Multivariable Isoperformance Methodology for Precision Opto-Mechanical Systems

Olivier L. de Weck and David W. Miller

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 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 isoperformance approach attempts to avoid situations, where very difficult requirements are levied onto one subsystem, while other subsystems hold substantial margins. An experimental validation is carried out on the DOLCE laboratory testbed trading disturbance excitation amplitude and payload mass. 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. 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.

Keywords— **Isoperformance, Multidisciplinary Design Optimization (MDO), Dynamics and Controls, Contour Mapping, Spacecraft Design, Optics, Sensitivity Analysis**

I. 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.

Thesis Committee: Prof. David Miller, Prof. Edward Crawley, Prof. Eric Feron, Dr. Marthinus van Schoor.

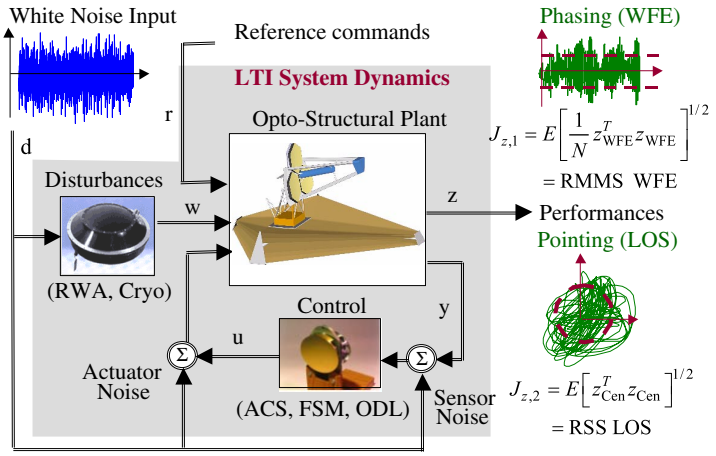

Fig. 1. Block diagram of science target observation mode of a space telescope. Inputs are white-noise unit-intensity disturbances d and reference commands r. Outputs are opto-scientific metrics of interest z. The performances, J_z, are typically expressed in terms of the root-mean-square (RMS) of the outputs.

This framework is first developed generically for LTI systems, which can be described in state space form. It is then applied specifically to dynamics and controls problems of precision opto-mechanical systems, such as the next generation of space-based observatories. These systems combine structures, optics and control systems such that stringent pointing and phasing requirements can be met in the presence of dynamic disturbance sources. The typical problem setting is depicted in Figure 1.

The goal of a disturbance analysis (= performance assessment) is to predict the expected values of the performances, $J_{z,i}$, where $i = 1, \ldots, n_z$ and n_z is the number of performance metrics. This has been previously developed and demonstrated by Gutierrez [7]. 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, $J_{z,req,i}$, including margins, a sensitivity analysis can provide partial derivatives $\partial J_{z,i}/\partial p_j$ which are used to identify in which direction important parameters p_j should be

changed. This is intended to drive the system to a design point which satisfies all requirements, i.e. a condition where $J_{z,i} \leq J_{z,req,i}$ is true for all i. This is as far as most existing tools and methodologies will go in the design process.

Once a nominal design, p_{nom}, 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 system parameters p_j will yield the same predicted system performance $J_{z,i}$. It is the essential idea of isoperformance to find and exploit these performance invariant solutions, p_{iso}, in the design space. A formal process and specific tools are needed, which will ensure that a required performance level is met, while minimizing the cost and risk of the system. This is the impetus for the following formal thesis problem formulation.

A. Thesis Problem Definition

The primary 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 $i = 1, \ldots, n_z$, attempt to find a set of independent solution vectors, $p_{iso} = [p_1, p_2, \ldots, p_{n_p}]$, whose elements are the variable parameters p_j, such that an efficient system design can be achieved. This can be formulated mathematically as follows.

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

$$\begin{aligned} \dot{q} &= A_{zd}(p_j)\, q + B_{zd}(p_j)\, d + B_{zr}(p_j)\, r \\ z &= C_{zd}(p_j)\, q + D_{zd}(p_j)\, d + D_{zr}(p_j)\, r \end{aligned} \tag{1}$$

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

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

where $i = 1, 2, ..., n_z$, are a definition of the performance metrics of interest, find a set of vectors, p_{iso}, such that the performance equality (isoperformance) constraint

$$J_{z,i}(p_{iso}) \equiv J_{z,req,i} \ \forall\ i = 1, 2, ..., n_z \tag{3}$$

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

$$n_p - n_z \geq 1 \tag{4}$$

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

$$p_{j,LB} \leq p_j \leq p_{j,UB} \ \forall\ j = 1, 2, ..., n_p \tag{5}$$

The isoperformance condition (3) has to be met subject to a numerical tolerance, τ

$$\left| \frac{J_z(p_{iso}) - J_{z,req}}{J_{z,req}} \right| \leq \frac{\tau}{100} \tag{6}$$

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

$$\begin{aligned} &\texttt{NLP} \\ &\min \left[\eta J_c^T Q_{cc} J_c + (1-\eta)\, J_r^T Q_{rr} J_r\right] \\ &\text{such that } p_{iso} \in \mathbf{I} \text{ and } p_{j,LB} \leq p_j \leq p_{j,UB} \\ &\text{and } \eta \in [0\ 1] \end{aligned} \tag{7}$$

where the weight η 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 (3).

Alternatively this can be formulated in terms of set theory. Figure 2 shows various sets in the vector space $p = \left[p_1\ p_2\ \ldots\ p_{n_p}\right]^T$ and their mutual relationship in the general case[1].

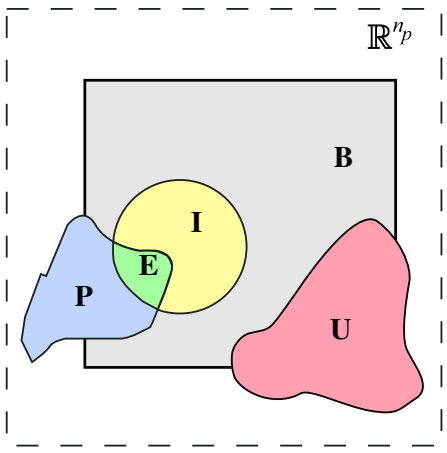

Fig. 2. Sets for thesis problem definition.

set	description
$\mathbb{R}^{n_p}$	n_p-dimensional **R**eal valued Euclidean vector space
$\mathbf{B} \subset \mathbb{R}^{n_p}$	subset of $\mathbb{R}^{n_p}$, which is **B**ounded by (5)
$\mathbf{I} \subset \mathbf{B}$	subset of **B**, which satisfies **I**soperformance, see (3),(6)
$\mathbf{U} \subset \mathbb{R}^{n_p}$	**U**nstable subspace, where $\max(\mathrm{Re}(\lambda_i)) > 0$
$\mathbf{P} \subset \mathbb{R}^{n_p}$	**P**areto optimal subset, satisfies (7) without constr.
$\mathbf{E} = \mathbf{I} \cap \mathbf{P}$	**E**fficient subset, satisfies (7) with constraints

[1] The eigenvalues λ_i are obtained by solving the eigenvalue problem $[A_{zd} - \lambda_i I]\phi_i = 0$.

The first task is to find the elements of the isoperformance set $\mathbf{I}$ in $\mathbf{B}$. Since the performance requirements are bounded, i.e. $|J_{z,req,i}| < \infty \; \forall \; i$, it is true that the intersection $\mathbf{U} \cap \mathbf{I} = \varnothing$. 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 $\overline{\mathbf{U}} = \{x | x \notin \mathbf{U}\}$. The ultimate goal is to find a family of designs p^*_{iso}, 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. $\mathbf{E} = \mathbf{I} \cap \mathbf{P}$.

Limitations of thesis scope: The disturbance sources d are assumed to be zero mean: $\mu_d = E[d] = 0$, while $E\left[d^T d\right] \neq 0$. Furthermore this thesis focuses on the science target observation mode, thus representing a steady state condition. 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 (5). The scope is limited to continuous time problems (s-domain). 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. Some system parameters need to be fixed ahead of time to make the problem formulation tractable for realistic systems.

B. Previous Work

The allocation of design requirements and resources (costs) as well as an assessment of risk during early stages of a program is based on preliminary analyses using simplified models that try to capture the behavior of interest [2]. 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 [7]. The $\mathcal{H}_2$-type performances used here are defined in accordance with Zhou, Doyle and Glover [21].

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 [5] for example presented a methodology for optimally selecting dampers for multidegree of freedom systems [5]. In that particular work (Fig.5) the contours of equal values of the objective function[2] 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 [16], [17], [18] on the need within the U.S. Department of Defense to improve 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. A systematic approach to isoperformance in complex, opto-mechanical systems such as the next generation of space observatories however is lacking at this time.

A relevant field that has received a lot of attention in recent years is **integrated modeling**. This encompasses efforts to simulate complex systems in a unified and multidisciplinary environment. 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) [8]. 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 [10]. The IMOS package is used extensively throughout this thesis for the generation and manipulation of finite element models.

The application of isoperformance to multiobjective design optimization draws on previous research results in **multidisciplinary design optimization**. A fundamental book on the theory of multiobjective optimization was published by Sawaragi, Nakayama and Tanino [20]. An important application of multiobjective optimization is concurrent control/structure optimization. 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., [13] 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 [12]. A good overview of structural and multidisciplinary optimization research is given in the volume "Structural Optimization: Status and Promise" edited by Kamat [9], with significant contributions by Haftka, Venkayya, Sobieszczanski-Sobieski and others.

C. Approach and Thesis Roadmap

A thesis roadmap is shown in Figures 3 and 4. The flow diagram in Figure 3 comprises the **development** of the isoperformance methodology and its implementation. The dashed box comprises essentially the performance assessment and enhancement framework developed by Gutierrez [7]. The analysis process starts with a given integrated model of the system of interest, which is populated by an initial design vector p^o. The performance assessment calculates the performance vector J_z^k and compares it to the requirements $J_{z,req}$. If the inequality $|\Delta J_z^k / J_z^k| < \tau$, where $\Delta J_z^k = J_z^k - J_{z,req}$, is met, we have found a solution that satisfies the isoperformance condition. We will call this solution the nominal design p_{nom}. If the relative error is larger than τ we perform a sensitivity analysis, which yields the gradient vector (Jacobian) ∇J_z^k at the k-th iteration.

[2] The objective function in reference [5] is called ITSE = integral of time multiplied by the sum of squares of displacements and velocities of the masses.

This is used in a gradient search 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 $n_z = 1$ (Section III). The generalization to the multivariable case with $n_p > 2$ is the topic of Section IV. The main result from the isoperformance analysis is a set of points, p_{iso}, which approximates the isoperformance set $\mathbf{I}$ in $\mathbb{R}^{n_p}$. If this set is empty it means that the algorithm was not able to detect elements in the isoperformance set. The recommended procedure is then 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 $J_{z,req}$.

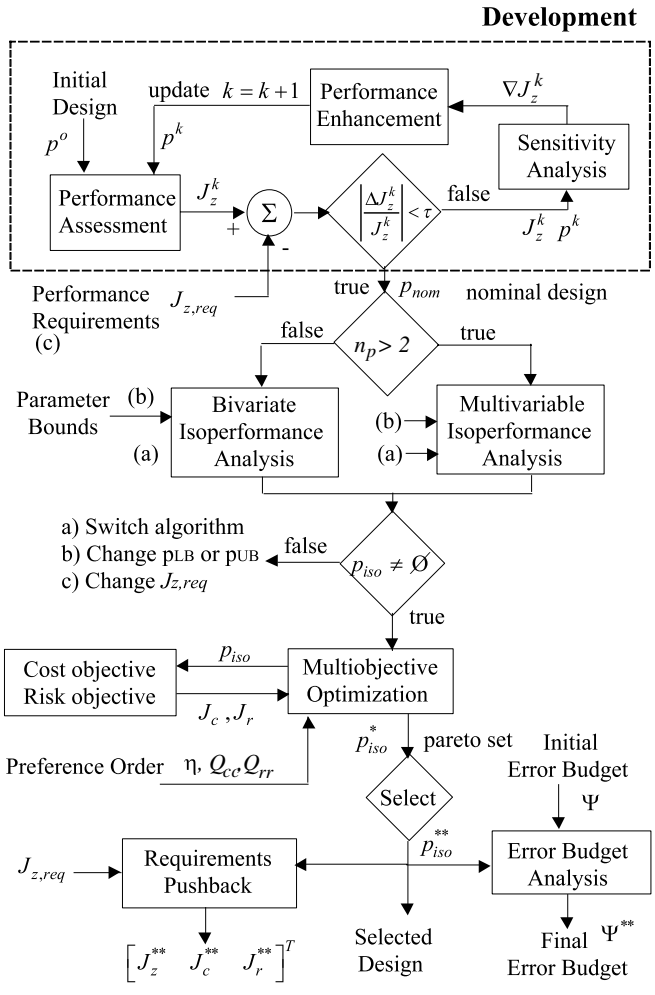

Fig. 3. Thesis Roadmap: Development

If an isoperformance solution was found the methodology proceeds to the multiobjective optimization step as described in Section VII. The solutions in the isoperformance set, p_{iso}, are evaluated for the cost objective function J_c and the risk objective function J_r. Note that a preference order can be formulated, since often multiple, possibly conflicting objectives exist. The solution is not a single "optimal" point design, but rather a family of pareto optimal designs p^*_{iso}, which make up the "efficient" set $\mathbf{E}$. 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 a requirements pushback analysis, which repeats a performance assessment and uncertainty analysis to verify that indeed all performance requirements $J_{z,req}$ are met with sufficient margins, while taking into account a known or assumed uncertainty v_i of the design parameters. The resulting vectors J_z^{**}, J_c^{**} and J_r^{**} are returned giving the performance, cost and risk of the selected design.

