July 13, 2004

Guest Lecture ESD.33

“Isoperformance”

Olivier de Weck

Why not performance-optimal ?

"The experience of the 1960's has shown that for military aircraft the cost of the final increment of performance usually is excessive in terms of other characteristics and that the overall system must be optimized, not just performance"

Ref: Current State of the Art of Multidisciplinary Design Optimization (MDO TC) - AIAA White Paper, Jan 15, 1991

⇨ TRW Experience

Industry designs not for optimal performance, but <u>according to targets</u> specified by a requirements document or contract - thus, optimize design for a set of GOALS.

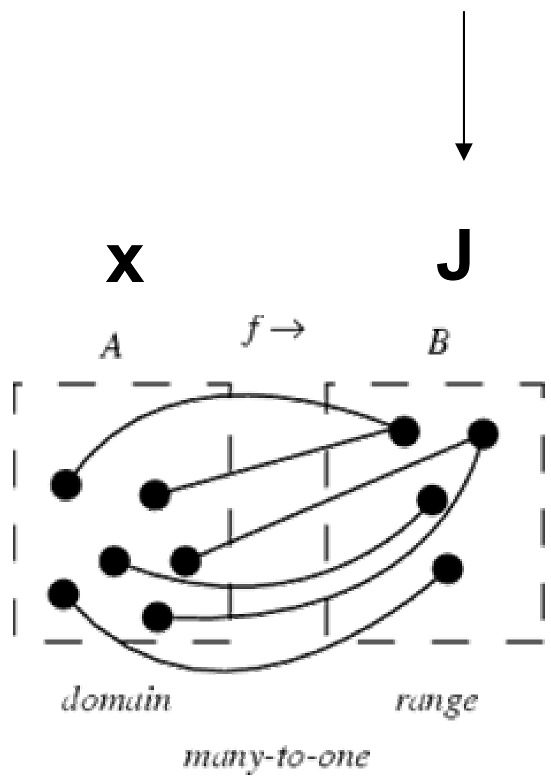
Lecture Outline

- Motivation - why isoperformance ?
- Example: Goal Seeking in Excel
- Case 1: Target vector **T** in Range
  = Isoperformance
- Case 2: Target vector **T** out of Range
  = Goal Programming
- Application to Spacecraft Design
- Stochastic Example: Baseball

Target
Vector

T

x **J**



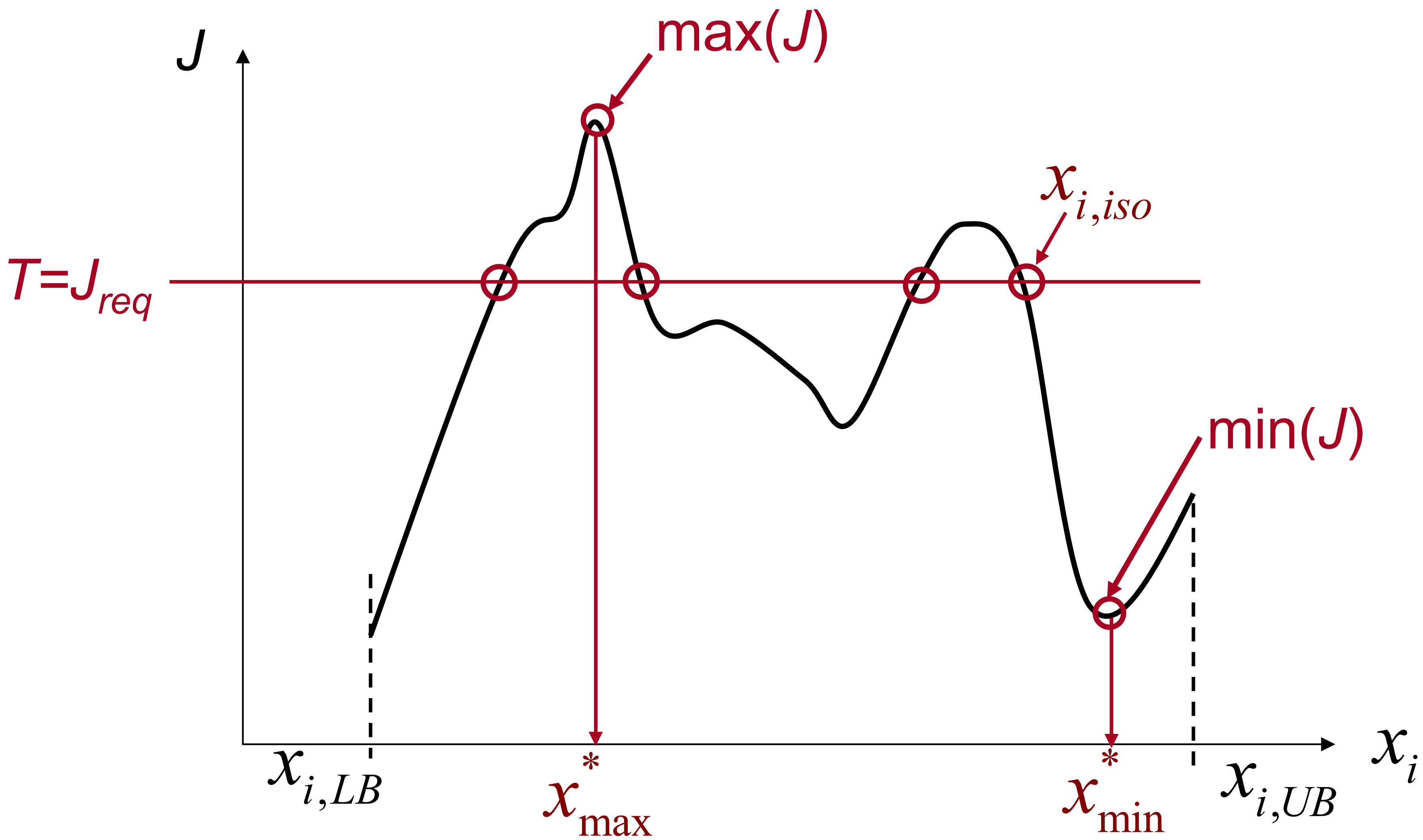
Forward Perspective

Choose **x** ⟶ What is **J** ?

Backward Perspective

Choose **J** ⟶ What **x** satisfy this?

Goal Seeking



Excel: Tools – Goal Seek

About Goal Seek

Goal Seek is part of a suite of commands sometimes called what-if analysis tools. When you know the desired result of a single formula but not the input value the formula needs to determine the result, you can use the Goal Seek feature available by clicking **Goal Seek** on the **Tools** menu. When goal seeking, Microsoft Excel varies the value in one specific cell until a formula that's dependent on that cell returns the result you want.

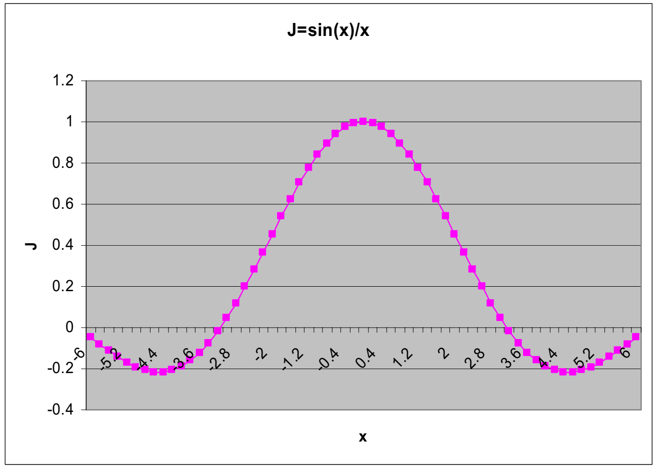
The value in cell B4 is the result of the formula =PMT(B3/12,B2,B1).

	A	B
1	Loan Amount	$ 100,000
2	Term in Months	180
3	Interest Rate	7.02%
4	Payment	($900.00)

Goal seek to determine the interest rate in cell B3 based on the payment in cell B4.

For example, use Goal Seek to change the interest rate in cell B3 incrementally until the payment value in B4 equals $900.00.

Excel - example



sin(x)/x - example

- single variable x
- no solution if *T* is out of range

Goal Seeking and Equality Constraints

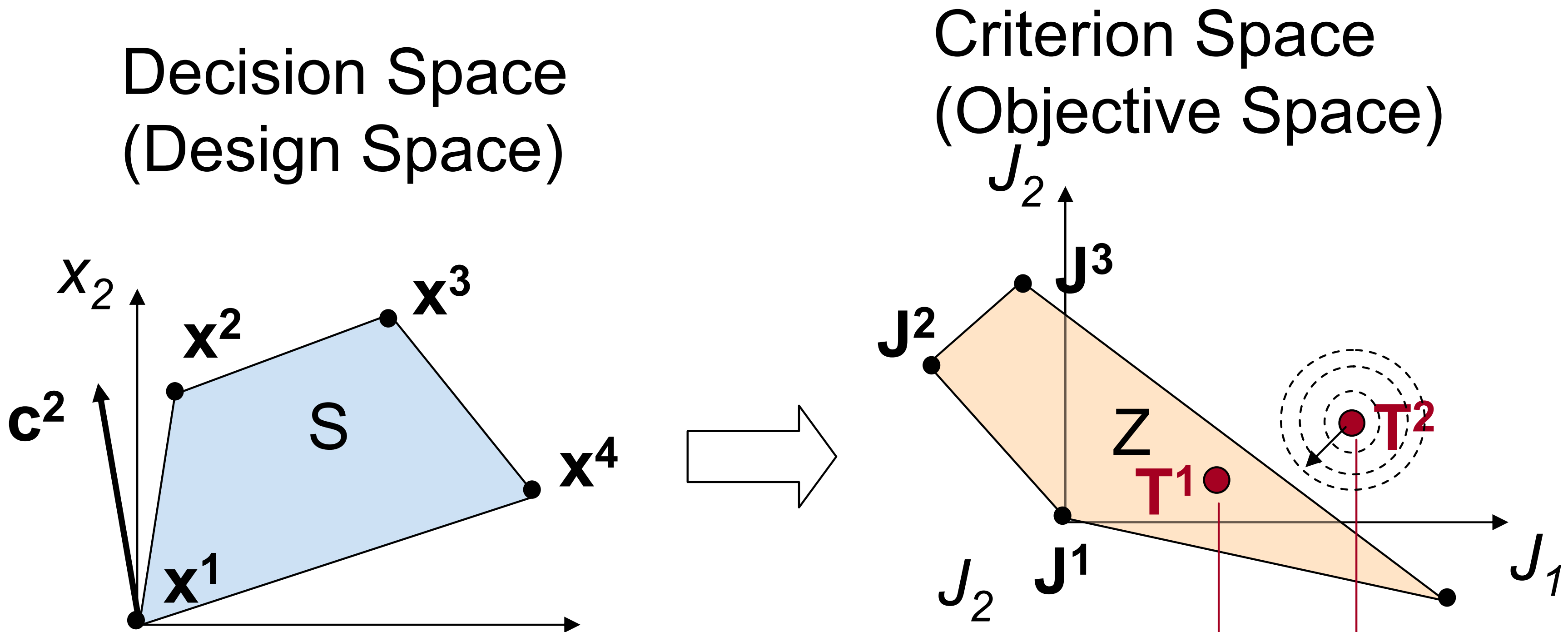
- Goal Seeking – is essentially the same as finding the set of points **x** that will satisfy the following "soft" equality constraint on the objective:

$$\text{Find all } \mathbf{x} \text{ such that } \left| \frac{J(\mathbf{x}) - J_{req}}{J_{req}} \right| \leq \varepsilon$$

Example Target Vector:

$$J_{req}(x) = \begin{bmatrix} m_{sat} \\ R_{data} \\ C_{sc} \end{bmatrix} \equiv \begin{bmatrix} 1000kg \\ 1.5Mbps \\ 15M\$ \end{bmatrix} \begin{matrix} \leftarrow \text{Target mass} \\ \leftarrow \text{Target data rate} \\ \leftarrow \text{Target Cost} \end{matrix}$$

Goal Programming vs. Isoperformance



Case 1: The target (goal) vector is in Z - usually get non-unique solutions = Isoperformance

Case 2: The target (goal) vector is not in Z - don't get a solution - find closest = Goal Programming

