MULTIVARIABLE ISOPERFORMANCE METHODOLOGY FOR PRECISION OPTO-MECHANICAL SYSTEMS

OLIVIER L. DE WECK
Department of Aeronautics and Astronautics
Engineering Systems Division (ESD)
Massachusetts Institute of Technology
77 Massachusetts Avenue
Cambridge, MA 02139, U.S.A.
email: deweck@mit.edu

DAVID W. MILLER
Space Systems Laboratory
Department of Aeronautics and Astronautics
Massachusetts Institute of Technology
77 Massachusetts Avenue
Cambridge, MA 02139, U.S.A.
email: millerd@mit.edu

ABSTRACT

Precision opto-mechanical systems, such as space telescopes, combine structures, optics and controls in order to meet stringent line-of-sight (LOS) pointing and wavefront error (WFE) 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 performance outputs are treated as equality constraints and the non-uniqueness of the design space is exploited by trading key system parameters with respect to each other. The goal is to find a performance invariant set of design options, **I**.

Three algorithms (branch-and-bound, tangential front following and vector spline approximation) are developed for the bivariate and multivariable isoperformance problem. An experimental validation is carried out on the DOLCE laboratory testbed and it is shown that the predicted performance contours match the experimental data well at low excitation levels. This paper focuses on the algorithms used to find **I**, rather than the systems engineering implications. The isoperformance approach enhances the understanding of complex multi-disciplinary systems by exploiting performance information beyond the local neighborhood of a particular point design.

KEY WORDS

Isoperformance, Multidisciplinary Design Optimization, Contouring, Opto-Mechanical Systems

1 Introduction

In designing complex engineering 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. One often voiced criticism of such a computational design is that it focuses too much on performance optimization, when a good system should merely "satisfice" the required performance levels and consider other objectives in order to achieve a balanced design.

The other approach is therefore 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 or risk. This paper explores the second approach by developing a framework termed the "isoperformance methodology" for dynamic, linear time-invariant (LTI) systems, such as the one in Figure 1.

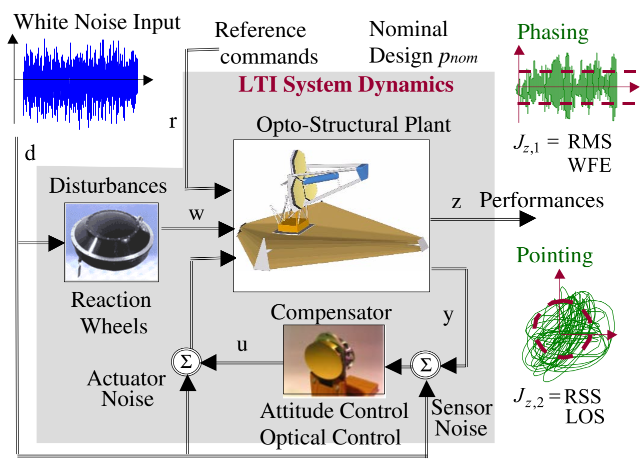

Figure 1. Block diagram of NEXUS spacecraft. The performances, J_z, are typically expressed in terms of the root-mean-square (RMS) or root-sum-square (RSS) values of the outputs.

Once a nominal design (vector), p_{nom}, has been found that meets all requirements with sufficient margins, it is important to realize that this design is generally not unique. This drives the following problem formulation.

1.1 Problem Definition

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 entries are the variable design parameters, p_j, where $j = 1, 2, ..., n_p$, 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 linear time-invariant (LTI) system is given as

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

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

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

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

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

is met, assuming that the number of parameters, n_p, 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 (percent) tolerance, τ

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

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

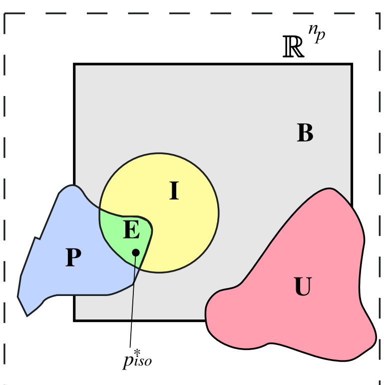

Figure 2. Sets for isoperformance problem definition.

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 $\max(\text{Re}(\lambda_i)) > 0$
$\mathbf{P} \subset \mathbb{R}^{n_p}$	**P**areto optimal subset
$\mathbf{E} = \mathbf{I} \cap \mathbf{P}$	**E**fficient subset

The first task is to find the elements of the isoperformance set **I** in **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 $\overline{\mathbf{U}} = \{x | x \notin \mathbf{U}\}$[1]. The ultimate goal is to find a family of designs p_{iso}^*, which are elements of the efficient set **E**. The efficient set is the intersection of the isoperformance set **I** and the pareto optimal set **P**, i.e. $\mathbf{E} = \mathbf{I} \cap \mathbf{P}$, see [1].

