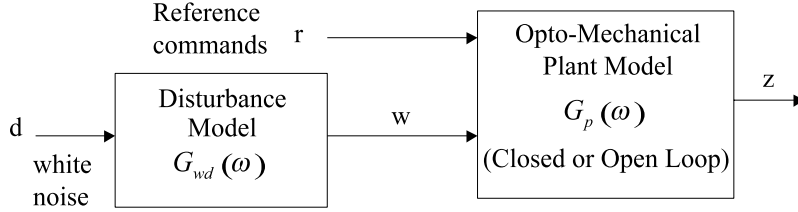
Note that the feedthrough matrix  $D_{wd}$  is generally zero, since white noise will otherwise feed through, which is not physically realizable since an ideal white noise process has infinite energy.

## A.2 Integrated Modeling

The plant, optics and control loops are modeled in a continuous linear time-invariant model as:

$$\begin{aligned} \dot{q}_p &= A_p q_p + B_{zw} w + B_{zr} r \\ z &= C_p q_p + D_{zw} w + D_{zr} r \end{aligned} \quad (\text{A.12})$$

where  $q_p$  is the appended state vector,  $r$  is the vector of reference inputs (commands) and  $z$  is a vector of performance outputs, from which the actual performance metrics  $J_z$  are formed. If the disturbances are cast in state space form as suggested in (A.11), they can be appended into (A.12). This overall state space system (A.13) then becomes the “integrated model” of the opto-mechanical system representing the closed or open loop dynamics and is shown in Figure A.1. In steady-state the reference commands are often  $r = 0$  and generally the



**Figure A.1:** General block diagram of system dynamics from white noise disturbances  $d$  to system performance signals  $z$ .

feedthrough matrix  $D_{zd}$  is also zero, since otherwise the white-noise input  $d$  will lead to an infinite variance  $\sigma_{z_i}^2$  of the  $i$ -th performance signal  $z_i$  that is affected by the feedthrough.

$$\begin{aligned} \dot{q} &= A_{zd}q + B_{zd}d + B_{zr}r \\ z &= C_{zd}q + D_{zd}d + D_{zr}r \end{aligned} \quad (\text{A.13})$$

In this thesis it is assumed that the feedthrough matrix  $D_{zd}$  is always zero. The vector  $q$  represents the state vector of length  $n_s$  and has to be ordered similar to the convention defined in [45, eq.(4.5)] as  $q = [q_d q_p q_c q_z]^T$ , where  $q_d$  are the disturbance filter states,  $q_p$  are the (structural) plant states,  $q_c$  are the controller states and  $q_z$  are performance weighting states, if applicable.

### A.3 Performance Assessment

Once an integrated model of a nominal system design is available, the next step is to assess the performance when the model is subjected to anticipated disturbances. In this thesis we will consider  $\mathcal{H}_2$  performance metrics according to Zhou [143] as follows:

$$\begin{aligned}
J_z &= \text{E} [z^T z]^{1/2} = \left( \frac{1}{T} \int_0^T z(t)^2 dt \right)^{1/2} \quad \text{RMS} \\
J_z &= \left[ \sum_{i=1}^N \text{E} [z_i^T z_i] \right]^{1/2} = \left( \sum_{i=1}^N \frac{1}{T} \int_0^T z_i(t)^2 dt \right)^{1/2} \quad \text{RSS} \\
J_z &= \left[ \frac{1}{N} \cdot \sum_{i=1}^N \text{E} [z_i^T z_i] \right]^{1/2} = \left( \frac{1}{N} \cdot \sum_{i=1}^N \frac{1}{T} \int_0^T z_i(t)^2 dt \right)^{1/2} \quad \text{RMMS}
\end{aligned} \tag{A.14}$$

The RMS (root-mean-square) metric is typically used to describe the “on-average” optical pathlength difference (OPD) in an interferometer. The RSS (root-sum-squared) can be employed to describe the line-of-sight (LOS) jitter of an observatory in xy-coordinates on the focal plane and, for example, the RMMS (root-mean-mean squared) metric is used to describe the “mean” wavefront error of a light bundle by averaging the phase errors of all rays modeled by an optical ray tracing program. Three disturbance analysis approaches are discussed below.

### A.3.1 Time Domain Analysis

A linear time-invariant system from (colored) disturbances  $w$  to performances  $z$  is given as:

$$\begin{aligned}
\dot{q}_p &= A_{zw} q_p(t) + B_{zw} w(t) \\
z(t) &= C_{zw} q_p(t) + D_{zw} w(t)
\end{aligned} \tag{A.15}$$

where  $q_p$  consists of structural states and compensator states, if applicable. Equivalently, the system can be described in the frequency-domain by the transfer function matrix

$$G_{zw}(\omega) = C_{zw} [j\omega I - A_{zw}]^{-1} B_{zw} + D_{zw} \tag{A.16}$$

The disturbances,  $w$ , are the inputs to the system, while the performances,  $z$ , are the outputs of interest. When measured time histories of the disturbances  $w(t)$  exist<sup>2</sup>, they can be used for time integration of the state space equations (A.15). Once the initial condition on the state vector,  $q_p(0)$ , is specified, numerical integration of (A.15) can then be performed to obtain estimates of the performance time histories  $z(t)$ . The standard difference method technique

---

<sup>2</sup>These can be obtained from spinup tests of reaction wheels, vibration testing of cryocoolers or noise floor measurements on sensors, among others.

approximates the continuous first-order equation (A.15) with a difference equation such as

$$\begin{aligned}\frac{(q_p)_{n+1} - (q_p)_n}{\Delta t} &= A_{zw}(q_p)_n + B_{zw}w_n \\ z_n &= C_{zw}(q_p)_n + D_{zw}w_n\end{aligned}\tag{A.17}$$

The state vector  $(q_p)_{n+1}$  at the  $n + 1$ -th time step can be found by the forward Euler method as

$$(q_p)_{n+1} = [\Delta t A_{zw} + I](q_p)_n + \Delta t B_{zw}w_n\tag{A.18}$$

This integration method is simple but can diverge easily when  $\Delta t \geq \Delta t_{\text{crit}}$ . For time integrations in this thesis it was found that the results obtained with the `ode45.m` solver according to Dormand and Prince [26] gave the best results. An advantage of the time-domain disturbance analysis is that transient effects can be observed, the compliance with time-domain performance specifications can be examined and threshold crossings can be determined from the time simulation. Generally the time domain analysis is computationally more expensive than the other methods. Other disadvantages are the dependency of the answers on the initial conditions of the state vector  $q_p(0)$  and on the seed used for the white noise random number generators.

### A.3.2 Frequency Domain Analysis

For linear systems in the time domain, the output can be expressed as a convolution of the input with the impulse-response function of the system. In the frequency domain (i.e. Laplace domain), the output is equal to the input multiplied by the transfer function (matrix). The disturbance spectral density matrix  $S_{ww}(\omega)$  can be measured experimentally or obtained from a shaping (pre-whitening) filter as  $S_{ww}(\omega) = G_d(\omega) G_d^H(\omega)$ . The performance spectral density matrix  $S_{zz}$  can be obtained from [141]

$$S_{zz}(\omega) = G_{zw}(\omega) S_{ww}(\omega) G_{zw}^H(\omega)\tag{A.19}$$

where  $S_{ww}$  is the disturbance spectral density matrix discussed above and  $G_{zw}$  is the open or closed loop plant transfer function matrix from (A.16).  $S_{ww}(\omega)$  can be a continuous function of frequency, or for the case of disturbances consisting of a series of discrete harmonics, it can contain impulses at the frequencies of the harmonics. In the latter case,  $S_{zz}(\omega)$  will also contain a series of impulses.  $S_{zz}(\omega)$  provides information on the frequency content of the

performances. The covariance matrix of the performances  $\Sigma_z$  (for zero-mean processes) is obtained as

$$\Sigma_z = \frac{1}{2\pi} \int_{-\infty}^{+\infty} S_{zz}(\omega) d\omega = \int_{-\infty}^{+\infty} S_{zz}(f) df \quad (\text{A.20})$$

The variance of the  $i$ -th performance is therefore given by

$$\begin{aligned} \sigma_{z_i}^2 &= [\Sigma_z]_{i,i} = \frac{1}{2\pi} \int_{-\infty}^{+\infty} [S_{zz}(\omega)]_{i,i} d\omega = \\ &= \int_{-\infty}^{+\infty} [S_{zz}(f)]_{i,i} df = 2 \int_0^{+\infty} [S_{zz}(f)]_{i,i} df \end{aligned} \quad (\text{A.21})$$

Taking the square root of  $\sigma_{z_i}^2$  produces the root-mean-square (RMS) value. It is important to specify whether a PSD is one or two sided and given in Hz or rad/sec [141]. In practice the upper and lower frequency integration limits are  $f_{min}$  and  $f_{max}$ , respectively.

$$\sigma_{z_i}^2 \approx 2 \int_{f_{min}}^{f_{max}} [S_{zz}(f)]_{i,i} df \quad (\text{A.22})$$

It is important to ensure that the frequency range that contributes most to the RMS value is sufficiently captured within these limits. One way to verify this is by computing the cumulative RMS function  $\sigma_{z_i,c}(f_o)$  as

$$\sigma_{z_i,c}(f_o) = \left[ 2 \int_{f_{min}}^{f_o} [S_{zz}(f)]_{i,i} df \right]^{\frac{1}{2}} \quad (\text{A.23})$$

where  $f_o \in [f_{min} \dots f_{max}]$ . If most of the energy lies in this range, then  $\sigma_{z_i,c}(f_{max})$  should be very close to the true value of  $\sigma_{z_i}$ . Generally, the frequency-domain approach is more efficient than a time-domain analysis. The method however requires high frequency resolution near lightly damped modes in order to arrive at correct RMS values. Also the frequency domain method is not well suited to assess the transient performance of a linear time-invariant system.

### A.3.3 Lyapunov Analysis

The third type of disturbance analysis can be conducted if the disturbances  $w$  are modeled as the outputs of a shaping filter in the form of equation (A.11). In order to keep the disturbance  $w$  from having infinite energy, there should be no feedthrough matrix  $D_{wd}$ . The state space system (A.11) containing the disturbance dynamics is appended to the plant model (A.12) resulting in the overall system model (A.13). If the system is asymptotically stable, the state covariance matrix obeys the Lyapunov equation [35].

$$A_{zd}\Sigma_q + \Sigma_q A_{zd}^T + B_{zd}B_{zd}^T = \dot{\Sigma}_q \quad (\text{A.24})$$

In order to do time integration of the above dynamics, the initial state covariance,  $\Sigma_{q_0}$ , would have to be specified. Since the white noise disturbance processes  $d$  are assumed to be stationary and the reference commands  $r$  have been set to zero, the statistics of the state vector are also stationary and  $\dot{\Sigma}_q = 0$ . This is not true for transient processes. One may then solve the steady-state Lyapunov equation of order  $n_s$  for the state covariance matrix  $\Sigma_q$  of the system (A.13).

$$A_{zd}\Sigma_q + \Sigma_q A_{zd}^T + B_{zd}B_{zd}^T = 0 \quad (\text{A.25})$$

For  $i = 1, 2, \dots, n_z$  one solves for each RMS value by extracting the  $i$ -th row from the  $C_{zd}$  matrix, pre- and post-multiplying  $\Sigma_q$  and by taking the square root. The RMS of the  $i$ -th performance metric is then given as:

$$\sigma_{z_i} = \left( C_{zd,i}\Sigma_q C_{zd,i}^T \right)^{1/2} \quad (\text{A.26})$$

where  $C_{zd,i}$  is the vector formed by the  $i$ -th row of the  $C_{zd}$  matrix and  $\Sigma_q$  is the state covariance matrix of the state space system that obeys the steady-state Lyapunov equation in (A.25). Alternatively one can pre- and post-multiply with the entire  $C_{zd}$  matrix to obtain the performance covariance matrix  $\Sigma_z$ .

$$\Sigma_z = E [zz^T] = E [C_{zd}qq^T C_{zd}^T] = C_{zd}E [qq^T] C_{zd}^T = C_{zd}\Sigma_q C_{zd}^T \quad (\text{A.27})$$

The variances of the individual performances (RMS squared) are then contained on the diagonal of  $\Sigma_z$ , where  $\Sigma_z$  is of the form

$$\Sigma_z = \begin{bmatrix} \sigma_{z_1}^2 & \sigma_{z_1 z_2} & \cdots & \sigma_{z_1 z_n} \\ \sigma_{z_2 z_1} & \sigma_{z_2}^2 & \cdots & \sigma_{z_2 z_n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{z_n z_1} & \sigma_{z_n z_2} & \cdots & \sigma_{z_n}^2 \end{bmatrix} \quad (\text{A.28})$$

and  $n$  is the total number of elements in the performance vector  $z$ . Thus, the Lyapunov method provides a relatively direct way of arriving at the RMS estimates (in the sense of statistical steady state) by solving one matrix equation (A.25) and computing a matrix triple product (A.27).

One problem is that the computational cost of solving (A.25) increases as  $\sim n_s^3$ , where  $n_s$  is the number of states. Chapter 5 addresses potential solutions for large order systems, which arise from modeling of complex systems such as precision opto-mechanical systems. It is true that often only a subset of the states in  $q$  will contribute significantly to the large entries in



the state covariance matrix  $\Sigma_q$ . If the performance metric of interest is a weighted sum of the performance outputs  $z$  and the  $z_i$  in (A.28) represent these weighted outputs, then the scalar performance cost  $J_z$  can be defined as

$$J_z = \text{trace} [\Sigma_z] = \sigma_{z_1}^2 + \sigma_{z_2}^2 + \dots + \sigma_{z_n}^2 \quad (\text{A.29})$$

The percent contribution of the  $i$ -th weighted output to the overall performance cost  $J_z$  is then

$$\% \text{ contribution} = \frac{\sigma_{z_i}^2}{J_z} \times 100\% \quad (\text{A.30})$$

The main drawback of the Lyapunov approach is that it does not provide insight into the frequency content of the outputs. The main advantage is that the answers provided are immune to the frequency resolution and time step,  $\Delta t$ , problems associated with the frequency-domain and time-domain approaches, respectively. Also, analytical sensitivities can be calculated as presented in the next section.

