Gaussian Process Emulator Approach for Engineering Mechanics

F. A. DiazDelaO & S. Adhikari

School of Engineering

Swansea University, U.K.

Swansea University
Prifysgol Abertawe

Outline

- Introduction
- Simulators and emulators
- Application of Gaussian process emulators
    - Linear dynamical systems (Paper 1, Schamburg)
    - Initial design (Paper 2, Osaka)
    - Random field discretisation (Paper 3, Athens)
    - Polynomial chaos expansion (Paper 4, Orlando)
- Webpage
- Future works
- Some references

Introduction

- About myself:
  - BSc in Applied Mathematics (ITAM, Mexico City)
  - MSc in Statistics and OR (University of Essex, UK)
  - 2^{nd} year PhD student (funded by EPSRC and CONACYT)
- This presentation summarizes the work carried out as part of the PhD project "Coupled Models: Expert Judgement, Emulators and Model Uncertainty".

Swansea University
Prifysgol Abertawe

Simulators

- **Problem:** There exist phenomena which are complex enough to make physical experimentation impossible.
- **Solution:** Study these systems running computer codes, also known as simulators (O'Hagan, 2006).
- A simulator is a function $\eta : \mathbb{R}^{d_1} \to \mathbb{R}^{d_2}$ that, given an input **x**, it produces an output **y**.
- **Potential drawback:** High cost of execution.

Simulators

- Such cost may be measured as:
  - CPU time employed
  - Number of floating point operations performed
  - Required computer capability
- Examples: Sophisticated climate models that take months to complete a single run (Goldstein, 2007). Prodigious computer power needed to simulate the outcome of a car accident (Thomke et al., 1999).

Swansea University
Prifysgol Abertawe

Emulators

- **A solution:** Build an emulator of the expensive simulator.
- An emulator is a statistical approximation to the simulator, i.e., it provides a probability distribution for $\eta(\cdot)$.
- Emulators have already been used in a number of scientific fields, which include:
  - Environmental science (Challenor et al., 2006)
  - Climate prediction (Rougier, 2007)
  - Medical science (Haylock and O'Hagan, 1996)

Emulators

- Emulators are built by choosing n design points in the input domain of the simulator and obtaining the training set $\mathcal{T} = \{\mathbf{y}_1 = \eta(\mathbf{x}_1), \ldots, \mathbf{y}_n = \eta(\mathbf{x}_n)\}$.
- Once this choice is made, an emulator should:
  - Reproduce the known output at any design point.
  - At any untried input, provide a distribution whose mean is a plausible interpolation of the training data. The probability distribution around this predictive mean should also express the uncertainty about how the emulator might interpolate.

Swansea University
Prifysgol Abertawe

Emulators

- The following figures illustrate how an emulator would approximate the simple simulator $\mathbf{y} = 0.5\mathbf{x} - \mathbf{x} sin(\mathbf{x})$, and estimate the uncertainty about such approximation.
- In Figure 1, the solid line is the output of the simulator. The circles represent the training runs, and the dots are the mean of the emulator, which provides the approximation.
- In Figure 2, the solid line is the output of the simulator. The circles represent the training runs, and the shaded areas are probability bounds of two standard deviations around the predictive mean.