1.2 Previous Work

According to Crawley et al. 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 and sensitivity analysis framework, which is used as a starting point for developing the isoperformance methodology was established by Gutierrez [3]. The $\mathcal{H}_2$-type performances used here are defined in accordance with Zhou, Doyle and Glover [4].

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 for example presented a methodology for optimally selecting dampers for multidegree of freedom systems [5]. In the field of **isoperformance** methodology, work has been done by Kennedy, Jones and coworkers [6]. They present the application of isoperformance analysis in military and aerospace systems design, by trading off equipment, training variables, and user characteristics.

The application of isoperformance draws on previous research results in **multidisciplinary design optimization**. Seminal contributions in this field were made by Messac [7], Sobieski [8] and others. A fundamental book on the theory of multiobjective optimization was published by Sawaragi, Nakayama and Tanino [1]. An important application of multiobjective optimization is concurrent control/structure optimization. The method developed by Milman et al., [9], does not seek the global optimal design, but rather generates a series of Pareto-optimal designs. This work comes closest to the spirit followed in this paper. A systematic approach to isoperformance in complex, opto-mechanical systems, however, is lacking at this time.

1.3 Research Approach - Roadmap

The roadmap in Figure 3 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 $|\Delta J_z^k / J_z^k| < \tau/100$, where $\Delta J_z^k = J_z^k - J_{z,req}$, is met, we have found a solution, p_{nom}, that satisfies the isoperformance condition. If the relative

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

error is larger than $\tau/100$ we perform a sensitivity analysis, which yields the gradient vector (Jacobian) ∇J_z^k. This is used in a gradient search algorithm, which attempts to drive all performances to the isoperformance condition by updating p^k.

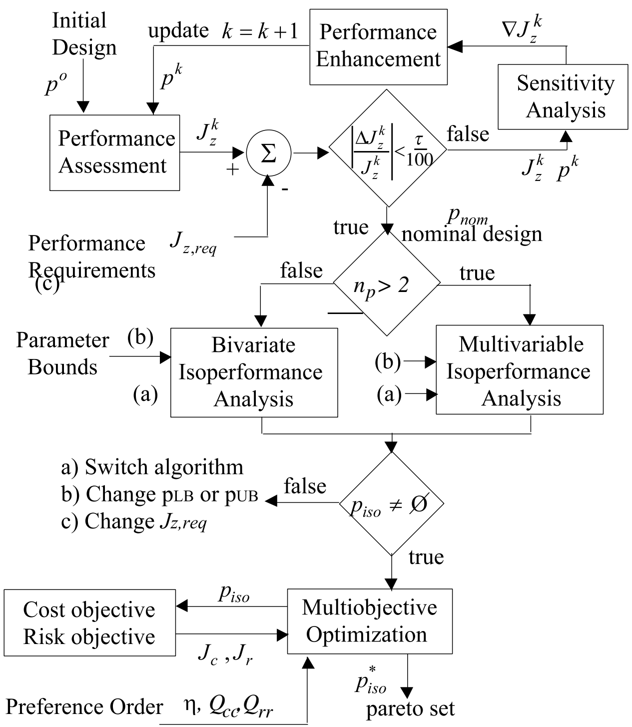

Figure 3. Isoperformance Roadmap

Once p_{nom} is found, we begin the actual isoperformance analysis. First, the problem space is restricted to only two variable parameters p_j, $j = 1, 2$, and one performance $n_z = 1$ (Section 2). The generalization to the multivariable case with $n_p > 2$ is the topic of Section 3. The main result from the isoperformance analysis is a set of points, p_{iso}, which approximate the isoperformance set **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 $J_{z,req}$ in order to obtain a feasible solution. The methodology then proceeds to the multiobjective optimization step, see [10]. The solution is not a single point design, but rather a family of pareto optimal designs, p^*_{iso}, which make up the "efficient" set **E**.

2 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.

2.1 Exhaustive Search Algorithm (I)

This method discretizes the parameter space 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. 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 as shown in Figure 4.

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 is used to find isoperformance points, $p_{iso,r}$[2]:

$$p_{iso,r} = \begin{bmatrix} p_{1,k} \\ p_{2,l} \end{bmatrix} + \frac{(J_z)_{k,l} - J_{z,req}}{(J_z)_{k,l} - (J_z)_{m,n}} \cdot \begin{bmatrix} p_{1,m} - p_{1,k} \\ p_{2,n} - p_{2,l} \end{bmatrix} \tag{7}$$

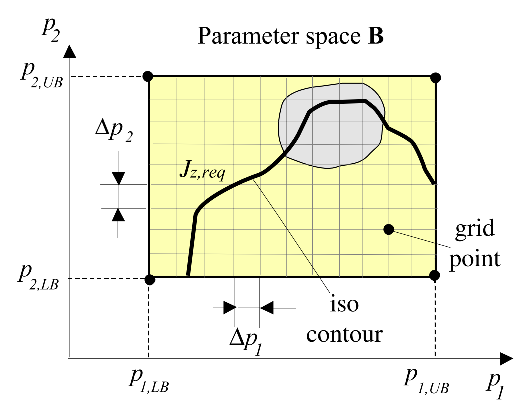

Figure 4. Algorithm I: Discretization of **B** in a linearly spaced grid with increments $\Delta p = [\Delta p_1, \Delta p_2]^T$.