## A.4 Sensitivity Analysis

Determining the sensitivity of certain outputs (e.g. performance metrics) of a system with respect to model parameters can provide useful information. This information can be used for model-to-test-data correlation, performance enhancement, uncertainty analysis and - the main topic of this thesis - isoperformance analysis. Gutierrez [45] proposes a Lagrangian approach for obtaining the sensitivities  $\partial\sigma_z/\partial p$  or  $\partial J_z/\partial p$  based on earlier work by Jacques [56]. Note that  $p$  can be a vector of modal or physical parameters of the system. The first step, for each performance metric  $z_i$ ,  $i = 1, 2, \dots, n_z$ , is to solve for the corresponding Lagrange multiplier matrix  $L_i$ . Again a steady state Lyapunov equation of order  $n_s$  has to be solved.

$$L_i A_{zd} + A_{zd}^T L_i + C_{zd,i}^T C_{zd,i} = 0 \quad (\text{A.31})$$

Next the governing sensitivity equation (GSE) is solved by substituting the results from Equations (A.25) and (A.31). Additionally the matrix derivatives with respect to the parameters of interest  $p_j$ ,  $j = 1, 2, \dots, n_p$ , need to be computed. The result is the partial derivative of the variance of the  $i$ -th performance  $z_i$  with respect to the  $j$ -th parameter  $p_j$ .

$$\frac{\partial\sigma_{z_i}^2}{\partial p_j} = \text{trace} \left[ \Sigma_q \frac{\partial (C_{zd,i}^T C_{zd,i})}{\partial p_j} \right] + \text{trace} \left[ L_i \left\{ \frac{\partial A_{zd}}{\partial p_j} \Sigma_q + \Sigma_q \frac{\partial A_{zd}^T}{\partial p_j} + \frac{\partial (B_{zd} B_{zd}^T)}{\partial p_j} \right\} \right] \quad (\text{A.32})$$

Here  $L_i$  is the Lagrange multiplier matrix for the  $i$ -th performance. The Lagrange multiplier matrix obeys the dual Lyapunov equation given in (A.31). Consult References [45] and [56] for a detailed derivation of the GSE. The above equation gives the sensitivity of the variance  $\sigma_{z_i}^2$ , but usually the sensitivity with respect to the RMS is desired. The results from (A.26) and (A.32) are substituted in

$$\frac{\partial \sigma_{z_i}}{\partial p_j} = \frac{1}{2\sigma_{z_i}} \cdot \frac{\partial \sigma_{z_i}^2}{\partial p_j} \quad (\text{A.33})$$

in order to obtain the desired sensitivity. Normalization with a factor  $p_{j,nom}/\sigma_{z_i,nom}$  is optional. This normalization allows comparing sensitivities with respect to parameters of different units:

$$\frac{p_{nom}}{(\sigma_{z_i})_{nom}} \frac{\partial \sigma_{z_i}}{\partial p} = \frac{\frac{\partial \sigma_{z_i}}{(\sigma_{z_i})_{nom}}}{\frac{\partial p}{p_{nom}}} \approx \frac{\frac{\Delta \sigma_{z_i}}{(\sigma_{z_i})_{nom}}}{\frac{\Delta p}{p_{nom}}} \approx \frac{\% \text{ change in } \sigma_{z_i}}{\% \text{ change in } p} \quad (\text{A.34})$$

The matrix partial derivatives such as  $\partial A_{zd}/\partial p_j$  in (A.32) represent the main difficulty in finding the sensitivities for large order systems. Initial work on sensitivities for modal parameters (frequencies, damping ratio, modal mass) and physical parameter (masses, stiffnesses) is presented by Gutierrez [45]. Progress in this area is reported in Chapter 5.

## A.5 Uncertainty Analysis

Disturbance analyses during early design stages of aerospace systems are typically conducted on simplified models of nominal point designs. The models attempt to capture the behavior of interest, and although they are generally suitable for judging relative merits between competing designs in a trade study, the validity of their use in making absolute performance predictions is not as clear. An uncertainty analysis is one way to account for uncertainties in an immature design model, thus establishing error bounds on the predicted performances.

If it is found that performance requirements are met even under worst-case model uncertainties, then confidence in design margins can be increased. In the isoperformance context an uncertainty analysis is beneficial, since it allows comparing designs which have the same nominal performance, but exhibit different performance sensitivity to parametric uncertainty. Optimization for minimum sensitivity to uncertain parameters has been investigated by Pritchard, Adelman and Sobieszczanski-Sobieski [119]. In the most general sense, a model uncertainty represents an unknown error in a model of a physical system. Model uncertainty

is typically divided into parametric and non-parametric uncertainties. Focusing on parametric uncertainty an estimate of the effect of parametric uncertainty on the system performances can be obtained via the first-order approach. The first-order approach expresses the performance RMS values as a Taylor series approximation about the nominal values.

$$\sigma_z = \sigma_{z,nom} + \frac{\partial \sigma_z}{\partial p} (p - p_{nom}) + \text{H.O.T.} \quad (\text{A.35})$$

where  $\sigma_z$  is a vector containing the RMS values for each performance output  $z$ . The vector  $p$  contains all of the uncertain parameters, and  $\partial \sigma_z / \partial p$  is referred to as the sensitivity matrix (or Jacobian). Neglecting the higher order terms and making the substitutions  $\Delta \sigma_z = \sigma_z - \sigma_{z,nom}$  and  $\Delta p = p - p_{nom}$  results in

$$\Delta \sigma_z \approx \frac{\partial \sigma_z}{\partial p} \Delta p \quad (\text{A.36})$$

The parameter uncertainty is captured by  $\Delta p$ , and it is assumed that  $\Delta p$  is specified based on empirical experience with similar systems or a best guess. For example, if all parameters are uncertain to within  $\pm 1\%$  of the nominal values, then for the  $j$ -th parameter,  $\Delta p_j \in [-0.01 + 0.01] \cdot p_{j,nom}$ . It is desirable to find the worst-case  $\Delta \sigma_z$  for  $\Delta p$  within the specified bounds. When several performance metrics with different units are present, first define a normalization vector  $r$ . For instance,  $r$  could contain the performance RMS requirements as follows

$$r^T = \left[ \frac{1}{\sigma_{z_1,req}} \quad \cdots \quad \frac{1}{\sigma_{z_{n_z},req}} \right] \quad (\text{A.37})$$

where  $n_z$  is the number of performance metrics. The maximum uncertainty ‘‘cost’’  $J_r$  can be defined as

$$J_r = r^T \Delta \sigma_z = \frac{\Delta \sigma_{z_1}}{\sigma_{z_1,req}} + \cdots + \frac{\Delta \sigma_{z_{n_z}}}{\sigma_{z_{n_z},req}} \quad (\text{A.38})$$

Expanding (A.36) yields,

$$\begin{aligned} \Delta \sigma_{z_1} &= \frac{\partial \sigma_{z_1}}{\partial p_1} \Delta p_1 + \frac{\partial \sigma_{z_1}}{\partial p_2} \Delta p_2 + \cdots + \frac{\partial \sigma_{z_1}}{\partial p_{n_p}} \Delta p_{n_p} \\ &\quad \vdots \\ \Delta \sigma_{z_{n_z}} &= \frac{\partial \sigma_{z_{n_z}}}{\partial p_1} \Delta p_1 + \frac{\partial \sigma_{z_{n_z}}}{\partial p_2} \Delta p_2 + \cdots + \frac{\partial \sigma_{z_{n_z}}}{\partial p_{n_p}} \Delta p_{n_p} \end{aligned} \quad (\text{A.39})$$

where  $n_p$  denotes the total number of uncertain parameters. Computing the uncertainty ‘‘cost’’ results in the following expression.

$$\begin{aligned}
J_r &= \underbrace{\left( \frac{1}{\sigma_{z_1, req}} \frac{\partial \sigma_{z_1}}{\partial p_1} + \dots + \frac{1}{\sigma_{z_{n_z}, req}} \frac{\partial \sigma_{z_{n_z}}}{\partial p_1} \right)}_{=a_1} \Delta p_1 + \dots \\
&+ \underbrace{\left( \frac{1}{\sigma_{z_1, req}} \frac{\partial \sigma_{z_1}}{\partial p_{n_p}} + \dots + \frac{1}{\sigma_{z_{n_z}, req}} \frac{\partial \sigma_{z_{n_z}}}{\partial p_{n_p}} \right)}_{=a_{n_p}} \Delta p_{n_p} = a_1 \Delta p_1 + \dots + a_{n_p} \Delta p_{n_p}
\end{aligned} \tag{A.40}$$

The vector  $\Delta p$  that maximizes  $J_r$  is therefore given by

$$\Delta p_j = \begin{cases} \Delta p_{j, LB} , & \text{if } \text{sgn}(a_j) = -1 \\ \Delta p_{j, UB} , & \text{if } \text{sgn}(a_j) = +1 \\ \text{undefined} , & \text{if } a_j = 0 \end{cases} \tag{A.41}$$

This assumes that the  $\Delta p$ 's can occur independently. Substituting the worst-case  $\Delta p$  into (A.36) produces an estimate of the increase in RMS values due to the parametric uncertainties. Note that other approaches such as “bad corner” evaluation, constrained optimization or robust control methods have been suggested [45]. The first order approach is used in Chapters 4 and 7 to distinguish between designs that have the same nominal performance  $J_z$ .

## Appendix B

# Derivations for Large Order Systems

### B.1 Proof of Similarity Transformation

It is true that for a similarity transformation the transfer function matrix  $G_{zd}(s)$  of the original system and the transfer matrix of the balanced system  $\tilde{G}_{zd}(s)$  are equal. This can be proven by showing that the difference between the transfer function matrices is zero.

$$G_{zd}(s) - \tilde{G}_{zd}(s) = C_{zd} [sI - A_{zd}]^{-1} B_{zd} + D_{zd} - C_{zd} T^{-1} [sI - T A_{zd} T^{-1}]^{-1} T B_{zd} + D_{zd} = C_{zd} \left( [sI - A_{zd}]^{-1} - T^{-1} [sI - T A_{zd} T^{-1}]^{-1} T \right) B_{zd} = 0 \quad (\text{B.1})$$

This is true if the inner term in round brackets can be shown to vanish, since we assume that  $C_{zd} \neq 0$  and  $B_{zd} \neq 0$ . Thus we have to show that

$$[sI - A_{zd}]^{-1} = T^{-1} [sI - T A_{zd} T^{-1}]^{-1} T \quad (\text{B.2})$$

We first pre-multiply both sides by  $T$

$$T [sI - A_{zd}]^{-1} = [sI - T A_{zd} T^{-1}]^{-1} T \quad (\text{B.3})$$

and then take the inverse of both sides, which leads to

$$[sI - A_{zd}] T^{-1} = T^{-1} [sI - T A_{zd} T^{-1}] \quad (\text{B.4})$$

Next we post-multiply both sides with  $T$  and show that the right hand side of (B.4) is equal to the left-hand side

$$RHS = T^{-1} [sI - TA_{zd}T^{-1}] T = s \underbrace{T^{-1}IT}_{I} - \underbrace{T^{-1}T}_{I} A_{zd} \underbrace{T^{-1}T}_{I} = sI - A_{zd} = LHS \quad (B.5)$$

q.e.d.

## B.2 Computation of Balancing Transformation

The goal of the internal balancing operation is to find a transformation matrix  $T$ , such that the observability and controllability gramians of the transformed system are diagonal and equal to each other. A number of different algorithms for finding  $T$ , starting with Moore's method [102] in 1981, have been suggested over the years. This method is implemented in the MATLAB function `xobalreal.m`. First the gramians are each computed as the solution of a Lyapunov equation.

Controllability Gramian  $W_c$ :

$$A_{zd}W_c + W_cA_{zd}^T + B_{zd}B_{zd}^T = 0 \quad (B.6)$$

Observability Gramian  $W_o$  :

$$A_{zd}^T W_o + W_o A_{zd} + C_{zd}^T C_{zd} = 0 \quad (B.7)$$

Moore then proposed directly performing a singular value decomposition (SVD) on  $W_c$ :

$$W_c = U_c \Sigma_c U_c^T \quad (B.8)$$

The elements of the diagonal matrix  $\Sigma_c$  are the sorted controllability singular values of the unbalanced system such that:

$$\Sigma_c = \text{diag} \{ \sigma_1^c, \sigma_2^c, \dots, \sigma_N^c \} \quad (B.9)$$

Similarly, a singular value decomposition is used on the matrix product

$$U_c \Sigma_c^{\frac{1}{2}} W_o \Sigma_c^{\frac{1}{2}} U_c^T = U_b \Sigma_H^2 U_b^T \quad (B.10)$$

where the diagonal elements of the diagonal matrix  $\Sigma_H$  are the Hankel singular values of the system and  $\sigma_i^H > \sigma_j^H$  for  $i > j$ . The square transformation matrix  $T$  is then formed as:

$$T = \Sigma_H^{\frac{1}{2}} U_b^T U_c \Sigma_c^{\frac{1}{2}} \quad (\text{B.11})$$

and the inverse can be directly computed as:

$$T^{-1} = \Sigma_c^{-\frac{1}{2}} U_c^T U_b \Sigma_H^{-\frac{1}{2}} \quad (\text{B.12})$$

The second algorithm presented here was developed by Laub and coworkers [77],[78] and it is also the one implemented in the MATLAB function `balreal.m`. Laub (1987) first computes the controllability and observability gramians  $W_o$ ,  $W_c$  according to (B.6) and (B.7) respectively. Next a Cholesky factorization of the gramians is performed such that:

$$W_c = R_c^T R_c \text{ and } W_o = R_o^T R_o \quad (\text{B.13})$$

The next step consists in performing a singular value decomposition of the mixed product of the Cholesky factors from (B.13) such that:

$$U \Sigma_H V^T = R_o R_c^T \quad (\text{B.14})$$

Of these matrices the matrix  $\Sigma_H$  is of importance, since it contains the Hankel singular values on the diagonal.

$$\Sigma_H = \text{diag} (\sigma_1^H, \sigma_2^H, \dots, \sigma_{n_s}^H) \quad (\text{B.15})$$

The resulting singular values, eigenvector matrices  $U$  and  $V$  as well as the Cholesky factors are used to determine  $T$  and its inverse  $T^{-1}$  as follows:

$$T = \Sigma_H^{-1/2} U^T R_o \text{ and } T^{-1} = R_c^T V \Sigma_H^{-1/2} \quad (\text{B.16})$$

The balancing matrix  $T$  can then be used to obtain the transformed gramians.

$$\tilde{W}_c = T W_c T^T \text{ and } \tilde{W}_o = (T^{-1})^T W_o T^{-1} \quad (\text{B.17})$$

It can be shown that the transformed gramians are equal to each other and equal to the Hankel singular value matrix.

$$\tilde{W}_c = \tilde{W}_o = \Sigma_H \text{ and thus } \Sigma_H^2 = T W_c W_o T^{-1} \quad (\text{B.18})$$

The original state space system can be written in terms of the  $S$  matrix, which is transformed into an internally balanced realization  $\tilde{S}$  :

$$S = \begin{bmatrix} A_{zd} & B_{zd} \\ C_{zd} & D_{zd} \end{bmatrix} \Rightarrow \tilde{S} = \begin{bmatrix} TA_{zd}T^{-1} & TB_{zd} \\ C_{zd}T^{-1} & D_{zd} \end{bmatrix} \quad (\text{B.19})$$

It is true that the matrices  $R_o, R_c, \Sigma_H, U, V$  all depend on the system parameters  $p_j$  and so does  $T$ . This knowledge is however not necessary in order to compute the performance RMS and sensitivity of the system as will be shown later. A prerequisite however is that the system be stable and minimal. Especially the last requirement is often not met for large order systems without prior model conditioning (e.g. pre-balancing according to [85]). Further details are available from Mallory [86] and Uebelhart [136].

*Benefits of Balancing:* A remarkable and desirable effect of the balancing transformation is the reduction of the condition number of the A-matrix. The condition number is the ratio of the largest to the smallest singular value of A. For the SDOF example in (5.29) the condition number is reduced from 1714 for the original system (left) to 1.001 for the balanced system (right). Also balancing is often used as a precursor operation for model reduction, since it ranks the states in descending order of disturbability/performability.

*Drawbacks of Balancing:* Since the balanced matrices are fully populated the parameters of interest do not appear explicitly anymore. This complicates the computation of the matrix derivatives in the sensitivity analysis, see Appendix A. The original realization can however be recovered by inverse transformation. For systems that are uncontrollable or unobservable the traditional computation of  $T$  fails. Also internal balancing is a computationally expensive and its computational cost was estimated to be roughly  $150 \cdot n_s^3$  floating point operations.