Isoperformance Analogy

Non-Uniqueness of Design if n > z

Performance: Buckling Load

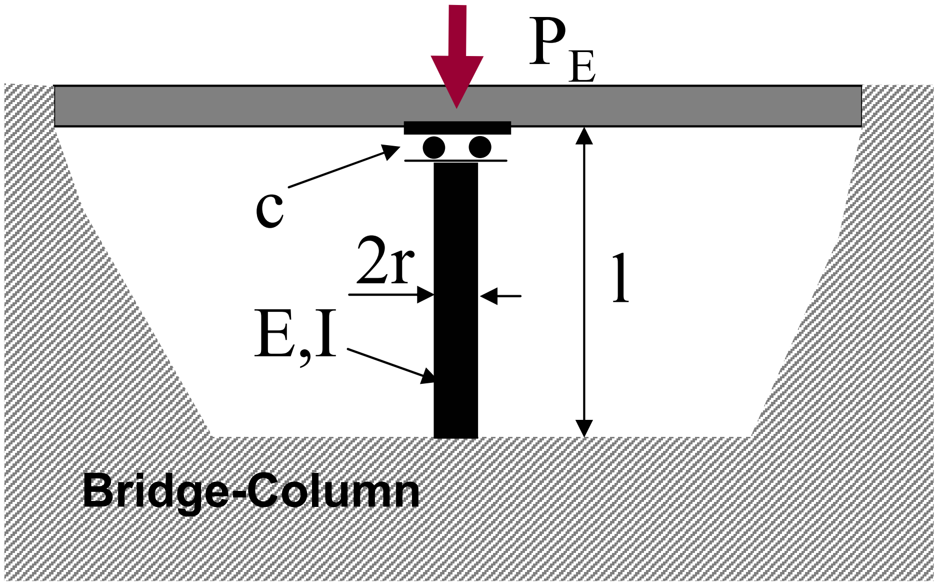
Constants: l=15 [m], c=2.05

$$P_E = \frac{c\pi^2 EI}{l^2}$$

Variable Parameters: ***E, I(r)***

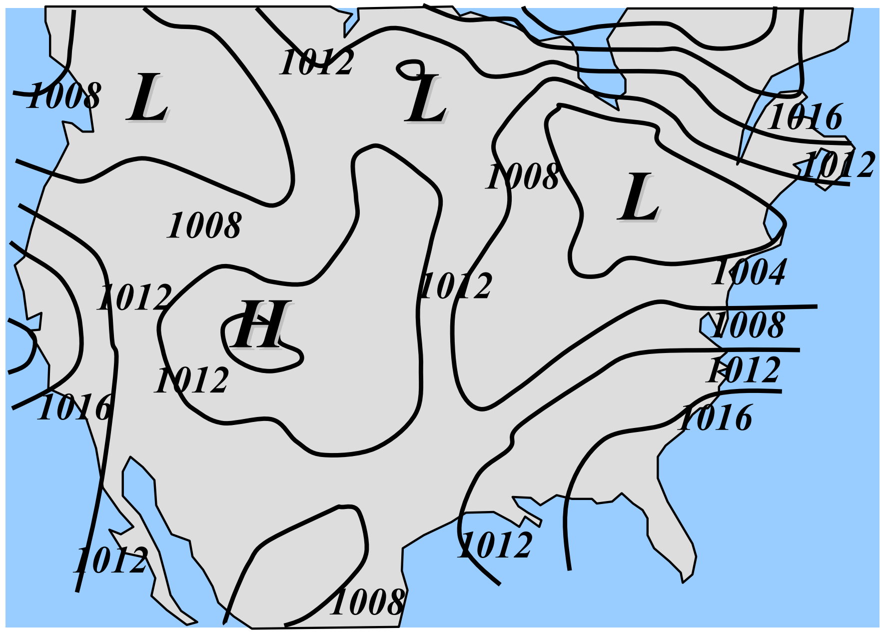
Requirement: $P_{E,REQ} = 1000$ metric tons

Solution 1: V2A steel, r=10 cm , E=19.1e+10
Solution 2: Al(99.9%), r=12.8 cm, E=7.1e+10



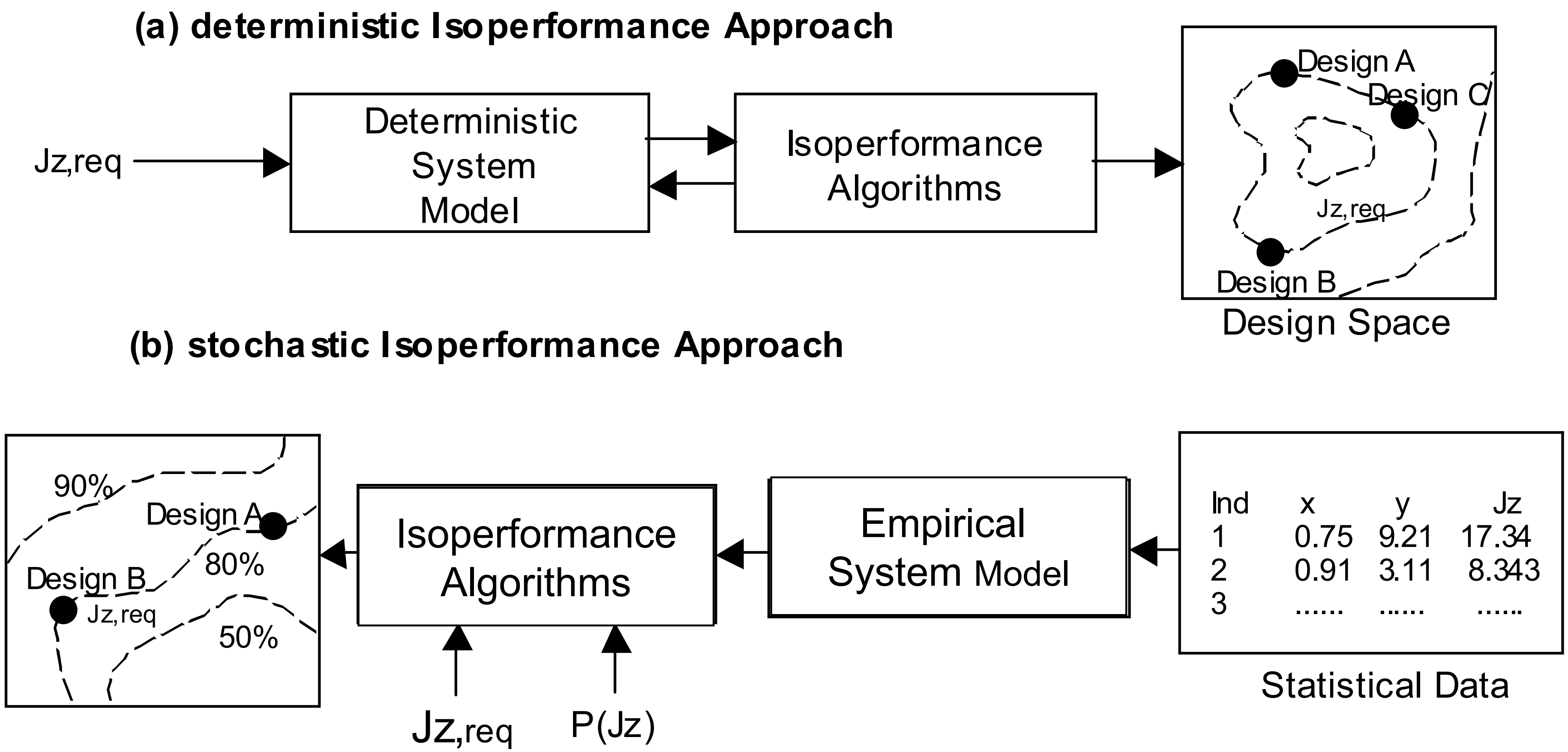
Analogy: Sea Level Pressure [mbar]
Chart: 1600 Z, Tue 9 May 2000

Isobars = Contours of Equal Pressure
Parameters = Longitude and Latitude



Isoperformance Contours = Locus of constant system performance
Parameters = e.g. Wheel Imbalance Us, Support Beam I_{xx}, Control Bandwidth ω_c

Isoperformance Approaches

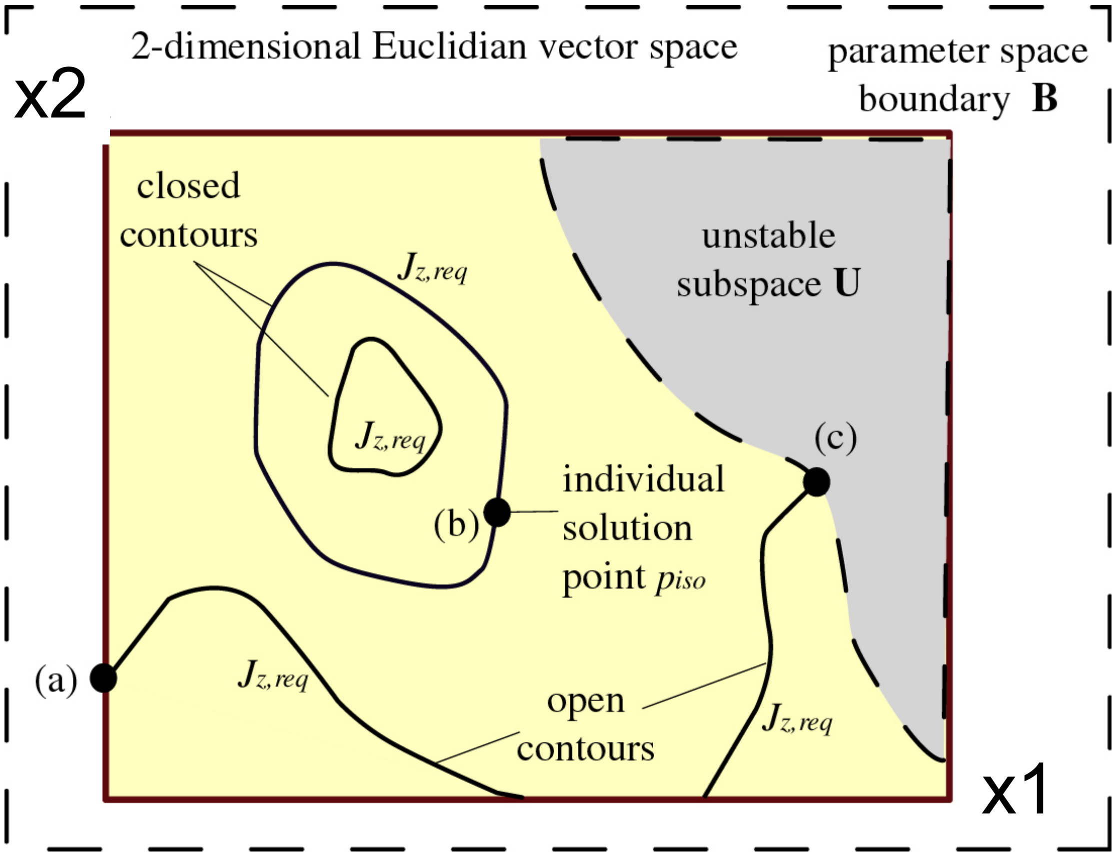


Bivariate Exhaustive Search (2D)

"Simple" Start: Bivariate Isoperformance Problem

$$\text{Performance } J_z(x_1, x_2):\ z = 1$$

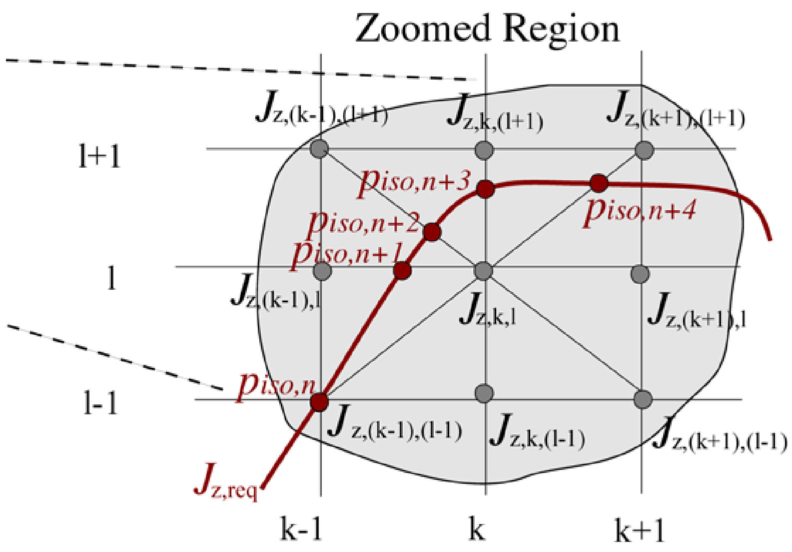
$$\text{Variables } x_j, j = 1,2:\quad n = 2$$



First Algorithm: **Exhaustive Search** coupled with bilinear interpolation

Number of points along j-th axis:

$$n_j = \left\lceil \frac{x_{j,UB} - x_{j,LB}}{\Delta x} \right\rceil$$



Can also use standard contouring code like MATLAB `contourc.m`

Contour Following (2D)

k-th isoperformance point: $x^k \mapsto J(x^k)$, where $\mathbb{R}^2 \mapsto \mathbb{R}$

Taylor series expansion

$$J_z(x) = J_z(x^k) + \underbrace{\nabla J_z^T\Big|_{x^k} \Delta x}_{\text{first order term}} + \underbrace{\frac{1}{2}\Delta x^T\, H\Big|_{x^k} \Delta x}_{\text{second order term}} + \text{H.O.T.}$$

$$\nabla J_z = \begin{bmatrix} \dfrac{\partial J_z}{\partial x_1} \\ \dfrac{\partial J_z}{\partial x_2} \end{bmatrix}$$

$$\nabla J_z^T\Big|_{p^k} \Delta x \equiv 0$$

$$t^k = \Re \cdot \underbrace{\frac{-\nabla J_z}{\left\|\nabla J_z\right\|}\Bigg|_{p^k}}_{n^k} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \cdot n^k$$

t^k: tangential step direction

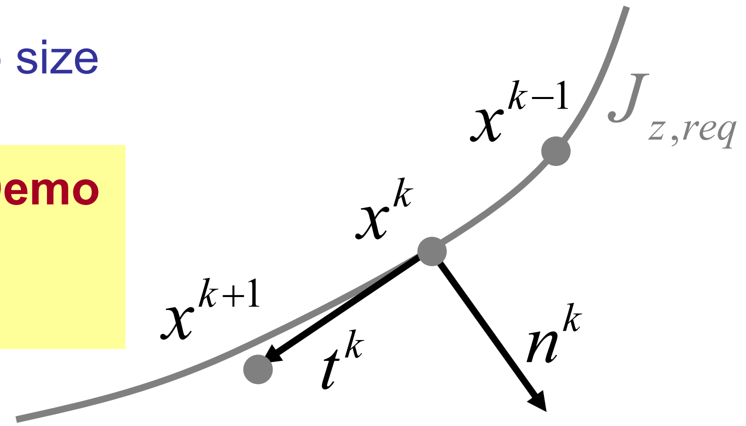
$$\alpha_k = \left[2\frac{\tau J_{z,req}}{100} \left(t_k^T\, H\Big|_{x^k} t_k \right)^{-1} \right]^{1/2}$$