2.2 Gradient-Based Contour Following (II)

The basic idea of gradient-based contour following is to first find an "isopoint", $p_{iso,k}$, which is known to yield the required performance $J_{z,req}$. A steepest gradient search is performed in order to intercept such a point, $p_{iso,k}$, starting from an initial guess [10]. One can then find a neighboring point $p_{iso,k+1} = p_{iso,k} + \Delta p_k$ such that $J_z(p_{iso,k} + \Delta p_k) = J_z(p_{iso,k+1}) = J_{z,req}$ by recalling the Taylor series expansion of the vector function $J_z(p)$ around $p_{iso,k}$:

$$J_z(p) = J_z(p_{iso,k}) + (\nabla J_z)^T\Big|_{p_{iso,k}} \cdot \Delta p + \frac{1}{2}\Delta p^T\, H\big|_{p_{iso,k}}\, \Delta p + \text{H.O.T.} \tag{8}$$

Note that $p = p_{iso,k} + \Delta p$ and that ∇J_z and H are the gradient vector and Hessian matrix, respectively. Neglect-

[2]This is similar to the MATLAB built-in function `contourc.m` used for contouring.

ing second-order and higher terms and setting the first order term (perturbation) to zero allows finding a neighboring point $p_{iso,k+1}$. Specifically, if

$$J_z\left(p_{iso,k+1}\right) = J_z\left(p_{iso,k} + \Delta p_k\right) \cong \\ J_z\left(p_{iso,k}\right) + \left(\nabla J_z\right)^T\Big|_{p_{iso,k}} \Delta p_k \equiv J_{z,req} \tag{9}$$

is to be true, then

$$\Delta J_{z,k} = \left(\nabla J_z\right)^T\Big|_{p_{iso,k}} \Delta p_k \equiv 0 \tag{10}$$

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

$$\Delta J_{z,k} = \frac{\partial J_z}{\partial p_1}\Big|_{p_{1,k}} \Delta p_{1,k} + \frac{\partial J_z}{\partial p_2}\Big|_{p_{2,k}} \Delta p_{2,k} \equiv 0 \tag{11}$$

Geometrically this condition corresponds to following the tangential vector t_k along the isocontour. Figure 5 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.

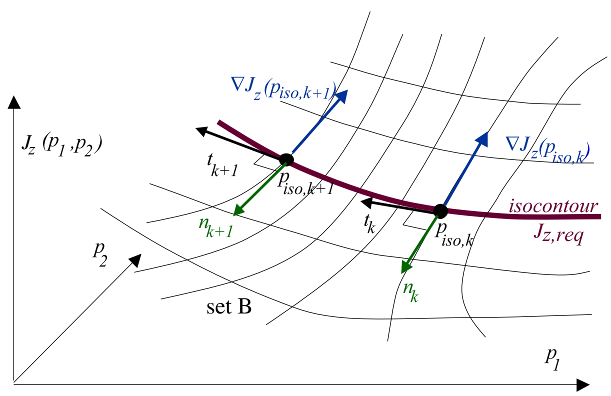

Figure 5. Algorithm II: Depiction of gradient vector ∇J_z, normal vector n and tangential vector t along the isoperformance contour.

There are two ways in which t_k can be obtained from $\nabla J_z(p_k)$. First one can compute the normal vector n_k and rotate it by 90 degrees to obtain the tangential vector t_k.

$$t_k = \mathcal{R} \cdot n_k = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \cdot n_k \tag{12}$$

The second method is more general, since it is also applicable to the case of $n_z > 1$ performances and $n_p > 2$ parameters and utilizes the singular value decomposition (SVD) of the gradient vector [10].

With the step direction t_k and a judicious choice of step size α_k 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.

2.3 Progressive Spline Approximation (III)

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 [11] are leveraged for this algorithm. The progressive spline approximation algorithm assumes that the two endpoints a, b are on the parameter space boundary **B**, see Figure 6.

A mathematical description of an isoperformance contour as a k-th order vector spline, $P_l(t)$, is given as

$$P_l\left(t\right) = \begin{bmatrix} p_{iso,1}\left(t\right) \\ p_{iso,2}\left(t\right) \end{bmatrix} = \begin{bmatrix} s_1\left(t\right) \\ s_2\left(t\right) \end{bmatrix} = p_{iso}(t) \tag{13}$$

where

$$t \in [0,1] \mapsto P_l\left(t\right) \in [a,b] \tag{14}$$

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

$$s_j\left(t\right) = f_{j,l}\left(t\right) \text{ for } j = 1,2 \text{ and } \ \forall \ l \tag{15}$$

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 $t = 1.0$. The functional approximation for each piece is then given as

$$f_{j,l}\left(t\right) = \sum_{i=1}^{k} \frac{\left(t-\zeta_l\right)^{k-i}}{(k-i)!} c_{j,l,i} \text{ where } \ t \in [\zeta_l \ldots \zeta_{l+1}] \tag{16}$$

Note that *all* relevant information is contained in the break point sequence, $\zeta_1 \ldots \zeta_{l+1}$ and in the polynomial coefficient array $c_{j,l,i}$. Next a bisection is performed at the mid-point of the first piece, $(t = 0.5)$, resulting in the point $p_{mid,1}$, which results in the closest point on the contour, $p_{iso,1}$, via gradient search. This bisection procedure is repeated until the midpoints of all pieces lie on the contour, subject to a tolerance τ as defined above, see Figure 6.