### B.3 Derivation of Transformed Governing Sensitivity Equation (TGSE)

The purpose of this section is to derive the governing sensitivity equation for a transformed system. This is done assuming that the similarity transformation is internal balancing. At no point is any property of balancing used, except for similarity, such that the results are valid for any similarity transformation, e.g. into Jordan form, 2nd order modal form, real modal



form, controller or observer realization etc. The first step is to recognize that the sensitivity of a balanced system is given as:

$$\frac{\partial \tilde{\sigma}_{z_i}}{\partial p_j} = \frac{1}{2\tilde{\sigma}_{z_i}} \cdot \frac{\partial \tilde{\sigma}_{z_i}^2}{\partial p_j} \quad (\text{B.20})$$

where the RMS of the balanced system  $\tilde{\sigma}_{z_i}$  is computed using equation (5.30). The derivative of the variance  $\tilde{\sigma}_{z_i}^2$  with respect to parameter  $p_j$  can be computed as follows. The first step is to substitute the transformed system matrices  $\tilde{A}_{zd}$ ,  $\tilde{B}_{zd}$  and  $\tilde{C}_{zd}$  into the governing sensitivity equation (A.32):

$$\frac{\partial \tilde{\sigma}_{z_i}^2}{\partial p_j} = \text{trace} \left[ \Sigma_{\tilde{q}} \frac{\partial \left( \tilde{C}_{zd,i}^T \tilde{C}_{zd,i} \right)}{\partial p_j} \right] + \text{trace} \left[ \tilde{L}_i \left\{ \frac{\partial \tilde{A}_{zd}}{\partial p_j} \Sigma_{\tilde{q}} + \Sigma_{\tilde{q}} \frac{\partial \tilde{A}_{zd}^T}{\partial p_j} + \frac{\partial \left( \tilde{B}_{zd} \tilde{B}_{zd}^T \right)}{\partial p_j} \right\} \right] \quad (\text{B.21})$$

whereby  $\Sigma_{\tilde{q}}$  is the balanced state covariance matrix and  $\tilde{L}_i$  is the balanced Lagrange multiplier matrix. These matrices are computed solving the Lyapunov equations (5.33) and (5.40), respectively. The difficulty lies in computing the partial derivatives of the system matrices  $\tilde{A}_{zd} = T A_{zd} T^{-1}$ ,  $\tilde{B}_{zd} = T B_{zd}$  and  $\tilde{C}_{zd,i} = C_{zd,i} T^{-1}$  with respect to the  $j$ -th parameter  $p_j$ . It shall be noted that  $p_j$  could be a modal or physical parameter of the system. The governing sensitivity equation can be rewritten in terms of the original system matrices and the balancing transformation matrix  $T$ :

$$\begin{aligned} \frac{\partial \tilde{\sigma}_{z_i}^2}{\partial p_j} = & \text{trace} \left[ \Sigma_{\tilde{q}} \frac{\partial \left( (T^{-1})^T C_{zd,i}^T C_{zd,i} T^{-1} \right)}{\partial p_j} \right] \\ & + \text{trace} \left[ \tilde{L}_i \left\{ \frac{\partial (T A_{zd} T^{-1})}{\partial p_j} \Sigma_{\tilde{q}} + \Sigma_{\tilde{q}} \frac{\partial \left( (T^{-1})^T A_{zd}^T T^T \right)}{\partial p_j} + \frac{\partial (T B_{zd} B_{zd}^T T^T)}{\partial p_j} \right\} \right] \end{aligned} \quad (\text{B.22})$$

Invoking the product rule of matrix calculus [143, p.24], which states that the derivative of the product of two matrices  $A(\alpha)$ ,  $B(\alpha)$  is given as:

$$\frac{d(AB)}{d\alpha} = \frac{dA}{d\alpha} B + A \frac{dB}{d\alpha} \quad (\text{B.23})$$

we can rewrite the above equation to yield

$$\begin{aligned}
\frac{\partial \tilde{\sigma}_{z_i}^2}{\partial p_j} = & \underbrace{\text{trace} \left[ \Sigma_{\tilde{q}} \frac{\partial (T^{-1})^T}{\partial p_j} C_{z_d,i}^T C_{z_d,i} T^{-1} \right]}_{\text{term1}} + \underbrace{\text{trace} \left[ \Sigma_{\tilde{q}} (T^{-1})^T \frac{\partial (C_{z_d,i}^T C_{z_d,i})}{\partial p_j} T^{-1} \right]}_{\text{term2}} + \\
& \underbrace{\text{trace} \left[ \Sigma_{\tilde{q}} (T^{-1})^T C_{z_d,i}^T C_{z_d,i} \frac{\partial T^{-1}}{\partial p_j} \right]}_{\text{term3}} + \underbrace{\text{trace} \left[ \tilde{L}_i \frac{\partial T}{\partial p_j} A_{z_d} T^{-1} \Sigma_{\tilde{q}} \right]}_{\text{term4}} + \\
& \underbrace{\text{trace} \left[ \tilde{L}_i T \frac{\partial A_{z_d}}{\partial p_j} T^{-1} \Sigma_{\tilde{q}} \right]}_{\text{term5}} + \underbrace{\text{trace} \left[ \tilde{L}_i T A_{z_d} \frac{\partial T^{-1}}{\partial p_j} \Sigma_{\tilde{q}} \right]}_{\text{term6}} + \underbrace{\text{trace} \left[ \tilde{L}_i \Sigma_{\tilde{q}} \frac{\partial (T^{-1})^T}{\partial p_j} A_{z_d}^T T^T \right]}_{\text{term7}} + \\
& \underbrace{\text{trace} \left[ \tilde{L}_i \Sigma_{\tilde{q}} (T^{-1})^T \frac{\partial A_{z_d}^T}{\partial p_j} T^T \right]}_{\text{term8}} + \underbrace{\text{trace} \left[ \tilde{L}_i \Sigma_{\tilde{q}} (T^{-1})^T A_{z_d}^T \frac{\partial T^T}{\partial p_j} \right]}_{\text{term9}} + \\
& \underbrace{\text{trace} \left[ \tilde{L}_i \frac{\partial T}{\partial p_j} B_{z_d} B_{z_d}^T T^T \right]}_{\text{term10}} + \underbrace{\text{trace} \left[ \tilde{L}_i T \frac{\partial (B_{z_d} B_{z_d}^T)}{\partial p_j} T^T \right]}_{\text{term11}} + \underbrace{\text{trace} \left[ \tilde{L}_i T B_{z_d} B_{z_d}^T \frac{\partial T^T}{\partial p_j} \right]}_{\text{term12}}
\end{aligned} \tag{B.24}$$

The above expression makes use of the fact that the trace of the sum of two matrices is equal to the sum of the traces of each individual matrix, i.e.

$$\text{trace} [A + B] = \text{trace} [A] + \text{trace} [B] \tag{B.25}$$

Then equation (B.24) can be reordered such that terms 2, 5, 8 and 11 appear first. We recognize that these terms correspond exactly to the governing sensitivity equation (A.32). The sum of the remaining terms are called residual. In order to prove that (A.32) is correct we must then show that the residual always vanishes. We rewrite (B.24) as

$$\begin{aligned}
\frac{\partial \tilde{\sigma}_{z_i}^2}{\partial p_j} = & \underbrace{\text{trace} \left[ \Sigma_{\tilde{q}} (T^{-1})^T \frac{\partial (C_{z_d,i}^T C_{z_d,i})}{\partial p_j} T^{-1} \right]}_{\text{term2}} + \\
& \underbrace{\text{trace} \left[ \tilde{L}_i \left\{ T \frac{\partial A_{z_d}}{\partial p_j} T^{-1} \Sigma_{\tilde{q}} + \Sigma_{\tilde{q}} (T^{-1})^T \frac{\partial A_{z_d}^T}{\partial p_j} T^T + T \frac{\partial (B_{z_d} B_{z_d}^T)}{\partial p_j} T^T \right\} \right]}_{\text{terms5+8+11}} + \text{residual}
\end{aligned} \tag{B.26}$$

The residual can be written as:

$$\begin{aligned}
\text{residual} = & \underbrace{\text{trace} \left[ \Sigma_{\tilde{q}} \frac{\partial (T^{-1})^T}{\partial p_j} C_{zd,i}^T C_{zd,i} T^{-1} + \Sigma_{\tilde{q}} (T^{-1})^T C_{zd,i}^T C_{zd,i} \frac{\partial T^{-1}}{\partial p_j} \right]}_{\text{terms1+3}} + \\
& \underbrace{\text{trace} \left[ \tilde{L}_i \frac{\partial T}{\partial p_j} A_{zd} T^{-1} \Sigma_{\tilde{q}} + \tilde{L}_i T A_{zd} \frac{\partial T^{-1}}{\partial p_j} \Sigma_{\tilde{q}} + \tilde{L}_i \frac{\partial T}{\partial p_j} B_{zd} B_{zd}^T T^T \right]}_{\text{term4+6+10}} + \\
& \underbrace{\text{trace} \left[ \tilde{L}_i \Sigma_{\tilde{q}} (T^{-1})^T A_{zd}^T \frac{\partial T^T}{\partial p_j} + \tilde{L}_i \Sigma_{\tilde{q}} \frac{\partial (T^{-1})^T}{\partial p_j} A_{zd}^T T^T + \tilde{L}_i T B_{zd} B_{zd}^T \frac{\partial T^T}{\partial p_j} \right]}_{\text{term9+7+12}}
\end{aligned} \tag{B.27}$$

At this point we make use of two important properties of the trace operator. First it is true that the trace of the transpose of the product of two matrices is equal to the trace of the product of the two matrices:

$$\text{trace} [AB^T] = \text{trace} [(AB^T)^T] = \text{trace} [BA^T] \tag{B.28}$$

This is obviously true, since the trace is the sum of the diagonal elements, which are not affected by the transpose operator. Secondly it is true that we can cycle the matrices inside the trace operator, without affecting the result, provided that the matrices are square, which they are in equation (B.27).

$$\text{trace} [ABC] = \text{trace} [CAB] = \text{trace} [BCA] \tag{B.29}$$

Invoking (B.28) for terms 3,9,7 and 12 and reordering according to (B.29), we realize that terms 1 and 3, 4 and 9, 6 and 7 as well as 10 and 12 are equal to each other. This will be illustrated for terms 4 and 9 and applied to the other terms.

$$\begin{aligned}
& \underbrace{\text{trace} \left[ \tilde{L}_i \Sigma_{\tilde{q}} (T^{-1})^T A_{zd}^T \frac{\partial T^T}{\partial p_j} \right]}_{\text{term9}} = \text{trace} \left[ \left( \tilde{L}_i \Sigma_{\tilde{q}} (T^{-1})^T A_{zd}^T \frac{\partial T^T}{\partial p_j} \right)^T \right] = \\
& \text{trace} \left[ \left( \frac{\partial T^T}{\partial p_j} \right)^T A_{zd} T^{-1} \Sigma_{\tilde{q}}^T \tilde{L}_i^T \right] = \text{trace} \left[ \frac{\partial T}{\partial p_j} A_{zd} T^{-1} \Sigma_{\tilde{q}} \tilde{L}_i \right] = \\
& \text{trace} \left[ \tilde{L}_i \frac{\partial T}{\partial p_j} A_{zd} T^{-1} \Sigma_{\tilde{q}} \right] = \underbrace{\text{trace} \left[ \tilde{L}_i \frac{\partial T}{\partial p_j} A_{zd} T^{-1} \Sigma_{\tilde{q}} \right]}_{\text{term4}}
\end{aligned} \tag{B.30}$$

Note that we have made use of some additional facts in order to arrive at equation (B.30). The state covariance matrix and the Lagrange multiplier matrix are each symmetric, since they are the solution of a steady state Lyapunov equation, i.e.

$$\Sigma_{\tilde{q}} = \Sigma_{\tilde{q}}^T \text{ and } \tilde{L}_i = \tilde{L}_i^T \quad (\text{B.31})$$

This has been used to arrive from the third to the fourth step in equation (B.30). Additionally we know from matrix calculus that the transpose operator and the (partial) derivative of a matrix are interchangeable, i.e.

$$\left( \frac{dA}{d\alpha} \right)^T = \left[ \frac{da_{ij}}{d\alpha} \right]^T = \left[ \frac{da_{ji}}{d\alpha} \right] = \frac{d(A^T)}{d\alpha} \quad (\text{B.32})$$

This has been used on the balancing transformation matrix to arrive from step three to step four in equation (B.30). Applying the same operations as shown above to terms 1, 7 and 12 we have only four distinct terms. Thus we can factor out a factor 2 and obtain:

$$\begin{aligned} \text{residual} = & 2 \cdot \text{trace} \left[ \Sigma_{\tilde{q}} (T^{-1})^T C_{zd,i}^T C_{zd,i} \frac{\partial T^{-1}}{\partial p_j} \right] + \\ & 2 \cdot \text{trace} \left[ \tilde{L}_i \frac{\partial T}{\partial p_j} A_{zd} T^{-1} \Sigma_{\tilde{q}} + \tilde{L}_i T A_{zd} \frac{\partial T^{-1}}{\partial p_j} \Sigma_{\tilde{q}} + \tilde{L}_i \frac{\partial T}{\partial p_j} B_{zd} B_{zd}^T T^T \right] \end{aligned} \quad (\text{B.33})$$

The difficulty with equation (B.33) is that there are mixed terms with partial derivatives of the transformation matrix  $T$  and its inverse. This can be resolved by recalling from matrix calculus that

$$\frac{dA^{-1}}{d\alpha} = -A^{-1} \frac{dA}{d\alpha} A^{-1} \quad (\text{B.34})$$

Thus equation (B.33) is rewritten as:

$$\begin{aligned} \text{residual} = & 2 \cdot \text{trace} \left[ -\Sigma_{\tilde{q}} (T^{-1})^T C_{zd,i}^T C_{zd,i} T^{-1} \frac{\partial T}{\partial p_j} T^{-1} \right] + \\ & 2 \cdot \text{trace} \left[ \tilde{L}_i \frac{\partial T}{\partial p_j} A_{zd} T^{-1} \Sigma_{\tilde{q}} - \tilde{L}_i T A_{zd} T^{-1} \frac{\partial T}{\partial p_j} T^{-1} \Sigma_{\tilde{q}} + \tilde{L}_i \frac{\partial T}{\partial p_j} B_{zd} B_{zd}^T T^T \right] \end{aligned} \quad (\text{B.35})$$

Next we can factor out the derivative term of  $T$  and the inverse of  $T$  and rearrange the remaining terms.

residual =

$$2 \cdot \text{trace} \left[ \frac{\partial T}{\partial p_j} T^{-1} \left( -\Sigma_{\tilde{q}} (T^{-1})^T C_{z_d,i}^T C_{z_d,i} T^{-1} + T A_{z_d} T^{-1} \Sigma_{\tilde{q}} \tilde{L}_i - \Sigma_{\tilde{q}} \tilde{L}_i T A_{z_d} T^{-1} + T B_{z_d} B_{z_d}^T T^T \tilde{L}_i \right) \right] \quad (\text{B.36})$$

We know that the state covariance matrix  $\Sigma_{\tilde{q}}$  obeys the Lyapunov Equation (5.34), from which we can isolate the term containing the  $B_{z_d}$  matrix, which results in

$$T B_{z_d} B_{z_d}^T T^T = -T A_{z_d} T^{-1} \Sigma_{\tilde{q}} - \Sigma_{\tilde{q}} (T^{-1})^T A_{z_d}^T T^T \quad (\text{B.37})$$

This expression (B.37) is then substituted into (B.36), which cancels the second term in the round brackets of (B.36). An additional term  $-\Sigma_{\tilde{q}}$  can then be factored out which leads to

$$\text{residual} = 2 \cdot \text{trace} \left[ -\frac{\partial T}{\partial p_j} T^{-1} \Sigma_{\tilde{q}} \left( (T^{-1})^T C_{z_d,i}^T C_{z_d,i} T^{-1} + (T^{-1})^T A_{z_d}^T T^T \tilde{L}_i + \tilde{L}_i T A_{z_d} T^{-1} \right) \right] \quad (\text{B.38})$$

We recognize that the matrices inside the round brackets are the transformed system matrices and that their sum contains all the elements of the left side of the Lyapunov Equation (5.40). After rewriting we obtain:

$$\text{residual} = 2 \cdot \text{trace} \left[ -\frac{\partial T}{\partial p_j} T^{-1} \Sigma_{\tilde{q}} \underbrace{\left( \tilde{A}_{z_d}^T \tilde{L}_i + \tilde{L}_i \tilde{A}_{z_d} + \tilde{C}_{z_d,i}^T \tilde{C}_{z_d,i} \right)}_0 \right] = 0 \quad (\text{B.39})$$

Thus the residual is zero due to the fact that the term in round brackets obeys the Lyapunov equation for the Lagrange multiplier matrix  $\tilde{L}_i$  and is always equal to zero. Hence it does not matter that the partial derivative of the transformation matrix  $T$  with respect to parameter  $p_j$  is non-zero, since it does not enter into the final expression for the sensitivity of an internally balanced system. We have thus shown that the expression for the sensitivity of a balanced system according to Equation (5.39) is correct. It shall also be noted that at no point in this derivation was it necessary to stipulate that  $T$  is an internal balancing transformation. Equation (5.39) is thus valid for other kinds of similarity transformations such as input or output normalization [102].