A second application of the isoperformance methodology is dynamics error budgeting. An initial error budget Ψ is usually established. The (l,i)-entry of the matrix Ψ is $\Psi_{l,i}$ and represents the fractional contribution of the l-th disturbance source to the i-th performance metric. The module returns a final error budget Ψ^{**} 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.

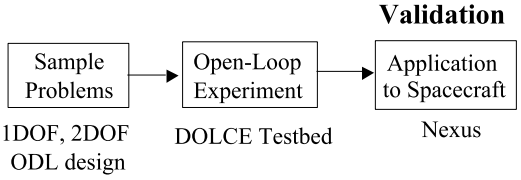

Fig. 4. Thesis Roadmap: Validation

Figure 4 contains the sequential steps used for the **validation** of the isoperformance methodology. In Section II we introduce three sample problems of increasing complexity. These help in gaining intuitive understanding and confidence in the correct implementation of the governing equations. An experimental investigation is presented in Section VI. The experiment uses the DOLCE testbed 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 contours for combinations of system parameters such as payload mass and disturbance excitation amplitude. Once confidence has been obtained that the methodology can yield useful results on physical systems it is applied to an actual spacecraft model. The NGST precursor mission NEXUS was chosen for an in-depth analysis including performance, sensitivity, uncertainty and isoperformance analyses (Section VII). Thesis contributions and recommendations for future work can be found in Section VIII.

II. Sample Problems

Three sample problems are used in the thesis: (1) Single degree-of-freedom oscillator, (2) Two degree-of-freedom oscillator and (3) ODL design problem. In this summary only the SDOF problem is discussed for brevity.

A. Single Degree-of-Freedom Example

Figure 5 shows a schematic representation of the single degree-of-freedom 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 ω_d [rad/sec].

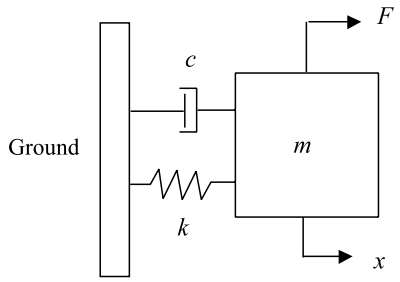

Fig. 5. Schematic of single degree-of-freedom (SDOF) oscillator.

The displacement x [m] of the mass is passed through a first order highpass filter (HPF) with corner frequency ω_o [rad/sec], simulating the effect of an optical controller. The resulting output z [m] is used to compute the performance. The performance is the RMS of z, specifically $J_z = (E[z^T z])^{1/2}$, where E[ ] denotes the expectation operator [1]. This system is shown in the block diagram of Figure 6.

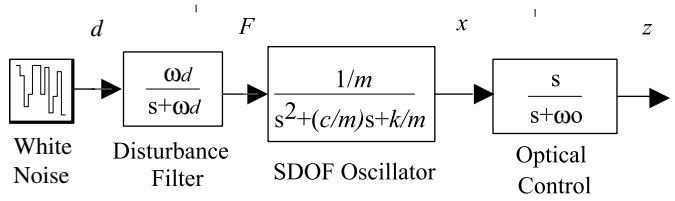

Fig. 6. SDOF block diagram. From left: white noise disturbance source, disturbance LPF, oscillator and optical control HPF.

The goal is to understand how this performance, J_z, depends on the variable design parameters, i.e. $p_i \mapsto J_z(p_i)$ for $i = 1, 2, ..., 5$, where $p = [\omega_d \; m \; k \; c \; \omega_o]^T$. Isoperformance results for this problem are presented in the next two sections.

III. Bivariate Isoperformance Methodology

This section 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 and compared. We want to find a set of solutions, p_{iso}, which satisfies the isoperformance condition (3).

A. Algorithm I: Exhaustive Search

This method discretizes the parameter space, defined by the upper and lower 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 dependent on the smoothness of $J_z(p_j)$, which is not known apriori. 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 $p_{k,l} = [\; p_{1,k} \quad p_{2,l} \;]^T$. 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[3]:

$$n_1 = \left\lceil \frac{p_{1,UB} - p_{1,LB}}{\Delta p_1} \right\rceil \text{ and } n_2 = \left\lceil \frac{p_{2,UB} - p_{2,LB}}{\Delta p_2} \right\rceil \quad (8)$$

The index k on the first parameter runs from 1 to $n_1 + 1$, the index l runs from 1 to $n_2 + 1$[4]. Thus a total number of $(n_1 + 1) \times (n_2 + 1)$ combinations has to be evaluated. This is algorithmically 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 **B** discretization is shown in Figure 7.

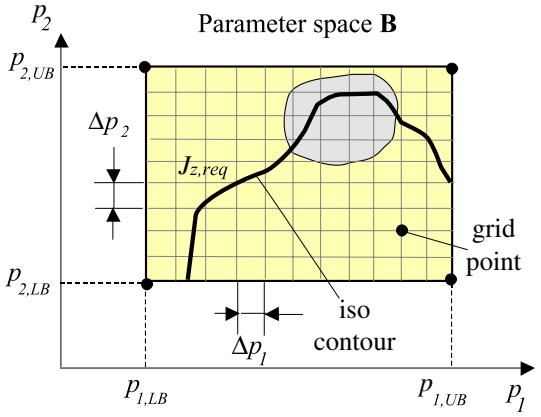

Fig. 7. Algorithm I: Discretization of **B** in a linearly spaced grid with increments $\Delta p = [\Delta p_1, \Delta p_2]^T$.

Note that the result of a particular parameter combination $p_{k,l}$ does not affect the computation of the next point. Once all the parameter combinations $p_{k,l}$ have been evaluated, linear interpolation between neighboring grid points

[3] The $\lceil \; \rceil$ operator denotes the ceiling function.
[4] If $k = n_1 + 1$ then $p_{1,k} = p_{1,UB}$ and if $l = n_2 + 1$ then $p_{2,l} = p_{2,UB}$.

is used to find isoperformance points $p_{iso,r}$. The linear interpolation algorithm uses the following equation to find the r-th isoperformance point:

$$p_{iso,r} = \begin{bmatrix} p_{1,k} \\ p_{2,l} \end{bmatrix} + \frac{(J_z)_{k,l} - J_{z,req}}{(J_z)_{k,l} - (J_z)_{m,n}} \cdot \begin{bmatrix} p_{1,m} - p_{1,k} \\ p_{2,n} - p_{2,l} \end{bmatrix} \quad (9)$$

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 grid point $p_{k,l}$ itself. 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.

B. Algorithm II: Gradient-based Contour Following

The basic idea of gradient-based contour following 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 computed by means of the gradient vector $\nabla J_z(p_1, p_2)$. Thus, a prerequisite is that $J_z(p_j)$ be continuous and differentiable at all points in the parameter space $p = [p_1, p_2]^T \in \mathbf{B}$. The desired step direction is colinear with the tangent vector t_k to the isoperformance contour. The derivation starts by considering the bivariate function

$$p_1, p_2 \mapsto J_z(p_1, p_2) \text{ , where } \mathbb{R}^2 \mapsto \mathbb{R} \text{ and } p_j \in \mathbf{B} \quad (10)$$

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:

$$\begin{aligned} &J_z\,(p) = J_z\,(p_{nom}) + \left(\nabla J_z\right)^T\Big|_{p_{nom}} \cdot \Delta p + \\ &\tfrac{1}{2}\Delta p^T\ H\big|_{p_{nom}}\,\Delta p + \text{H.O.T.} \end{aligned} \quad (11)$$

Note that $p = p_{nom} + \Delta p$ and that ∇J_z and H are the gradient vector and Hessian matrix, respectively. The parameter vector increment, Δp, can be written as the product of a step size, α, and a step direction (vector), d. Note that d is normalized to unit length

$$\Delta p = \alpha \cdot d \quad (12)$$

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 [4] is used to obtain a first isopoint $p_{iso,1}$ on the isoperformance contour. 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

$$J_z(p_o + \alpha \cdot d) < J_z(p_o) \quad (13)$$

for all sufficiently small positive values of α. The step size α 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 J_z value). Conversely if the initial guess p_o or any subsequent iterate is "below" the isoperformance level, α 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. Thus, one can write the first order approximation at the point p_o as:

$$J_z(p_o + \alpha_o \cdot n_o) \cong J_z(p_o) + \nabla J_z(p_o)^T \cdot \alpha_o n_o \quad (14)$$

Recall from the Cauchy-Schwartz inequality that

$$J_z + \nabla J_z^T \left(\frac{-\nabla J_z}{\|\nabla J_z\|}\right) \leq J_z + \nabla J_z^T \left(\frac{d}{\|d\|}\right) \quad (15)$$

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

$$n_o = \left(\frac{-\nabla J_z\,(p_o)}{\|\nabla J_z\,(p_o)\|}\right) \quad (16)$$

The step size, α_o, is found by assuming linearity from the initial guess p_o to the first point on the isoperformance contour $p_{iso,1}$. From the expression

$$J_z\,(p_o + \alpha_o d_o) \cong J_z\,(p_o) + \nabla J_z^T \cdot \alpha_o n_o \equiv J_{z,req} \quad (17)$$

one can solve for α_o , such that

$$\alpha_o = \left(\frac{-\nabla J_z\,(p_o)^T\,\nabla J_z\,(p_o)}{\|\nabla J_z\,(p_o)\|}\right)^{-1} \cdot \left(J_{z,req} - J_z\,(p_o)\right) \quad (18)$$

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 $\mathbf{B}$, at a point $p_{iso,1}$ 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 $\mathbf{B}$.

Given that $J_z(p_{iso,k}) = J_{z,req}$, i.e. 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 (11), neglecting second-order and higher terms and setting the first order term (perturbation) to zero. Specifically, if

$$\begin{aligned} &J_z\,(p_{iso,k+1}) = J_z\,(p_{iso,k} + \Delta p_k) \cong \\ &J_z\,(p_{iso,k}) + \left(\nabla J_z\right)^T\Big|_{p_{iso,k}} \Delta p_k \equiv J_{z,req} \end{aligned} \quad (19)$$

is to be true, then

$$\Delta J_{z,k} = \left(\nabla J_z\right)^T\Big|_{p_{iso,k}} \Delta p_k \equiv 0 \quad (20)$$

In other words, one must choose the vector Δp_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

$$\Delta J_{z,k} = \left.\frac{\partial J_z}{\partial p_1}\right|_{p_{1,k}} \Delta p_{1,k} + \left.\frac{\partial J_z}{\partial p_2}\right|_{p_{2,k}} \Delta p_{2,k} \equiv 0 \qquad (21)$$

Geometrically this condition corresponds to following the tangential vector t_k along the isocontour. Figure 8 shows that t_k can be considered the tangential vector at point $p_{iso,k}$ and that it is orthogonal to the normal vector n_k. There are two ways in which t_k can be obtained from

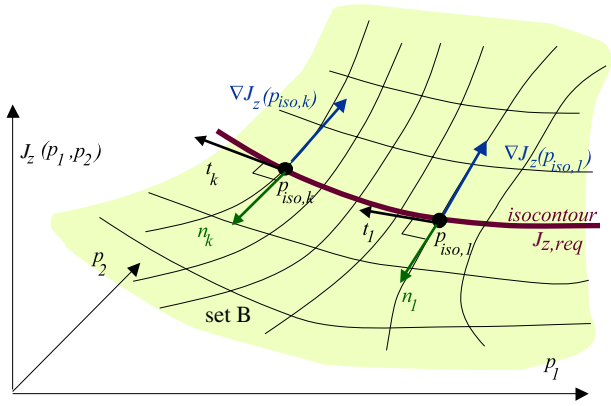

Fig. 8. Algorithm II: Depiction of gradient vector ∇J_z, normal vector n and tangential vector t along the isoperformance contour.