2.4 Algorithm Comparison

This subsection applies the three algorithms developed above to a single DOF sample problem and compares the answers. Figure 7 shows a single degree-of-freedom (SDOF) 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 with unity DC-gain and a corner frequency ω_d [rad/sec]. The displacement x [m] of the mass is passed through a first order highpass filter with corner frequency ω_o [rad/sec], simulating the effect of an optical controller. The performance

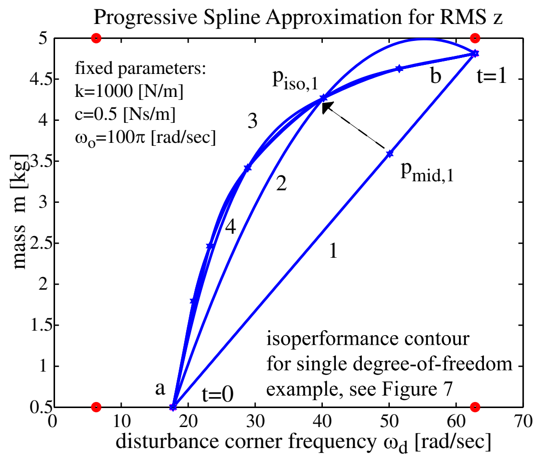

Figure 6. Progressive (cubic) spline approximation. Isoperformance analysis of SDOF problem with variables ω_d and m. The required performance is $J_{z,req} = 0.0008$ [m].

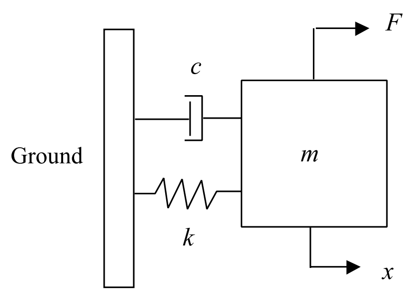

Figure 7. Single degree-of-freedom (SDOF) oscillator.

is the RMS of the filtered displacement output z, specifically $J_z = (E[z^T z])^{1/2}$, where E[ ] denotes the expectation operator [12]. 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 $p = [\omega_d \; m \; k \; c \; \omega_o]^T$. Here, 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.

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

$$\Upsilon_{iso} = \frac{100}{J_{z,req}} \cdot \left[\frac{\sum_{k=1}^{n_{iso}} \left[J_z(p_{iso,k}) - J_{z,req} \right]^2}{n_{iso}} \right]^{1/2} \quad (17)$$

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.

The isoperformance results for exhaustive search (I) are shown in Figure 8. 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. The isoperformance contours obtained with contour

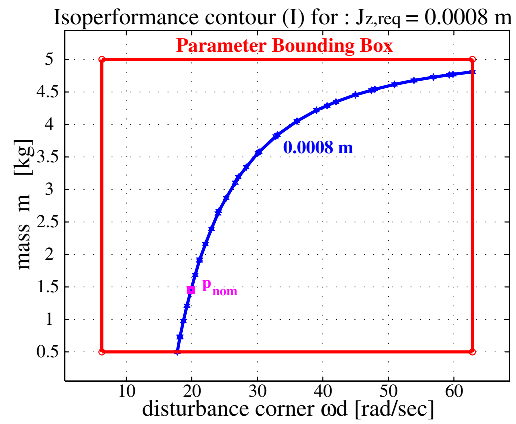

Figure 8. Algorithm I (Exhaustive Search): Isoperformance contour for single DOF problem (ω_d, m) with discretization $\Delta p = (1/20)[p_{UB} - p_{LB}]$ and a tolerance of $\tau = 1\%$.

following (not shown) and progressive spline approximation (Fig. 6) are very similar. A comparison of the computational cost among algorithms is shown in Table 1.

Table 1. Comparison of Algorithms I-III for SDOF problem.

Result	Exhaustive Search	Contour Following	Spline Approximation
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

Algorithm I is the most computationally expensive. 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 **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. A more detailed comparison is available in [10].

3 Multivariable Isoperformance Algorithms

This section generalizes the algorithms developed in the previous section to the multivariable case. Specifically, there can be more than two variable design 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 $n_p - n_z \geq 1$ has to be maintained in order for there to be a non-zero isoperformance set.