**Note:** A simpler proof can be carried out by invoking the similarity transformation property. By substituting the inverse transformation of the balanced state covariance matrix (5.32) into the original GSE (A.32), the TGSE (5.39) can be obtained directly.

## B.4 Computation of Sensitivity Error Bound due to Model Reduction

The goal of this appendix is to derive the error in the RMS sensitivity prediction  $\partial\bar{\sigma}_{z_i}/\partial p_j$  due to balanced model reduction. The RMS value for the  $i$ -th performance metric  $z_i$  of a balanced, reduced system is computed as:

$$\bar{\sigma}_{z_i} = (\bar{C}_{z_d,i}\Sigma_{\bar{q}}\bar{C}_{z_d,i}^T)^{1/2} = \left(C_{z_d,i}T^{-1}P^T\Sigma_{\bar{q}}P(T^{-1})^T C_{z_d,i}^T\right)^{1/2} \quad (\text{B.40})$$

The sensitivity of a reduced, internally balanced state space system is given as:

$$\frac{\partial\bar{\sigma}_{z_i}}{\partial p_j} = \frac{1}{2\bar{\sigma}_{z_i}} \cdot \frac{\partial\bar{\sigma}_{z_i}^2}{\partial p_j} \quad (\text{B.41})$$

Here the partial derivative with respect to the variance is obtained as:

$$\begin{aligned} \frac{\partial\bar{\sigma}_{z_i}^2}{\partial p_j} = & \text{trace} \left[ T^{-1}P^T\Sigma_{\bar{q}}P(T^{-1})^T \frac{\partial \left( C_{z_d,i}^T C_{z_d,i} \right)}{\partial p_j} \right] + \\ & \text{trace} \left[ \bar{L}_i \left\{ PT \frac{\partial A_{z_d}}{\partial p_j} T^{-1} P^T \Sigma_{\bar{q}} + \Sigma_{\bar{q}} P (T^{-1})^T \frac{\partial A_{z_d}^T}{\partial p_j} T^T P^T + PT \frac{\partial (B_{z_d} B_{z_d}^T)}{\partial p_j} T^T P^T \right\} \right] \end{aligned} \quad (\text{B.42})$$

The Lagrange multiplier matrix for the reduced system obeys:

$$\bar{A}_{z_d}^T \bar{L}_i + \bar{L}_i \bar{A}_{z_d} + \bar{C}_{z_d,i}^T \bar{C}_{z_d,i} = 0 \quad (\text{B.43})$$

As was the case for the computation of the reduced system RMS performance  $\bar{\sigma}_{z_i}$ , we can expect that the reduced system sensitivity  $\partial\bar{\sigma}_{z_i}/\partial p_j$  is in error due to the removal of system states. The quantity we are trying to determine is:

$$\varepsilon_{RMS}^{sens} = \left| \frac{\partial\bar{\sigma}_{z_i}}{\partial p_j} - \frac{\partial\bar{\sigma}_{z_i}}{\partial p_j} \right| = \left| \frac{1}{2\bar{\sigma}_{z_i}} \cdot \frac{\partial\bar{\sigma}_{z_i}^2}{\partial p_j} - \frac{1}{2\bar{\sigma}_{z_i}} \cdot \frac{\partial\bar{\sigma}_{z_i}^2}{\partial p_j} \right| \quad (\text{B.44})$$

With the following substitutions we can derive an expression for  $\varepsilon_{RMS}^{sens}$  more easily:

$$a = \frac{\partial \bar{\sigma}_{z_i}^2}{\partial p_j}, \quad b = \frac{\partial \tilde{\sigma}_{z_i}^2}{\partial p_j}, \quad c = \bar{\sigma}_{z_i}, \quad d = \tilde{\sigma}_{z_i} = \bar{\sigma}_{z_i} + \Delta \sigma_{z_i} = c + \Delta c \quad (\text{B.45})$$

We can then rewrite (B.44) as:

$$\begin{aligned} \varepsilon_{RMS}^{sens} &= \left| \frac{1}{2d}b - \frac{1}{2c}a \right| = \left| \frac{1}{2(c + \Delta c)}b - \frac{1}{2c}a \right| = \\ &= \left| \frac{c \cdot b - (c + \Delta c)a}{2(c + \Delta c)c} \right| = \left| \frac{c(b - a) - \Delta c \cdot a}{2(c^2 + \Delta c \cdot c)} \right| \end{aligned} \quad (\text{B.46})$$

Substituting in the original variables we obtain an expression for the absolute error on the sensitivity  $\partial \bar{\sigma}_{z_i} / \partial p_j$  due to balanced model reduction.

$$\varepsilon_{RMS}^{sens} = \left| \frac{\partial \bar{\sigma}_{z_i}}{\partial p_j} - \frac{\partial \tilde{\sigma}_{z_i}}{\partial p_j} \right| = \left| \frac{\bar{\sigma}_{z_i} \left( \frac{\partial \tilde{\sigma}_{z_i}^2}{\partial p_j} - \frac{\partial \bar{\sigma}_{z_i}^2}{\partial p_j} \right) - \Delta \sigma_{z_i} \cdot \frac{\partial \bar{\sigma}_{z_i}^2}{\partial p_j}}{2(\bar{\sigma}_{z_i}^2 + \Delta \sigma_{z_i} \cdot \bar{\sigma}_{z_i})} \right| \quad (\text{B.47})$$

Of the quantities in equation (B.47), we can obtain  $\bar{\sigma}_{z_i}$  from equation (5.47), the RMS error  $\Delta \sigma_{z_i}$  from equation (5.62) and the partial derivative of the variance  $\partial \bar{\sigma}_{z_i}^2 / \partial p_j$  from equation (B.42). An expression for the difference of partial derivatives of the variances is derived below.

$$\begin{aligned} \frac{\partial \tilde{\sigma}_{z_i}^2}{\partial p_j} - \frac{\partial \bar{\sigma}_{z_i}^2}{\partial p_j} &= \underbrace{\text{trace} \left[ T^{-1} \Sigma_{\tilde{q}} (T^{-1})^T \frac{\partial (C_{zd,i}^T C_{zd,i})}{\partial p_j} \right]}_{\text{term1}} + \\ &\underbrace{\text{trace} \left[ \tilde{L}_i \left\{ T \frac{\partial A_{zd}}{\partial p_j} T^{-1} \Sigma_{\tilde{q}} + \Sigma_{\tilde{q}} (T^{-1})^T \frac{\partial A_{zd}^T}{\partial p_j} T^T + T \frac{\partial (B_{zd} B_{zd}^T)}{\partial p_j} T^T \right\} \right]}_{\text{term2}} \\ &- \underbrace{\text{trace} \left[ T^{-1} P^T \Sigma_{\tilde{q}} P (T^{-1})^T \frac{\partial (C_{zd,i}^T C_{zd,i})}{\partial p_j} \right]}_{\text{term3}} \\ &- \underbrace{\text{trace} \left[ \bar{L}_i \left\{ P^T \frac{\partial A_{zd}}{\partial p_j} T^{-1} P^T \Sigma_{\tilde{q}} + \Sigma_{\tilde{q}} P (T^{-1})^T \frac{\partial A_{zd}^T}{\partial p_j} T^T P^T + P^T \frac{\partial (B_{zd} B_{zd}^T)}{\partial p_j} T^T P^T \right\} \right]}_{\text{term4}} \end{aligned} \quad (\text{B.48})$$

Terms 1 and 3 correspond to each other and the result of their subtraction is:

term1 – term3 =

$$\begin{aligned} & \text{trace} \left[ T^{-1} \Sigma_{\tilde{q}} (T^{-1})^T \frac{\partial (C_{zd,i}^T C_{zd,i})}{\partial p_j} \right] - \text{trace} \left[ T^{-1} P^T \Sigma_{\tilde{q}} P (T^{-1})^T \frac{\partial (C_{zd,i}^T C_{zd,i})}{\partial p_j} \right] = \\ & \text{trace} \left[ \left( T^{-1} \Sigma_{\tilde{q}} (T^{-1})^T - T^{-1} P^T \Sigma_{\tilde{q}} P (T^{-1})^T \right) \frac{\partial (C_{zd,i}^T C_{zd,i})}{\partial p_j} \right] \end{aligned} \quad (\text{B.49})$$

since

$$\Sigma_{\tilde{q}} = \Sigma_H \text{ and } \Sigma_{\tilde{q}} = P \Sigma_H P^T \quad (\text{B.50})$$

we can rewrite (B.49) as

$$\begin{aligned} & \text{term1} - \text{term3} = \\ & \text{trace} \left[ \left( T^{-1} \Sigma_H (T^{-1})^T - T^{-1} P^T P \Sigma_H P^T P (T^{-1})^T \right) \frac{\partial (C_{zd,i}^T C_{zd,i})}{\partial p_j} \right] = \\ & \text{trace} \left[ T^{-1} \Sigma_R (T^{-1})^T \frac{\partial (C_{zd,i}^T C_{zd,i})}{\partial p_j} \right] \end{aligned} \quad (\text{B.51})$$

The difference of terms 2 and 4 from equation (B.48) is computed as

term2 – term4 =

$$\begin{aligned} & \text{trace} \left[ \tilde{L}_i \left\{ T \frac{\partial A_{zd}}{\partial p_j} T^{-1} \Sigma_{\tilde{q}} + \Sigma_{\tilde{q}} (T^{-1})^T \frac{\partial A_{zd}^T}{\partial p_j} T^T + T \frac{\partial (B_{zd} B_{zd}^T)}{\partial p_j} T^T \right\} \right] \\ & - \text{trace} \left[ \bar{L}_i \left\{ P T \frac{\partial A_{zd}}{\partial p_j} T^{-1} P^T \Sigma_{\tilde{q}} + \Sigma_{\tilde{q}} P (T^{-1})^T \frac{\partial A_{zd}^T}{\partial p_j} T^T P^T + P T \frac{\partial (B_{zd} B_{zd}^T)}{\partial p_j} T^T P^T \right\} \right] = \\ & \text{trace} \left[ \tilde{L}_i \left\{ T \frac{\partial A_{zd}}{\partial p_j} T^{-1} \Sigma_H + \Sigma_H (T^{-1})^T \frac{\partial A_{zd}^T}{\partial p_j} T^T + T \frac{\partial (B_{zd} B_{zd}^T)}{\partial p_j} T^T \right\} \right] - \\ & \text{trace} \left[ \bar{L}_i \left\{ P T \frac{\partial A_{zd}}{\partial p_j} T^{-1} P^T P \Sigma_H P^T + P \Sigma_H P^T P (T^{-1})^T \frac{\partial A_{zd}^T}{\partial p_j} T^T P^T + P T \frac{\partial (B_{zd} B_{zd}^T)}{\partial p_j} T^T P^T \right\} \right] \end{aligned} \quad (\text{B.52})$$

The Lagrange multiplier matrix for the i-th performance metric in the balanced and the reduced case are related to each other as follows

$$\bar{L}_i = P \tilde{L}_i P^T \quad (\text{B.53})$$



This relationship is invoked to rewrite (B.52) as

$$\begin{aligned}
& \text{term2} - \text{term4} = \\
& \text{trace} \left[ \tilde{L}_i T \frac{\partial A_{zd}}{\partial p_j} T^{-1} \Sigma_H + \tilde{L}_i \Sigma_H (T^{-1})^T \frac{\partial A_{zd}^T}{\partial p_j} T^T + \tilde{L}_i T \frac{\partial (B_{zd} B_{zd}^T)}{\partial p_j} T^T \right] - \\
& \text{trace} \left[ P \tilde{L}_i P^T P T \frac{\partial A_{zd}}{\partial p_j} T^{-1} P^T P \Sigma_H P^T + P \tilde{L}_i P^T P \Sigma_H P^T P (T^{-1})^T \frac{\partial A_{zd}^T}{\partial p_j} T^T P^T + \right. \\
& \left. P \tilde{L}_i P^T P T \frac{\partial (B_{zd} B_{zd}^T)}{\partial p_j} T^T P^T \right] = \\
& \text{trace} \left[ \tilde{L}_i \left( T \frac{\partial A_{zd}}{\partial p_j} T^{-1} \Sigma_R + \Sigma_R (T^{-1})^T \frac{\partial A_{zd}^T}{\partial p_j} T^T + I_R T \frac{\partial (B_{zd} B_{zd}^T)}{\partial p_j} T^T \right) \right]
\end{aligned} \tag{B.54}$$

Combining corresponding terms and rearranging allows expressing the difference in terms of the matrices  $\Sigma_R$  and  $I_R$  as defined in (5.63) and (B.55) respectively.

$$I_R = \begin{bmatrix} \mathbf{0}^{n_k \times n_k} & \mathbf{0}^{n_k \times n_r} \\ \mathbf{0}^{n_r \times n_k} & I^{n_r \times n_r} \end{bmatrix} \tag{B.55}$$

Then the difference of the partial derivatives of the variances can be written using (B.54) and (B.51) as follows.

$$\begin{aligned}
& \frac{\partial \tilde{\sigma}_{z_i}^2}{\partial p_j} - \frac{\partial \sigma_{z_i}^2}{\partial p_j} = \text{trace} \left[ T^{-1} \Sigma_R (T^{-1})^T \frac{\partial (C_{zd,i}^T C_{zd,i})}{\partial p_j} \right] + \\
& \text{trace} \left[ \tilde{L}_i \left\{ T \frac{\partial A_{zd}}{\partial p_j} T^{-1} \Sigma_R + \Sigma_R (T^{-1})^T \frac{\partial A_{zd}^T}{\partial p_j} T^T + I_R T \frac{\partial (B_{zd} B_{zd}^T)}{\partial p_j} T^T \right\} \right]
\end{aligned} \tag{B.56}$$