Approximation

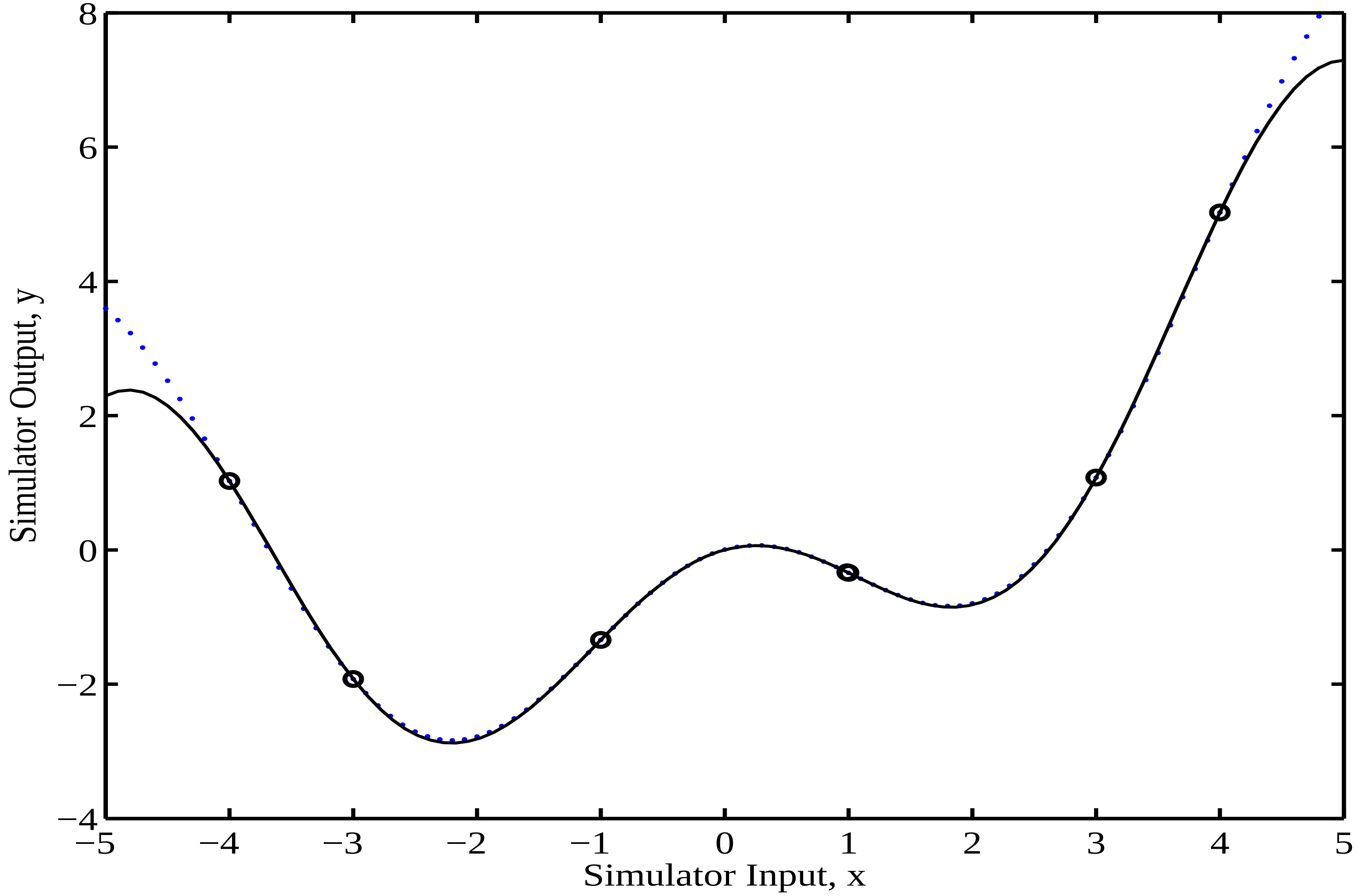


Figure 1: Predictive mean (· · ·) using 6 training runs (o).