3.1 Branch and Bound Algorithm (I)

The exhaustive search algorithm in the multivariable case ($n_p > 2$) discretizes the parameter set **B** with a fine grid and evaluates all grid points. For $n_p >> 2$ this is not practical, even for relatively modest problems. Assume for example that $n_p = 6$ and that 50 grid points are used per parameter. Then the performance evaluation $p_j \mapsto J_z$ has to be carried out $50^6 = 1.56 \cdot 10^{10}$ times. If it took one second of CPU time per performance evaluation it would take 495.5 years to evaluate the entire trade space on a single computer.

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

$$[J_z(p_m) \geq J_{z,req} \geq J_z(p_n)] \cup [J_z(p_m) \leq J_{z,req} \leq J_z(p_n)] \tag{18}$$

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 9 for two dimensions. In the multivariable case the squares shown in Figure 9 are actually hyper-rectangles. The refinement continues with each generation, n_g, until the exit criterion

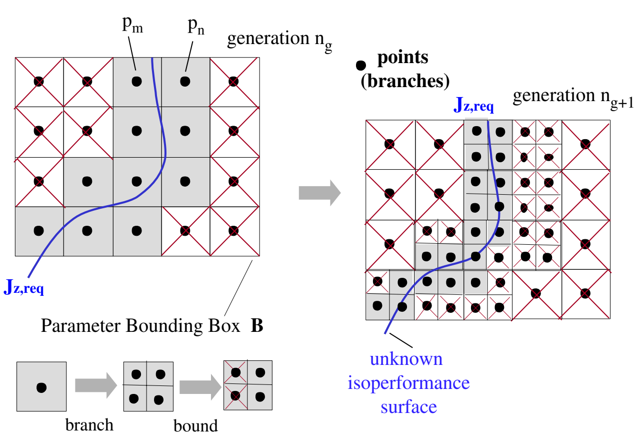

Figure 9. Multivariable Isoperformance (I): Branch-and-Bound graphic representation. Crossed out points (branches) are dropped in the next generation.

$$\Upsilon_{iso,n_g} < \tau \tag{19}$$

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.

3.2 Tangential Front Following (II)

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:

$$J_z\left(p^{k+1}\right) = J_z\left(p^k + \Delta p\right) = J_z\left(p^k\right) + \nabla J_z^T\Big|_{p^k} \Delta p + \text{HOT} \tag{20}$$

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

$$\nabla J_z = \begin{bmatrix} \frac{\partial J_{z,1}}{\partial p_1} & \frac{\partial J_{z,2}}{\partial p_1} & \cdots & \frac{\partial J_{z,n_z}}{\partial p_1} \\ \frac{\partial J_{z,1}}{\partial p_2} & \frac{\partial J_{z,2}}{\partial p_2} & \cdots & \frac{\partial J_{z,n_z}}{\partial p_2} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial J_{z,1}}{\partial p_{n_p}} & \frac{\partial J_{z,2}}{\partial p_{n_p}} & \cdots & \frac{\partial J_{z,n_z}}{\partial p_{n_p}} \end{bmatrix} \tag{21}$$

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.

$$U \Sigma V^T = \nabla J_z^T \tag{22}$$

and the individual matrices are as follows:

$$\begin{aligned} U &= \underbrace{\left[\; u_1 \;\; \cdots \;\; u_{n_z} \;\right]}_{n_z \times n_z} \\ \Sigma &= \underbrace{\left[\; \mathrm{diag}\left(\; \sigma_1 \;\; \cdots \;\; \sigma_{n_z} \;\right) \;\; 0_{n_z \times (n_p - n_z)} \;\right]}_{n_z \times n_p} \\ V &= \left[\; \underbrace{v_1 \;\; \cdots \;\; v_{n_z}}_{\text{column space}} \;\; \underbrace{v_{n_z+1} \;\; \cdots \;\; v_{n_p}}_{\text{null space } V_t} \;\right] \end{aligned} \tag{23}$$

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.

$$\Delta p = \alpha \cdot \left(\beta_1 v_{n_z+1} + \ldots + \beta_{n_p - n_z} v_{n_p}\right) = \alpha V_t \beta \tag{24}$$

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. The principal front points, as shown in Figure 10, 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 idea is to gradually explore the isoperformance set **I**, starting from an initial point, p_{nom}, and*

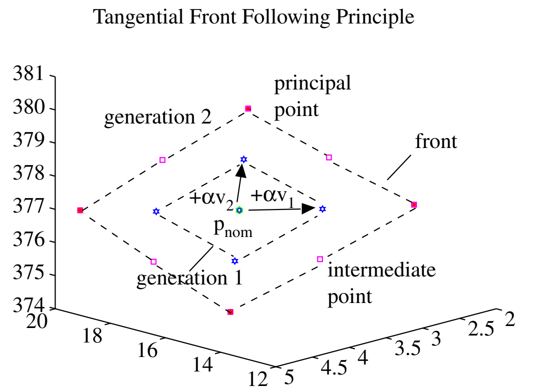

Figure 10. Tangential Front Following (II) principle.