Where  $\Sigma_R$  is the matrix containing the Hankel singular values corresponding to the removed states, as defined in (5.63) and  $I_R$  is the identity matrix with ones as diagonal entries corresponding to the removed states and zeros everywhere else. The computation of the sensitivity error  $\varepsilon_{RMS}^{sens}$  according to (B.47) requires many terms. Especially the computation of the Lagrange multiplier matrix of the balanced, unreduced system  $\tilde{L}_i$  as shown in equation (5.40) is undesirable. What is needed is therefore an (upper) error bound for the *sensitivity error* similar to the bound for the RMS performance error shown in (5.65). Such a bound can be found by first computing the relative sensitivity error. This is done by dividing equation (B.47) with the reduced model sensitivity  $\partial \tilde{\sigma}_{z_i} / \partial p_j$  such that

$$\left| \frac{\Delta \frac{\partial \sigma_{z_i}}{\partial p_j}}{\frac{\partial \bar{\sigma}_{z_i}}{\partial p_j}} \right| = \left| \frac{\frac{\partial \tilde{\sigma}_{z_i}}{\partial p_j} - \frac{\partial \bar{\sigma}_{z_i}}{\partial p_j}}{\frac{\partial \bar{\sigma}_{z_i}}{\partial p_j}} \right| = \left| \frac{\bar{\sigma}_{z_i} \left( \frac{\partial \tilde{\sigma}_{z_i}^2}{\partial p_j} - \frac{\partial \bar{\sigma}_{z_i}^2}{\partial p_j} \right) - \Delta \sigma_{z_i} \cdot \frac{\partial \bar{\sigma}_{z_i}^2}{\partial p_j}}{2 \frac{\partial \bar{\sigma}_{z_i}}{\partial p_j} (\bar{\sigma}_{z_i}^2 + \Delta \sigma_{z_i} \cdot \bar{\sigma}_{z_i})} \right| \quad (\text{B.57})$$

Substituting in (5.67) for  $\partial \bar{\sigma}_{z_i} / \partial p_j$  in the denominator and rearranging yields

$$\left| \frac{\Delta \frac{\partial \sigma_{z_i}}{\partial p_j}}{\frac{\partial \bar{\sigma}_{z_i}}{\partial p_j}} \right| = \left| \frac{\bar{\sigma}_{z_i} \left( \frac{\partial \tilde{\sigma}_{z_i}^2}{\partial p_j} - \frac{\partial \bar{\sigma}_{z_i}^2}{\partial p_j} \right) - \Delta \sigma_{z_i} \cdot \frac{\partial \bar{\sigma}_{z_i}^2}{\partial p_j}}{\frac{\partial \bar{\sigma}_{z_i}^2}{\partial p_j} (\bar{\sigma}_{z_i} + \Delta \sigma_{z_i})} \right| = \left| \frac{\bar{\sigma}_{z_i} \left( \frac{\partial \tilde{\sigma}_{z_i}^2}{\partial p_j} / \frac{\partial \bar{\sigma}_{z_i}^2}{\partial p_j} - 1 \right) - \Delta \sigma_{z_i}}{\tilde{\sigma}_{z_i}} \right| \quad (\text{B.58})$$

Even though the above expression is valid, it does not yet represent the simple error bound we are trying to find. A first simplification is introduced by replacing  $\tilde{\sigma}_{z_i}$  with  $\bar{\sigma}_{z_i}$  in the denominator. Since  $\tilde{\sigma}_{z_i} > \bar{\sigma}_{z_i}$  we obtain an inequality. We also keep the absolute value of both sides, since the relative sensitivity error from (B.47) can be positive or negative and we are only interested in the magnitude of the relative sensitivity error. This results in:

$$\left| \frac{\Delta \frac{\partial \sigma_{z_i}}{\partial p_j}}{\frac{\partial \bar{\sigma}_{z_i}}{\partial p_j}} \right| \leq \left| \left( \frac{\frac{\partial \tilde{\sigma}_{z_i}^2}{\partial p_j}}{\frac{\partial \bar{\sigma}_{z_i}^2}{\partial p_j}} - 1 \right) - \frac{\Delta \sigma_{z_i}}{\bar{\sigma}_{z_i}} \right| \quad (\text{B.59})$$

We recognize that the second term on the right side of (B.59) corresponds to the expression for the relative RMS error according to (5.65). Since we have an upper bound for the relative RMS error we flip the negative sign in (B.59), which preserves the inequality. The ratio of partial derivatives of the variances is approximated as follows:

$$\frac{\frac{\partial \tilde{\sigma}_{z_i}^2}{\partial p_j}}{\frac{\partial \bar{\sigma}_{z_i}^2}{\partial p_j}} \approx \frac{\text{trace} [\Sigma_H^2]}{\text{trace} [\Sigma_K^2]} = \frac{\sum_{i=1}^{n_s} (\sigma_i^H)^2}{\sum_{i=1}^{n_k} (\sigma_i^H)^2} \quad (\text{B.60})$$

Finally we arrive at an upper error bound for the sensitivity error by substituting (B.60) and (5.65) into (B.59).

$$\left| \frac{\Delta \frac{\partial \sigma_{z_i}}{\partial p_j}}{\frac{\partial \bar{\sigma}_{z_i}}{\partial p_j}} \right| < \frac{\sum_{i=1}^{n_s} (\sigma_i^H)^2}{\sum_{i=1}^{n_k} (\sigma_i^H)^2} - 1 + \frac{1}{2} \cdot \frac{\sum_{i=n_k+1}^{n_s} \sigma_i^H}{\sum_{i=1}^{n_k} \sigma_i^H} \quad (\text{B.61})$$

## B.5 Summary

This appendix contains derivations in support of Chapter 5. These include a proof for similarity transformations, the derivation of the Transformed Governing Sensitivity Equation (TGSE) and the derivation of error bounds for reduced state space systems.



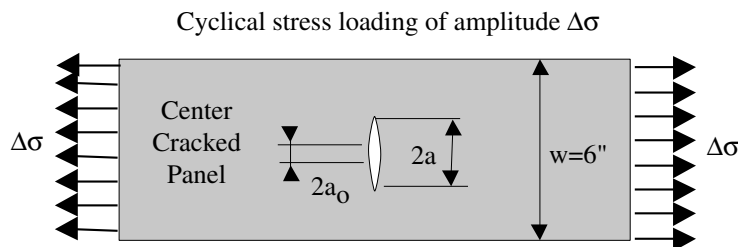
## Appendix C

# Application of Isoperformance to other Fields of Engineering

The focus of this thesis has been on applying the isoperformance methodology to dynamics and controls problems. The kernel isoperformance algorithms developed in Chapters 3 and 4 are, however, applicable to a wide range of engineering problems. The purpose of this appendix is to demonstrate the versatility of the isoperformance technique. The application to crack growth in a metallic structure is presented by means of a cracked panel sample problem.

### C.1 Crack Growth in Metallic Structures

Consider the problem shown in Figure C.1, where a flat, infinitely long metallic panel of width,  $w$ , is subjected to a longitudinal stress loading,  $\Delta\sigma = \sigma_{max} - \sigma_{min}$ .



**Figure C.1:** Crack growth sample problem: Center cracked infinitely long panel.

The stress varies sinusoidally as a function of time with a stress range  $\Delta\sigma$ , whereby  $R = \sigma_{min}/\sigma_{max} = 0$ . The amplitude  $\Delta\sigma/2$  is assumed constant. Any thermal effects which might

come into play above a cycling frequency of about 5 [Hz] are ignored. Since this is a crack growth calculation, an initial transverse crack of length,  $2a_o$ , is assumed to exist at the center of the panel. The crack will subsequently propagate to length,  $2a$ , until it reaches the critical crack length,  $2a_c$ . This is due to stress concentration at the crack tip, giving rise to a stress intensity  $\Delta K$ , see Reference [134]:

$$\Delta K = \Delta\sigma\sqrt{\pi a} \cdot \sqrt{\sec\left(\frac{\pi a}{w}\right)} = \Delta\sigma\sqrt{\pi a} \cdot \left[\cos\left(\frac{\pi a}{w}\right)\right]^{-1/2} \quad (\text{C.1})$$

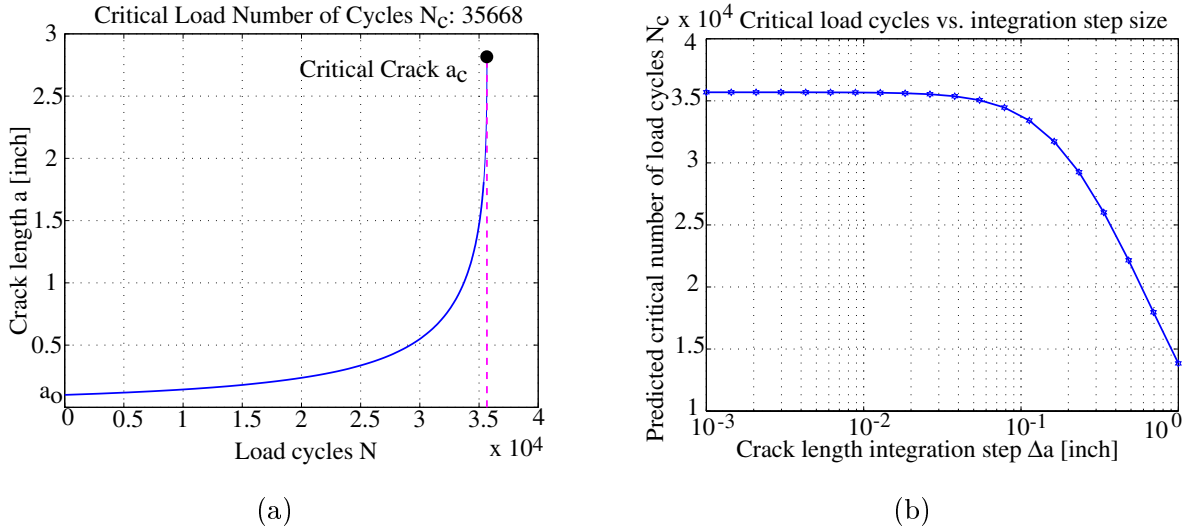
The rate of crack growth as a function of the stress intensity,  $\Delta K$ , is given by the Paris Law as:

$$\frac{da}{dN} = C\Delta K^m \quad (\text{C.2})$$

where  $C$  and  $m$  are material dependent crack growth parameters. The goal of the calculation is to determine the critical number of load cycles,  $N_c$ , from initial to critical crack length,  $a_o \rightarrow a_c$ . The critical number of load cycles can be computed by integrating from  $a_o$  to  $a_c$ :

$$N_c = \int_{a_o}^{a_c} \left(\frac{da}{dN}\right)^{-1} da = \int_{a_o}^{a_c} C^{-1}\Delta K^{-m} da = C^{-1}\pi^{-\frac{m}{2}}\Delta\sigma^{-m} \int_{a_o}^{a_c} \left[\cos\left(\frac{\pi a}{w}\right)\right]^{\frac{m}{2}} a^{-\frac{m}{2}} da \quad (\text{C.3})$$

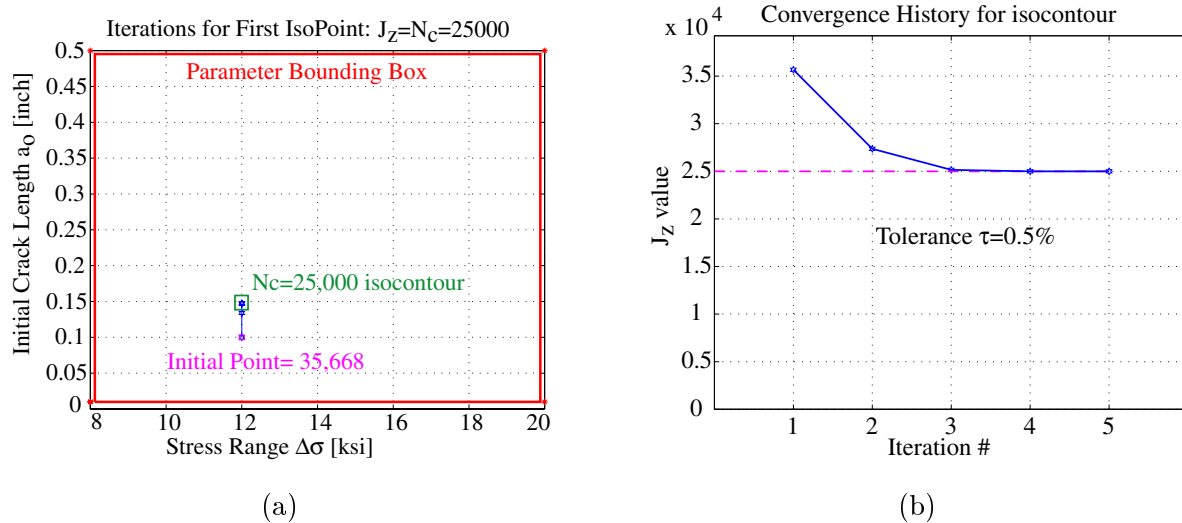
Unfortunately Equation (C.3) is transcendental, such that a solution can only be found numerically. The solution,  $N_c$ , is quite sensitive to the integration step,  $\Delta a$ . If  $\Delta a$  is chosen smaller and smaller the solution will asymptote to the correct value. This is shown in Figure C.2(b).



**Figure C.2:** (a) Nominal crack growth curve from  $a_o$  to  $a_c$  with integration step  $\Delta a = 0.01$ ". (b) Dependence of predicted critical number of load cycles,  $N_c$ , on crack length integration step size,  $\Delta a$ .

The nominal case assumes a stress range,  $\Delta\sigma = \sigma_{max} - \sigma_{min} = 12$  [ksi],  $R = 0$ , an initial crack length  $a_o = 0.1$ " [inch], a critical crack length  $a_c = 2.8$ " as well as material coefficients  $C = 4 \cdot 10^{-9}$  and  $m = 3.5$ . The panel width is  $w = 6$ " as shown in Figure C.1. The crack propagation from  $a_o$  to  $a_c$  for this nominal case is plotted in Figure C.2(a). It can be seen that the crack grows slowly at first and that the rate of growth increases until the crack goes unstable at  $a_c = 2.8$ ", causing the panel to fail. The dependence of the answer on the crack length increment,  $da \simeq \Delta a$ , is depicted in Figure C.2(b). One can see that the number of critical load cycles,  $N_c$ , is underpredicted for large step sizes. Below  $\Delta a \approx 0.01$ " the number of load cycles asymptotes to a value of  $N_c = 35,668$ . This is why this step size is used.

Applying the isoperformance technique to this problem, we may consider the critical number of load cycles to be the "performance" of the system, i.e.  $J_z = N_c$ . Assume that a performance of  $J_{z,req} = N_{c,req} = 25,000$  load cycles to failure is required for safety and design life considerations. For a bivariate problem we consider the initial crack length,  $a_o$ , and the stress range,  $\Delta\sigma$ , to be the variable design parameters. The intervals over which the parameters can vary are  $[0.01", 0.5"]$  and  $[8 \text{ ksi}, 20 \text{ ksi}]$ , respectively.