$\nabla J_z(p_k)$. First one can compute the normal vector n_k from equation (16) and then rotate it by 90 degrees to obtain the tangential vector t_k.

$$t_k = \mathcal{R} \cdot n_k = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \cdot n_k \qquad (22)$$

The second method is more general, since it is also applicable to the case of $n_z > 1$ performances and $n_p > 2$ parameters. A singular value decomposition (SVD) [19] is performed on the transpose of the gradient vector.

$$U_k S_k V_k^T = \nabla J_k^T \qquad (23)$$

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 matrix $V_k = [n_k \; t_k]$. An appropriate step size α_k needs to be chosen. An estimate of the linearization error incurred due to a step of size Δp_k can be written as:

$$\epsilon_k = \frac{1}{2} \Delta p_k^T \; H|_{p_k} \Delta p_k + \text{H.O.T.} \qquad (24)$$

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

$$\alpha_k = \left[\frac{2\tau J_{z,req}}{100} \left(t_k^T \cdot H|_{p_k} \cdot t_k \right)^{-1} \right]^{1/2} \qquad (25)$$

The quantity τ is a user defined tolerance and is defined as the $\pm$ % acceptable deviation from the nominal "centerline", $J_{z,req}$.

With equations (22) and (25) the step direction t_k and the step size α_k have been determined and one can find the next point on the isoperformance contour $p_{iso,k+1} = p_{iso,k} + \alpha_k t_k$. 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 **B** are reached, the solution reaches the unstable subspace **U** or the isoperformance contour closes on itself.

C. Algorithm III: Progressive Spline Approximation

The progressive spline approximation algorithm assumes that the isoperformance contour intersects the boundary **B**, i.e. that 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. 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 de Boor [3] as well as the resulting MATLAB spline toolbox are leveraged for this algorithm.

A mathematical description of a spline, $P_l(x)$, is given in terms of its break points (`breaks`) $\zeta_1, \ldots, \zeta_{l+1}$ and the local polynomial coefficients $c_{l,i}$ of its `pieces`.

$$P_l(x) = \sum_{i=1}^{k} \frac{(x - \zeta_j)^{k-i}}{(k-i)!} c_{l,i} \qquad (26)$$

This form (`ppform`) 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 **B**. The initial estimate of the isoperformance contour consists of a single piece. The isoperformance contours are parameterized with parameter t from endpoint a to endpoint b. 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 (26) the algorithm works with *vector* splines such that

$$P_l(t) = \begin{bmatrix} p_{iso,1}(t) \\ p_{iso,2}(t) \end{bmatrix} = \begin{bmatrix} s_1(t) \\ s_2(t) \end{bmatrix} = p_{iso}(t) \qquad (27)$$

where

$$t \in [0,1] \mapsto P_l(t) \in [a,b] \qquad (28)$$

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

$$s_j(t) = f_{j,l}(t) \;\text{ for } j = 1,2 \text{ and } \;\forall\; l \qquad (29)$$

The functional approximation for each piece is then given as

$$f_{j,l}(t) = \sum_{i=1}^{k} \frac{(t - \zeta_l)^{k-i}}{(k-i)!} c_{j,l,i} \text{ where } \; t \in [\zeta_l \ldots \zeta_{l+1}] \qquad (30)$$

Note that *all* relevant information is contained in the break point sequence, $\zeta_1 \ldots \zeta_{l+1}$ 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.

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 ζ_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 τ as defined above. This is graphically shown in Figure 9 for the single degree-of-freedom example introduced in Section II.

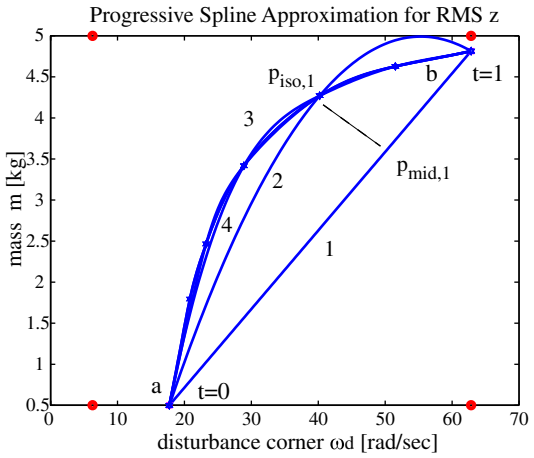

Fig. 9. Progressive (cubic) spline approximation. Isoperformance analysis of SDOF problem with variables ω_d and m. The required performance is $J_{z,req} = 0.0008$ [m].

D. 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 provide guidance for applications to larger problems and the multivariable case. We choose the disturbance corner frequency, ω_d, and oscillator mass, m, as the variable parameters in order to find the isoperformance contour at the $J_z = 0.8$ [mm] level.

D.1 Quality of Isoperformance Solution

In order to assess how well the resulting isoperformance points, p_{iso}, actually meet the isoperformance condition (3) it is necessary to define a solution "quality" metric. The "quality" of the isoperformance solution can be quantified as follows. Let

$$\Upsilon_{iso} = \frac{100}{J_{z,req}} \cdot \left[\frac{\sum_{k=1}^{n_{iso}} \left[J_z(p_{iso,k}) - J_{z,req} \right]^2}{n_{iso}} \right]^{1/2} \tag{31}$$

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, Υ_{iso}, can then be directly compared to the desired isoperformance contour tolerance, τ, 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 $[(1-\tau/100) \cdot J_{z,req}, (1+\tau/100) \cdot J_{z,req}]$.

D.2 Algorithm Comparison

The isoperformance results for exhaustive search are shown in Figure 10. The isoperformance curve shows that a small increase in the disturbance filter corner frequency ω_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.

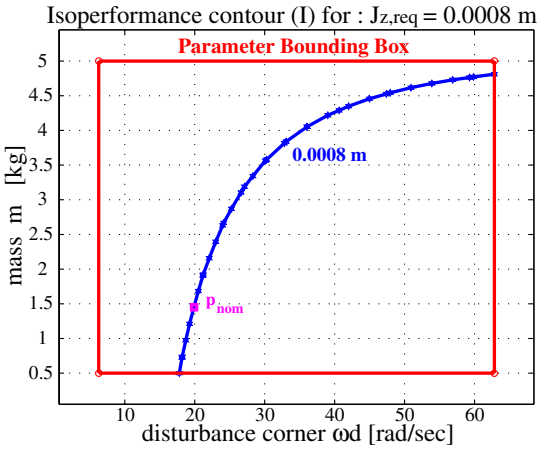

Fig. 10. Algorithm I (Exhaustive Search): Isoperformance contour for single DOF problem (ω_d, m) with discretization $\Delta p = (1/20)[p_{UB} - p_{LB}]$ and a tolerance of $\tau = 1\%$.

The quality of the isoperformance contour is very dependent on the discretization level. The smaller Δp, 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 11.

The isoperformance contours obtained with contour following (not shown) and progressive spline approximation (Fig. 9) are very similar. A comparison of the computational cost among algorithms is shown in Table I. In

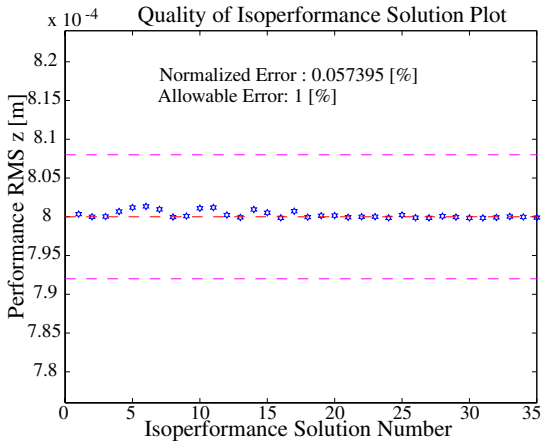

Fig. 11. Quality: Contour solution quality according to (31).

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 Υ_{iso} metric. Algorithm I is the most computationally expensive. This

TABLE I
COMPARISON OF ALGORITHMS I-III FOR SINGLE DOF PROBLEM.

Result	Ex Search	Co Follow	Sp Approx
FLOPS	2,140,897	783,761	377,196
CPU [sec]	1.15	0.55	0.33
Tolerance: τ	1.0 %	1.0 %	1.0 %
Error: Υ_{iso}	0.057 %	0.379 %	0.087 %
isopoints	35	41	7

is due to the fact that in the SDOF case 441 points had to be evaluated, but only 35 points form the isoperformance contour. Algorithm III (progressive spline approximation) is clearly the fastest, however it only works for open segments and assumes that there is only a single isoperformance contour, which intersects the boundary $\mathbf{B}$. Thus, it is the most restrictive (least general) of the three algorithms. 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 several random trial points $p_{nom,i}$, where $i = 1, 2, \ldots,$ #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, α_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. 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 performance and gradient calculations are presented in Section V. The generalization of these algorithms to the multivariable case is discussed in the next section.

IV. Multivariable Isoperformance Methodology

This section generalizes the algorithms developed in the previous section to the multivariable case. This generalization is essential in order to render isoperformance a useful technique for realistic problems. 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. There are two main challenges in the multivariable case:

- Computational complexity as a function of n_p and n_z
- Visualization of isoperformance set $\mathbf{I}$ in n_p-space

A. Branch and Bound Search Algorithm (Ib)

The exhaustive search algorithm (Ia) in the multivariable case ($n_p > 2$) discretizes the parameter set $\mathbf{B}$, defined by the lower and upper bounds $p_{LB,j}$ and $p_{UB,j}$, where $j = 1, 2, ..., n_p$, with a fine grid and evaluates all grid points. This was presented for the case, when $n_p = 2$ in subsection III-A. Subsequently each grid point is tested, and if the isoperformance condition (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. Note that the value of the j-th parameter in these loops is given as

$$p_{j,i_j} = p_{j,LB} + (i_j - 1) \cdot \Delta p_j \text{ where } \quad j = 1, 2, \ldots, n_p \quad (32)$$

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

A remedy is found by modifying exhaustive search as a branch-and-bound algorithm (Ia). The branch-and-bound algorithm starts with an initial population (branches), which are evenly but coarsely 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:

$$[J_z(p_m) \geq J_{z,req} \geq J_z(p_n)] \cup [J_z(p_m) \leq J_{z,req} \leq J_z(p_n)] \quad (33)$$

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 12 for two dimensions.

In the multivariable case the squares shown in Figure 12 are actually hyper-rectangles. The size of the hyper-rectangles is reduced by a factor of two along edges with each generation. This refinement continues with each generation, n_g, until the exit criterion

$$\Upsilon_{iso,n_g} < \tau \quad (34)$$

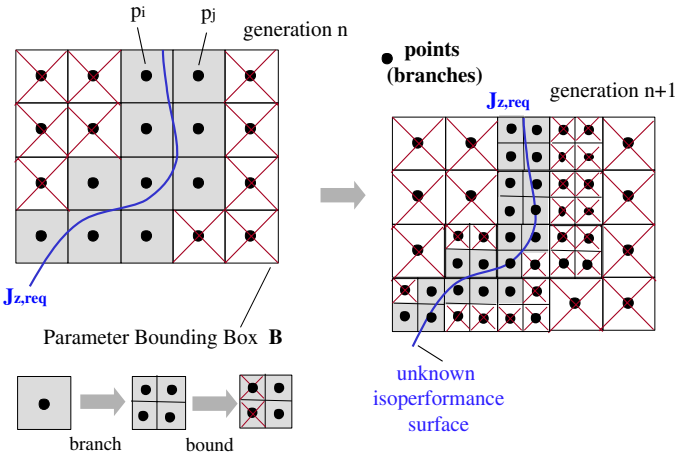

Fig. 12. Multivariable Isoperformance (Ib): Branch-and-Bound graphic representation. Crossed out points (branches) are dropped in the next generation.

is met.

It was empirically found 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.