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 **B** is intercepted. The main advantage of this algorithm is that it converts the computational complexity from a n_p to a $n_p - n_z$ problem.

3.3 Vector Spline Approximation (III)

This 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 **I**.* A t-parameterized vector spline in n_p-dimensional space connecting two points A and B can be written as

$$p(t) = \begin{bmatrix} p_1(t) \\ p_j(t) \\ \vdots \\ p_{n_p}(t) \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{k} \frac{(t-t_A)^{k-i}}{(k-i)!} \cdot c_{1,i} \\ \sum_{i=1}^{k} \frac{(t-t_A)^{k-i}}{(k-i)!} \cdot c_{j,i} \\ \vdots \\ \sum_{i=1}^{k} \frac{(t-t_A)^{k-i}}{(k-i)!} \cdot c_{n_p,i} \end{bmatrix} = C \cdot \hat{t} \tag{25}$$

where C is the vector spline coefficient matrix and $\hat{t}$ is a vector, which depends on the parameter t ,whereby $t \in [t_A, t_B]$. The vector spline approximation algorithm uses cubic splines of order, $k = 4$, we can then write:

$$\hat{t}(t) = \begin{bmatrix} 1 & t - t_A & \frac{(t-t_A)^2}{2} & \frac{(t-t_A)^3}{6} \end{bmatrix}^T \tag{26}$$

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 **B**. These points are found by first computing the performance vector, J_z, at all 2^{n_p} corner points and searching for border points, $p_{iso,border}$, which lie on an edge connecting two corner points. The next step is to connect the isoperformance border points with cubic splines along the boundary of **B**. In this step the mid-points of the border splines are also determined. Finally interior points of the isoperformance set **I** are obtained by computing the *centroid*. This can be considered to be the center point of **I**. An initial guess for the centroid is:

$$\hat{p}_{cent} = \begin{bmatrix} \hat{p}_{c,1} & \cdots & \hat{p}_{c,j} & \cdots & \hat{p}_{c,n_p} \end{bmatrix}^T \quad \text{where} \quad \hat{p}_{c,j} = \frac{1}{n_b} \sum_{i=1}^{n_b} p_{iso,border,i,j} \tag{27}$$

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

3.4 Multivariable Algorithm Comparison

The multivariable SDOF problem with three variable (design) parameters, ω_d, m and ω_o is considered. Again, the desired performance level is $J_{z,req} = 0.8$ [mm] RMS. Results for the single DOF oscillator problem are shown in Figure 11. The outline of the isoperformance surface can clearly be seen.

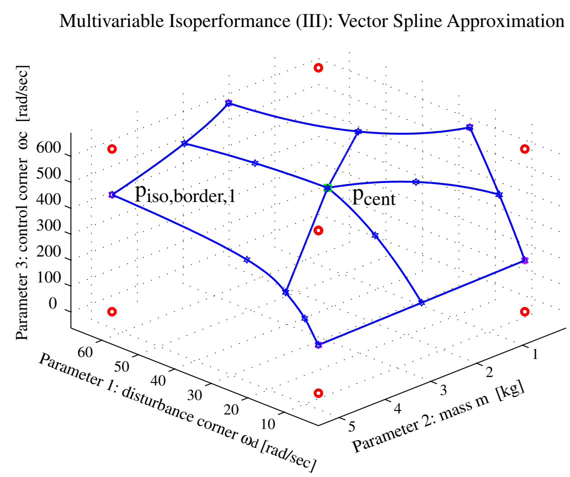

Figure 11. Multivariable Isoperformance (III): Vector Spline Approximation for SDOF sample problem.

A comparison of the multivariable algorithms is presented in Table 2.

Table 2. Comparison of multivariable algorithms for SDOF problem: Exhaustive Search, (I) Branch-and-Bound, (II) Tangential Front Following and (III) Vector Spline Approximation.

Metric	Ex Search	I	II	III
MFLOPS	6,163	891	106	1.5
CPU time [sec]	5078.19	498.56	69.59	4.45
Tolerance τ	1.5 %	2.5 %	1.5 %	1.5%
Error Υ_{iso} %	0.87	2.43	0.22	0.42
# of isopoints	2073	7421	4999	20

As expected the exhaustive search is the most expensive algorithm and requires almost 1.5 hours to run[3]. 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 (~ 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.

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 **B** is expected to exist, but cannot be found. Complexity considerations [10] suggest that the isoperformance problem is intrinsically non-polynomial in n_p. 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 expensive to solve.

4 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.

4.1 Testbed Description

The main feature of the DOLCE testbed is that system parameters can be varied over a large range. Figure 12 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. The performance is the root-mean-square (RMS) of the base plate displacement

$$J_z = E\left[z^T z\right]^{1/2} \tag{28}$$

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 gap sensor.