H: Hessian

α^k : Step size

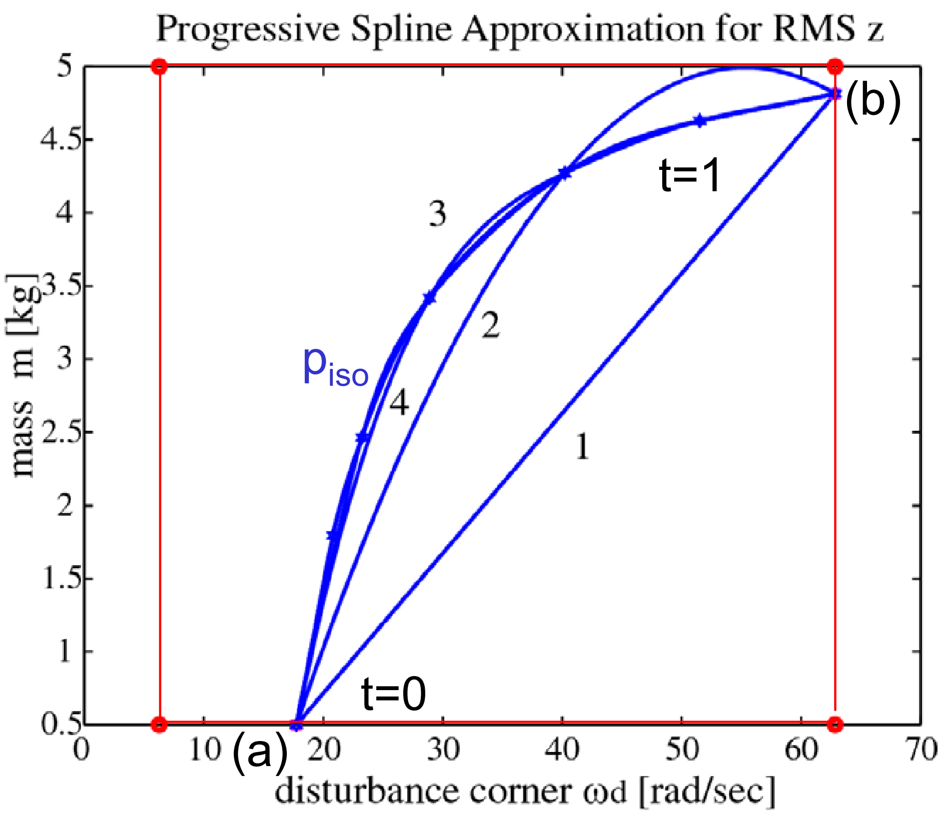
$$\Delta x = \alpha^k \cdot t^k$$

Demo



k+1-th isoperformance point: $x^{k+1} = x^k + \Delta x$

Progressive Spline Approximation (III)



- First find iso-points on boundary
- Then progressive spline approximation via segment-wise bisection
- Makes use of MATLAB spline toolbox , e.g. function **`csape.m`**

$$t \mapsto P_l(t) = \begin{bmatrix} x_{iso,1}(t) \\ x_{iso,2}(t) \end{bmatrix} = \begin{bmatrix} f_1(t) \\ f_2(t) \end{bmatrix}$$

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

Demo

Use cubic splines: k=4

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

Bivariate Algorithm Comparison

Metric	Exhaustive Search (I)	Contour Follow (II)	Spline Approx (III)
FLOPS	2,140,897	783,761	377,196
CPU time [sec]	1.15	0.55	0.33
Tolerance τ	1.0%	1.0%	1.0%
Actual Error γ_{iso}	0.057%	0.347%	0.087%
# of isopoints	35	45	7

Results for SDOF Problem

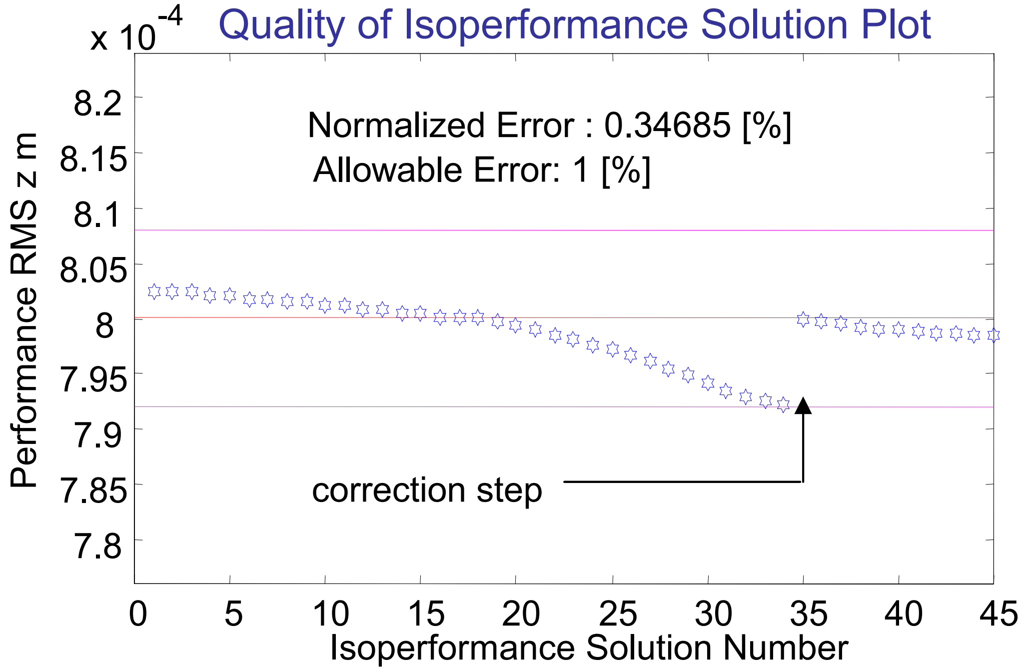
Conclusions:

(I) most general but expensive
(II) robust, but requires guesses
(III) most efficient, but requires monotonic performance J_z

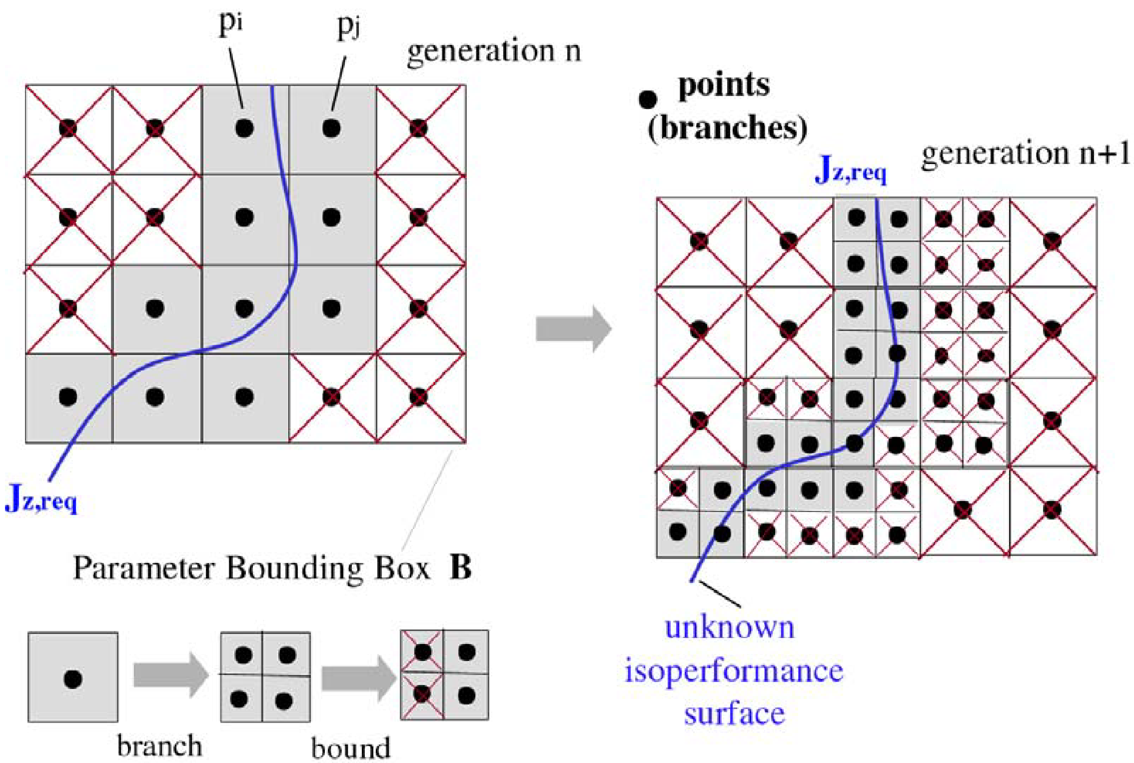
Isoperformance Quality Metric

"Normalized Error"

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

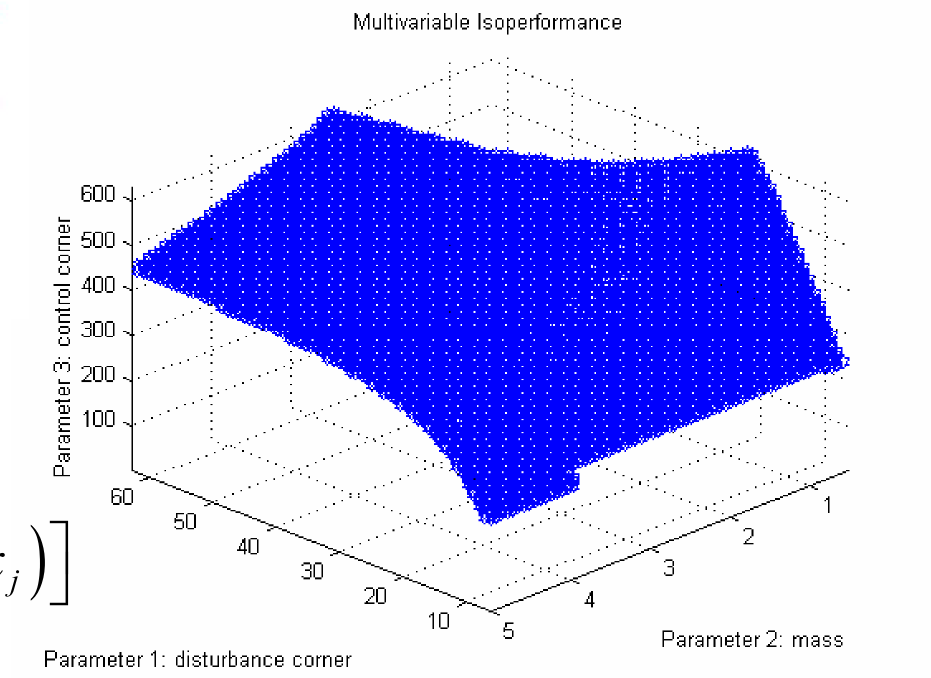


Multivariable Branch-and-Bound



Exhaustive Search requires n_p-nested loops -> NP-cost: e.g.

$$N = \prod_{j=1}^{n_p} \left[\frac{x_{UB,j} - x_{LB,j}}{\Delta x_j} \right]$$



Branch-and-Bound only retains points/branches which meet the condition:

$$\left[J_z(x_i) \ge J_{z,req} \ge J_z(x_j) \right] \cup \left[J_z(x_i) \le J_{z,req} \le J_z(x_j) \right]$$

Expensive for small tolerance τ

Need initial branches to be fine enough

Tangential Front Following

$$U = \underbrace{\begin{bmatrix} u_1 & \cdots & u_{n_z} \end{bmatrix}}_{zxz}$$

$$U\Sigma V^T = \nabla J_z^T$$

$$\Sigma = \underbrace{\begin{bmatrix} \operatorname{diag}\left(\sigma_1 \quad \cdots \quad \sigma_{n_z}\right) & 0_{n_z x (n_p - n_z)} \end{bmatrix}}_{zxn}$$

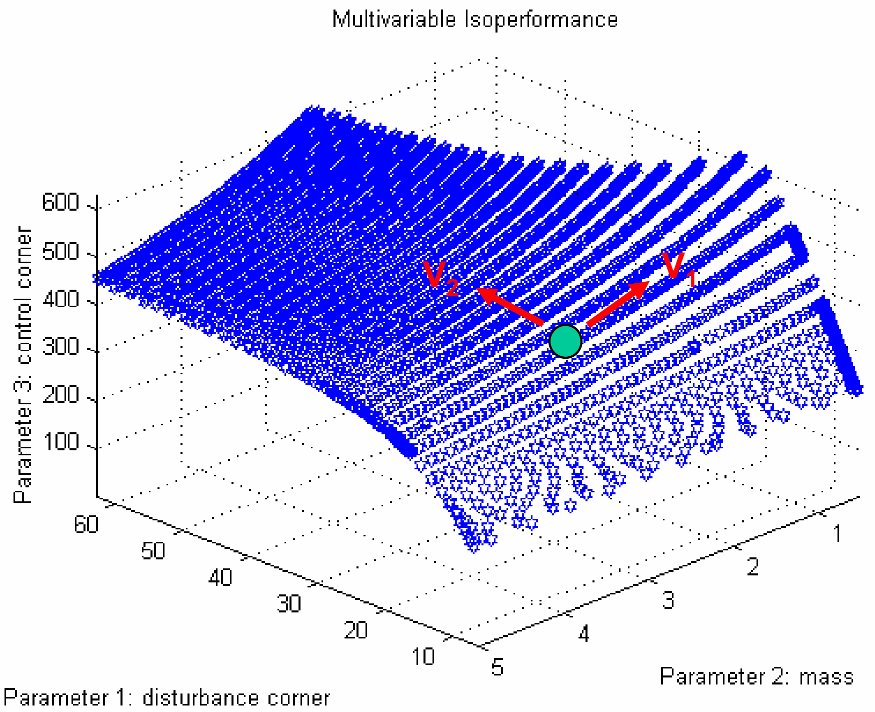
$$\nabla J_z = \begin{bmatrix} \frac{\partial J_1}{\partial x_1} & & \frac{\partial J_z}{\partial x_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial J_1}{\partial x_n} & & \frac{\partial J_z}{\partial x} \end{bmatrix}$$

$$V = \begin{bmatrix} \underbrace{v_1 \quad \cdots \quad v_z}_{\text{column space}} & \underbrace{v_{z+1} \quad \cdots \quad v_n}_{\text{nullspace}} \end{bmatrix}$$

$$\Delta x = \alpha \cdot \left(\beta_1 v_{z+1} + \ldots + \beta_{n-z} v_n\right) = \alpha V_t \beta$$

SVD of Jacobian provides V-matrix
V-matrix contains the orthonormal vectors of the nullspace.