B. Tangential Front Following Algorithm

In the multivariable case there will be n_z performance metrics and n_p parameters, where $n_p - n_z \geq 1$. A first order Taylor approximation of the vector performance function J_z at a point $p^k = [p_1^k \; p_2^k \; \ldots p_{n_p}^k]^T \in \mathbf{B}$ can be written as:

$$J_z\left(p^{k+1}\right) = J_z\left(p^k + \Delta p\right) = J_z\left(p^k\right) + \nabla J_z^T\big|_{p^k} \Delta p + \text{HOT} \tag{35}$$

The Jacobian, ∇J_z, is the matrix of first order partial derivatives of J_z with respect to p:

$$\nabla J_z = \begin{bmatrix} \frac{\partial J_{z,1}}{\partial p_1} & \frac{\partial J_{z,2}}{\partial p_1} & \cdots & \frac{\partial J_{z,n_z}}{\partial p_1} \\ \frac{\partial J_{z,1}}{\partial p_2} & \frac{\partial J_{z,2}}{\partial p_2} & \cdots & \frac{\partial J_{z,n_z}}{\partial p_2} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial J_{z,1}}{\partial p_{n_p}} & \frac{\partial J_{z,2}}{\partial p_{n_p}} & \cdots & \frac{\partial J_{z,n_z}}{\partial p_{n_p}} \end{bmatrix} \tag{36}$$

The 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, respectively.

$$U \Sigma V^T = \nabla J_z^T \tag{37}$$

and the individual matrices are as follows:

$$\begin{aligned} U &= \underbrace{\begin{bmatrix} u_1 & \cdots & u_{n_z} \end{bmatrix}}_{n_z \times n_z} \\ \Sigma &= \underbrace{\begin{bmatrix} \text{diag}\begin{pmatrix} \sigma_1 & \cdots & \sigma_{n_z} \end{pmatrix} & 0_{n_z \times (n_p - n_z)} \end{bmatrix}}_{n_z \times n_p} \\ V &= \begin{bmatrix} \underbrace{v_1 \; \cdots \; v_{n_z}}_{\text{column space}} & \underbrace{v_{n_z+1} \; \cdots \; v_{n_p}}_{\text{null space}} \end{bmatrix} \end{aligned} \tag{38}$$

Thus, at each point there are $n_p - n_z$ directions in the null space. It is a linear combination of the vectors in the null space, V_t, which is used to determine a tangential step, Δp, in a performance invariant direction.

$$\Delta p = \alpha \cdot \left(\beta_1 v_{n_{z+1}} + \ldots + \beta_{n_p - n_z} v_{n_p}\right) = \alpha V_t \beta \tag{39}$$

where Δp is the performance invariant step increment in $\mathbb{R}^{n_p}$, β is a vector of coefficients, which determines the linear combination of directions in the nullspace, V_t, and α is a step size. Currently, in the multivariable case the step size, α, is set by the user. Values in the range from 0.1-2.0 were empirically found to give satisfactory results. An automatic step size determination could be added as a refinement in the future. The coefficient vector, β, is determined as follows

$$\beta = \begin{cases} \beta_i = \pm 1, \beta_j = 0 \text{ for } i \neq j \\ \beta_i = \pm \dfrac{1}{\sqrt{n_p - n_z}} \;\; \forall \quad i = 1, \ldots, n_p - n_z \end{cases} \tag{40}$$

The principal front points, as shown in Figure 13, 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 β_i determines in which "quadrant" the front point propagates.

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 III-B. *The idea is to gradually explore the isoperformance set **I**, starting from a random initial point, p_{nom}, and subsequently stepping in tangential, orthogonal directions, v_j, where $j = n_z + 1, \ldots, n_p$, which lie in the null space of the Jacobian.* 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 similar to "moss", which grows from an initial seed to gradually cover the entire exposed surface of an imaginary n_p-dimensional rock. This is shown graphically in Figure 13.

The main advantage of this algorithm, is that it converts the computational complexity from a n_p to a $n_p - n_z$ problem, albeit still in non-polynomial time. 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 $\mathbf{B}$. The underlying performance function $J_z(p)$ has to be continuous and differentiable over the entire set $\mathbf{B}$.

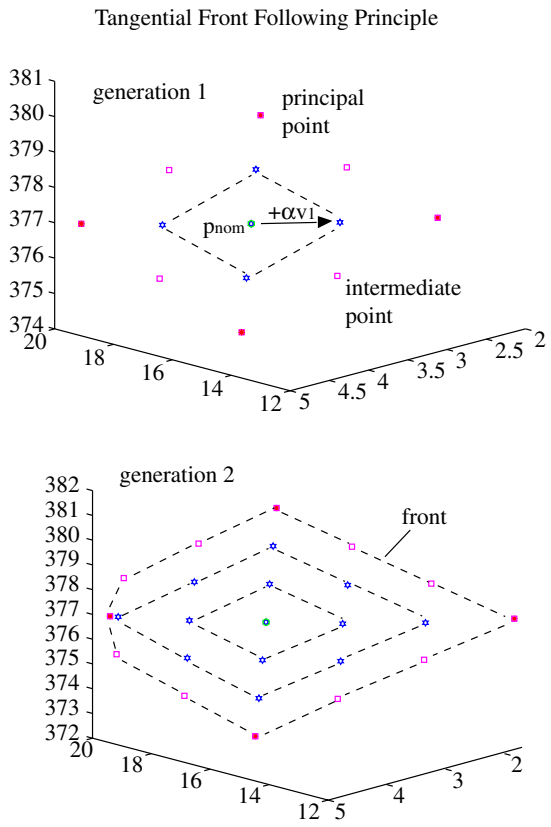

Fig. 13. Tangential Front Following (II) principle.

C. Vector Spline Approximation

Even though the tangential front following algorithm is more efficient than branch-and-bound, it will still be computationally expensive if $n_p - n_z$, is large. An estimate of the computational expense of each algorithm is given in subsection IV-E. Hence it is desirable to find an algorithm with a further significant increase in efficiency. Such an algorithm is constructed by generalizing the bivariate progressive spline approximation. *The basic idea of vector spline approximation is to only capture important border and interior points of the isoperformance set* $\boldsymbol{I}$. A t-parameterized vector spline in n_p-dimensional space connecting two points A and B can be written as

$$p(t) = \begin{bmatrix} p_1(t) \\ p_j(t) \\ \vdots \\ p_{n_p}(t) \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{k} \frac{(t-t_A)^{k-i}}{(k-i)!} \cdot c_{1,i} \\ \sum_{i=1}^{k} \frac{(t-t_A)^{k-i}}{(k-i)!} \cdot c_{j,i} \\ \vdots \\ \sum_{i=1}^{k} \frac{(t-t_A)^{k-i}}{(k-i)!} \cdot c_{n_p,i} \end{bmatrix} = C \cdot \hat{t} \tag{41}$$

where C is the vector spline coefficient matrix and $\hat{t}$ is a vector, which depends on the parameter t

$$\hat{t} = \begin{bmatrix} 1 & \cdots & \frac{(t-t_A)^{k-i}}{(k-i)!} & \cdots & \frac{(t-t_A)^{k-1}}{(k-1)!} \end{bmatrix}^T \tag{42}$$

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

$$\hat{t}(t) = \begin{bmatrix} 1 & t - t_A & \frac{(t-t_A)^2}{2} & \frac{(t-t_A)^3}{6} \end{bmatrix}^T \tag{43}$$

and the cubic spline coefficient matrix, C, simplifies to

$$C = \begin{bmatrix} c_{1,1} & c_{1,2} & c_{1,3} & c_{1,4} \\ \vdots & \vdots & \vdots & \vdots \\ c_{j,1} & c_{j,2} & c_{j,3} & c_{j,4} \\ \vdots & \vdots & \vdots & \vdots \\ c_{n_p,1} & c_{n_p,2} & c_{n_p,3} & c_{n_p,4} \end{bmatrix} \tag{44}$$

The first step of the vector spline approximation algorithm is to find the border points, $p_{iso,border}$, which meet the isoperformance condition (3) 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

$$\begin{aligned} &J_z(p_{corner,i}) \le J_{z,req} \le J_z(p_{corner,j}) \cup \\ &J_z(p_{corner,i}) \ge J_{z,req} \ge J_z(p_{corner,j}) \end{aligned} \tag{45}$$

The next step is to connect the isoperformance border points with cubic splines along the boundary of $\mathbf{B}$. 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:

$$\begin{aligned} &\hat{p}_{cent} = \begin{bmatrix} \hat{p}_{c,1} & \cdots & \hat{p}_{c,j} & \cdots & \hat{p}_{c,n_p} \end{bmatrix}^T \\ &\text{where} \qquad \hat{p}_{c,j} = \frac{1}{n_b} \sum_{i=1}^{n_b} p_{iso,border,i,j} \end{aligned} \tag{46}$$

and n_b is the number of border points. The actual centroid, p_{cent}, is found by steepest gradient search, see subsection III. Finally the cubic splines connecting the centroid and the mid-points of the border splines are found, subject to tolerance, τ.

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. 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.

The multivariable SDOF problem was tackled by the vector spline approximation algorithm. The three variable (design) parameters, ω_d, m and ω_o are considered. The

desired performance level is $J_{z,req} = 0.8$ [mm] RMS. Results for the single DOF oscillator problem are shown in Figure 14. The outline of the isoperformance surface can clearly be seen.

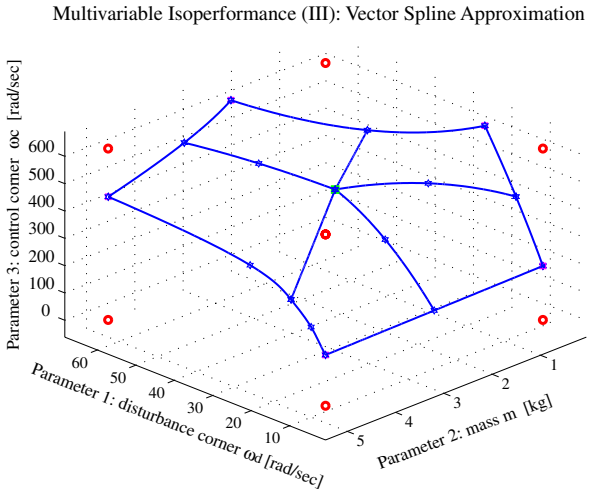

Fig. 14. Multivariable Isoperformance (III): Vector Spline Approximation for SDOF sample problem.

D. Multivariable Algorithm Comparison

A comparison of the multivariable algorithms using the single degree-of-freedom problem is presented in Table II. The algorithms are compared based on the CPU runtime, the number of floating-point operations required, the solution quality expressed as Υ_{iso} and the number (quantity) of isoperformance points, p_{iso}, found.

TABLE II
COMPARISON OF MULTIVARIABLE ALGORITHMS FOR SDOF PROBLEM: (IA) EXHAUSTIVE SEARCH, (IB) BRANCH-AND-BOUND, (II) TANGENTIAL FRONT FOLLOWING AND (III) VECTOR SPLINE APPROXIMATION.

Metric	Ia	Ib	II	III
MFLOPS	6,163.72	891.35	106.04	1.49
CPU time [sec]	5078.19	498.56	69.59	4.45
Tolerance τ	1.5 %	2.5 %	1.5 %	1.5%
Actual Error Υ_{iso}	0.87 %	2.43 %	0.22 %	0.42 %
# of isopoints	2073	7421	4999	20

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 tangential 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, Υ_{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.

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 $p_j = 1, \ldots, n_p$, 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 existed but cannot be found. This algorithm switching strategy was suggested in the thesis roadmap, see Figure 3.

E. Complexity Theory

One of the aims of complexity theory [6] 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. We attempt to estimate the asymptotic growth of the number of floating point operations required as a function of the algorithm used, 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 exhaustive search approach requires the following number of performance function $J_z(p_j)$ evaluations N_{exs}:

$$N_{exs} = \prod_{j=1}^{n_p} \left\lceil \frac{p_{UB,j} - p_{LB,j}}{\Delta p_j} \right\rceil \tag{47}$$

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 Δp_j is the discretization step size of the j-th parameter. As an approximation we can assume that the main computational cost for computing J_z comes from solving the Lyapunov equation (56) for the state covariance Σ_q. Section V 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

$$J_{exs} = \alpha^{n_p} \cdot 50 n_s^3 \tag{48}$$

Looking at the asymptotic growth of the algorithm can be accomplished by taking the `log` of (48) such that

$$log\,(J_{exs}) = n_p log(\alpha) + 3log(n_s) + \text{ const.} \qquad (49)$$

Thus, exhaustive search is solvable in polynomial time as a function of n_s ("size of the model"), but it is non-polynomial in n_p.

The complexity of branch-and-bound, as developed in section IV, is more difficult to assess than exhaustive search, since the number of branches kept at each generation is problem dependent. The computational cost for branch-and-bound is approximated as

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

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

$$log\,(J_{bab}) = n_g\,(n_p log 2 + log\beta) + 3log(n_s) + \text{ const.} \quad (51)$$

Note that the number of generations, n_g, is difficult to predict apriori but is strongly dependent on the isoperformance tolerance, τ.