[3]Pentium III PC with 700 MHz processor

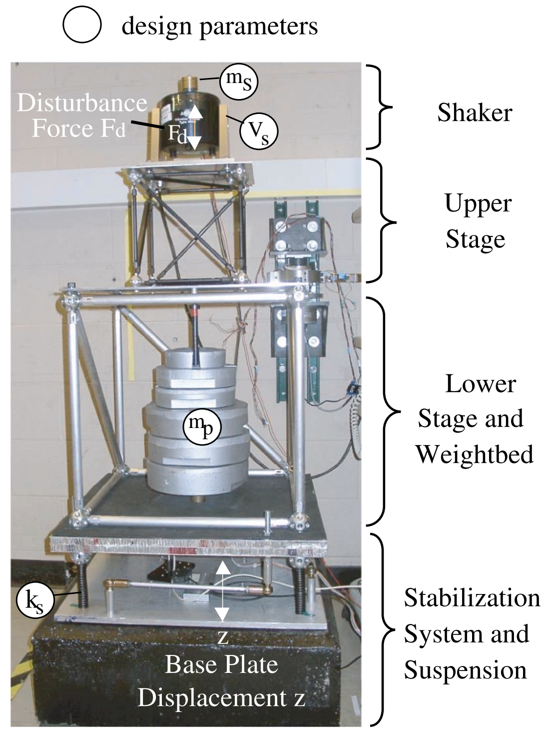

Figure 12. DOLCE Testbed

4.2 Experimental Approach

The experimental approach is presented in Figure 13. First the testbed was assembled, instrumented and calibrated. It was decided to conduct a bivariate isoperformance test, with the performance given by Equation (28). 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 2.1.

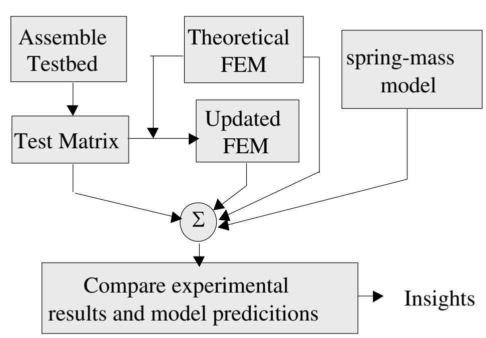

Figure 13. Experimental Approach

Independently and without knowledge of the experimental results an apriori finite element model (FEM) was constructed ("Theoretical 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 opto-mechanical systems in the conceptual and preliminary design phases. 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 at one design point in **B**. 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. The hope is that insights can be gained by comparing different performance contours for the experiment with the ones predicted for the models.

4.3 Testbed Characterization

The transfer function 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 14.

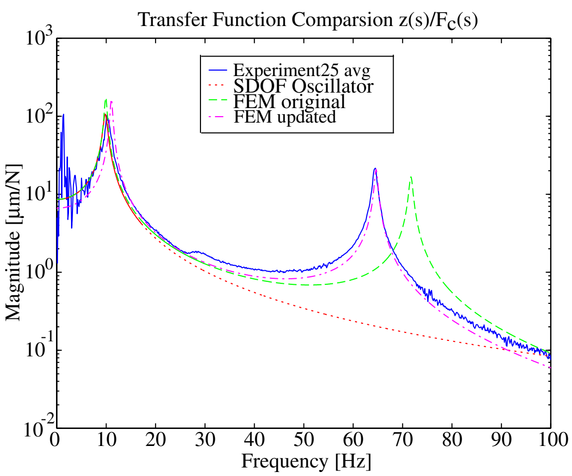

Figure 14. DOLCE transfer function $G_{zd} = Z(s)/F_d(s)$ for $m_p = 0$, $V_s = 1.0$

As can be seen, there are two observable modes in the bandwidth up to 100 Hz. The first mode at 10 Hz is the axial base suspension mode, where the testbed translates vertically up and down on the four suspension (compression) springs. The second mode at 65 Hz is the upper coupling plate bending mode, see Figure 15.

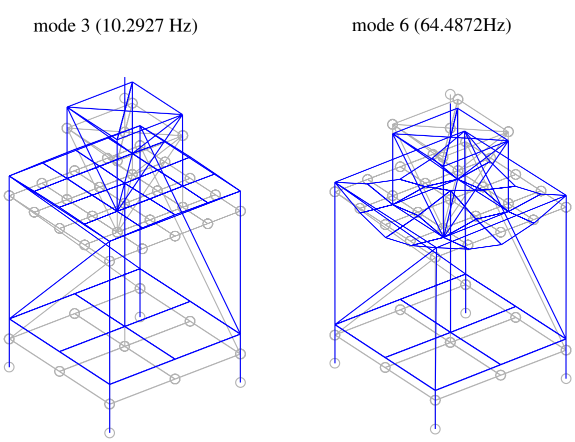

Figure 15. DOLCE Testbed Observable Modes

4.4 Isoperformance Results

The basis for obtaining the experimental isoperformance contours is the test matrix with V_s and m_p as described in Subsection 4.2. At each parameter combination the time histories of $F_d(t)$ and $z(t)$ were recorded and the performance $J_z = J_z(V_s, m_p)$ was computed with 25 averages. 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. 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] RMS displacement levels (Figure 16).