Isoperformance set I is obtained by following the nullspace of the Jacobian !



Vector Spline Approximation

Tangential front following is more efficient than branch-and-bound but can still be expensive for n_p large.

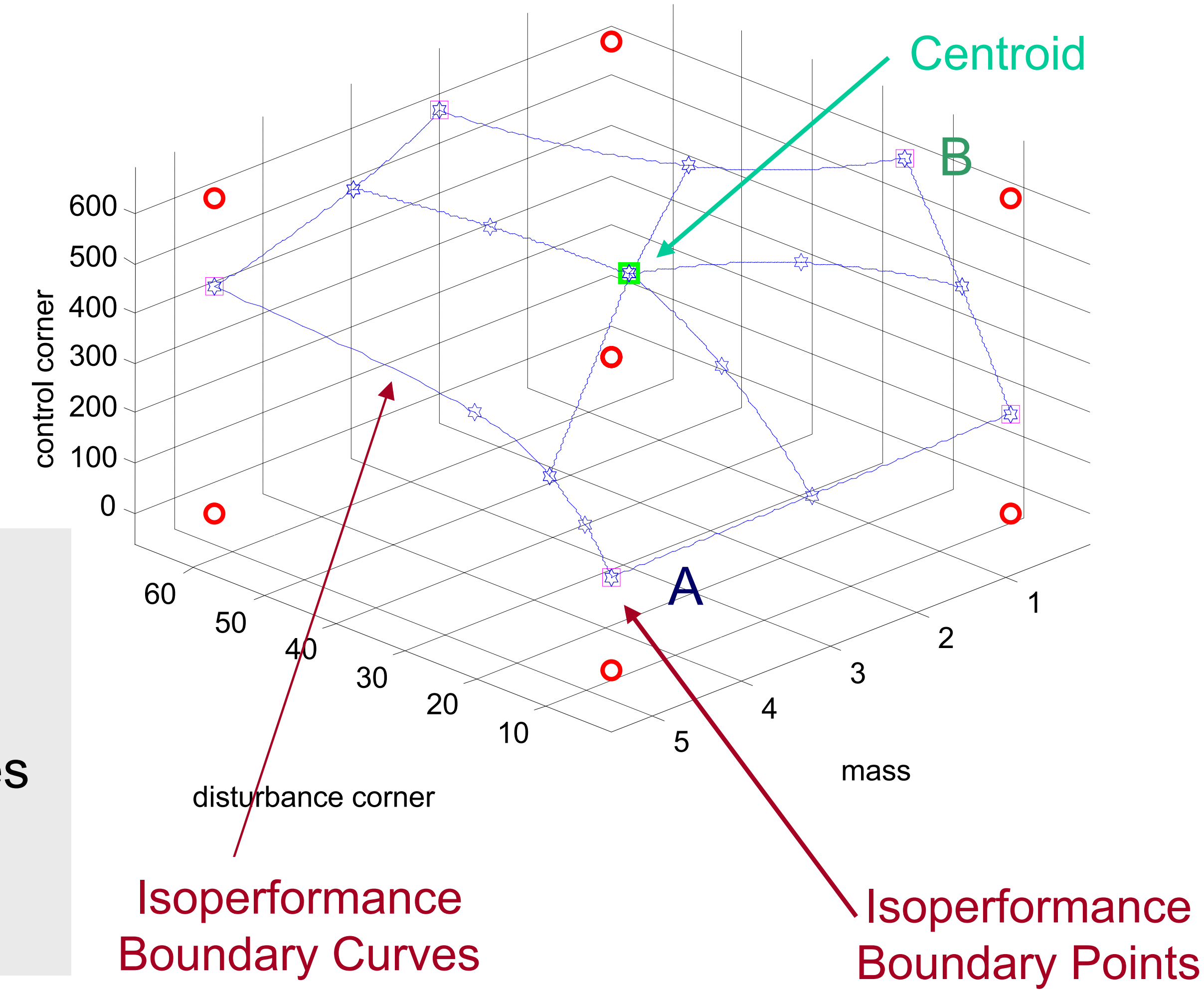
Idea: Find a representative subset off all isoperformance points, which are different from other.

“Frame-but-not-panels” analogy in construction

Algorithm:

1. Find Boundary (Edge) Points
2. Approximate Boundary curves
3. Find Centroid point
4. Approximate Internal curves

Vector Spline Approximation of Isoperformance Set



Multivariable Algorithm Comparison

Challenges if $n_p > 2$

- Computational complexity as a function of [n_z n_d n_p n_s]
- Visualization of isoperformance set in n_p-space

Problem Size:

z = **# of performances**

d = **# of disturbances**

n = **# of variables**

n_s = **# of states**

Table: Multivariable Algorithm Comparison for SDOF (n_p=3)

Metric	Exhaustive Search	Branch-and-Bound	Tang Front Following	V- Spline Approx
MFLOPS	6,163.72	891.35	106.04	1.49
CPU [sec]	5078.19	498.56	69.59	4.45
Error Y_{iso}	0.87 %	2.43%	0.22%	0.42%
# of points	2073	7421	4999	20

From Complexity Theory: Asymptotic Cost [FLOPS]

Exhaustive Search: $$\log\left(J_{exs}\right) \to n_p \log\alpha + 3\log n_s + c$$

Branch-and-Bound: $$\log\left(J_{bab}\right) \to n_g\left(n_p \log 2 + \log\beta\right) + 3\log n_s + c$$

Tang Front Follow: $$\log\left(J_{tff}\right) \to \left(n_p - n_z\right)\log\gamma + \log\left(1+n_z\right) + 3\log n_s + c$$

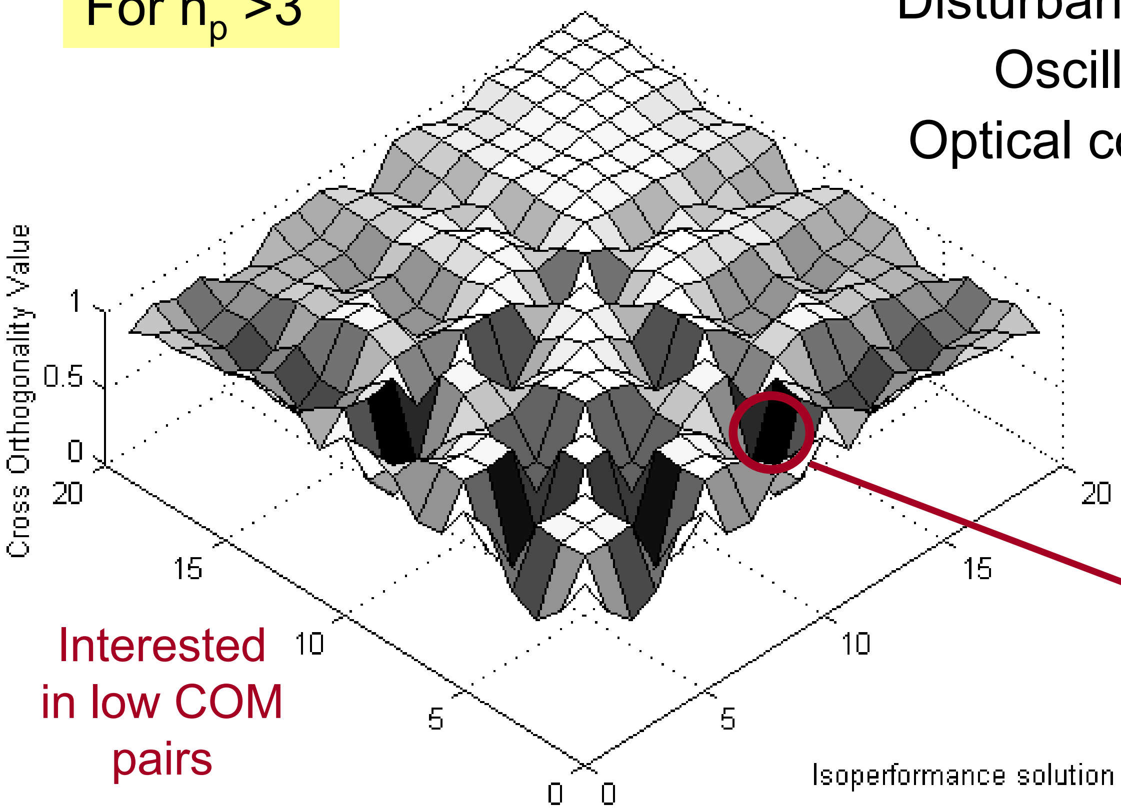
V-Spline Approx: $$\log\left(J_{vsa}\right) \to n_p \log 2 + 3\log n_s + \log(n_z+1) + c$$

Conclusion: Isoperformance problem is non-polynomial in n_p

Graphics: Radar Plots

For $n_p > 3$

	A	B
Disturbance corner ω_d	6.2832	21.3705
Oscillator mass m	5.0000	0.5000
Optical control bw ω_o	186.5751	628.3185

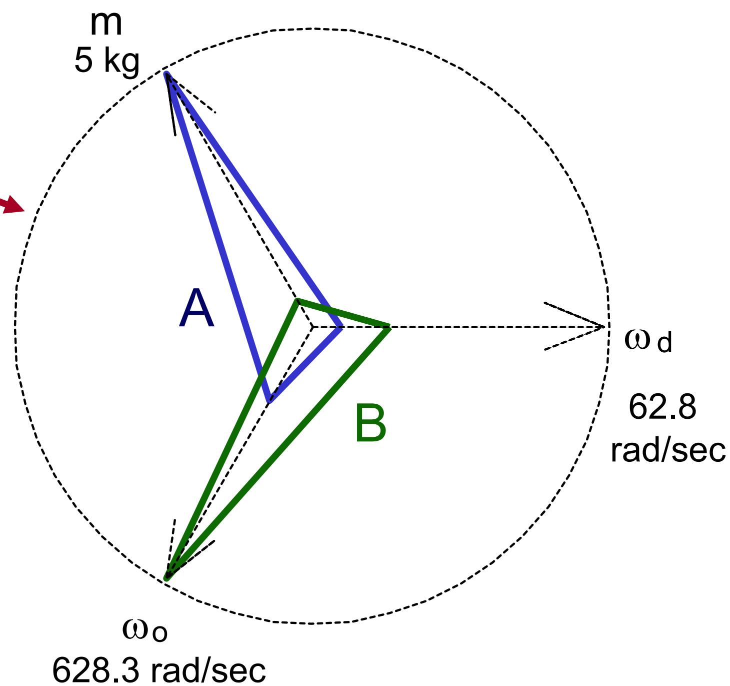


Interested in low COM pairs

Cross Orthogonality Matrix

$$COM(i,j) = \frac{p_{iso,i} \cdot p_{iso,j}}{\left|p_{iso,i}\right| \cdot \left|p_{iso,j}\right|}$$

Multi-Dimensional Comparison of Isoperformance Points



Nexus Case Study

Purpose of this case study:

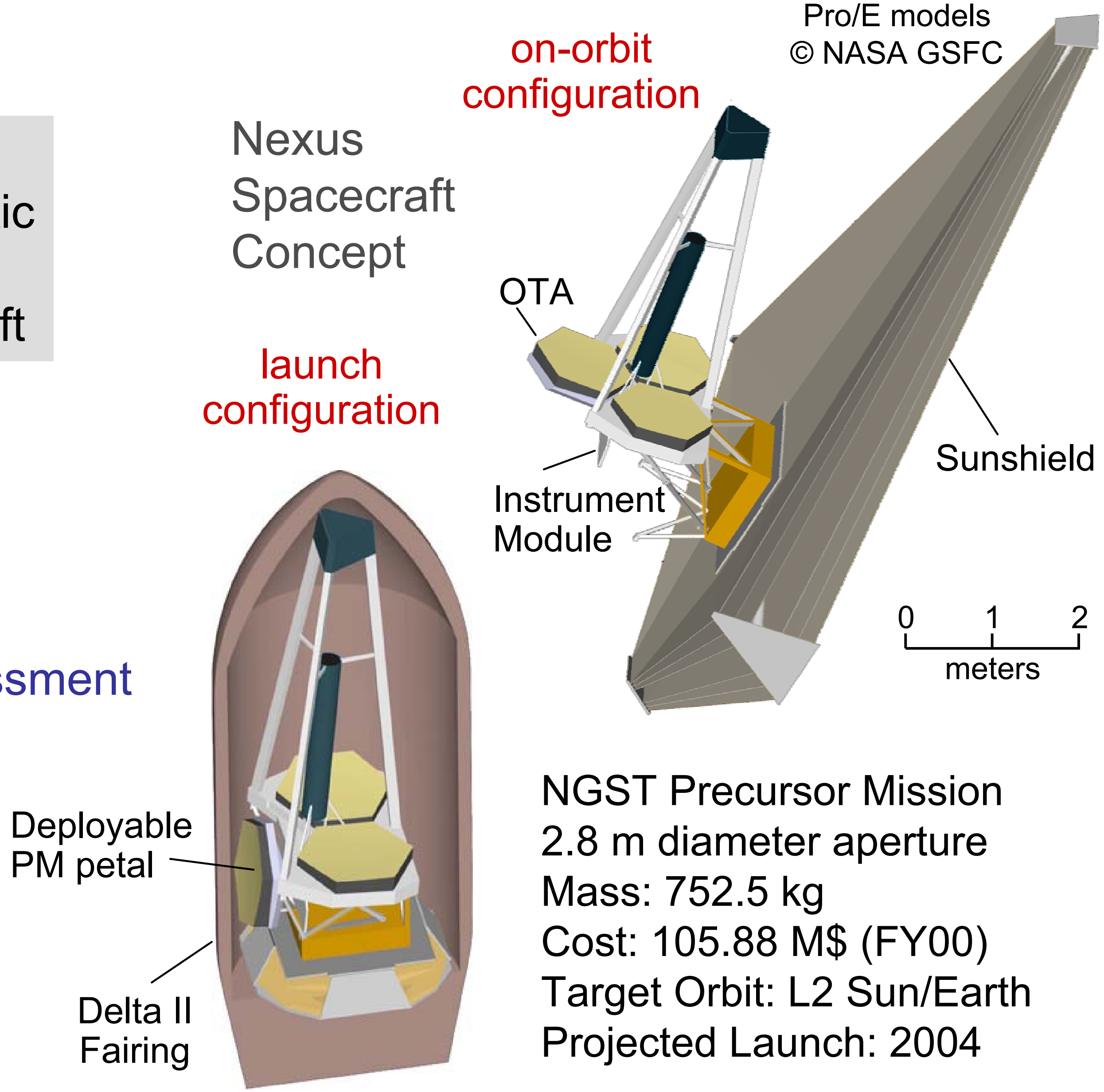
Demonstrate the usefulness of Isoperformance on a realistic conceptual design model of a high-performance spacecraft

The following results are shown:

- Integrated Modeling
- Nexus Block Diagram
- Baseline Performance Assessment
- Sensitivity Analysis
- Isoperformance Analysis (2)
- Multiobjective Optimization
- Error Budgeting

Details are contained in CH7

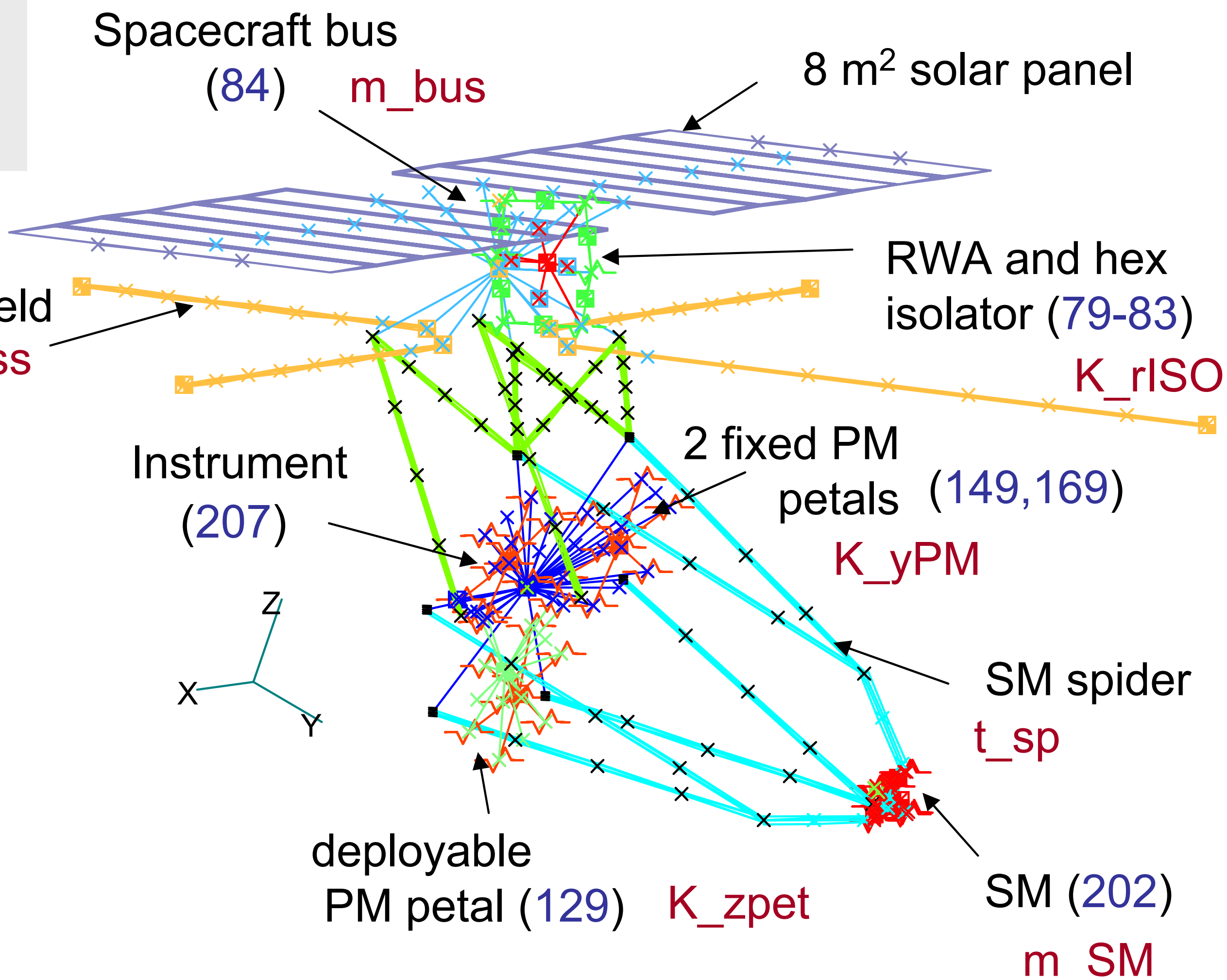
Nexus Spacecraft Concept



NGST Precursor Mission
2.8 m diameter aperture
Mass: 752.5 kg
Cost: 105.88 M$ (FY00)
Target Orbit: L2 Sun/Earth
Projected Launch: 2004

Nexus Integrated Model

Legend:
Design Parameters
(I/O Nodes)



Cassegrain Telescope:

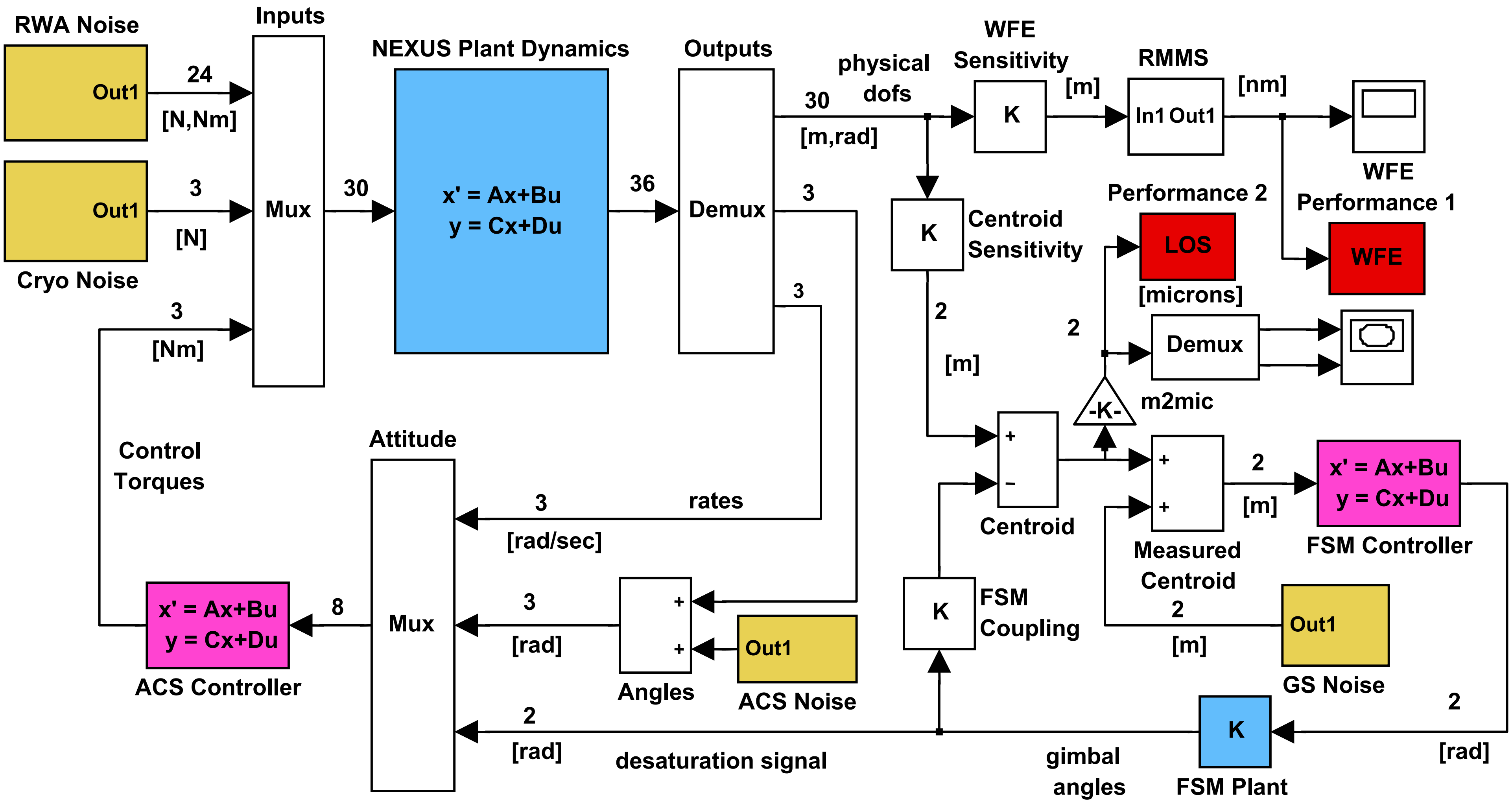
PM (2.8 m)
PM f/# 1.25
SM (0.27 m)
f/24 OTA