Uncertainty

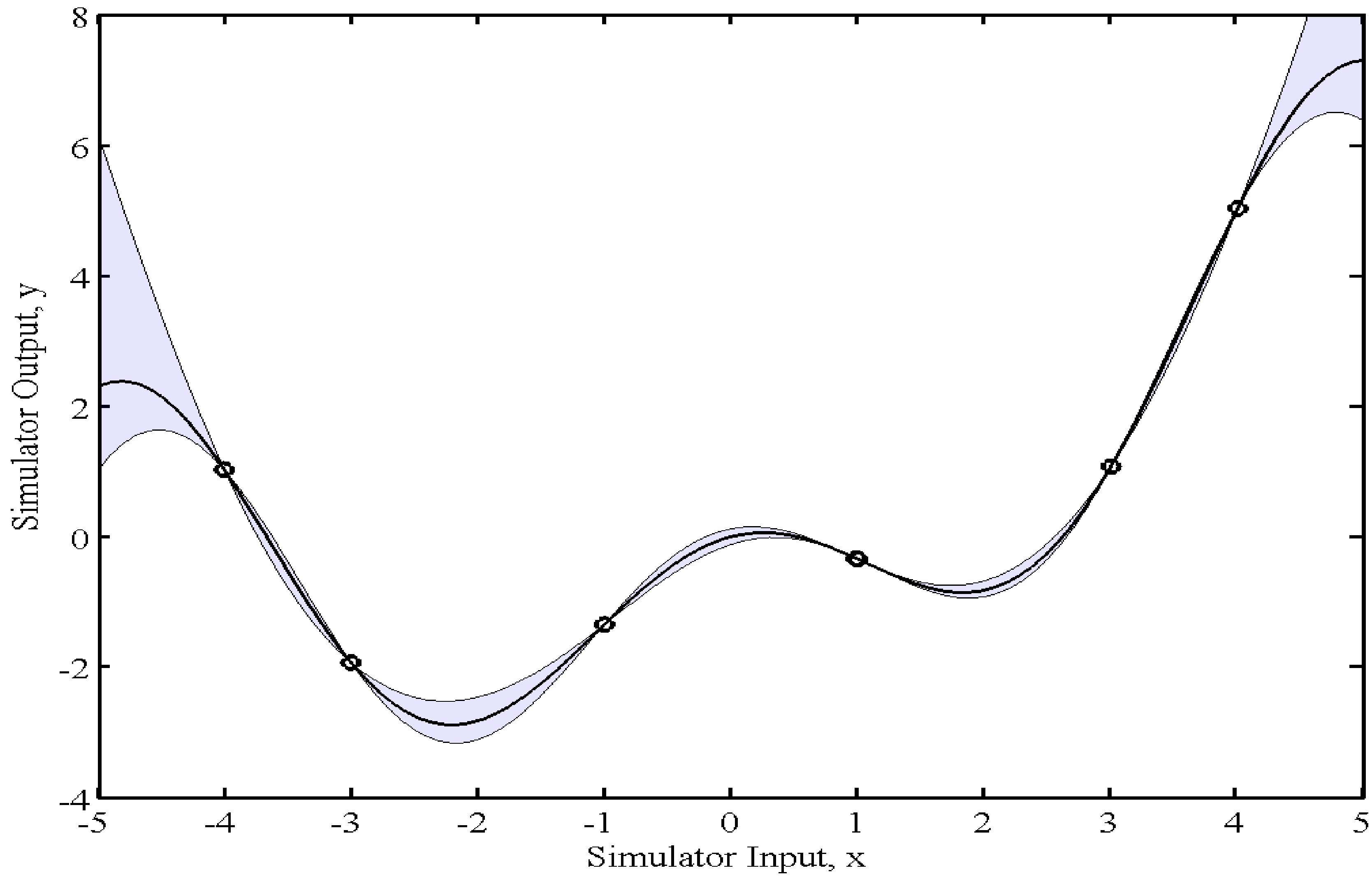


Figure 2: Uncertainty bounds using 6 training runs (o).

Emulators

- From a Bayesian perspecive, $\eta(\cdot)$ is regarded as a random variable in the sense that it is unknown until the simulator is run.
- Assume that $\eta(\cdot)$ deviates from the mean of its distribution in the following way

$$\eta(\mathbf{x}) = \sum_{j=1}^{n} \beta_j h_j(\mathbf{x}) + Z(\mathbf{x}) \qquad (1)$$

where for all j, $h_j(\mathbf{x})$ is a known function and β_j is an unknown coefficient.

Emulators

- The function $Z(\cdot)$ in Eq.(1) is assumed to be a Gaussian stochastic process (GP) with mean zero and covariance given by

$$Cov(\eta(\mathbf{x}), \eta(\mathbf{x}^{'})) = \sigma^2 e^{-(\mathbf{x}-\mathbf{x}^{'})^T \mathbf{B}(\mathbf{x}-\mathbf{x}^{'})} \tag{2}$$

  where **B** is a positive definite diagonal matrix that contains smoothness parameters.

- If the mean of $\eta(\cdot)$ is of the form $m(\cdot) = \mathbf{h}(\cdot)^T \boldsymbol{\beta}$ then $\eta(\cdot)$ has a GP distribution with mean $m(\cdot)$ and covariance given by Eq.(2).

Swansea University
Prifysgol Abertawe

Emulators

- The latter is expressed as

$$\eta(\cdot)|\boldsymbol{\beta}, \sigma^2 \sim N(\mathbf{h}(\cdot)^T\boldsymbol{\beta}, \sigma^2 C(\cdot,\cdot)) \tag{3}$$

- Subjective information about the relationship between input and the unknown output is contained in this prior distribution.
- The next step is to update this belief by adding objective information, represented by the vector of observations $\mathbf{y} = [\mathbf{y}_1 = \eta(\mathbf{x}_1), \ldots, \mathbf{y}_n = \eta(\mathbf{x}_n)]^T$.

Swansea University
Prifysgol Abertawe

Emulators

- Using standard integration techniques, it can be shown (Haylock and O'Hagan, 1996) that such update is

$$\eta(\cdot)|\mathbf{y},\sigma^2 \sim N(m^{**}(\cdot),\sigma^2 C^{**}(\cdot,\cdot)) \tag{4}$$

  where $m^{**}(\cdot)$ constitutes the fast approximation of $\eta(\mathbf{x})$ for any **x** in the input domain.