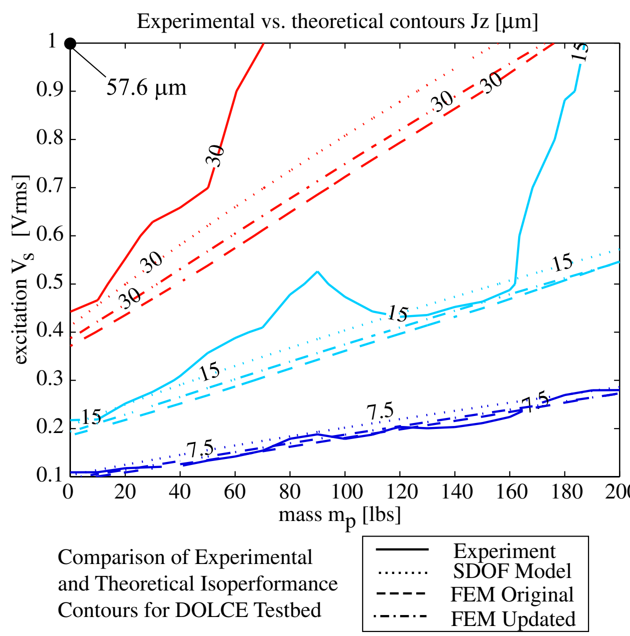

Figure 16. DOLCE Testbed Comparison of Experimental versus Theoretical Isoperformance Contours

Similar contours are predicted for the SDOF and FEM models. 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. This will require further analysis. In conclusion the isoperformance prediction capability is good at low disturbance levels which are representative of the vibration environment on precision opto-mechanical systems. Caution must be exercised if non-linearities are present in the system.

5 Summary and Recommendations

5.1 Summary

This paper attempts to develop and validate a novel design approach for complex multi-disciplinary systems. The isoperformance approach enhances the understanding of opto-mechanical systems by exploiting physical parameter sensitivity and performance information beyond the local neighborhood of a particular point design. It seeks to avoid situations, where very difficult requirements are levied onto one subsystem, while other subsystems hold substantial margins. It accomplishes this by fixing the desired output performance levels of a system apriori and searching for design points on the performance invariant surfaces in $\mathbb{R}^{n_p}$-space. An application of isoperformance to the large order NEXUS spacecraft model shown in Figure 1 is contained in [10].

5.2 Limitations

The current limitations of the isoperformance framework are that it assumes Linear-Time-Invariant (LTI) systems and operates on $\mathcal{H}_2$-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 in $\mathbb{R}^{n_p}$-space and work within a given topology or architecture.

5.3 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 "satisficing". 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 are:

- Perform a closed loop experimental validation
- Extend methodology to discrete parameter problems
- Extend methodology to non-steady-state processes
- Allow more complex constraints $g(p_j) \leq 0$
- Link to system architecture and conceptual design

Acknowledgments

This research was supported by the NASA Goddard Space Flight Center under contract No. NAG5-7839 and by the Jet Propulsion Laboratory under research contract No. JPL 961123. The above research contracts were monitored by Mr. Gary Mosier (GSFC) 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.

References

[1] Yoshikazu Sawaragi, Hirotaka Nakayama, and Tetsuzo Tanino. *Theory of Multiobjective Optimization*, volume 176 of *Mathematics in Science and Engineering*. Academic Press Inc., London, United Kingdom, 1 edition, 1985.

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

[3] Homero 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.

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

[5] John J. Gilheany. Optimum selection of dampers for freely vibrating multidegree of freedom systems. *Proceedings of Damping '89*, II(WRDC-TR-89-3116):FCC–1:18, February 8-10 1989.

[6] Marshall B. Jones and Robert S. Kennedy. Isoperformance curves in applied psychology. *Human Factors*, 38(1):167–182, 1996.

[7] A. Messac, R. Gueler, and K. Malek. Control-structure integrated design: A computational approach. In *Proceedings of the* $32^{\rm nd}$ *AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, pages 553–567, Baltimore, MD, April 1991. AIAA Paper No. 91-1161.

[8] J. I. Pritchard, H. M. Adelman, and J. Sobieszczanski-Sobieski. Optimization for minimum sensitivity to uncertain parameters. *AIAA Journal*, 34(7):1501–1504, July 1996.

[9] M. Milman, M. Salama, R. Scheid, and J. S. Gibson. Combined control-structural optimization. *Computational Mechanics*, (8):1–18, 1991.

[10] Olivier L. de Weck. *Multivariable Isoperformance Methodology for Precision Opto-Mechanical Systems*. PhD thesis, Massachusetts Institute of Technology, August 2001.

[11] Carl de Boor. *A practical guide to splines*, volume I of *Applied mathematical sciences*. Springer Verlag, New York, 1 edition, 1978.

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