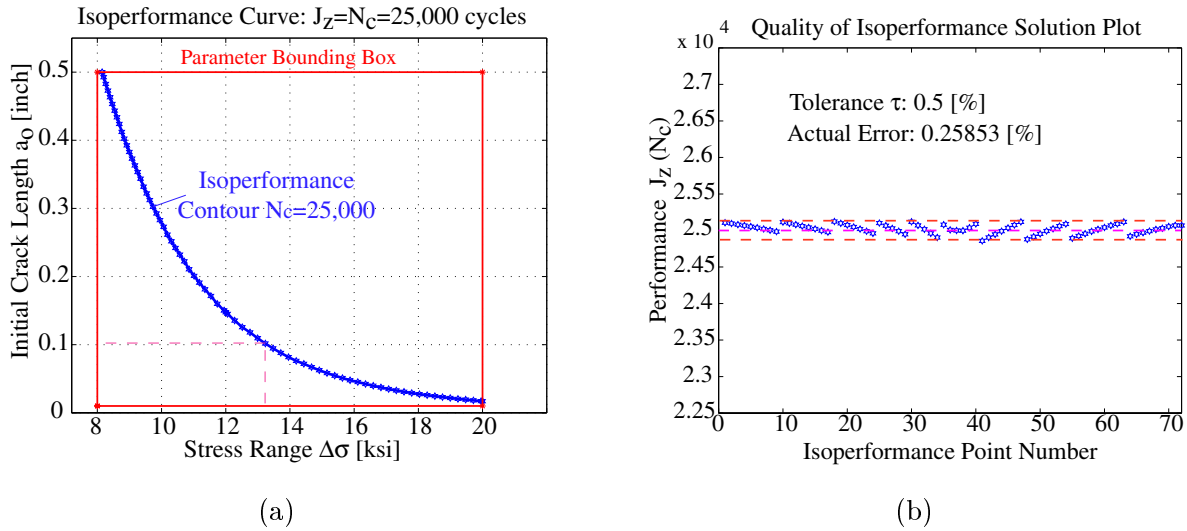


**Figure C.3:** (a) Crack growth problem: Gradient search for  $J_z = 25,000$  isoperformance contour. (b) Iterations to intercept.

Figure C.3(a) shows the gradient search from the initial guess ( $a_o = 0.1$ ",  $\Delta\sigma = 12$  [ksi]) until the  $N_c = 25,000$  isoperformance contour is intercepted. Since the first order derivatives of the integral in Equation (C.3) cannot be obtained analytically, we approximate the gradient

vector  $\nabla J_z = [\partial N_c / \partial \Delta\sigma, \partial N_c / \partial a_o]^T$  via finite differences<sup>1</sup>. The isocontour (=“curve of equal crack growth life”) is found in five iterations, given a numerical tolerance of  $\tau = 0.5\%$ , see Figure C.3(b).

Once the isoperformance contour is intercepted we apply the Gradient-Based Contour Following Algorithm which was developed in Subsection 3.2.2. The performance-invariant curve is computed until the parameter bounding box,  $\mathbf{B}$ , is intercepted. The result is shown in Figure C.4(a). Every combination along this curve will (theoretically) produce a crack growth life of  $N_c = 25,000$ . A total of 72 isoperformance points,  $p_{iso}$ , are obtained. This required 1.79 million floating point operations and 15.87 seconds of CPU time (Pentium III, 650 MHz computer).



**Figure C.4:** (a) Crack growth problem: Isoperformance contour for  $N_c = 25,000$  and variable parameters  $\Delta\sigma$  and  $a_o$ . (b) Quality of solution plot for tolerance  $\tau = 0.5\%$ .

As expected, the isoperformance curve shows that as the initial crack length,  $a_o$ , increases the stress range,  $\Delta\sigma$ , must be decreased in order to achieve the same performance. This is a non-linear relationship and assumes that the critical crack length,  $a_c$ , is the same for all configurations and does not depend on the maximum stress  $\sigma_{max}$ . The “quality of solution”, using the metric,  $\Upsilon_{iso}$ , from Equation (3.40), is shown in Figure C.4(b). It can be seen that the computed isoperformance curve is within the required numerical tolerance,  $\tau$ .

The curve in Figure C.4(a) could support important design decisions. Assume for example

<sup>1</sup>A 1% perturbation size is used.



that the detectable crack length at manufacture is 0.1” or larger and that the panel could not be inspected during operations; this would then restrict the stress range to 13 [ksi] or smaller. A multivariable analysis involving the other parameters ( $a_c$ ,  $w$ ,  $C$  and  $m$ ) could be conducted using the multivariable algorithms from Chapter 4. In this way tradeoffs between loading conditions, material properties and geometry could be made, while holding the performance in terms of crack growth life (fixed by a “customer” requirement) constant. This demonstrates the applicability of isoperformance to problems other than dynamics and controls.



## Appendix D

# DOLCE Finite Element Model

This listing is for the DOLCE testbed finite element model and is compatible with IMOS version 5.0 [62].

```
% dolce_fem
% Simple FEM of the DOLCE testbed
% see lab notebook page 46 and following
% dWo, 6/30/2001

flops(0) tic;
% constants
diagnostics=0;
lbs2kg=0.45;
mp=0*lbs2kg;

%grid locations [m]
xyz=[...
    1    -0.25    -0.25    0.0
    2     0.25    -0.25    0.0
    3     0.25     0.25    0.0
    4    -0.25     0.25    0.0
    5    -0.25    -0.25    0.5
    6     0.25    -0.25    0.5
```

7	0.25	0.25	0.5
8	-0.25	0.25	0.5
9	-0.125	-0.125	0.5
10	0.125	-0.125	0.5
11	0.125	0.125	0.5
12	-0.125	0.125	0.5
13	-0.125	-0.125	0.75
14	0.125	-0.125	0.75
15	0.125	0.125	0.75
16	-0.125	0.125	0.75
17	0.0	0.0	0.0
18	0.0	0.0	0.5
19	0.0	0.0	0.75
20	0.0	-0.25	0.0
21	0.25	0.0	0.0
22	0.0	0.25	0.0
23	-0.25	0.0	0.0
24	-0.125	-0.25	0.5
25	0.0	-0.25	0.5
26	0.125	-0.25	0.5
27	0.25	-0.125	0.5
28	0.25	0.0	0.5
29	0.25	0.125	0.5
30	0.125	0.25	0.5
31	0.0	0.25	0.5
32	-0.125	0.25	0.5
33	-0.25	0.125	0.5
34	-0.25	0.0	0.5
35	-0.25	-0.125	0.5
36	0.0	-0.125	0.5
37	0.125	0.0	0.5
38	0.0	0.125	0.5

```

39   -0.125    0.0    0.5
40    0.0      0.0    0.868
101  -0.25    -0.25   -0.1
102   0.25    -0.25   -0.1
103   0.25     0.25   -0.1
104  -0.25     0.25  -0.1 ];

```

```
% connectivity beam elements
```

```
nibar=[ ...
```

```

1    1    2    0    1    0    0    1    1    1
2    2    3    0    1    0    0    1    1    1
3    3    4    0    1    0    0    1    1    1
4    4    1    0    1    0    0    1    1    1
5    1    5    0    1    0    0    1    1    1
6    2    6    0    1    0    0    1    1    1
7    3    7    0    1    0    0    1    1    1
8    4    8    0    1    0    0    1    1    1
9    5    6    0    1    0    0    1    1    1
10   6    7    0    1    0    0    1    1    1
11   7    8    0    1    0    0    1    1    1
12   8    5    0    1    0    0    1    1    1
13   4    5    0    1    0    0    1    1    1
14   3    8    0    1    0    0    1    1    1
15   2    7    0    1    0    0    1    1    1
16   9    10   0    2    0    0    1    1    1
17  10    11   0    2    0    0    1    1    1
18  11    12   0    2    0    0    1    1    1
19  12     9    0    2    0    0    1    1    1
20   9    13   0    2    0    0    1    1    1
21  10    14   0    2    0    0    1    1    1
22  11    15   0    2    0    0    1    1    1
23  12    16   0    2    0    0    1    1    1

```

```

24      13      14      0      2      0      0      1      1      1
25      14      15      0      2      0      0      1      1      1
26      15      16      0      2      0      0      1      1      1
27      16      13      0      2      0      0      1      1      1
28      9       11      0      2      0      0      1      1      1
29      9       16      0      2      0      0      1      1      1
30      16      11      0      2      0      0      1      1      1
31      11      14      0      2      0      0      1      1      1
32      9       14      0      2      0      0      1      1      1
33      13      15      0      2      0      0      1      1      1
34      17      18      0      3      0      0      1      1      1];

```

```
% Create bar element property matrix
```

```
Alarge=((0.02212/2)^2)*pi-(((0.02212/2)-0.001016)^2)*pi;
```

```
Asmall=((0.009398/2)^2)*pi-(((0.009398/2)-0.001524)^2)*pi;
```

```
Iblarge=(pi*(0.02212/2)^4/4)-(pi*((0.02212/2)-0.001016)^4/4);
```

```
Ibsmall=(pi*(0.009398/2)^4/4)-(pi*((0.009398/2)-0.001524)^4/4);
```

```
Jlarge=2*Iblarge; %approximation
```

```
Jsmall=2*Ibsmall; %approximation
```

```
propbar=[ ...
```

```

% pid      mid      A      I22      I33      J
      1      1      Alarge      Iblarge      Iblarge      Jlarge
      2      1      Asmall      Ibsmall      Ibsmall      Jsmall
      3      1  0.005^2*pi  pi*0.005^4/12  pi*0.005^4/12  pi*0.005^4/6
];

```

```
% connectivity plate elements
```

```
%sandwich
```

```
nisplate=[...
```

```

35      1      20      17      23      4

```

```

36  20  2  21  17  4
37  17  21  3  22  4
38  23  17  22  4  4];
%top Al Plate
niuplate=[...
39   5  24   9  35  5
40  24  25  36   9  5
41  25  26  10  36  5
42  26   6  27  10  5
43  35   9  39  34  5
44   9  36  18  39  5
45  36  10  37  18  5
46  10  27  28  37  5
47  34  39  12  33  5
48  39  18  38  12  5
49  18  37  11  38  5
50  37  28  29  11  5
51  33  12  32   8  5
52  12  38  31  32  5
53  38  11  30  31  5
54  11  29   7  30  5];
tface=0.0019; tcore=0.03146; tplate=0.004572;
msp=6.751; % mass of sandwich plate
propsplate=[...
4      1      tface  1      tcore  msp/0.25];
propuplate=[...
5      1      tplate  0];

% springs
lbs2kg=0.45; in2m=0.0254; lbspin2Npm=lbs2kg*9.81/in2m;
k_green=168; %catalogue value lbs/in
k_spring=k_green*lbspin2Npm;

```

```

nicelas2=[...
    55  1  1  101  1  1e3*k_spring 0
    56  1  2  101  2  1e3*k_spring 0
    57  1  3  101  3  k_spring 0
    58  1  4  101  4  0.1*k_spring 0
    59  1  5  101  5  0.1*k_spring 0
    60  1  6  101  6  1e-3*k_spring 0
    61  2  1  102  1  1e3*k_spring 0
    62  2  2  102  2  1e3*k_spring 0
    63  2  3  102  3  k_spring 0
    64  2  4  102  4  0.1*k_spring 0
    65  2  5  102  5  0.1*k_spring 0
    66  2  6  102  6  1e-3*k_spring 0
    67  3  1  103  1  1e3*k_spring 0
    68  3  2  103  2  1e3*k_spring 0
    69  3  3  103  3  k_spring 0
    70  3  4  103  4  0.1*k_spring 0
    71  3  5  103  5  0.1*k_spring 0
    72  3  6  103  6  1e-3*k_spring 0
    73  4  1  104  1  1e3*k_spring 0
    74  4  2  104  2  1e3*k_spring 0
    75  4  3  104  3  k_spring 0
    76  4  4  104  4  0.1*k_spring 0
    77  4  5  104  5  0.1*k_spring 0
    78  4  6  104  6  1e-3*k_spring 0];

%concentrated masses
m_bn=0.2281; m_sn=0.03475; mshk=11.921+0.33975;
niconm2= [ ...
    79  17  lbs2kg*mp      0.0E+00  0.0E+00  0.0E+00  0  0  0  0  0  0
    80   1  m_bn          0.0E+00  0.0E+00  0.0E+00  0  0  0  0  0  0
    81   2  m_bn          0.0E+00  0.0E+00  0.0E+00  0  0  0  0  0  0

```



```

82   3  m_bn  0.0E+00  0.0E+00  0.0E+00  0  0  0  0  0  0
83   4  m_bn  0.0E+00  0.0E+00  0.0E+00  0  0  0  0  0  0
84   5  m_bn  0.0E+00  0.0E+00  0.0E+00  0  0  0  0  0  0
85   6  m_bn  0.0E+00  0.0E+00  0.0E+00  0  0  0  0  0  0
86   7  m_bn  0.0E+00  0.0E+00  0.0E+00  0  0  0  0  0  0
87   8  m_bn  0.0E+00  0.0E+00  0.0E+00  0  0  0  0  0  0
88   9  m_sn  0.0E+00  0.0E+00  0.0E+00  0  0  0  0  0  0
89  10  m_sn  0.0E+00  0.0E+00  0.0E+00  0  0  0  0  0  0
90  11  m_sn  0.0E+00  0.0E+00  0.0E+00  0  0  0  0  0  0
91  12  m_sn  0.0E+00  0.0E+00  0.0E+00  0  0  0  0  0  0
92  13  m_sn  0.0E+00  0.0E+00  0.0E+00  0  0  0  0  0  0
93  14  m_sn  0.0E+00  0.0E+00  0.0E+00  0  0  0  0  0  0
94  15  m_sn  0.0E+00  0.0E+00  0.0E+00  0  0  0  0  0  0
95  16  m_sn  0.0E+00  0.0E+00  0.0E+00  0  0  0  0  0  0
96  19  mshk  0.0E+00  0.0E+00  0.0E+00  0  0  0  0  0  0];

```

```
% material Al 2219-T851
```

```
mat=0; mid=      1; E=  72e9; nu=  0.31000E+00; rho=  2.85e3;
```

```
alpha=  22.1e-6; Tref=  20.00000E+00;
```

```
mat=mat1(mid,E,nu,rho,alpha,Tref,mat);
```

```
% boundary conditions - initialize m and k
```

```
% Create initial boundary condition matrix
```

```
% (0= constrain, 1= free)
```

```
bci=ones(size(xyz,1),6);
```

```
% fix nodes 101-104 to concrete foundation (ground)
```

```
bci(size(bci,1)-3:end,:)=zeros(4,6);
```

```
% fix z-rotation for plate nodes
```

```
bci([17:18 20:39],6)=zeros(22,1);
```

```
% Compute the number of dofs and dof numbers, initialize k and m
```

```
[bc,ndof]=bcond(bci); nset=[1:ndof]; k=sparse(ndof,ndof); m=k;
```

```

g=k;

% assemble m and k
if diagnostics
    disp('Assembling m for conm2s')
end
    m=conm2(niconm2,bc,m,xyz);
if diagnostics
    disp('Assembling k and m for beams')
end
    [k,m]=beam_lump(nibar,xyz,propbar,mat,bc,k,m);
if diagnostics
    disp('Assembling k and m for plates')
end
    [k,m] = uplate(niuplate,xyz,propuplate,mat,bc,k,m);
    [k,m]= splate(nisplate,xyz,propsplate,mat,bc,k,m);
if diagnostics
    disp('Assembling k for spring elements')
end
    [k,g]=celas2(nicelas2,xyz,bc,k,g);

% mass properties
[xyzcg]=cg_calc(m,xyz,bc); mass=wtcg(bc,xyz,m,xyzcg);

% process rigid body elements
rg=[0];
mset=[0];
if diagnostics
    disp('Processing RBE2 rigid body elements')
end
gn=[40]; cm=[ ...
    1

```

```

        2
        3
        4
        5
        6
    ];
gm=[ ...
    13
    14
    15
    16
    19
];
    [nset,mset,rg]=rbe2(bc,xyz,nset,mset,rg,gn,cm,gm);
    rg=sparse(rg);
if diagnostics
    disp('Reducing k and m to independent dofs')
end
    [gm,k,m]=mce1(nset,mset,rg,k,m);
    bc=bcnset(bc,nset,mset);
    fset=nset;
    % solve eigenproblem
[phi,omeg]=eigfem(k,m); omeg=abs(omeg); phi=real(phi);
phi=normphi(phi,m); % mass normalize
phi=mce_dis(nset,mset,rg,phi);
phi_full=zeros(prod(size(bci)),size(phi,2));
phi_full(find(bci'),:)=phi; phi=phi_full;

% select modes for analysis
nm=20; % number of modes in FEM
omeg=omeg(1:nm); phi=phi(:,1:nm);