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, ∇J_z, which requires $(1+n_z)\cdot 50 n_s^3$ floating point operations. The number of directions in the nullspace is $n_p - n_z$ and the distance to search to the parameter boundary $\mathbf{B}$ can be approximated as some constant, γ, which depends on the parameter bounds, p_{LB}, p_{UB} and the step size. The cost of tangential front following is thus approximated as

$$J_{tff} = \gamma^{n_p - n_z} \cdot (1+n_z) \cdot 50 n_s^3 \qquad (52)$$

The logarithmic cost is

$$log\,(J_{tff}) = (n_p - n_z)\,log\gamma + log(1+n_z) + 3log(n_s) + \text{ const.} \qquad (53)$$

The first step in the vector spline approximation algorithm is to compute the performance at all 2^{n_p} corner points. An expression for the cost of vector spline approximation, using constants where appropriate, is

$$J_{vsa} = 2^{n_p} \cdot 50 n_s^3 + 2^{n_p} \cdot \frac{1}{2} \cdot 2^{n_g - 1} \cdot (1+n_z) \cdot 50 n_s^3 \quad (54)$$

The logarithmic cost is

$$log\,(J_{vsa}) = n_p log 2 + log(1+n_z) + 3log(n_s) + \text{ const.} \quad (55)$$

From this one can see that the isoperformance problem is intrinsically non-polynomial ($\mathcal{NP}$) in n_p. The benefit of tangential front following is that it reduces the logarithmic asymptote from n_p to $n_p - n_z$. The actual number of floating point operations (FLOPS) required is problem dependent. There is no doubt, that isoperformance problems with more than 10 parameters are still expensive to solve.

V. Large Order Models

A key element of the isoperformance methodology is the ability to compute performances, J_z, and analytical sensitivities, ∇J_z. This has to be done in a computationally efficient manner for large order, typically numerically ill-conditioned systems. It is likely that hundreds or even thousands of Lyapunov equations have to be solved during a comprehensive isoperformance analysis.

The cost of solving a Lyapunov equation (56) or (57) is roughly $50 \cdot n_s^3$. 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), and to apply the new, fast Lyapunov solver presented in Subsection V-A. This drops the exponent in the Lyapunov solution cost from 3 to 2. The second approach is to reduce the number of states, n_s, while retaining the important information in the model. Subsection V-B derives error bounds for performance and sensitivity analysis for reduced systems. Finally Subsection V-C is devoted to the derivation of the transformed governing sensitivity equation (TGSE). Using the TGSE 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 sum of these contributions enables a meaningful isoperformance analysis for realistic, large order systems.

A. Efficient Solution of Lyapunov Equation

Empirical considerations 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}. A fast Lyapunov solver for diagonalizable systems is presented here, which can reduce the exponent from a value of 3 to 2.

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

$$A_{zd}\Sigma_q + \Sigma_q A_{zd}^T + B_{zd} B_{zd}^T = 0 \qquad (56)$$

and the Lagrange multiplier matrix

$$L_i A_{zd} + A_{zd}^T L_i + C_{zd,i}^T C_{zd,i} = 0 \qquad (57)$$

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 (1). Note that matrix A_{zd} is of size $n_s \times n_s$, where n_s is a measure of the model order (size). The general form of the Lyapunov equations (56) and (57) can be written as

$$AX + XA^T + Q = 0 \qquad (58)$$

If A can be diagonalized into smaller diagonal block of size m, e.g. 2×2 blocks by eigenvalue decomposition, the sys-

tem can be rewritten as

$$\begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix} \begin{bmatrix} X_{11} & X_{12} \\ X_{12}^T & X_{22} \end{bmatrix} + \begin{bmatrix} X_{11} & X_{12} \\ X_{12}^T & X_{22} \end{bmatrix} \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix}^T + \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{12}^T & Q_{22} \end{bmatrix} = 0 \tag{59}$$

Instead of looking for the global Lyapunov solution, X, we can solve the following four $m \times m$ matrix equations with $m < n_s$.

$$\begin{aligned} &1)\ A_1 X_{11} + X_{11} A_1^T + Q_{11} = 0 \\ &2)\ A_1 X_{12} + X_{12} A_2^T + Q_{12} = 0 \\ &3)\ A_2 X_{12}^T + X_{12}^T A_1^T + Q_{12}^T = 0 \\ &4)\ A_2 X_{22} + X_{22} A_2^T + Q_{22} = 0 \end{aligned} \tag{60}$$

Notice that equations 1 and 4 in Equation (60) 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 in less time than the full $n_s \times n_s$ system. This is due to the fact that the Lyapunov 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 decoupled modal form resulting from a normal modes analysis can be easily written in such a diagonal 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 [21]. This is also sometimes referred to as the real modal form

$$A_i = \begin{bmatrix} -\zeta_i \omega_i & \omega_i \sqrt{1-\zeta_i^2} \\ -\omega_i \sqrt{1-\zeta_i^2} & -\zeta_i \omega_i \end{bmatrix} \tag{61}$$

where ω_i and ζ_i are the modal frequency and damping ratio of the i-th mode, respectively. Using the 2×2 modal system, there are now $n_s/2$ blocks along the diagonal of A. Keeping in mind the symmetry of X, this means

$$\frac{\frac{n_s}{2}\left(\frac{n_s}{2}+1\right)}{2}$$

separate 2×2 Lyapunov solutions X_{ij} must be solved. 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 blocks can be written.

$$N_{blocks} = \frac{\frac{n_s}{m}\left(\frac{n_s}{m}+1\right)}{2} \tag{62}$$

An example of the improvements in computation times for large order Lyapunov solutions is given in Table III. The computational cost can be estimated as

$$J_{newlyap} = 25(mn_s^2 + m^2 n_s) \tag{63}$$

TABLE III
TIME SAVINGS FOR DIAGONAL LYAPUNOV SOLVER (SIM MODEL V2.2)

n_s	CPU time n_s system [min]	CPU time m blocks [min]	fastest block size m	savings ratio t_{max}/t_{min}
600	3.5	0.13	24	26.9
1000	16.1	0.37	20	43.5
1500	53.7	0.90	20	59.7
2000	145.5	1.65	40	88.2

The quality of the solution is checked by placing the answer back into the Lyapunov equation. 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 the systems investigated. 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.

B. Model Reduction apriori Error Bound

The second approach to dealing with large order state space systems is to reduce the number of states, n_s, while retaining the important information in the model. This can be done by internally balancing the systems (1) and partitioning it into states that are going to be "kept" and states that are going to be "removed" as follows:

$$\begin{aligned} \dot{\tilde{q}} &= \tilde{A}_{zd}\tilde{q} + \tilde{B}_{zd} d = \begin{bmatrix} \tilde{A}_{kk} & \tilde{A}_{kr} \\ \tilde{A}_{rk} & \tilde{A}_{rr} \end{bmatrix} \begin{bmatrix} \tilde{q}_k \\ \tilde{q}_r \end{bmatrix} + \begin{bmatrix} \tilde{B}_k \\ \tilde{B}_r \end{bmatrix} d \\ z &= \tilde{C}_{zd}\tilde{q} = \begin{bmatrix} \tilde{C}_k & \tilde{C}_r \end{bmatrix} \begin{bmatrix} \tilde{q}_k \\ \tilde{q}_r \end{bmatrix} \end{aligned} \tag{64}$$

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. The reduced state space system can then be written as:

$$\begin{aligned} \dot{\bar{q}} &= \bar{A}_{zd}\bar{q} + \bar{B}_{zd} d = \tilde{A}_{kk}\tilde{q}_k + \tilde{B}_k d \\ z &= \bar{C}_{zd}\bar{q} = \tilde{C}_k \tilde{q}_k \end{aligned} \tag{65}$$

This can be graphically demonstrated by looking at the structure of the A_{zd} matrix (see Figure 15).

Assuming that the performance is the RMS of the i-th metric, $J_{z,i} = \sigma_{z,i}$, the RMS can be computed using the reduced system matrices directly by substituting (65) into (56). The RMS performance of the i-th performance metric is then

$$\bar{\sigma}_{z_i} = \left(\bar{C}_{zd,i} \Sigma_{\bar{q}} \bar{C}_{zd,i}^T\right)^{1/2} \tag{66}$$

Now we have introduced an approximation, since only the n_k states that have Hankel singular values[5] σ_i^H above a

[5] See Moore [14] on details of computing the Hankel singular values.

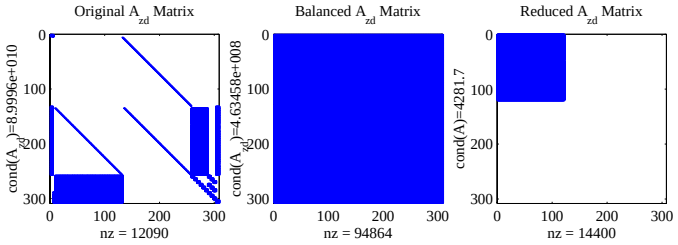

Fig. 15. Sparsity structure of (a) the original SIM-Classic A-matrix (308 states) according to equation (68), (b) Balanced A-matrix (308 states, fully populated) and (c) the reduced A-matrix (120 states, fully populated).

threshold value are kept in the model. Thus, it is evident that $\bar{\sigma}_{z_i} \neq \sigma_{z_i}$, since states have been removed and so has their contribution to the resulting RMS. An apriori error bound for deciding on the number of states to keep, n_k, was developed. This bound is less conservative than the ones found in the literature, see Zhou [21], p.159.

$$\frac{\Delta\sigma_{z_i}}{\bar{\sigma}_{z_i}} < \frac{1}{2} \cdot \frac{\sum_{i=n_k+1}^{n_s} \sigma_i^H}{\sum_{i=1}^{n_k} \sigma_i^H} \tag{67}$$

The inequality shows the relative RMS error due to model reduction on the left side. This is smaller half the ratio of the sum of removed Hankel singular values over the sum of kept singular values. This inequality is useful, since it can be used to determine the number of states n_k that have to be kept in the model in order to achieve a desired accuracy on the RMS prediction. Gutierrez [7] for example has previously stated that *"In actuality, the model reduction process should be iterative in nature, and states should be removed until performance predictions begin to deviate by a predetermined amount"*. This suggests that the model should be run several times until the correct level of reduction is found. This time consuming procedure can be avoided by applying the error bound in (67) apriori.

The relative RMS error (MDOF example explained in thesis) as a function of the truncation level n_k is shown in Figure 16. It can be seen that the conservative error bound from inequality (67) is valid. The conservatism ratio, defined as the ratio of error bound (67) over the actual relative RMS error, was observed to decrease from 46794 with only 2 states removed to a ratio of 1.169 with only 2 states kept in the system. This means that the RMS error bound (67) becomes increasingly accurate the more states are truncated. A similar error bound was developed for the analytical sensitivity error due to model reduction.

C. Transformed Governing Sensitivity Equation

The assumed structure of the appended A_{zd} matrix in (1), which can span on the order of 100-10000 states, as stipulated by Gutierrez [7], is given as follows:

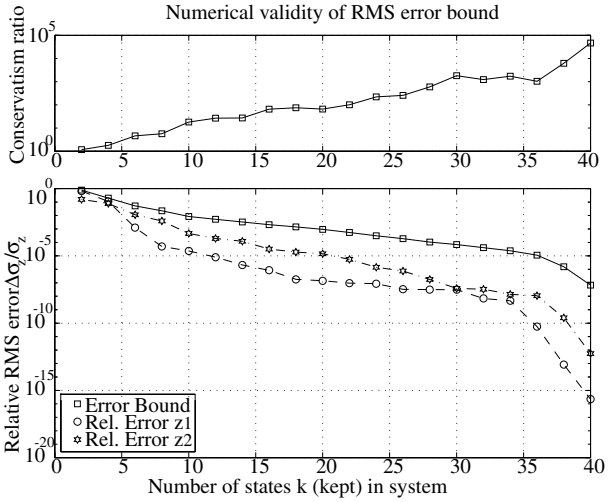

Fig. 16. Validity of RMS error bound for various truncation levels according to inequality (67)

$$A_{zd} = \begin{bmatrix} A_d & 0 \\ \begin{bmatrix} B_w \\ B_c D_{yw} \end{bmatrix} C_d & \begin{bmatrix} A_p & B_u C_c \\ B_c C_y & A_c + B_c D_{yu} C_c \end{bmatrix} \end{bmatrix} \tag{68}$$

Any kind of similarity transformation, $\tilde{q} = Tq$, suggested in [14],[11] removes the explicit dependence of the state space matrices on the parameters of interest. This is the case both for the diagonalization in Subsections V-A and the balanced reduction in V-B. 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.