- Moreover, it can be shown that

$$\frac{\eta(\mathbf{x}) - m^{**}(\mathbf{x})}{\widehat{\sigma}\sqrt{\frac{(n-q-2)C^{**}(\mathbf{x})}{n-q}}} \sim t_{n-q} \tag{5}$$

Swansea University
Prifysgol Abertawe

Linear dynamical systems

- **Title:** Bayesian Emulator Approach for Complex Dynamical Systems
- **Conference:** 49^{th} AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics & Materials Conference; Schamburg, Illinois, USA.
- **Rationale:** Running a simulator to compute the frequency response function (FRF) of a complex engineering system can be very expensive, even for obtaining the response at few frequency points. A Gaussian process emulator (GPE) can approximate the FRF over a frequency range, using only a few response values, obtained either from experimental measurements or by running a finite element model at carefully selected inputs.

Linear dynamical systems

- The FRF of a simple spring-mass system, shown in Figure 3, was emulated for illustration.
- The associated simulator is of the form

$$\eta(\omega) = \left| \left[-\omega^2 \mathbf{M} + i\omega \mathbf{C} + \mathbf{K} \right]^{-1} \bar{\mathbf{f}}(\omega) \right| \qquad (6)$$

  where **M**, **C** and **K** are respectively the mass, damping and stiffness matrices, ω is the frequency level, and $\bar{\mathbf{f}}(\omega)$ is the Fourier transform of the forcing vector.
- The predictive mean of the emulator and its probability bounds are shown in Figures 4 and 5.

Linear dynamical systems

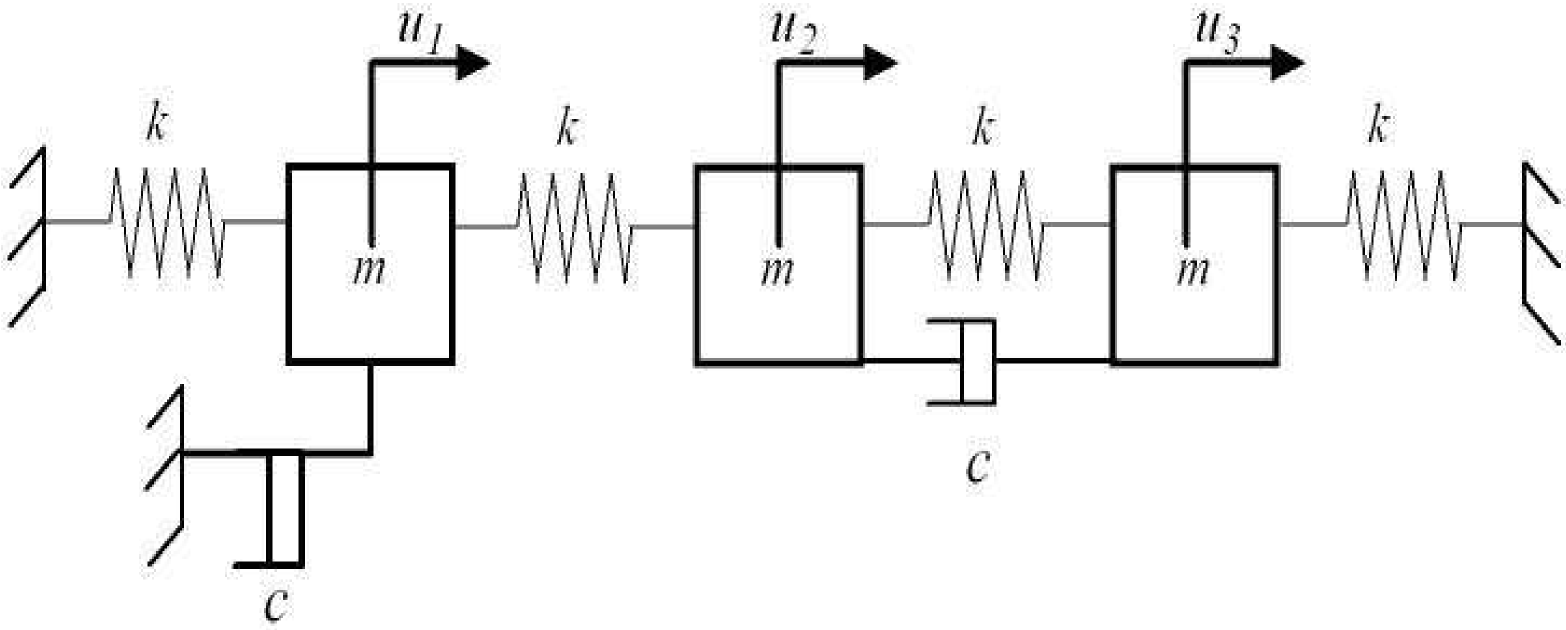


Figure 3: Simple spring-mass system.