```

```

% (1) build state space model using mode2ss2
% define input dofs
% input node: 40 direction 3
ig=[ 40  0 0 1 0 0 0 ];
% ig= input vector 1- shaker Force Fc
str_in= str2mat('1 Fc');

% define displacement output dofs
% output nodes: 17 in 3 direction
dg=[ 17  0 0 1 0 0 0];
str_out=str2mat('1 z');
za=0.005*ones(1,nm); % flexible prop damping
za(1:3)=0.025*ones(1,3); % Suspension modes have higher damping (refine later)
nrbm=0;
vg=[];
% compute state space model
[Ap,Bp,Cp,Dp,lb,lc] =
mode2ss2(xyz,bc,nm,ig,dg,vg,nrbm,za,phi,omeg);
sysp=ss(Ap,Bp,1e6*Cp,Dp); if 0
    f=logspace(-1,2,1000);
    [mag_fem,phs_fem]=bode(sysp,f*2*pi);
    mag_fem=squeeze(mag_fem);
    figure
    semilogy(f,mag_fem)
end

toc flops

```

# References

- [1] K. J. ASTRÖM AND B. WITTENMARK, *Computer-Controlled Systems: Theory and Design*, Prentice-Hall, Inc., 1997.
- [2] R. H. BARTELS AND G. STEWART, *Solution of the matrix equation  $ax+xb=c$* , Communications of ACM, 15 (1972).
- [3] I. BASDOGAN AND R. GROGAN, *Preliminary optical performance analysis of the space interferometer mission using an integrated modeling methodology*, in Control of Vibration and Noise - New Millenium: International Mechanical Engineering Congress and Exposition, 6th Biennial Symposium on Active Control of Vibration and Noise, 2000.
- [4] K. J. BATHE, *Finite Element Procedures*, Prentice-Hall, Inc., 1996.
- [5] W. BEITZ AND K.-H. GROTE, *DUBBEL-Taschenbuch für den Maschinenbau*, Springer Verlag, 20 ed., January 2001.
- [6] P. R. BÉLANGER, *Control Engineering*, vol. 1, Saunders College Publishing, Orlando, Florida, first ed., 1995.
- [7] B. BIALKE, *A compilation of reaction wheel induced spacecraft disturbances*, in Proceedings of the 20th Annual AAS Guidance and Control Conference, Breckenridge, CO, February 5–9, 1997. AAS Paper No. 97-038.
- [8] B. BIALKE, *A compilation of reaction wheel induced spacecraft disturbances*, 20th Annual American Astronautical Society Guidance and Control Conference, 1 (1997).
- [9] G. H. BLACKWOOD, R. N. JACQUES, AND D. W. MILLER, *The MIT multipoint alignment testbed: technology development for optical interferometry*, in Proceedings of the SPIE Conference on Active and Adaptive Optical Systems, vol. 1542, San Diego, CA, July 1991, pp. 371–391.
- [10] M. BORN AND E. WOLF, *Principles of Optics*, Cambridge University Press, sixth (corrected) ed., 1998.
- [11] R. G. BROWN AND P. Y. C. HWANG, *Introduction to Random Signals and Applied Kalman Filtering*, John Wiley & Sons, Inc., 1997.
- [12] A. E. BRYSON AND R. A. MILLS, *Linear-quadratic-gaussian controllers with specified parameter robustness*, Journal of Guidance, Control, and Dynamics, 21 (1998), pp. 11–18.
- [13] M. E. CAMPBELL AND E. F. CRAWLEY, *Uncertainty modeling for structural control analysis and synthesis*, Tech. Rep. 2-96, MIT Space Engineering Research Center, January 1996.
- [14] M. E. CAMPBELL AND E. F. CRAWLEY, *Development of structural uncertainty models*,

Journal of Guidance, Control, and Dynamics, 20 (1997), pp. 841–849.

- [15] M. COLAVITA, *Visibility and phasing*, JPL Interoffice Memorandum, To SITE distribution, August 10, 1994.
- [16] R. D. COOK, D. S. MALKUS, AND M. E. PLESHA, *Concepts and Applications of Finite Element Analysis*, John Wiley & Sons, Inc., 1989.
- [17] R. CRAIG JR., *Structural Dynamics: An Introduction to Computer Methods*, John Wiley & Sons, Inc., 1981.
- [18] S. H. CRANDALL AND W. D. MARK, *Random Vibration in Mechanical Systems*, Academic Press, 1963.
- [19] E. F. CRAWLEY, B. P. MASTERS, AND T. T. HYDE, *Conceptual design methodology for high performance dynamic structures*, in Proceedings of the 36<sup>th</sup> AIAA Structures, Structural Dynamics, and Materials Conference, New Orleans, LA, April 1995, pp. 2768–2787. AIAA Paper No. 95-2557.
- [20] L. P. DAVIS, D. CUNNINGHAM, AND J. HARRELL, *Advanced 1.5 Hz passive viscous isolation system*, in Proceedings of the 35<sup>th</sup> AIAA/ASME/ASCE/AAS/ASC Structures, Structural Dynamics, and Materials Conference, Hilton Head, SC, April 18–24, 1994, pp. 2655–2665. AIAA Paper No. 94-1651.
- [21] L. P. DAVIS, J. F. WILSON, R. E. JEWELL, AND J. J. RODEN, *Hubble space telescope reaction wheel assembly vibration isolation system*, in Vibration Damping Workshop, Marshall Space Flight Center, March 1986.
- [22] C. DE BOOR, *A practical guide to splines*, vol. I of Applied mathematical sciences, Springer Verlag, New York, 1 ed., 1978.
- [23] O. L. DE WECK, *Integrated Modeling and Dynamics Simulation for the Next Generation Space Telescope*, Master's thesis, Massachusetts Institute of Technology, June 1999.
- [24] O. L. DE WECK, D. MILLER, AND H. GUTIERREZ, *Structural dynamics and controls for NGST*, in Proceedings of the 34<sup>th</sup> Liège International Astrophysics Colloquium, Liège, Belgium, June 15–18, 1998.
- [25] O. L. DE WECK AND D. W. MILLER, *Performance and Sensitivity Analysis for Internally Balanced and Reduced Dynamic Linear Time-Invariant (LTI) Systems*. Research Memorandum, February 18 2000. Space Systems Laboratory, MIT.
- [26] J. R. DORMAND AND P. J. PRINCE, *A family of embedded Runge-Kutta formulae*, J. Comp. Appl. Math., 6 (1980), pp. 19–26.
- [27] A. DRESSLER, *HST and Beyond - Exploration and the Search for Origins: A Vision for Ultraviolet-Optical-Infrared Space Astronomy*, vol. 1, Association of Universities for Research in Astronomy, Washington D.C., 1 ed., May 1996.
- [28] C. H. EDWARDS AND D. E. PENNEY, *Multivariable Calculus with Analytic Geometry*, Prentice Hall, Upper Saddle River, NJ 07458, 5 ed., 1998.
- [29] L. E. EL GHAOU AND A. E. BRYSON, *Worst parameter changes for stabilized conservative systems*, in AIAA Guidance, Navigation, and Control Conference, New Orleans, LA, 1991, pp. 1490–1495.
- [30] C. E. EYERMAN AND J. F. SHEA, *A systems engineering approach to disturbance minimization for spacecraft utilizing controlled structures technology*, MIT SERC Report

#2-90, MIT, June 1990.

- [31] R. L. FOX AND M. P. KAPOOR, *Rates of change of eigenvalues and eigenvectors*, AIAA Journal, 6 (1968), pp. 2426–2429.
- [32] G. F. FRANKLIN AND J. D. POWELL, *Digital Control of Dynamic Systems*, Addison-Wesley Publishing Co., Inc., 1980.
- [33] R. FREUND, *Course Notes 16.910J/6.631 - Introduction to Simulation and Optimization - Massachusetts Institute of Technology*. Module 1, Sep-Oct 1999.
- [34] B. FRIEDLAND, *Control System Design: An Introduction to State-Space Methods*, McGraw-Hill, Inc., 1986.
- [35] A. GELB, ed., *Applied Optimal Estimation*, The M.I.T. Press, 1974.
- [36] J. J. GILHEANY, *Optimum selection of dampers for freely vibrating multidegree of freedom systems*, Proceedings of Damping '89, II (1989), pp. FCC-1:18.
- [37] O. GOLDREICH, *Complexity theory survey*. Internet: <http://www.wisdom.weizmann.ac.il>, accessed 5-4-2001 2001.
- [38] A. GRACE, *Optimization Toolbox User's Guide*, The MathWorks, Inc., 1994.
- [39] C. Z. GREGORY, *Reduction of large flexible spacecraft models using internal balancing theory*, Journal of Guidance, Control, and Dynamics, 7 (1984), pp. 17–32.
- [40] S. C. O. GROCOTT, *Comparision of control techniques for robust performance on uncertain structural systems*, Master's thesis, Massachusetts Institute of Technology, January 1994. SERC Report #2-94.
- [41] B. GROGAN AND I. BASDOGAN, *Preliminary Interface Description for SIM Classic Model Version 2.0*. Interoffice Memorandum, March 24 2000. Jet Propulsion Laboratory.
- [42] R. L. GROGAN AND R. A. LASKIN, *On multidisciplinary modeling of the space interferometry mission*, in Proceedings of the American Control Conference, Philadelphia, PA, June 1998.
- [43] H. GUTIERREZ, M. LEVINE, AND R. GROGAN, *Analysis of IPEX-2 pre-flight ground integration test data*, Tech. Rep. D-14905, NASA Jet Propulsion Laboratory, October 1997.
- [44] H. L. GUTIERREZ, *Damping formulations in substructure synthesis*, Master's thesis, Massachusetts Institute of Technology, May 1994.
- [45] H. L. GUTIERREZ, *Performance Assessment and Enhancement of Precision Controlled Structures During Conceptual Design*, PhD thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, 1999.
- [46] H. L. GUTIERREZ AND D. W. MILLER, *Disturbance analysis of integrated structural/optical models with uncertainties*, in Proceedings of the American Control Conference, Philadelphia, PA, June 1998.
- [47] T. K. HASSELMAN AND J. D. CHROSTOWSKI, *Evaluation of predictive accuracy in structural dynamic models*, in Proceedings of the International Modal Analysis Conference, 1989, pp. 360–366.
- [48] T. K. HASSELMAN, J. D. CHROSTOWSKI, AND T. J. ROSS, *Interval prediction in structural dynamic analysis*, in Proceedings of the 33<sup>rd</sup> AIAA/ASME/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Dallas, TX, April 1992, pp. 1272–1284.

AIAA Paper No. 92-2215.

- [49] T. K. HASSELMAN, J. D. CHROSTOWSKI, AND T. J. ROSS, *Propagation of Modeling Uncertainty Through Structural Dynamic Models*, in Proceedings of the 35<sup>th</sup> AIAA/ASME/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Hilton Head, SC, April 1994.
- [50] E. HECHT, *Optics*, Addison-Wesley Publishing Company, second ed., 1990.
- [51] G. J. HOU AND S. P. KENNY, *Eigenvalue and eigenvector approximate analysis for repeated eigenvalue problems*, AIAA Journal, 30 (1992).
- [52] G. J. HOU AND G. KOGANTI, *Sensitivity analysis of Lyapunov and Riccati equations with application to controls-structures integrated design*, in Proceedings of the 34<sup>th</sup> AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, LaJolla, CA, April 1993, pp. 1906–1915. AIAA Paper No. 93-1529.
- [53] J. P. HOW, *Robust Control Design with Real Parameter Uncertainty Using Absolute Stability Theory*, PhD thesis, Massachusetts Institute of Technology, February 1993. SERC Report #1-93.
- [54] T. T. HYDE, *Active Vibration Isolation for Precision Space Structures*, PhD thesis, Massachusetts Institute of Technology, January 1996. SERC Report #4-96.
- [55] M. INGHAM, *Microdynamics and thermal snap response of deployable space structures*, Master's thesis, Massachusetts Institute of Technology, June 1998. SERC Report #2-98.
- [56] R. N. JACQUES, *An approach to the preliminary design of controlled structures*, Master's thesis, Massachusetts Institute of Technology, February 1991. SERC Report #1-91.
- [57] R. N. JACQUES, *On-line System Identification and Control Design for Flexible Structures*, PhD thesis, Massachusetts Institute of Technology, May 1995. SERC Report #3-95.
- [58] L. JANDURA, *Analysis of hexapod isolator requirements necessary to meet SIM OPD requirements for astrometry and nulling*, Interoffice Memorandum SIM/ITP LJAN9705151, Jet Propulsion Laboratory, May 15, 1997.
- [59] JET PROPULSION LABORATORY, *Origins Missions*, URL <<http://origins.jpl.nasa.gov/missions/sbobs.html>>.
- [60] JET PROPULSION LABORATORY, *Space Interferometry Mission*, URL <<http://huey.jpl.nasa.gov/sim/index.html>>. Accessed November 11, 1998.
- [61] JET PROPULSION LABORATORY, *Modeling and Analysis for Controlled Optical Systems User's Manual*, 1997.
- [62] JET PROPULSION LABORATORY, *Integrated Modeling of Optical Systems User's Manual V5.0*, Jet Propulsion Laboratory, February 2000. JPL Publication 98-12, Rev. A.
- [63] J. L. JUNKINS AND Y. KIM, *Introduction to Dynamics and Control of Flexible Structures*, AIAA Education Series, 1992.
- [64] M. P. KAMAT, *Structural Optimization: Status and Promise*, vol. 150 of Progress in Aeronautics and Astronautics, American Institute of Aeronautics and Astronautics, Washington, D.C., 1992.
- [65] H. KARTTUNEN, P. KRÖGER, H. OJA, M. POUTANEN, AND K. J. DONNER, *Fundamental Astronomy*, vol. 1, Springer, Berlin, Germany, 3 ed., 1996.