Pre-multiplying with the transformation matrix T (T can be rectangular if the model is being transformed and reduced) we obtain the transformed state space system as

$$\begin{aligned} \dot{\tilde{q}} &= TA_{zd}T^{-1}\tilde{q} + TB_{zd}d = \tilde{A}_{zd}\tilde{q} + \tilde{B}_{zd}d \\ z &= C_{zd}T^{-1}\tilde{q} = \tilde{C}_{zd}\tilde{q} \end{aligned} \tag{69}$$

The sensitivity of the i-th performance RMS with respect to the j-th parameter for a similarity transformed system is then computed as follows:

$$\frac{\partial \tilde{\sigma}_{z_i}}{\partial p_j} = \frac{1}{2\tilde{\sigma}_{z_i}} \cdot \frac{\partial \tilde{\sigma}_{z_i}^2}{\partial p_j} \tag{70}$$

The RMS in the denominator of the first term is directly obtained by substituting the transformed matrices in (56) and (66). 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:

$$\frac{\partial \tilde{\sigma}^2_{z_i}}{\partial p_j} = trace\left[T^{-1}\Sigma_{\tilde{q}}\left(T^{-1}\right)^T \frac{\partial\left(C^T_{zd,i}C_{zd,i}\right)}{\partial p_j}\right] + trace\left[\tilde{L}_i\left\{T\frac{\partial A_{zd}}{\partial p_j}T^{-1}\Sigma_{\tilde{q}} + \Sigma_{\tilde{q}}\left(T^{-1}\right)^T\frac{\partial A^T_{zd}}{\partial p_j}T^T\right\}\right] + trace\left[\tilde{L}_i\left\{T\frac{\partial\left(B_{zd}B^T_{zd}\right)}{\partial p_j}T^T\right\}\right] \quad (71)$$

The TGSE (71) 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 transformed system. The matrix derivatives in (71) 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 (71) is surprising, since we expect to find derivative terms of the transformation matrix T in this equation. Proof of the TGSE is contained in full length thesis document.

VI. 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 1g environment.

A. Testbed Description

The DOLCE testbed shown in Figure 17 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 [7], which was used for physical parameter sensitivity validation via small perturbations of masses and stiffnesses. The four variable parameters on DOLCE are:

- V_s excitation RMS voltage [V]
- m_p payload mass [lbs]
- m_s seismic mass [g]
- k_s suspension spring stiffness [lbs/in]

Figure 17 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 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. 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

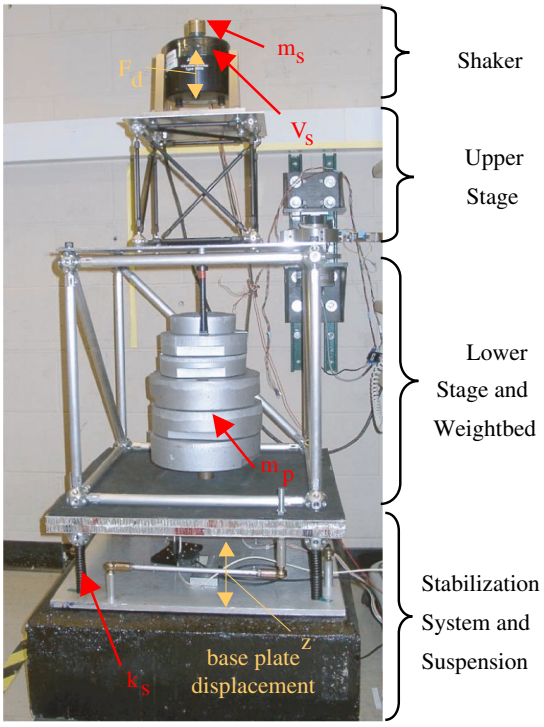

Fig. 17. DOLCE Testbed

level. The performance is the root-mean-square (RMS) of the base plate displacement

$$J_z = E\left[z^T z\right]^{1/2} \quad (72)$$

This would correspond to jitter of the spacecraft bus in a real space system. The primary instrumentation consists of a uniaxial load cell, which is attached to the seismic mass and measures the disturbance force, F_d. The performance is measured via an inductive proximitor, which acts as a gap sensor (eddy current gap sensor 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 suite below the sandwich plate is shown in Figure 18.

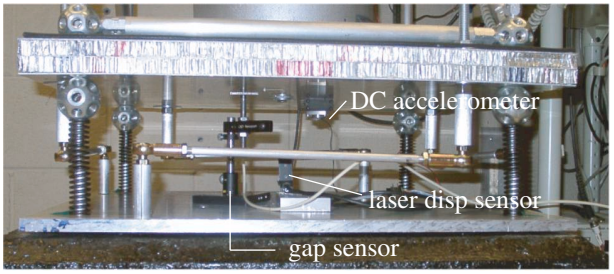

Fig. 18. DOLCE Testbed Sensors

B. Experimental Approach

The experimental approach is presented in Figure 19. First the testbed was assembled, instrumented and cali-

brated. It was decided to conduct a bivariate isoperformance test, with the performance given by Equation 72. 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]. A test matrix was run on the testbed and recorded with parameter increments $\Delta V_s = 0.1$ and $\Delta m_p = 10$, respectively. From this gridded data isoperformance contours were extracted via linear interpolation, see Subsection III-A.

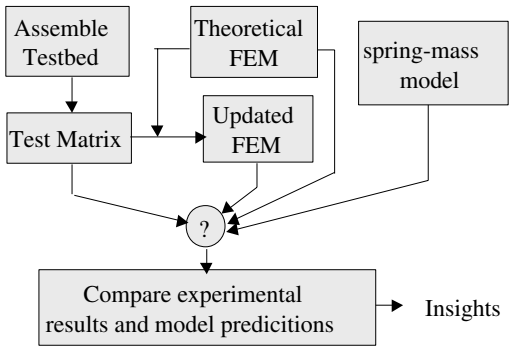

Fig. 19. Experimental Approach

Independently and without knowledge of the experimental results an apriori finite element model (FEM) was constructed ("original FEM"). This model only used assembly drawings, masses from scale measurements and catalogue values for material properties and spring stiffnesses. 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. 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. Finally the isoperformance contours for DOLCE are predicted with a single degree-of-freedom (SDOF) model, which lumps the entire testbed mass together with the payload mass m_p over the four suspension springs (in parallel) represented as a single compliance. The hope is that insights can be gained by comparing different performance contours for the experiment with the ones predicted for the three models.

C. Testbed Characterization

The transfer function (FRF) from disturbance (shaker) force to base plate displacement, $G_{zd}(s) = Z(s)/F_d(s)$, where $s = j\omega$, is obtained experimentally and by model prediction, see Figure 20.

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 translated vertically up and down on the 4 suspension (compression) springs. The second mode at 65 Hz is the upper coupling plate bending mode, which causes a vertical displacement via the center rod. Mode shapes for these two modes are contained in Figure 21.

As expected the SDOF model can only predict the first resonance. The original FEM overpredicts the upper plate

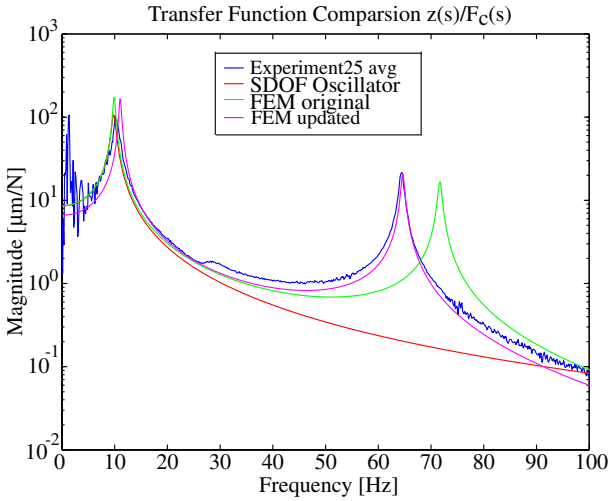

Fig. 20. DOLCE transfer function $G_{zd} = Z(s)/F_d(s)$ for $m_p = 0$, $V_s = 1.0$

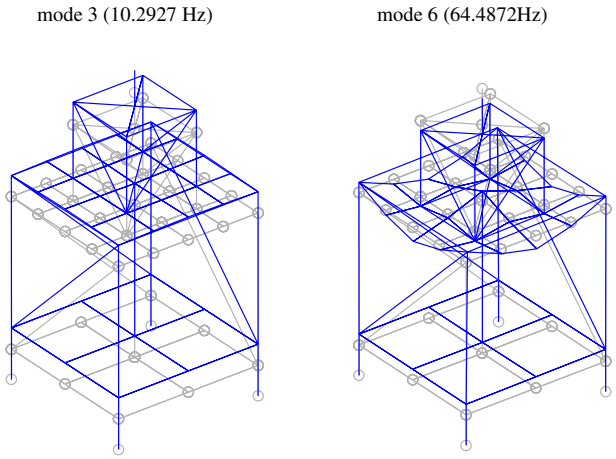

Fig. 21. DOLCE Testbed Observable Modes

mode by roughly 10 Hz. The agreement between the updated FEM and the experimental transfer function is very good.

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 22.

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. The performance J_z can be computed by integrating under S_{zz} and taking the square root.

$$J_z = \left[2 \int_{f_{\min}}^{f_{\max}} S_{zz}(f)\, df \right]^{1/2} \tag{73}$$

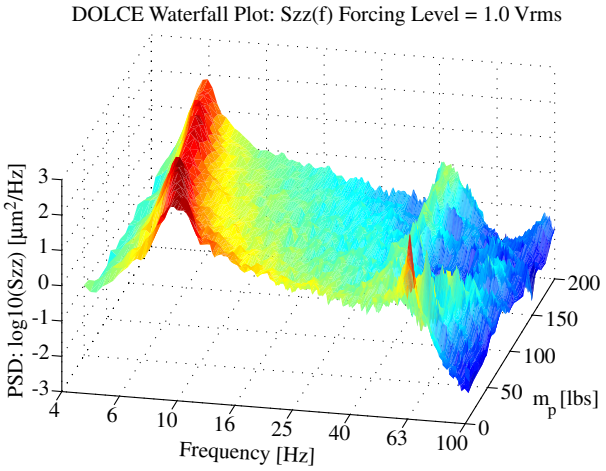

Fig. 22. Waterfall Plot for 1.0 Vrms forcing level

D. Isoperformance Results and Interpretation

The basis for obtaining the experimental isoperformance contours is the test matrix with V_s and m_p as described under VI-B. At each parameter combination the time histories of $F_d(t)$ and $z(t)$, where recorded and the performance $J_z = J_z(V_s, m_p)$ was computed with 25 averages. The results from the test matrix are shown in Figure 23.

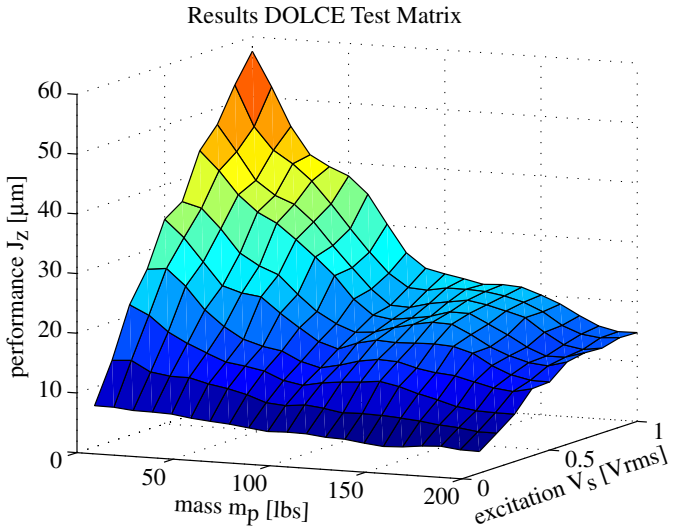

Fig. 23. DOLCE Test Matrix

The peak displacement RMS value of 57.6 [μm] is obtained for the maximum excitation level ($V_s = 1.0$ [Vrms]) with an empty weight bed ($m_p = 0$ [lbs]). This is intuitively satisfactory, since at this point the maximum disturbance energy enters the system (about 7 N of force F_d 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 [μm] levels (Figure 24).