Structural Model (FEM)
(Nastran, IMOS) → Ω, Φ

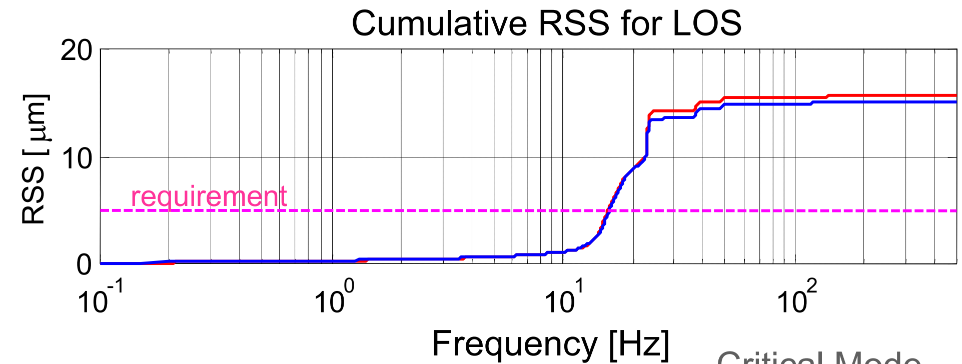
Nexus Block Diagram

Number of performances: n_z=2
Number of design parameters: n_p=25

Number of states n_s= 320
Number of disturbance sources: n_d=4



Initial Performance Assessment $J_z(p^o)$

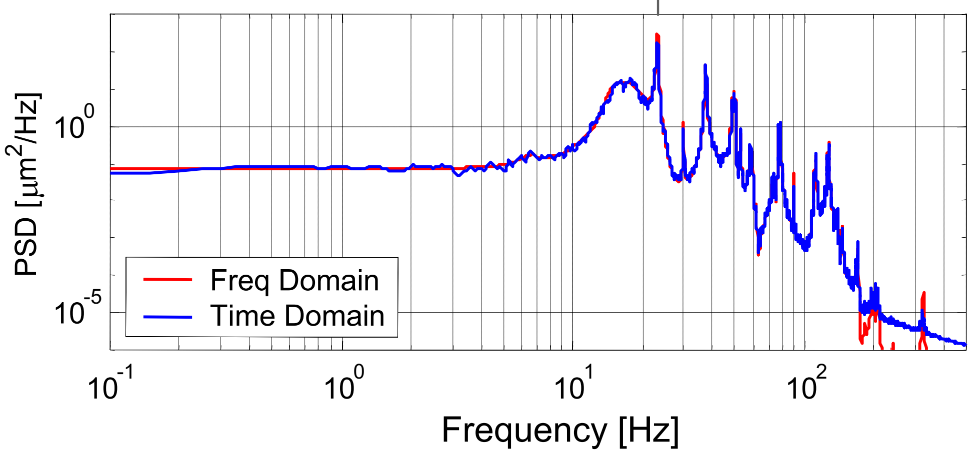


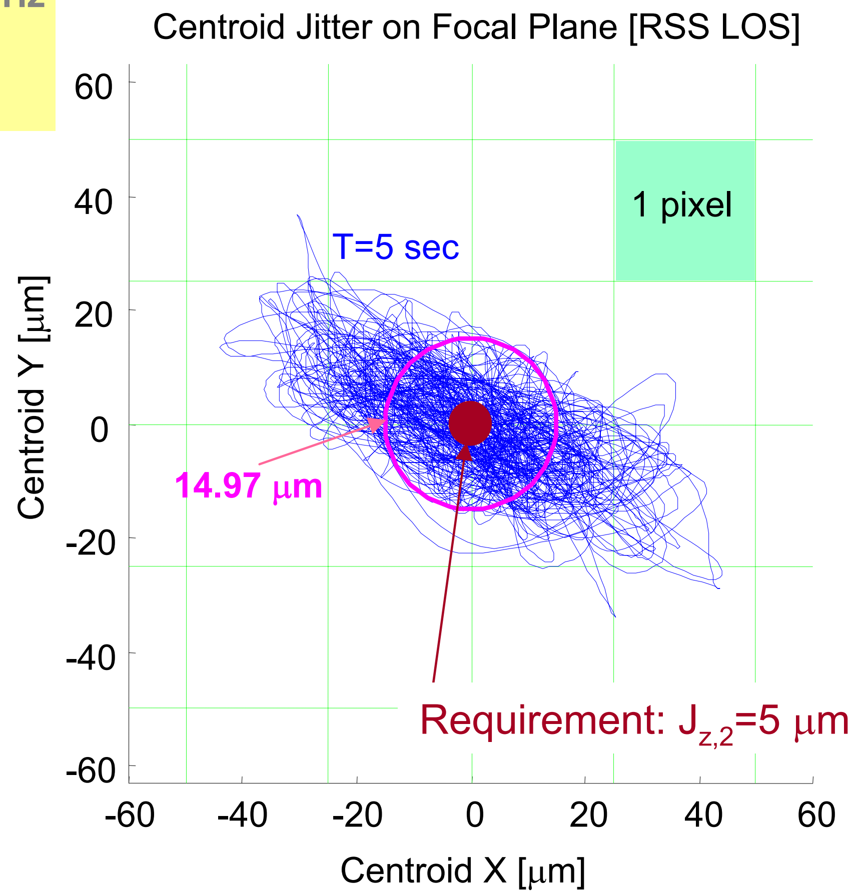
Results

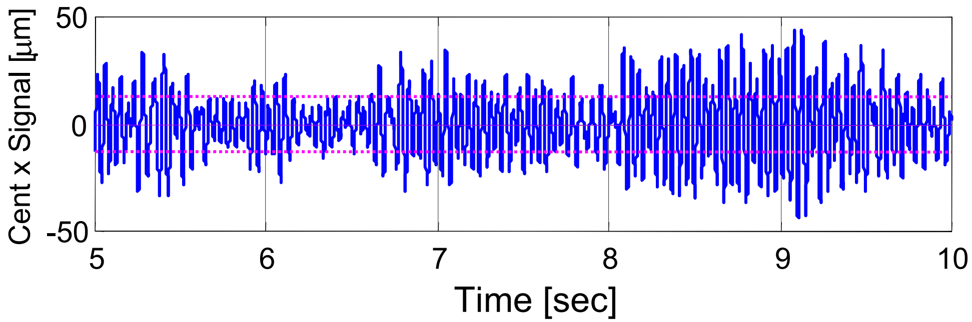
	Lyap/Freq	Time	
$J_{z,1}$ (RMMS WFE)	**25.61**	**19.51**	[nm]
$J_{z,2}$ (RSS LOS)	**15.51**	**14.97**	[µm]

Critical Mode

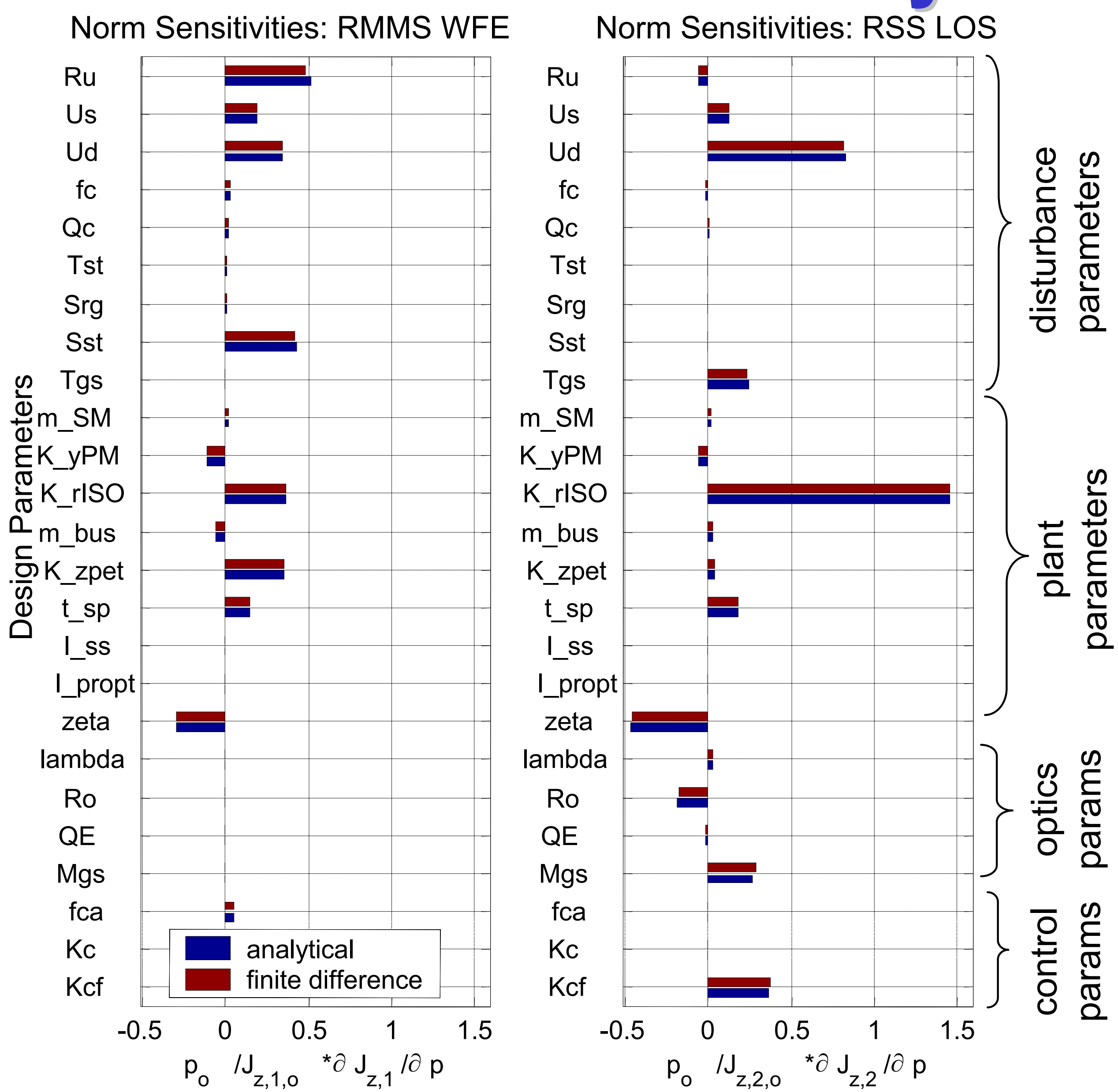
23.1 Hz







Nexus Sensitivity Analysis



Graphical Representation of Jacobian evaluated at design p_o, normalized for comparison.

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

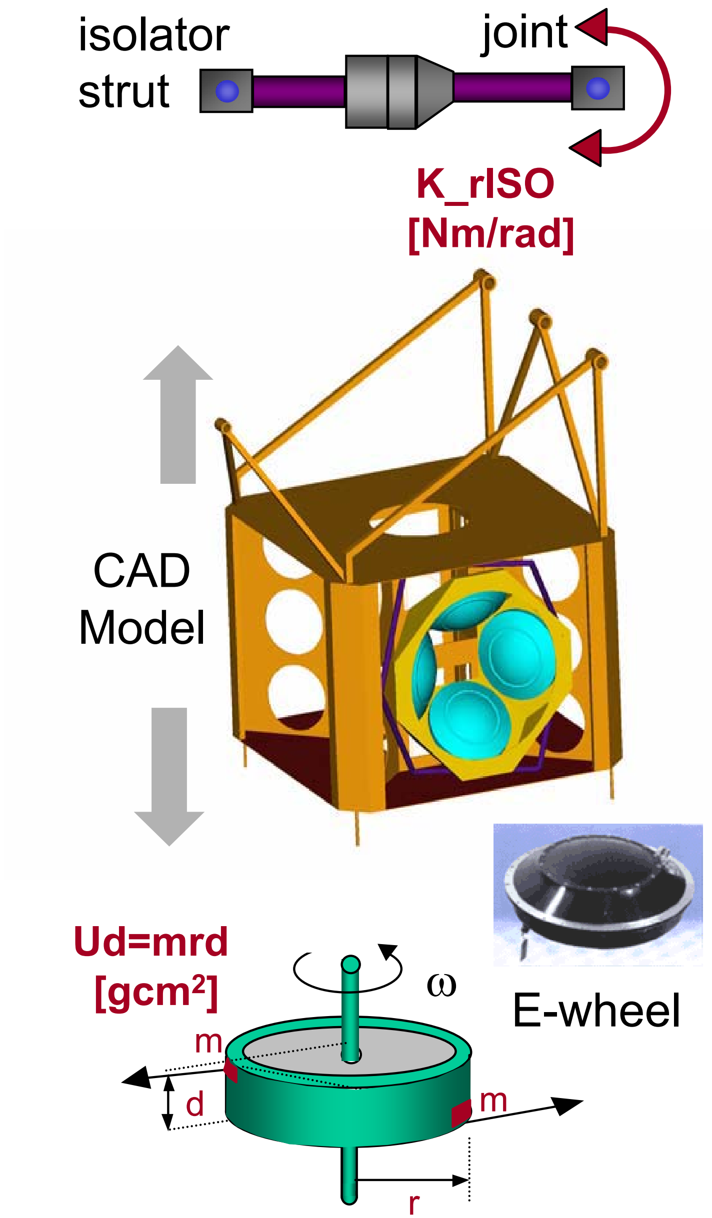
RMMS WFE most sensitive to:

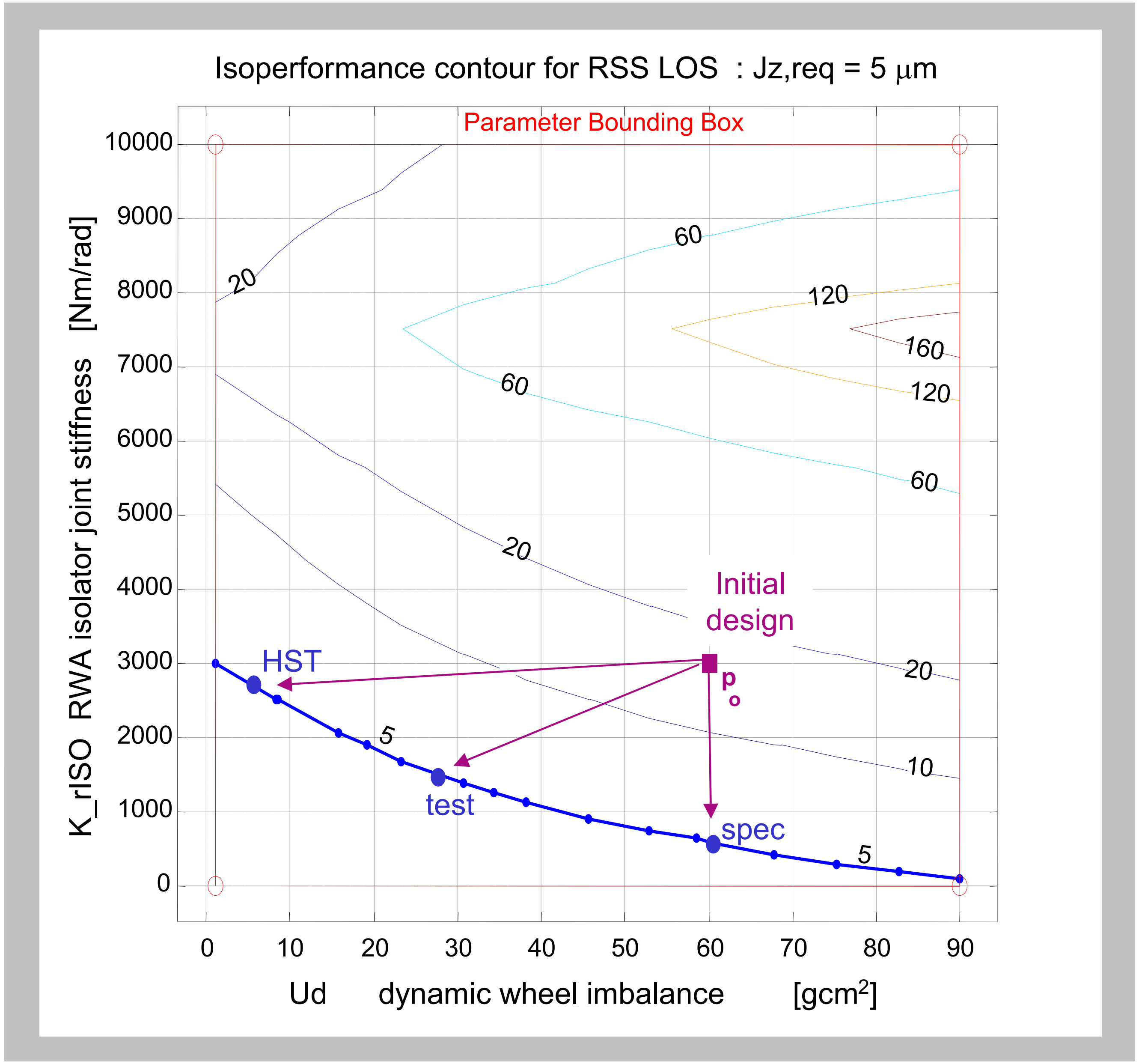
Ru - upper op wheel speed [RPM]
Sst - star track noise 1σ [asec]
K_rISO - isolator joint stiffness [Nm/rad]
K_zpet - deploy petal stiffness [N/m]