- [66] A. KEANE, *Passive vibration control via unusual geometries: The application of genetic algorithm optimization to structural design*, Journal of Sound and Vibration, 185 (1995), pp. 441–453.
- [67] A. J. KEANE, *Experiences with optimizers in structural design*, Adaptive Computing in Engineering Design and Control, (1994).
- [68] A. J. KEANE AND A. P. BRIGHT, *Passive vibration control via unusual geometries: Experiments on model aerospace structures*, Journal of Sound and Vibration, 192 (1996).
- [69] R. KENNEDY, J. TURNAGE, AND M. B. JONES, *A meta-model for systems development through life cycle phases - coupling the isoperformance methodology with utility analysis*, SAE, Aerospace Technology Conference and Exposition, Long Beach, CA, AIAA Technical Library (1990), p. 9.
- [70] R. S. KENNEDY, M. B. JONES, AND D. R. BALTZLEY, *Empirical demonstration of isoperformance methodology preparatory of an interactive expert computerized decision aid. final report*, Tech. Rep. AIAA Technical Library, NTIS HC A03/MF A01, Essex Corporation, Orlando, FL, Feb 1988. AD-A202439; ARI-RN-88-93; Contract: MDA903-87-C-0603; Contract: DA PROJ. 2Q6-65502-M-770.
- [71] R. S. KENNEDY, M. B. JONES, AND D. R. BALTZLEY, *Optimal solutions for complex design problems: Using isoperformance software for human factors trade offs*, In NASA. Lyndon B. Johnson Space Center, 2nd Annual Workshop on Space Operations Automation and Robotics (SOAR 1988), N89-19817 12-59 (1988), pp. 313–319.
- [72] S. KENNY, *Eigenvalue and eigenvector derivatives for structures*, Final Report, MIT Course 2.093: Computer Methods in Dynamics, April 29, 1997.
- [73] W. J. LARSON AND J. R. WERTZ, eds., *Space Mission Analysis and Design*, Microcosm, Inc. and Kluwer Academic Publishers, 1991.
- [74] R. A. LASKIN, *Technology for space optical interferometry*, in Proceedings of the 33<sup>rd</sup> AIAA Aerospace Science Meeting and Exhibit, Reno, NV, January 1995. AIAA Paper No. 95-0825.
- [75] R. A. LASKIN, *SIM Dynamics & Control Requirements Flowdown Process*. Presentation at the SIM Project Preliminary Instrument System Requirements Review, JPL, March 17–18, 1998.
- [76] R. A. LASKIN AND M. SAN MARTIN, *Control/structure system design of a spaceborne optical interferometer*, in Proceedings of the AAS/AIAA Astrodynamics Specialist Conference, Stowe, VT, August 1989. AAS Paper No. 89-424.
- [77] A. LAUB, *Computation of balancing transformations*, Proceedings of JACC, 1 (1980).
- [78] A. LAUB, M. HEATH, C. PAIGE, AND R. WARD, *Computation of system balancing transformations and other applications of simultaneous diagonalization algorithms*, IEEE Trans. Automatic Control, (1987), pp. 115–122.
- [79] A. J. LAUB, M. T. HEATH, C. C. PAIGE, AND R. C. WARD, *Computation of system balancing transformations and other applications of simultaneous diagonalization algorithms*, IEEE Trans. on Automatic Control, AC-32 (1987), pp. 115–122.
- [80] M. LEVINE, R. BRUNO, AND H. GUTIERREZ, *Interferometry program flight experiment #1: Objectives and results*, in Proceedings of the 16<sup>th</sup> International Modal Analysis Conference, Santa Barbara, CA, 1998, pp. 1184–1190.

- [81] M. B. LEVINE, *Interferometry program flight experiments: IPEX I and II*, in Proceedings of the SPIE Conference on Astronomical Interferometry, R. D. Reasenberg, ed., vol. 3350, Kona, HI, March 1998, pp. 707–718.
- [82] K. B. LIM, D. E. COX, G. J. BALAS, AND J.-N. JUANG, *Validation of an experimentally derived uncertainty model*, in Proceedings of the 35<sup>th</sup> AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, January 1997. AIAA Paper No. 97-0244.
- [83] F. LIU, *NEXUS Optical Sensitivities*, Tech. Rep. NEXUS:99-007, Swales Aerospace & Associates Inc., 5050 Powder Mill Road, Beltsville, MD 20705, December 1999.
- [84] P. MAGHAMI, S. GUPTA, K. B. ELLIOT, S. M. JOSHI, AND J. E. WALZ, *Experimental validation of an integrated controls-structures design methodology for a class of flexible space structures*, in NASA Technical Paper 3462, November 1994.
- [85] G. MALLORY AND D. W. MILLER, *Decentralized state estimation for flexible space structures*, 40th AIAA Structural Dynamics and Materials Conference, (1999).
- [86] G. J. MALLORY, *Development and Experimental Validation of Direct Controller Tuning for Spaceborne Telescopes*, PhD thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, April 2000. Report SERC 1-2000.
- [87] B. MASTERS AND E. CRAWLEY, *An experimental investigation of optimized precision optical controlled-structures*, in Proceedings of the SPIE Conference on Smart Structures and Materials, vol. 3041, San Diego, CA, March 1997, pp. 348–359. Paper No. 3041-28.
- [88] B. P. MASTERS AND E. F. CRAWLEY, *Evolutionary design of controlled structures systems*, in Proceedings of the 38<sup>th</sup> AIAA Structures, Structural Dynamics, and Materials Conference, Kissimmee, FL, April 1997. AIAA Paper No. 97-1263.
- [89] R. MASTERTSON, *Development and Validation of Empirical and Analytical Reaction Wheel Disturbance Models*, Master’s thesis, Massachusetts Institute of Technology, June 1999.
- [90] J. MELODY AND M. LEVINE, *Model updating and evolutionary structures*, in Proceedings of the 15<sup>th</sup> Biennial Conference on Mechanical Engineering and Noise, Boston, MA, September 1995.
- [91] J. W. MELODY, *Discrete-frequency and broadband reaction wheel disturbance models*, Interoffice Memorandum 3411-95-200csi, Jet Propulsion Laboratory, June 1995.
- [92] J. W. MELODY, *SIM wavefront tilt disturbance analysis and angle feedforward*, Interoffice Memorandum 3411-96-336 ITP, Jet Propulsion Laboratory, November 1996.
- [93] J. W. MELODY AND H. C. BRIGGS, *Analysis of structural and optical interactions of the precision optical interferometer in space (POINTS)*, in Proceedings of the SPIE Conference on Spaceborne Interferometry, Orlando, FL, April 1993.
- [94] J. W. MELODY AND G. W. NEAT, *Integrated modeling methodology validation using the micro-precision interferometer testbed: Assessment of closed-loop performance prediction capability*, in Proceedings of the American Control Conference, Albuquerque, NM, June 1997.
- [95] A. MESSAC, R. GUELER, AND K. MALEK, *Control-structure integrated design: A computational approach*, in Proceedings of the 32<sup>nd</sup> AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Baltimore, MD, April 1991, pp. 553–567. AIAA Paper No. 91-1161.

- [96] D. MILLER, A. CURTIS, AND O. DE WECK *et al.*, *Architecting the Search for Terrestrial Planets and related Rrigins (astro)*, in SPIE International Symposium on Astronomical Telescopes and Instrumentation 2000, Munich, Germany, 2000.
- [97] D. MILLER, O. DE WECK, AND S. UEBELHART, *SIM Classic Version 2.0 - Preliminary Model Reduction, Disturbance and Sensitivity Analysis*. Presentation with Annotations, April 2000. Submitted to Dr. Sanjay Joshi, JPL.
- [98] D. MILLER, O. DE WECK, S. UEBELHART, R. GROGAN, AND I. BASDOGAN, *Integrated Dynamics and Controls Modeling for the Space Interferometry Mission (SIM)*, in IEEE Aerospace Conference, Big Sky, Montana, 2001.
- [99] M. MILMAN, M. SALAMA, R. SCHEID, R. BRUNO, AND J. GIBSON, *Integrated control-structure design: A multiobjective approach*, Tech. Rep. D-6767, JPL, October 1989.
- [100] M. MILMAN, M. SALAMA, R. SCHEID, AND J. S. GIBSON, *Combined control-structural optimization*, Computational Mechanics, (1991), pp. 1–18.
- [101] M. MILMAN, M. SALAMA, M. WETTE, AND R. BRUNO, *A multiobjective approach to integrated control, structure and optical design*, in Proceedings of the 32<sup>nd</sup> AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Baltimore, MD, April 1991, pp. 846–854. AIAA Paper No. 91-1136.
- [102] B. C. MOORE, *Principal component analysis of linear systems: Controllability, observability, and model reduction*, IEEE Transactions on Automatic Control, AC-26 (1981), pp. 17–32.
- [103] G. E. MOSIER, M. FEMIANO, K. HA, P. Y. BELY, R. BURG, D. C. REDDING, A. KISSIL, J. RAKOCZY, AND L. CRAIG, *Integrated modeling environment for systems-level performance analysis of the Next Generation Space Telescope*, in SPIE International Symposium on Space Telescopes and Instruments V, Kona, Hawaii, August 1998. Proc. SPIE Vol. 3356 p. 89-97.
- [104] D. V. MURTHY AND R. T. HAFTKA, *Survey of methods for calculating sensitivity of general eigenproblems*, in Sensitivity Analysis in Engineering, Langley Research Center, September 1986, pp. 177–196. NASA CP 2457.
- [105] NASA, *Origins: Roadmap for the Office of Space Science Origins Theme*, National Aeronautics and Space Administration, National Aeronautics and Space Administration, 1997.
- [106] G. W. NEAT, J. F. O'BRIEN, N. M. NERHEIM, AND R. J. CALVET, *Micro-Precision interferometer testbed: First stabilized fringes*, in Proceedings of the SPIE International Symposium on AeroSense, Conference on Spaceborne Interferometry II, vol. 2477, Orlando, FL, April 1995, pp. 104–115.
- [107] R. B. NELSON, *Simplified calculations of eigenvector derivatives*, AIAA Journal, 14 (1976), pp. 1201–1205.
- [108] D. E. NEWLAND, *An Introduction to Random Vibrations, Spectral and Wavelet Analysis*, John Wiley & Sons, Inc., 1993.
- [109] J. F. O'BRIEN AND G. W. NEAT, *Micro-Precision interferometer: Pointing control system*, in Proceedings of the 4th Conference on Control Applications, Albany, NY, September 1995, pp. 464–469.
- [110] D. OF ATMOSPHERIC SCIENCES, *Ww2010 (the weather world 2010 project)*. University

of Illinois at Urbana-Champaign, March 2000.

- [111] K. OGATA, *Modern Control Engineering*, Prentice-Hall, Inc., 1990.
- [112] M. O'NEAL, D. ELDRED, D. LIU, AND D. REDDING, *Experimental verification of nanometer level optical pathlength control on a flexible structure*, in Proceedings of the 14<sup>th</sup> Annual Rocky Mountain Guidance and Control Conference, Keystone, CO, February 2–6, 1991, pp. 231–249.
- [113] M. C. O'NEAL AND J. T. SPANOS, *Optical pathlength control in the nanometer regime on the JPL phase B interferometer testbed*, in Proceedings of the SPIE Conference on Active and Adaptive Optical Systems, vol. 1542, San Diego, CA, July 1991, pp. 359–370.
- [114] *OSI: Orbiting Stellar Interferometer*, Jet Propulsion Laboratory, March 1996. Report for NASA Science and Technical Review.
- [115] S. L. PADULA, B. B. JAMES, G. P. C., AND S. E. WOODARD, *Multidisciplinary optimization of controlled space structures with global sensitivity equations*, Tech. Rep. NASA TP-3130, National Aeronautics and Space Administration, November 1991.
- [116] C. G. PARK AND D. W. MILLER, *Structural design and control of high performance optical delay lines*, Master's thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, August 1995. SERC Report 9-95.
- [117] A. PARKINSON, C. SORENSEN, AND N. POURHASSAN, *A general approach for robust optimal design*, Journal of Mechanical Design, 115 (1993), pp. 74–80.
- [118] W. H. PRESS, S. A. TENKOLSKY, W. T. VETTERLING, AND B. P. FLANNERY, *Numerical Recipes in C - The Art of Scientific Computing*, Cambridge University Press, 2nd ed., 1996.
- [119] J. I. PRITCHARD, H. M. ADELMAN, AND J. SOBIESZCZANSKI-SOBIESKI, *Optimization for minimum sensitivity to uncertain parameters*, AIAA Journal, 34 (1996), pp. 1501–1504.
- [120] D. C. REDDING AND W. G. BRECKENRIDGE, *Optical modeling for dynamics and control analysis*, Journal of Guidance, Control, and Dynamics, 14 (1991), pp. 1021–1032.
- [121] J. ROARK AND W. C. YOUNG, *Roark's Formulas for Stress and Strain*, McGraw-Hill, New York, 5th ed., 1975.
- [122] H. ROGERS, *Multivariable Calculus with Vectors*, Massachusetts Institute of Technology, Prentice Hall, Upper Saddle River, New Jersey 07458, 1999.
- [123] H. G. RUTTEN AND M. A. VAN VENROOIJ, *Telescope Optics - A comprehensive Manual for Amateur Astronomers*, Willmann-Bell Inc., Richmond, Virginia, fourth ed., 1999.
- [124] Y. SAWARAGI, H. NAKAYAMA, AND T. TANINO, *Theory of Multiobjective Optimization*, vol. 176 of Mathematics in Science and Engineering, Academic Press Inc., London, United Kingdom, i ed., 1985.
- [125] S. B. SHAKLAN, J. W. YU, AND H. C. BRIGGS, *Integrated structural and optical modeling of the orbiting stellar interferometer*, in Proceedings of the SPIE Conference on Space Astronomical Telescopes and Instruments II, P. Y. Bely and J. B. Breckinridge, eds., vol. 1945, Orlando, FL, April 1993, pp. 123–132.
- [126] R. E. SKELTON, *Model error concepts in control design*, International Journal of Control, 49 (1989), pp. 1725–1753.

- [127] R. E. SKELTON, P. C. HUGHES, AND H. B. HABLANI, *Order reduction for models of space structures using modal cost analysis*, AIAA Journal of Guidance, 5 (1982), pp. 351–357. AIAA 82-4194.
- [128] M. J. SMITH, K. M. GRIGORIADIS, AND R. E. SKELTON, *Optimal mix of passive and active control in structures*, Journal of Guidance, Control, and Dynamics, 15 (1992), pp. 912–919.
- [129] *The Space Interferometry Mission: Taking the Measure of the Universe*, April 5, 1996. Final Report of the Space Interferometry Science Working Group. Available from URL <<http://www.ess.sunysb.edu/simswg/siswg/siswg.html>> (accessed November 29, 1998).
- [130] J. SPANOS, Z. RAHMAN, AND G. BLACKWOOD, *A soft 6-axis active vibration isolator*, in Proceedings of the American Control Conference, Seattle, WA, June 1995.
- [131] G. STRANG, *Introduction to Applied Mathematics*, Wellesley-Cambridge Press, 1986.
- [132] G. STRANG, *Introduction to Linear Algebra*, Wellesley-Cambridge Press, 1998.
- [133] T. R. SUTTER, C. J. CAMARDA, J. L. WALSH, AND H. M. ADELMAN, *Comparison of several methods for calculating vibration mode shape derivatives*, AIAA Journal, 26 (1988).
- [134] H. TADA, P. PARIS, AND G. IRWIN, *The stress analysis of cracks handbook*, Del Research Corporation, Hellertown, Pennsylvania, 1973.
- [135] J. TING AND E. F. CRAWLEY, *Characterization of damping of materials and structures from nanostrain levels to one thousand microstrain*, AIAA Journal, 30 (1992), pp. 1856–1863.
- [136] S. A. UEBELHART, *Conditioning, Reduction, and Disturbance Analysis of Large Order Integrated Models of Space-Based Telescopes*, Master’s thesis, Massachusetts Institute of Technology, February 2001.
- [137] J. VAN DE VEGTE, *Feedback Control Systems*, Prentice-Hall, Inc., 1990.
- [138] S. L. VENNARI, *Future directions and needs in engineering tools*, ICASE/LaRC/NSF/ARO Workshop on Computational Aerosciences in the 21st Century, (1998). Hampton, Virginia.
- [139] M. WEHNER, *Personal Communication, TRW, Redondo Beach, CA., 1999.*
- [140] K. WHITING AND E. S. KOVALCIK, *AIAA Aerospace Engineers Guide*, American Institute of Aeronautics and Astronautics Inc., 370 L’Enfant Promenade, SW, Washington DC, 20024-2518, 3rd ed., March 1993.
- [141] P. H. WIRSCHING, T. L. PAEZ, AND H. ORTIZ, *Random Vibrations: Theory and Practice*, John Wiley & Sons, Inc., 1995.
- [142] K. Y.-L. YANG, *Efficient Design of Robust Controllers for  $H_2$  Performance*, PhD thesis, Massachusetts Institute of Technology, October 1996. SERC Report #8-96.
- [143] K. ZHOU, J. C. DOYLE, AND K. GLOVER, *Robust and Optimal Control*, Prentice-Hall, Inc., 1996.