Theory of mimoSHORSA

Multi-Input Multi-Output Stochastic High Order Response Surface Algorithm

A Comprehensive Theoretical Overview

Table of Contents

---

1. Introduction and Motivation

1.1 The Response Surface Problem

In many engineering and scientific applications, we have: - **Input variables** $X = [X_1, X_2, ..., X_n] \in \mathbb{R}^n$ - **Output variables** $Y = [Y_1, Y_2, ..., Y_m] \in \mathbb{R}^m$ - A complex, possibly nonlinear relationship $Y = f(X)$

The goal is to approximate this relationship with a computationally efficient model that: 1. Captures nonlinear behavior 2. Handles high-dimensional inputs 3. Provides uncertainty quantification 4. Remains computationally tractable

1.2 Why Response Surfaces?

Traditional approaches face challenges: - **Direct simulation**: Computationally expensive (finite element, CFD, etc.) - **Low-order polynomials**: Inadequate for complex nonlinear behavior - **Neural networks**: Black-box nature, difficult uncertainty quantification - **Kriging/Gaussian processes**: Computational scaling issues for large datasets

Response surface methods provide a middle ground: explicit polynomial models with: - Analytical derivatives - Uncertainty quantification - Interpretable coefficients - Efficient evaluation

---

2. Mathematical Foundation

2.1 General Polynomial Representation

mimoSHORSA approximates the input-output relationship as:

```
Y(X) = Σ cⱼ Ψⱼ(X) j=1
```

Where: - **c_j**: Model coefficients to be determined - **$\Psi_j(X)$**: Basis functions (products of Hermite functions) - **nTerm**: Total number of terms in the expansion

2.2 Detailed Polynomial Structure

The full expansion includes:

1. **Constant term**: c_0
2. **Pure power terms**: $\Sigma\ \Sigma\ b_{i\,j}\ \Psi_j(X_i)$ i=1 j=1 where k_i is the maximum order for variable X_i
3. **Mixed terms** (cross-products): $\Sigma\ cq\ \Pi\ \Psi_{p\,i}q(X_i)$ q=1 i=1

2.3 Multi-Output Extension

For multiple outputs, mimoSHORSA fits **separate models** for each output:

$Y_k(X) = \Sigma\ c_{k\,j}\ \Psi_{k\,j}(X)$, k = 1, 2, ..., m j=1

Each output can have: - Different numbers of terms ($nTerm_k$) - Different coefficient values - Different retained terms (after culling)

---

3. Hermite Polynomial Basis

3.1 Why Hermite Functions?

Standard power polynomials ($1, x, x^2, x^3, \ldots$) have problems: - **Numerical instability** for high orders - **Poor conditioning** of basis matrices - **Lack of orthogonality**

Hermite functions provide: - **Orthogonality** with respect to Gaussian weight - **Numerical stability** for higher orders - **Natural scaling** for standardized variables

3.2 Hermite Function Definition

The Hermite functions are defined as:

$\psi_n(z) = (1/\sqrt{(2^n\ n!\ \sqrt{\pi})})\ H_n(z)\ \exp(-z^2/2)$

Where: - **$H_n(z)$**: Hermite polynomial of order n (physicist's version) - **$\exp(-z^2/2)$**: Gaussian weight function - Normalization ensures orthonormality

3.3 Low-Order Hermite Functions

``` $\psi_0(z) = \pi$^(-1/4) $\exp(-z^2/2)$

$\psi_1(z) = \sqrt{2}\ \pi$^(-1/4) $z \exp(-z^2/2)$

$\psi_2(z) = (1/\sqrt{2})\ \pi$^(-1/4) $(2z^2 - 1) \exp(-z^2/2)$

$\psi_3(z) = (1/\sqrt{3})\ \pi$^(-1/4) $(2z^3 - 3z) \exp(-z^2/2)$

$\psi_4(z) = (1/(2\sqrt{6}))\ \pi$^(-1/4) $(4z^4 - 12z^2 + 3) \exp(-z^2/2)$

$\psi_5(z) = (1/(2\sqrt{15}))\ \pi$^(-1/4) $(4z^5 - 20z^3 + 15z) \exp(-z^2/2)$ ```
```

3.4 Orthogonality Property

$\int_{-\infty}^{\infty} \psi_m(z)\ \psi_n(z)\ dz = \delta_{mn}$

This orthonormality is key to numerical stability.

3.5 Multi-Dimensional Basis Functions

For multi-dimensional inputs, basis functions are **products** of 1D Hermite functions:

$\Psi(Z) = \psi_{p_1}(Z_1) \times \psi_{p_2}(Z_2) \times \ldots \times \psi_{p_n}(Z_n)$

Where **p** = [p_1, p_2, ..., p_n] is the **order vector** specifying the polynomial order in each dimension.