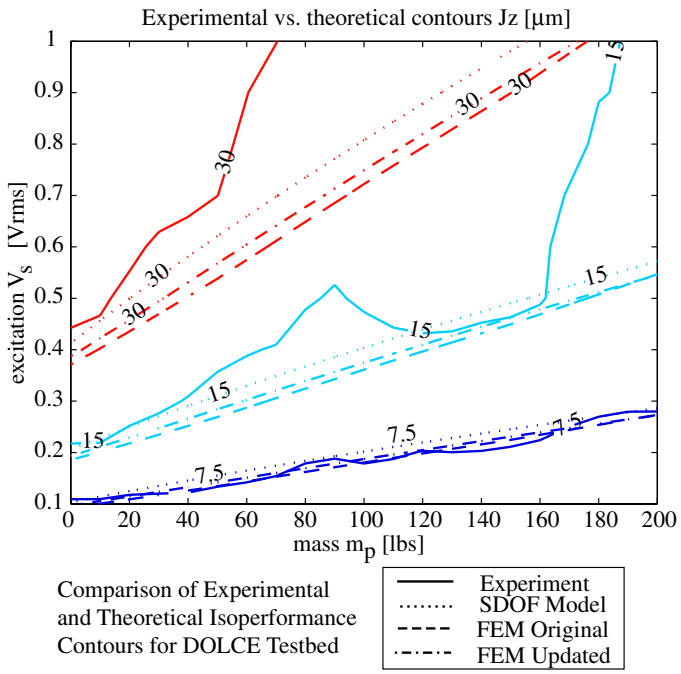

Fig. 24. DOLCE Testbed Comparison of Experimental versus Theoretical Isoperformance Contours

Similar contours are predicted for the SDOF and 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 μm contour. Deviations are found for larger forcing levels (15 and 30 μm contours), even though the general trends are still predicted correctly by the isoperformance models. The cause for this deviation is likely due to non-linear effects in the structural plant as the shaker amplitude increases. 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 opto-mechanical systems. Caution must be exercised if non-linearities are suspected in any part of the system.

VII. NEXUS Spacecraft Study

At this point confidence has been gained 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 case study is to demonstrate the usefulness of the isoperformance methodology on a realistic conceptual design model of a high-performance spacecraft.

A. NEXUS Description

A graphical representation of the launch and on-orbit configurations of NEXUS is shown in Figure 25.

NEXUS was planned as a technology risk-reduction experiment in space and as a precursor to NGST. 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 con-

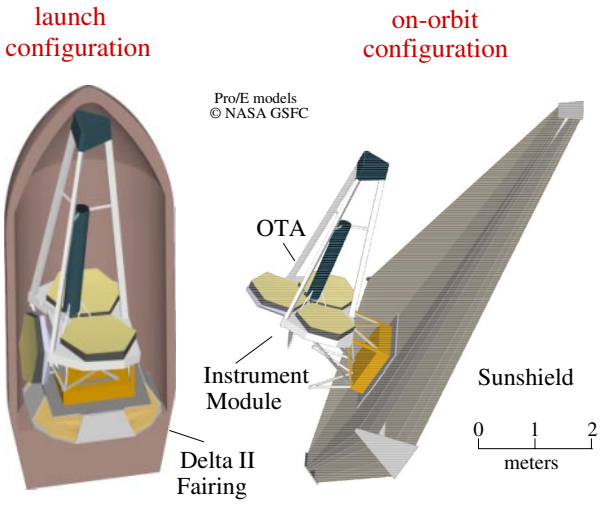

Fig. 25. NEXUS Spacecraft Concept

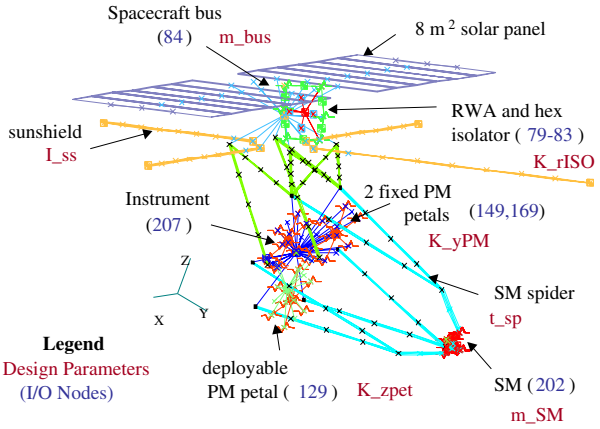

Fig. 26. NEXUS Finite Element Model. Important I/O grid points (nodes) and variable design parameters are shown.

ceptual model is well developed and many lessons learned from previous NGST Yardstick models were incorporated.

NEXUS features a 2.8 m diameter primary mirror, consisting of three AMSD-sized primary mirror (PM) petals. Two of these are fixed and one is deployable as shown in Figure 25 on the left side. The total mass of the spacecraft is nominally 752.8 [kg] at a cost of $M 105.88 (FY00). The target orbit is 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 light-weight,thus accounting for the first flexible mode of the spacecraft structure around 0.2 Hz.

B. Integrated Modeling

The integrated model for NEXUS contains a structural finite element model (FEM), see Figure 26. The model was initially created in FEMAP/NASTRAN and subsequently translated to IMOS [8]. The figure 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.

The two performance metrics of interest are the root-mean-mean-square wavefront error, $J_{z,1}$ = RMMS WFE, and the root-sum-square line-of-sight jitter, $J_{z,2}$ = RSS LOS. The optical linear sensitivity matrices for these performance with respect to the translations and rotations of the optical elements were computed with MACOS. There are four expected disturbance sources in the 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. 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}. The appended dynamics of this system are shown in the block diagram of Figure 27.

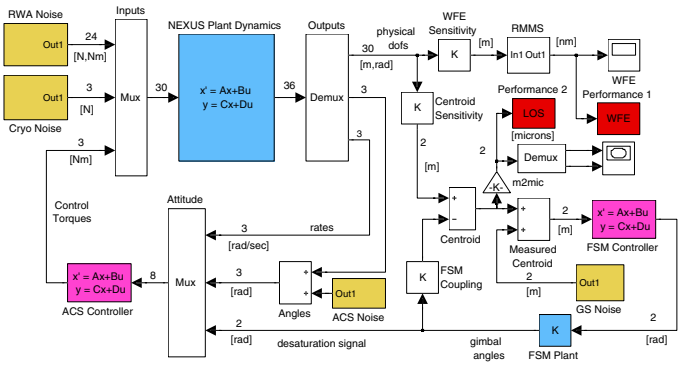

Fig. 27. NEXUS block diagram with 4 disturbance sources (RWA, Cryo, ACS noise, GS noise) and 2 performances (RMMS WFE, RSS LOS). Simulation implemented in Simulink as well as state space.

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 were 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 IV summarizes the variable design parameters in the NEXUS case study.

C. Disturbance Analysis

A disturbance analysis was carried out with the initial parameters, p_o, given in Table IV. Results for LOS jitter are contained in Figure 28. 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

TABLE IV
NEXUS VARIABLE DESIGN PARAMETERS p_j, $j = 1, \ldots, 25$.

Symbol	Nom	Description	Units
		disturbance parameters	
Ru	3000	upr op wheel speed	[RPM]
Us	1.8	stat whl imbalance	[gcm]
Ud	60	dyn whl imbalance	[gcm^2]
fc	30	cryo drive freq	[Hz]
Qc	0.005	cryo attenuation	[-]
Tst	20	star track update	[sec]
Srg	3e-14	RG noise intensity	[rad^2/s]
Sst	2	Tracker one sigma	[arcsec]
Tgs	0.04	Guider int time	[sec]
		plant parameter	
m_SM	2.49	mass of SM	[kg]
K_yPM	0.8e6	PM bipod stiffness	[N/m]
K_rISO	3000	RW Isolator stiff	[Nm/rad]
m_bus	0.3e3	S/C bus mass	[kg]
$Kzpet$	0.9e8	petal hinge stiff	[N/m]
t_sp	0.003	Spider wall thick	[m]
I_ss	0.8e-8	SS bend inertia	[m^4]
I_propt	5.11	prop sys inertia	[kgm^2]
ζ	0.005	modal damping	[-]
		optics parameters	
λ	1e-6	CL opt wavelength	[m]
Ro	0.98	opt surf trans	[-]
QE	0.80	CCD quantum eff	[-]
Mgs	15.0	mag of guide star	[mag]
		controls parameters	
fca	0.01	ACS control BW	[Hz]
Kc	0.0	FSM/ACS coupling	[0-1]
Kcf	2000	FSM controller gain	[-]

(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.

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 29. The predicted RSS LOS is 14.97 μm, versus a requirement of 5 μm[6].

The wavefront error performance is omitted here for simplicity, but it is discussed in Chapter 7 of the thesis document. Table V shows an overview of the predicted performance, using the initial parameters p_o.

TABLE V
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	[μ m]

[6]This requirement comes from the assumption of 25 μm pixel pitch and a desire to maintain LOS jitter below 1/5 of a pixel.

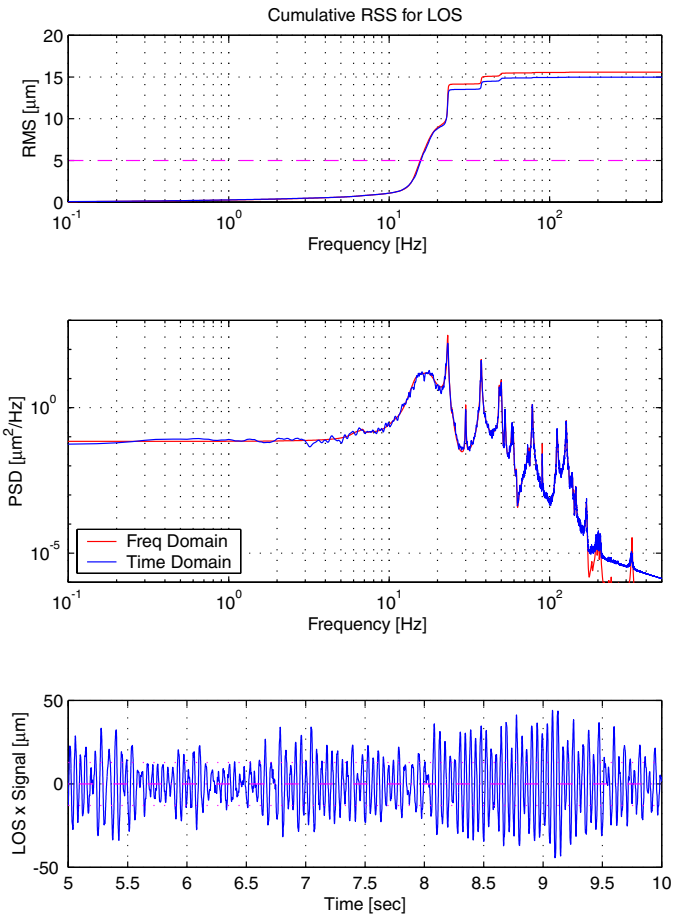

Fig. 28. LOS Jitter initial disturbance analysis

The wavefront error requirement ($\lambda/50$) is nearly met, but the pointing performance has to improve by a factor of roughly 3.

D. Sensitivity Analysis

The next step is a comprehensive sensitivity analysis for the 25 variable design parameters of NEXUS. The sensitivity produces the normalized Jacobian matrix evaluated at p^o

$$\overline{\overline{\nabla}} J_z = \frac{p_o}{J_{z,o}} \begin{bmatrix} \frac{\partial J_{z,1}}{\partial R_u} & \frac{\partial J_{z,2}}{\partial R_u} \\ \cdots & \cdots \\ \frac{\partial J_{z,1}}{\partial K_{cf}} & \frac{\partial J_{z,2}}{\partial K_{cf}} \end{bmatrix} \tag{74}$$

which is graphically shown in Figure 30. 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.

The RMMS WFE is most sensitive to the upper operational wheel speed, Ru, the RWA isolator stiffness, K_rISO, and the deployable petal hinge stiffness, K_zpet. The RSS LOS is most sensitive to the dynamic wheel imbalance, Ud, the RWA isolator stiffness, K_rISO, structural damping, $zeta$, the guide star magnitude, Mgs and the FSM (fine pointing loop) control gain, Kcf. Interpreting these

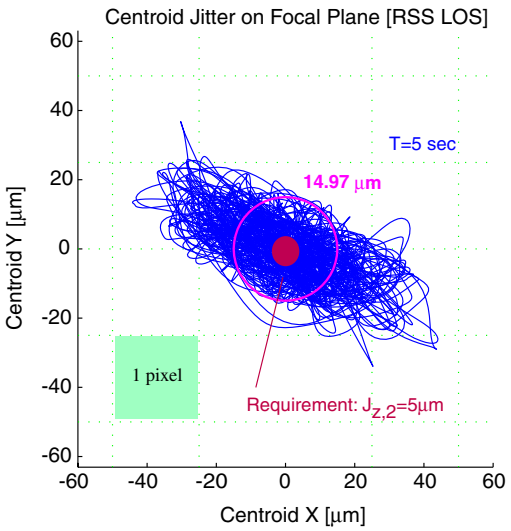

Fig. 29. RSS LOS Centroid Jitter Plot on Focal Plane

results one would expect for example that a 1.0 % decrease in the isolator stiffness, K_rISO 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.