RSS LOS most sensitive to:

Ud - dynamic wheel imbalance [gcm^2]
K_rISO - isolator joint stiffness [Nm/rad]
zeta - proportional damping ratio [-]
Mgs - guide star magnitude [mag]
Kcf - FSM controller gain [-]

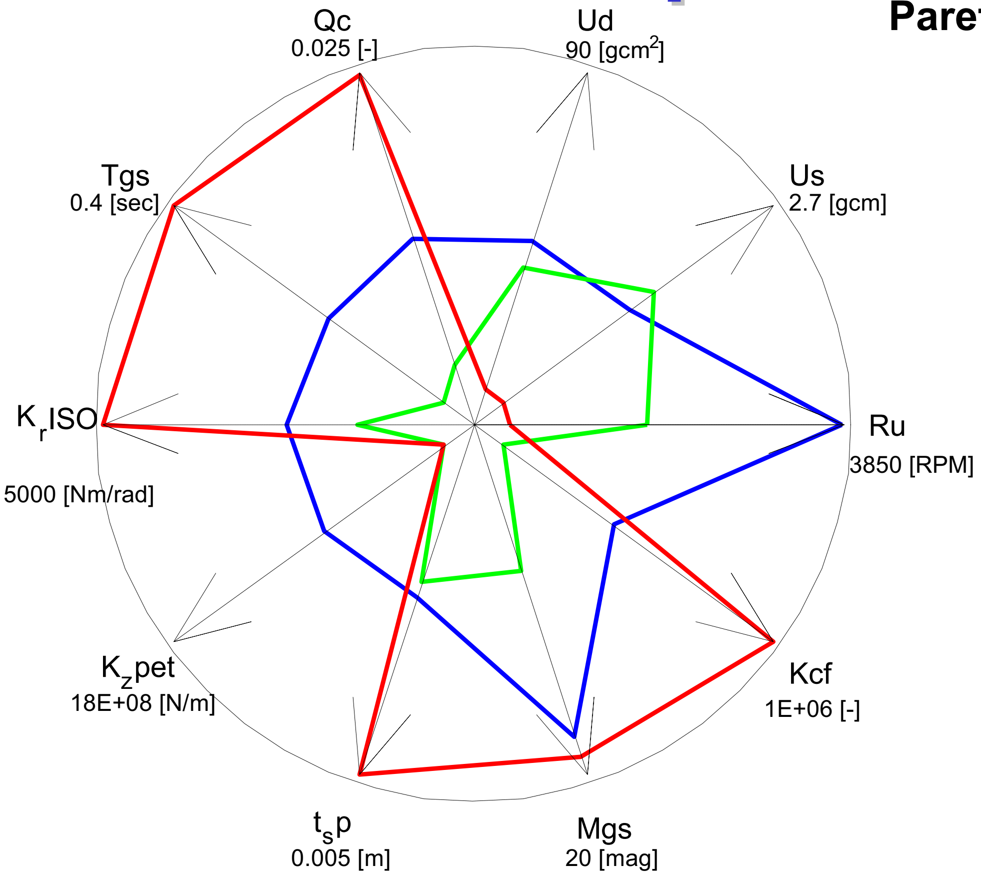
2D-Isoperformance Analysis





Nexus Multivariable Isoperformance n_p=10

Pareto-Optimal Designs p^*_{iso}



Design A

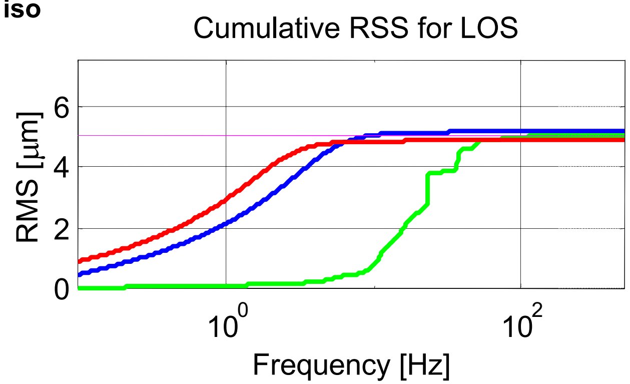
Best "mid-range" compromise

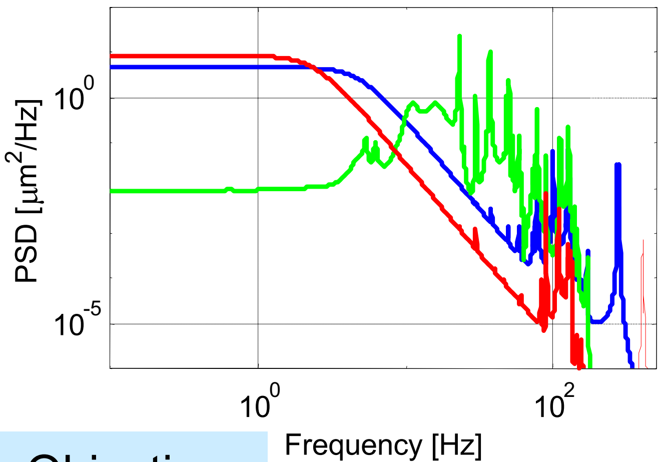
Design B

Smallest FSM control gain

Design C

Smallest performance uncertainty

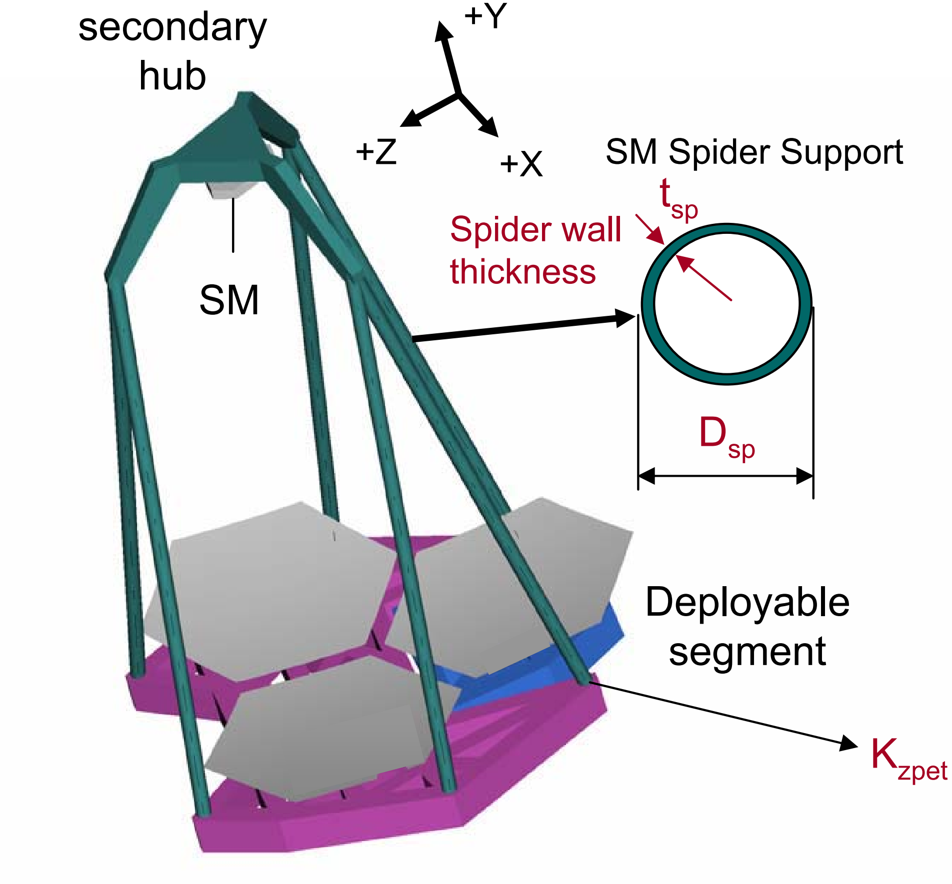




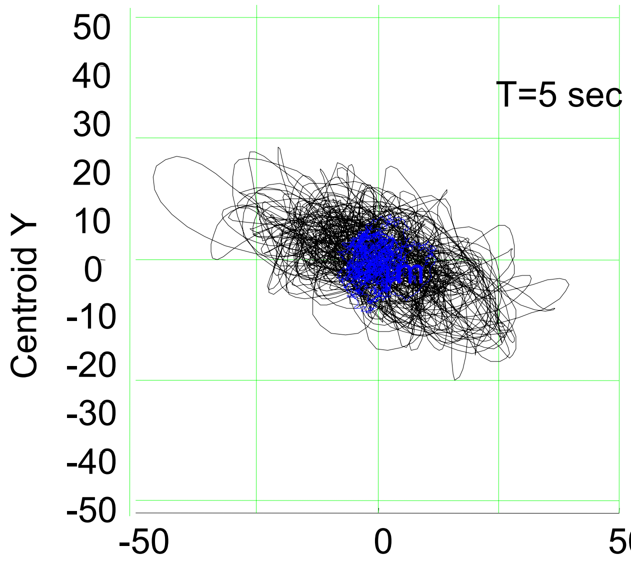
- A: min(J_{c1})
- B: min(J_{c2})
- C: min(J_{r1})

	Performance		Cost and Risk Objectives		
	Jz,1	Jz,2	Jc,1	Jc,2	Jr,1
Design A	20.0000	5.2013	0.6324	0.4668	+/- 14.3218 %
Design B	20.0012	5.0253	0.8960	0.0017	+/- 8.7883 %
Design C	20.0001	4.8559	1.5627	1.0000	+/- 5.3067 %

Nexus Initial p° vs. Final Design p^{**}_{iso}



Parameters	Initial	Final	
Ru	3000	3845	[RPM]
Us	1.8	1.45	[gcm]
Ud	60	47.2	[gcm^2]
Qc	0.005	0.014	[-]
Tgs	0.040	0.196	[sec]
KrISO	3000	2546	[Nm/rad]
Kzpet	0.9E+8	8.9E+8	[N/m]
tsp	0.003	0.003	[m]
Mgs	15	18.6	[Mag]
Kcf	2E+3	4.7E+5	[-]



Centroid Jitter on Focal Plane [RSS LOS]

Initial: 14.97

Final: 5.155

Improvements are achieved by a well balanced mix of changes in the disturbance parameters, structural redesign and increase in control gain of the FSM fine pointing loop.

Isoperformance with Stochastic Data

Example: Baseball season has started

What determines success of a team ?

Pitching ⇩ ERA *"Earned Run Average"*

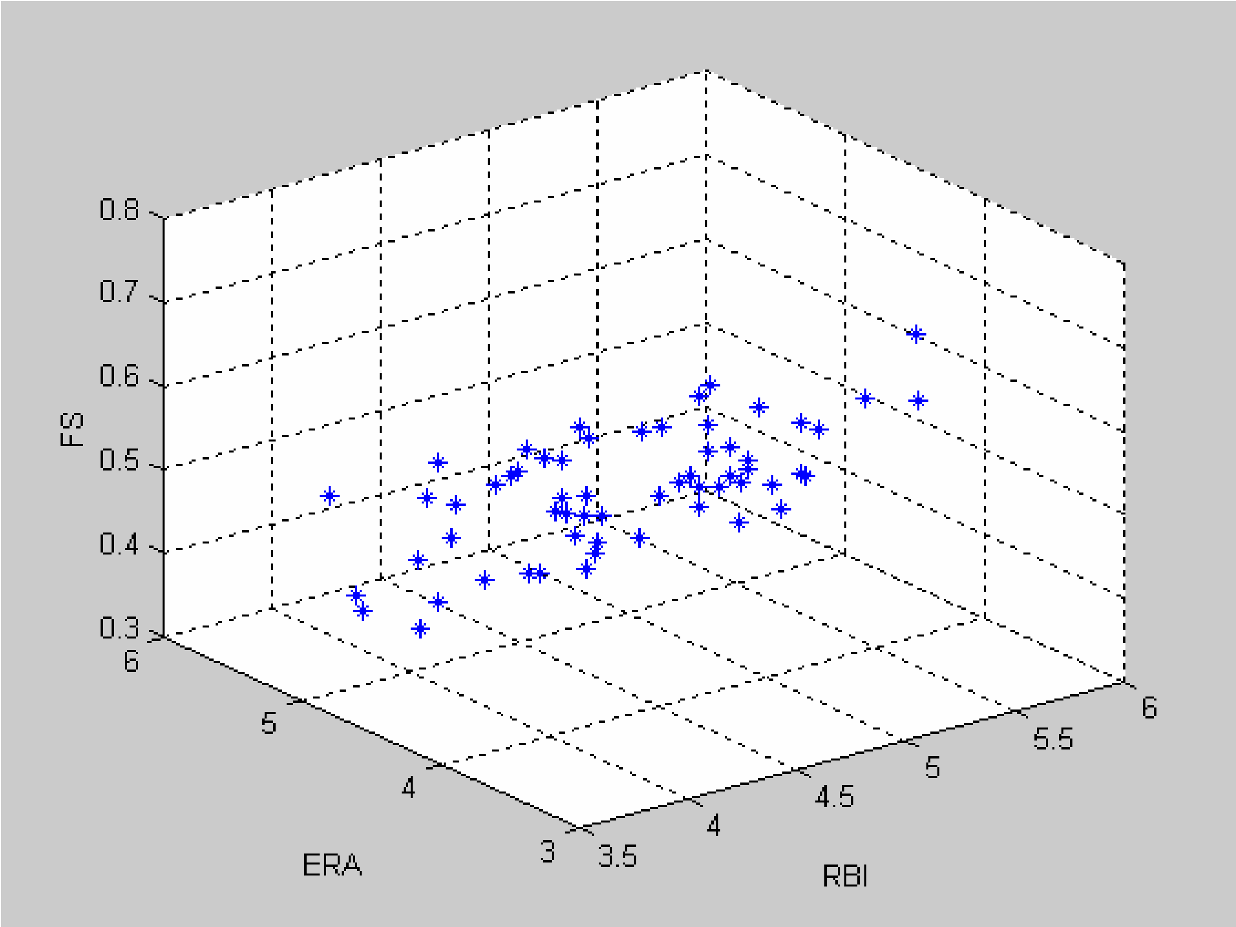
Batting ⇩ RBI *"Runs Batted In"*

How is success of team measured ?