---

4. Model Construction Process

4.1 Three-Stage Algorithm

mimoSHORSA follows a three-stage process:

Stage 1: Polynomial Order Determination (Optional)

- Determine optimal maximum order k_i for each variable
- Uses 1D Chebyshev sampling and curve fitting
- Currently disabled in practice (uses uniform maximum order)

Stage 2: Mixed Term Enumeration

- Generate all possible term combinations
- Filter based on total order constraint
- Create order matrix specifying each term

Stage 3: Model Fitting and Reduction

- Fit full model using least squares
- Iteratively remove uncertain terms
- Continue until tolerance criteria met

4.2 Term Generation Algorithm

The algorithm generates all combinations where:

``` $p_1 \leq \text{maxOrder}_1$ $p_2 \leq \text{maxOrder}_2$ ... $p_n \leq \text{maxOrder}_n$

AND

$p_1 + p_2 + \ldots + p_n \leq \max(\text{maxOrder})$ ```

This creates a structured polynomial space with controlled complexity.

**Example** (n=2, maxOrder=[2,2]): [0,0] → $\psi_0(Z_1)\psi_0(Z_2)$ = constant [1,0] → $\psi_1(Z_1)\psi_0(Z_2)$ = linear in $Z_1$ [0,1] → $\psi_0(Z_1)\psi_1(Z_2)$ = linear in $Z_2$ [2,0] → $\psi_2(Z_1)\psi_0(Z_2)$ = quadratic in $Z_1$ [1,1] → $\psi_1(Z_1)\psi_1(Z_2)$ = bilinear term [0,2] → $\psi_0(Z_1)\psi_2(Z_2)$ = quadratic in $Z_2$
```

4.3 Least Squares Fitting

Given: - **Z**: Scaled input data (nInp × mData) - **Y**: Output data (nOut × mData) - **B**: Basis matrix (mData × nTerm)

The coefficients are found by solving:

$$\min ||B \cdot c - Y||^2 \; c$$

Solution (via SVD for numerical stability): $c = (B^T B)^{(-1)} B^T Y$

In practice, singular value decomposition handles ill-conditioned cases.

4.4 Design Matrix Construction

The basis matrix **B** has structure:

$$B = [\Psi_1(Z^1)\ \Psi_2(Z^1)\ \dots\ \Psi_n T_{erm}(Z^1)]\ [\Psi_1(Z^2)\ \Psi_2(Z^2)\ \dots\ \Psi_n T_{erm}(Z^2)]\ [\ \dots\ \dots\ \dots\ \dots\ ]\ [\Psi_1(Z^m)\ \Psi_2(Z^m)\ \dots\ \Psi_n T_{erm}(Z^m)]$$

Where: - Rows correspond to data points - Columns correspond to terms - Each entry is the basis function evaluated at that data point

---

5. Iterative Model Reduction

5.1 Motivation for Reduction

Initial full model often has problems: - **Overfitting**: Too many parameters relative to data - **Uncertain coefficients**: High variance estimates - **Poor generalization**: Fits noise rather than signal - **Computational cost**: Unnecessary terms

5.2 Coefficient of Variation (COV)

For each coefficient, compute:

$$COV(c_j) = \sigma(c_j) / |c_j|$$

Where the standard error is:

$$\sigma(c_j) = \sqrt{[(RSS/(m - nTerm)) \times (B^T B)^{-1}_{jj}]}$$

Components: - **RSS**: Residual sum of squares = $||Y - \hat{Y}||^2$ - **m**: Number of data points - **nTerm**: Number of model terms - **$(B^T B)^{-1}_{jj}$**: Diagonal element of inverse Gram matrix

5.3 Culling Strategy

At each iteration: 1. Identify term with **largest COV** (most uncertain) 2. Remove that term from model 3. Refit remaining coefficients 4. Re-evaluate COV for all remaining terms 5. Repeat until: COV < tolerance OR max iterations reached

Why largest COV? - High COV indicates coefficient is poorly determined - Removal has minimal impact on model quality - Improves model parsimony and generalization

5.4 Stopping Criteria

Iteration stops when: $\max(COV) < tolerance \text{ AND } \rho_{test} > 0$

Where: - **tolerance**: User-specified (typically 0.05 - 0.15) - **ρ_test**: Model-data correlation on test set (must be positive)

This ensures both: - Coefficient certainty (low COV) - Predictive capability (positive correlation)