E. Bivariate Isoperformance

A bivariate isoperformance analysis is conducted for NEXUS using $J_{z,1}$ = RSS LOS as the performance and the two most sensitive parameters from Figure 30, right column, as the parameters. Hence, dynamic wheel imbalance, Ud, is traded versus RWA isolator joint stiffness, K_rISO, while constraining the performance the the requirement, $J_{z,2,req} = 5[\mu m]$. The results are contained in Figure 31.

The isoperformance contour at RSS LOS = 5 μm 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 struts and larger imbalances. The region in the upper right of Figure 31, where LOS jitter of 160 μm is predicted, occurs, when the isolator modes coincide with other flexible modes of the NEXUS structure.

F. Multiobjective Optimization

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 VII-E 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,

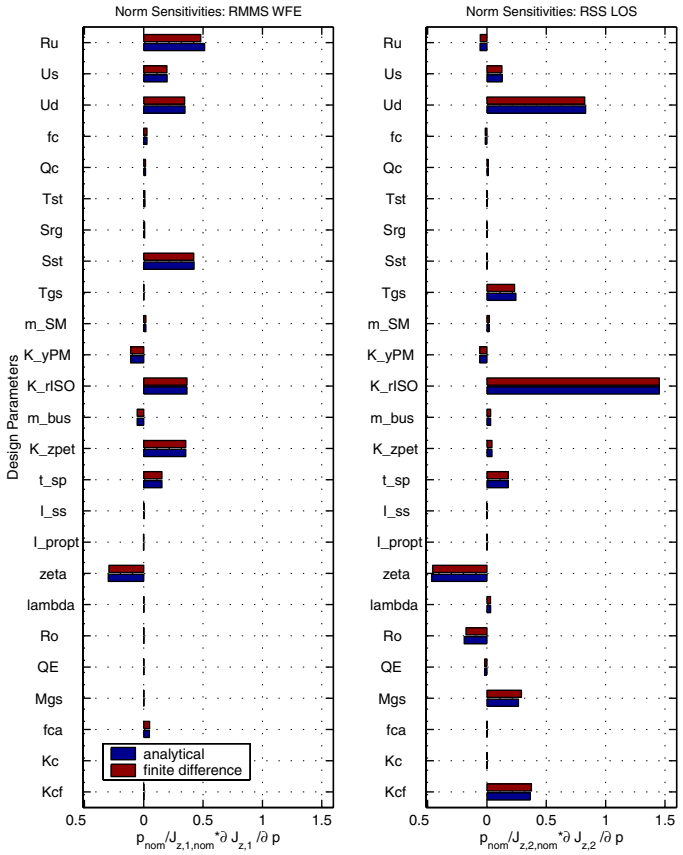

Fig. 30. NEXUS normalized sensitivity analysis results at p^o.

$J_z(p_{iso}) = J_{z,req}$, 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 (7) may be solved, whereby Q_{cc} and Q_{rr} are weighting matrices among the cost and risk objectives and η 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.

Such a multivariable analysis was conducted for a subset of 10 out of the 25 design parameters for NEXUS. 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")
- $J_{c,2}$ = Smallest FSM control gain
- $J_{r,1}$ = Percent performance uncertainty

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 32.

Specifically, the isoperformance condition leads to the fact that all designs, p^*_{iso}, asymptote to the same value

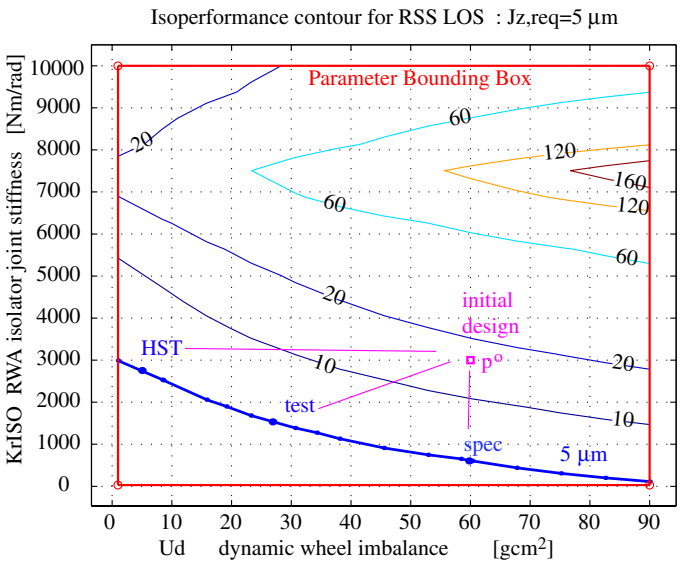

Fig. 31. NEXUS Bivariate Isoperformance analysis with $p_1 = U_d$, $p_2 = K_r ISO$ and $J_z = RSSLOSS$.

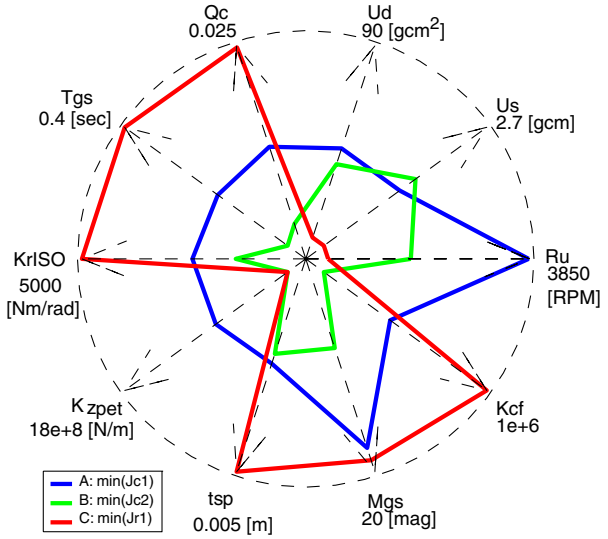

Fig. 32. NEXUS Multivariable Isoperformance. Radar plot of 3 pareto optimal designs.

in the cumulative RMS plot, as shown for RSS LOS in the thesis (Chapter 7). The results for the NEXUS pareto optimal designs are summarized in Table VI.

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.

G. 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 VII shows the apriori allocation and the actual disturbance contributions to the variance of RSS LOS for Design "A", which is chosen as the final design, p^{**}_{iso}.

TABLE VI
NEXUS PARETO OPTIMAL DESIGNS

	$J_{z,1}$	$J_{z,2}$	$J_{c,1}$	$J_{c,2}$	$J_{r,1}$
A	20.0000	5.2013	0.6324	0.4668	± 14.3 %
B	20.0012	5.0253	0.8960	0.0017	± 8.8 %
C	20.0001	4.8559	1.5627	1.0000	± 5.3 %

TABLE VII
NEXUS ERROR BUDGET

Error Source	VAR %	Budget	VAR %	Capability
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
Total	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

$$\Psi_i = \sum_{j=1}^{n_d} \Psi_{i,j} = J^2_{z,req,i} \tag{75}$$

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 33.

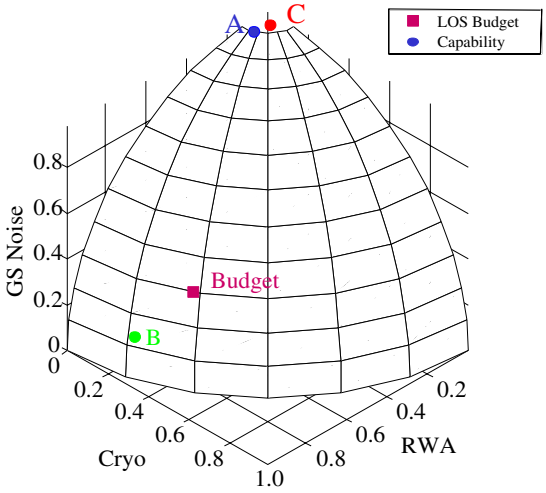

Fig. 33. NEXUS Error Sphere for RSS LOS. Note: ACS sensor noise contributions not shown.

Error Budgeting is an obvious application of isoperformance, since an apriori error budget will always result in the desired performance level. The advantage of using isoperformance in this context is that a "capability" error bud-

get, Ψ^{**}, can be found, which is theoretically achievable since it is based on the underlying integrated model.

VIII. Thesis Contributions and Recommendations

A. Contributions

This thesis develops and validates a novel approach to the design of complex multi-disciplinary systems. 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. The following specific thesis contributions can be identified:

1. Developed a methodology for identifying the locus of parameters that yields constant 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.

B. Limitations

The limitations of the isoperformance framework are that it assumes Linear-Time-Invariant (LTI) systems and operates on H2-performance metrics for zero-mean random processes. 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.

C. Recommendations

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 "satisfising". In this approach not a product that optimizes the performance is sought, but rather a product 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
- Link to System Architecture, GINA and product design

Acknowledgments

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. The undergraduate students who participated in fabrication and assembly of the DOLCE testbed are Dusty DeQuine and Cemocan Yesil.

References

[1] R. G. Brown and P. Y. C. Hwang, *Introduction to Random Signals and Applied Kalman Filtering*, John Wiley & Sons, Inc., 1997.

[2] E. F. Crawley, B. P. Masters, and T. T. Hyde, *Conceptual design methodology for high performance dynamic structures*, in Proceedings of the 36th AIAA Structures, Structural Dynamics, and Materials Conference, New Orleans, LA, April 1995, pp. 2768–2787. AIAA Paper No. 95-2557.

[3] C. de Boor, *A practical guide to splines*, vol. I of Applied mathematical sciences, Springer Verlag, New York, 1 ed., 1978.

[4] P. R. Freund, *Course Notes 16.910J/6.631 - Introduction to Simulation and Optimization - Massachusetts Institute of Technology*. Module 1, Sep-Oct 1999.

[5] J. J. Gilheany, *Optimum selection of dampers for freely vibrating multidegree of freedom systems*, Proceedings of Damping '89, II (1989), pp. FCC–1:18.

[6] O. Goldreich, *Complexity theory survey*. Internet: http://www.wisdom.weizmann.ac.il, accessed 5-4-2001 2001.

[7] 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.

[8] Jet Propulsion Laboratory, *Integrated Modeling of Optical Systems User's Manual*, January 1998. JPL D-13040.

[9] 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.

[10] 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.

[11] A. Laub, *Computation of balancing transformations*, Proceedings of JACC, 1 (1980).

[12] B. P. Masters and E. F. Crawley, *Evolutionary design of controlled structures systems*, in Proceedings of the 38th AIAA Structures, Structural Dynamics, and Materials Conference, Kissammee, FL, April 1997. AIAA Paper No. 97-1263.

[13] M. Milman, M. Salama, R. Scheid, and J. S. Gibson, *Combined control-structural optimization*, Computational Mechanics, (1991), pp. 1–18.

[14] 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.

[15] J. I. Pritchard, H. M. Adelman, and J. Sobieszczanski-Sobieski, *Optimization for minimum sensitivity to uncertain parameters*, AIAA Journal, 34 (1996), pp. 1501–1504.

[16] M. B. J. Robert S. Kennedy 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.

[17] M. B. J. Robert S. Kennedy 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.

[18] J. T. R.S. Kennedy 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.

[19] W. T. V. William H. Press, Saul A. Tenkolsky and B. P. Flannery, *Numerical Recipes in C - The Art of Scientific Computing*, Cambridge University Press, 2nd ed., 1996.

[20] H. N. Yoshikazu Sawaragi and T. Tanino, *Theory of Multiobjective Optimization*, vol. 176 of Mathematics in Science and Engineering, Academic Press Inc., London, United Kingdom, i ed., 1985.

[21] K. Zhou, J. C. Doyle, and K. Glover, *Robust and Optimal Control*, Prentice-Hall, Inc., 1996.

Olivier de Weck is a Ph.D. candidate at the Department of Aeronautics and Astronautics at MIT. His research interests include integrated modeling and simulation, spacecraft design, multiobjective optimization and system architecture. Before joining MIT he was a liaison engineer and later engineering program manager for the Swiss F/A-18 fighter aircraft program. He holds a masters of science degree in Aeronautics and Astronautics from MIT (1999) and a Dipl. Ingenieur degree (1993) in Industrial Engineering from the Swiss Federal Institute of Technology, ETH Zurich, Switzerland.