FS= Wins/Decisions

Raw Data

Team results for 2000, 2001 seasons: RBI,ERA,FS



Stochastic Isoperformance (I)

Step-by-step process for obtaining (bivariate) isoperformance curves given statistical data:

Starting point, need:

- Model - derived from empirical data set

- (Performance) Criterion

- Desired Confidence Level

Model

Step 1: Obtain an expression from model for expected performance of a "system" for individual design i as a function of design variables $x_{1,I}$ and $x_{2,i}$

1.1 assumed model

$$E\left[J_i\right] = a_0 + a_1\left(x_{1,i}\right) + a_2\left(x_{2,i}\right) + a_{12}\left(x_{1,i} - \overline{x_1}\right)\left(x_{2,i} - \overline{x_2}\right) \qquad (1)$$

1.2 model fitting

General mean

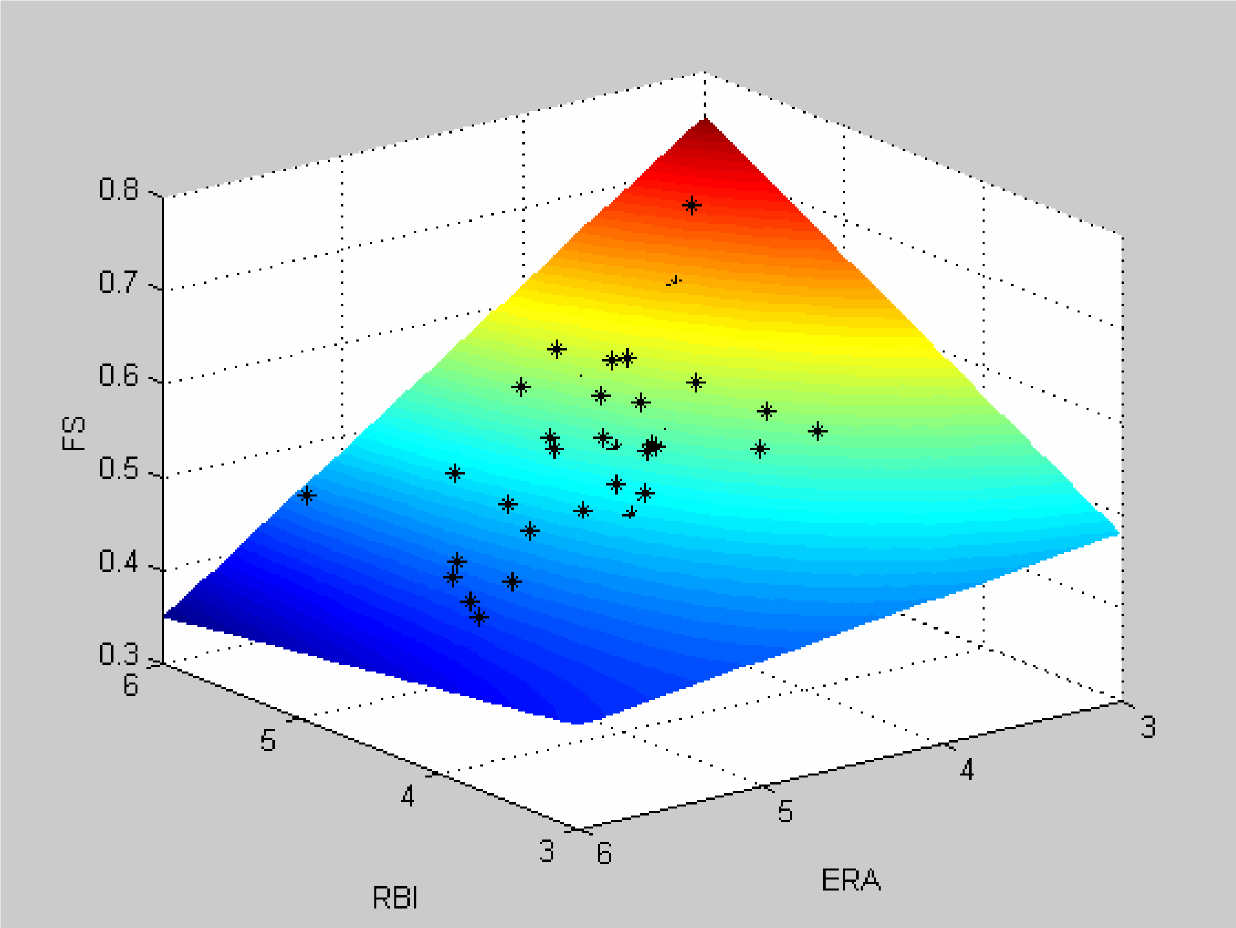
$$a_o = \frac{1}{N}\sum_{j=1}^{N} J_j$$

Used Matlab fminunc.m for optimal surface fit

Baseball: Obtain an expression for expected final standings (FS_i) of individual Team *i* as a function of RBI_i and ERA_i

$$E\left[FS_i\right] = m + a\left(RBI_i\right) + b\left(ERA_i\right) + c\left(RBI_i - \overline{RBI}\right)\left(ERA_i - \overline{ERA}\right)$$

Fitted Model

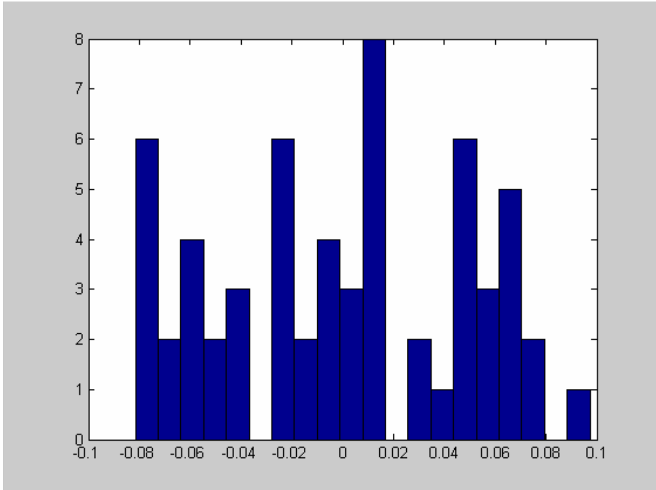


Coefficients:

ao=0.7450
a1=0.0321
a2=-0.0869
a12= -0.0369

RMSE:
Error
σ_e= 0.0493

Error
Distribution



Expected Performance

Step 2: Determine expected level of performance for design i such that the probability of adequate performance is equal to specified confidence level

$$E[J_i] = J_{req} + z\sigma_\varepsilon \qquad (2)$$

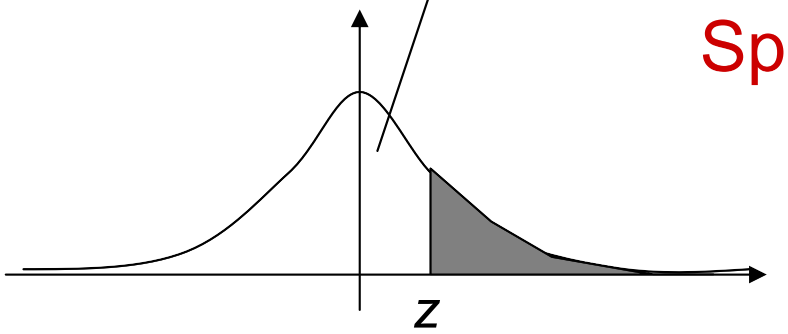
Required performance level

Confidence level normal variable z (Lookup Table)

Error Term (total variance)

$\Phi(z)$

Specify



$$\Phi(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{z} e^{-\frac{z^2}{2}} dz$$

Expected Performance

Baseball:

Performance criterion

- User specifies a final desired standing of FS_i=0.550

Confidence Level

- User specifies a .80 confidence level that this is achieved

Spec is met if for Team *i*:

From normal table lookup → 0.84; Error term from data → 0.0493

$$E\left[FS_i\right] = .550 + z\sigma_r = .550 + 0.84\left(0.0493\right) = 0.5914$$

If the final standing of team *i* is to equal or exceed .550 with a probability of .80, then the expected final standing for Team i must equal 0.5914

Get Isoperformance Curve

Step 3: Put equations (1) and (2) together

$$J_{req} + z\sigma_r = E[J_i] = a_0 + a_1(x_{1,i}) + a_2(x_{2,i}) + a_{12}(x_{1,i} - \overline{x_1})(x_{2,i} - \overline{x_2})$$

(3)

- Four constant parameters: a_o, a_1, a_2, a_{12}
- Two sample statistics: $\overline{x}_1, \overline{x}_2$
- Two design variables: $x_{1,i}$ and $x_{2,i}$

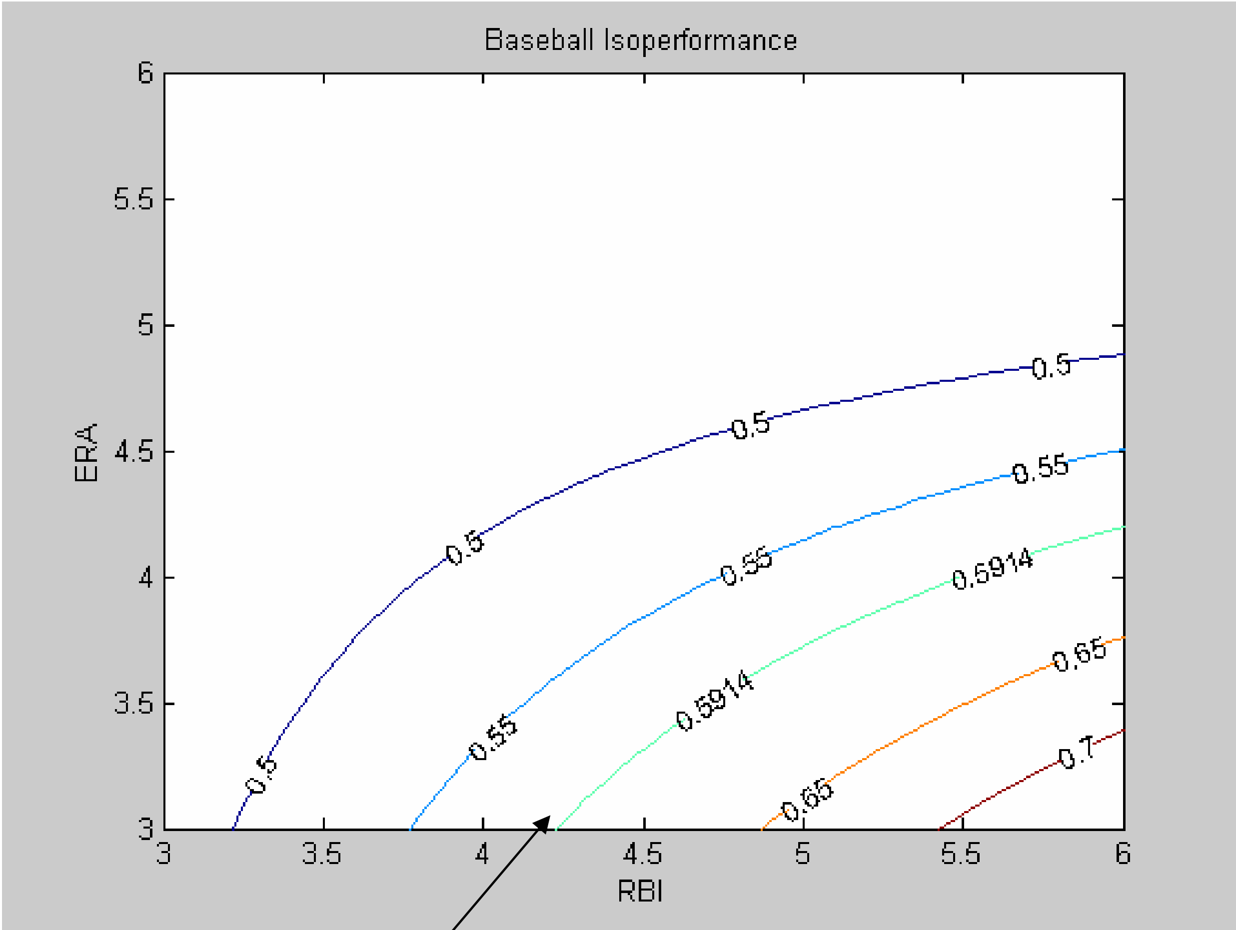
Then rearrange: $x_{2,i} = f(x_{1,i})$

Baseball:

$$RBI_i = \frac{.5914 - m - bERA_i + c\overline{RBI}(ERA_i - \overline{ERA})}{a + c(ERA_i - \overline{ERA})}$$

Equation for isoperformance curve

Stochastic Isoperformance



This is our desired tradeoff curve

Summary

- Isoperformance fixes a target level of "expected" performance and finds a set of points (contours) that meet that requirement
- Model can be physics-based or empirical
- Helps to achieve a "balanced" system design, rather than an "optimal design".