---

6. Statistical Evaluation Metrics

6.1 R-Squared (R^2)

Measures explained variance:

$$R^2 = 1 - (RSS / TSS)$$

Where: - **RSS** = $\Sigma(Y_i - \hat{Y}_i)^2$ (residual sum of squares) - **TSS** = $\Sigma(Y_i - \bar{Y})^2$ (total sum of squares)

Interpretation: - $R^2 = 1$: Perfect fit - $R^2 = 0$: Model no better than mean - $R^2 < 0$: Model worse than mean (on test data)

6.2 Adjusted R-Squared (R^2_adj)

Penalizes model complexity:

$$R^2\_adj = ((m-1) \cdot R^2 - nTerm) / (m - nTerm)$$

Why adjust? - Raw R^2 always increases with more terms - R^2_adj accounts for degrees of freedom - Prevents overfitting through complexity penalty

6.3 Model-Data Correlation (ρ)

Pearson correlation between predictions and observations:

$$\rho = Cov(Y, \hat{Y}) / (\sigma\_Y \cdot \sigma\_\hat{Y})$$

Advantages over R^2: - Scale-invariant - More interpretable for practitioners - Robust to offset errors

6.4 Coefficient of Variation (COV)

For each coefficient:

$$COV(c_j) = SE(c_j) / |c_j|$$

Interpretation: - COV < 0.10: Well-determined coefficient - 0.10 < COV < 0.30: Moderate uncertainty - COV > 0.30: Highly uncertain, candidate for removal

6.5 Condition Number

Measures numerical stability:

$$\kappa(B) = ||B|| \cdot ||B^{-1}||$$

Interpretation: - $\kappa < 100$: Well-conditioned - $100 < \kappa < 1000$: Moderate conditioning - $\kappa > 1000$: Ill-conditioned, numerical issues possible

---

7. Scaling and Preprocessing

7.1 Why Scale?

Raw data problems: - **Different units**: Variables on vastly different scales - **Numerical instability**: Extreme values cause conditioning issues - **Poor Hermite basis fit**: Hermite functions optimal for ~N(0,1) data

7.2 Scaling Options

Option 0: No Scaling

$Z = X$ Use when data already normalized.

Option 1: Standardization

$Z = (X - \mu) / \sigma$ Centers and scales to unit variance.

Option 2: Decorrelation (Whitening)

$Z = V \Lambda^{(-1/2)} V^T (X - \mu)$ Where $V \cdot \Lambda \cdot V^T = \text{Cov}(X)$

Removes linear correlations between variables.

Option 3: Log-Standardization

$Z = (\log_{10}(X) - \mu_{log}) / \sigma_{log}$ For data with multiplicative structure or log-normal distributions.

Option 4: Log-Decorrelation

$Z = V \Lambda^{(-1/2)} V^T (\log_{10}(X) - \mu_{log})$ Combines logarithmic and linear decorrelation.

7.3 Transformation Matrices

The transformations are stored as: $Z = T^{-1} (X - \mu)$

Where: - **μ**: Mean vector - **T**: Transformation matrix

Inverse transformation (for predictions): $X = T \cdot Z + \mu$

For log transformations: $X = 10^{(T \cdot Z + \mu)}$

7.4 Outlier Removal

After scaling, remove data points where: $|Z_{ij}| > \text{threshold (typically 4)}$

Rationale: - Hermite functions designed for ~N(0,1) - Extreme values degrade approximation - Removes potential data errors

---

8. Computational Considerations

8.1 Computational Complexity

Term generation: $O(k^n)$ where k = maxOrder, n = nInp - Combinatorial explosion for high dimensions - Filtering reduces to manageable size

Basis construction: O(m·nTerm·n) where m = mData - Linear in number of data points - Dominates for large datasets

Least squares solve: $O(m \cdot nTerm^2 + nTerm^3)$ - SVD used for numerical stability - Can be expensive for many terms

Per-iteration cost: Dominated by least squares - Typically 10-50 iterations - Each iteration removes one term

8.2 Conditioning and Stability

Sources of ill-conditioning: 1. Highly correlated input variables 2. Insufficient data (m < nTerm) 3. Extreme polynomial orders

Mitigation strategies: 1. **Hermite basis**: Better conditioned than power basis 2. **Decorrelation** (scaling option 2 or 4) 3. **SVD-based solve**: Handles near-singular systems 4. **Model reduction**: Removes problematic terms

8.3 Train-Test Split

Critical for validation: - **Training set**: Used for coefficient estimation - **Test set**: Used for performance evaluation

Typical split: 50-80% training, 20-50% testing

Why separate? - Training metrics overestimate performance - Test metrics indicate generalization - Prevents overfitting bias

8.4 Memory Requirements

Storage needs: - **B matrix**: O(m·nTerm) - largest structure - **Data**: O(m·(n+p)) - **Coefficients**: O(nTerm·p)

For large problems ($m > 10^6$), consider: - Batch processing - Out-of-core algorithms - Reduced precision storage

---

9. Applications and Use Cases

9.1 Structural Reliability Analysis

Original motivation (Gavin & Yau, 2005): - Approximate limit state functions - Compute failure probabilities - Sensitivity analysis

Advantages: - Explicit failure surface - Analytical gradients for importance sampling - Handles high-dimensional random variables

9.2 Uncertainty Quantification

Applications: - **Forward propagation**: Input uncertainty → output uncertainty - **Sensitivity analysis**: Which inputs matter most? - **Reliability**: Probability of exceeding thresholds

mimoSHORSA provides: - Coefficient uncertainties (COV) - Analytical variance propagation - Importance measures via coefficients

9.3 Design Optimization

Use response surface as surrogate: `minimize Y(X) subject to constraints on X`

Benefits: - Cheap function evaluations (vs. simulation) - Analytical gradients available - Global optimization feasible

9.4 Model Reduction for Complex Simulations

When expensive simulations (FEA, CFD) exist: 1. Run limited design of experiments 2. Fit mimoSHORSA surrogate 3. Use surrogate for: - Optimization - Monte Carlo analysis - Real-time prediction

9.5 Multi-Physics Problems

Natural fit for coupled systems: - Multiple outputs from single input - Each output modeled independently - Maintains physical intuition

Examples: - Thermal-structural coupling - Fluid-structure interaction - Electro-mechanical systems

9.6 Practical Considerations

When mimoSHORSA excels: - Smooth, continuous responses - Moderate dimensions ($n < 20$) - Sufficient data ($m > 5 \cdot nTerm$) - Need for interpretability

When to use alternatives: - Discontinuous responses → Classification methods - Very high dimensions → Dimension reduction first - Sparse data → Bayesian approaches - Black-box OK → Neural networks, tree methods

---

10. Mathematical Appendix

10.1 Hermite Polynomial Recurrence

Higher-order Hermite polynomials via recurrence:

$$H_{n+1}(z) = 2z H_n(z) - 2n H_{n-1}(z)$$

Starting values: $H_0(z) = 1$ $H_1(z) = 2z$

10.2 Standard Error Derivation

For linear model $Y = Bc + \varepsilon$ where $\varepsilon \sim N(0, \sigma^2 I)$:

$$Var(c) = \sigma^2 (B^T B)^{-1}$$

Estimate σ^2 from residuals: $\sigma^2 = RSS / (m - nTerm)$

Therefore: $SE(c_j) = \hat{\sigma}\sqrt{[(B^T B)^{-1}]_{jj}}$

10.3 R^2 Relationship to Correlation

For centered data: $R^2 = \rho^2$

But for general case (with intercept): $R^2 \neq \rho^2$

R^2_adj provides better comparison across models.

---

11. References and Further Reading

Primary Reference

Gavin, H.P. and Yau, S.C., "High order limit state functions in the response surface method for structural reliability analysis," *Structural Safety*, 2008, Vol. 30, pp. 162-179.

Theoretical Background

Response Surface Methods: - Box, G.E.P. and Draper, N.R., "Response Surfaces, Mixtures, and Ridge Analyses," 2007 - Myers, R.H. et al., "Response Surface Methodology," 2016

Hermite Polynomials: - Szegö, G., "Orthogonal Polynomials," 1939 - Abramowitz, M. and Stegun, I., "Handbook of Mathematical Functions," 1964

Uncertainty Quantification: - Sudret, B., "Global sensitivity analysis using polynomial chaos expansions," 2008 - Xiu, D. and Karniadakis, G.E., "The Wiener-Askey polynomial chaos," 2002

Model Selection: - Akaike, H., "A new look at the statistical model identification," 1974 - Burnham, K.P. and Anderson, D.R., "Model Selection and Multimodel Inference," 2002

12. Summary

mimoSHORSA provides a powerful framework for approximating complex input-output relationships through:

1. **Hermite function basis** for numerical stability
2. **Systematic term generation** for comprehensive coverage
3. **Iterative model reduction** for parsimony
4. **Statistical validation** for confidence
5. **Multiple scaling options** for diverse data types

The method excels in applications requiring: - Explicit functional forms - Uncertainty quantification - Computational efficiency - Physical interpretability

By combining classical polynomial approximation theory with modern statistical model selection, mimoSHORSA achieves both accuracy and practicality for real-world engineering and scientific problems.

Document prepared October 23, 2025 Based on translation and analysis of mimoSHORSA MATLAB implementation Duke University, Department of Civil and Environmental Engineering