Linear dynamical systems

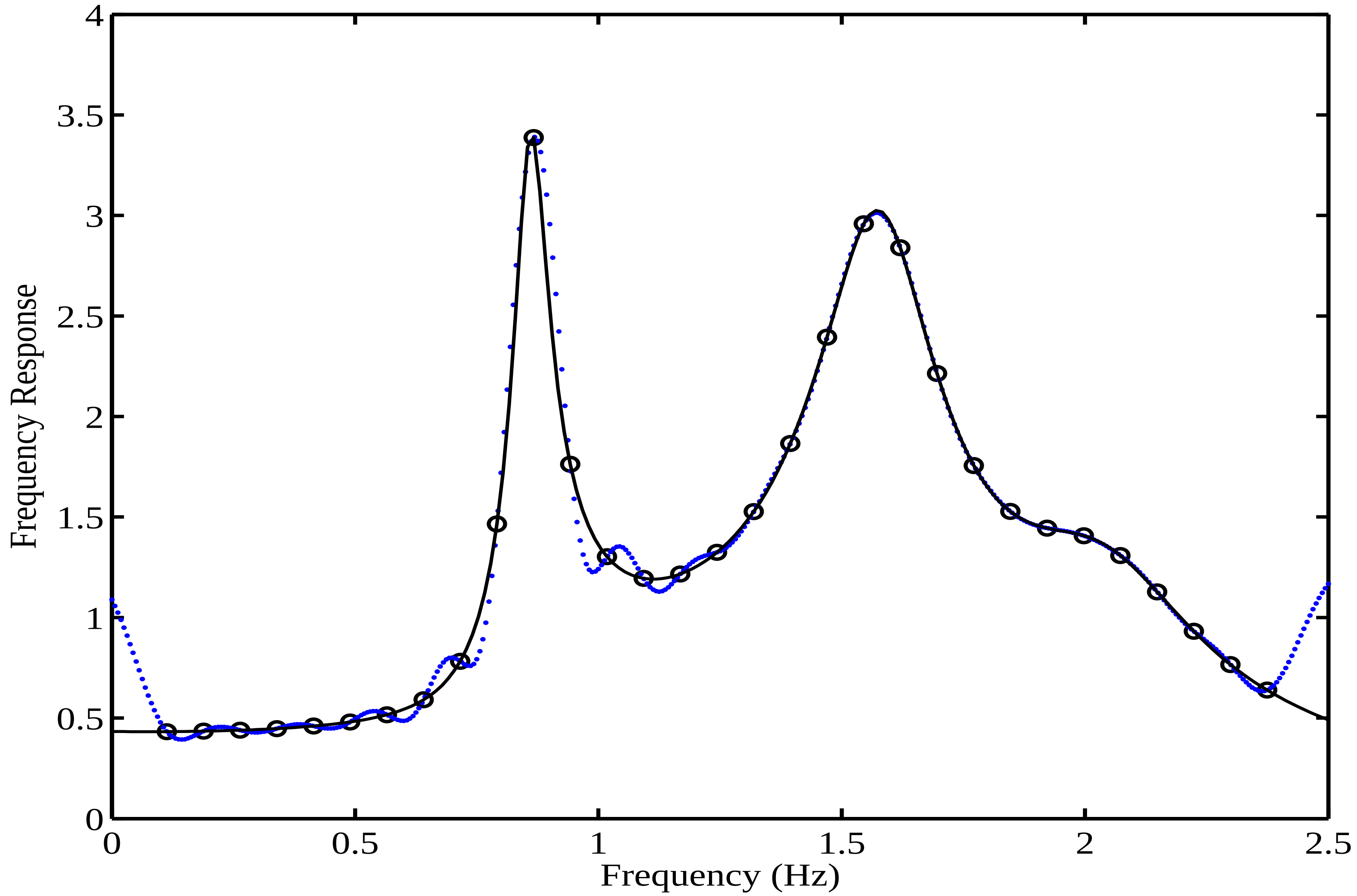


Figure 4: Predictive mean (· · ·) using 31 training runs (o).

Linear dynamical systems

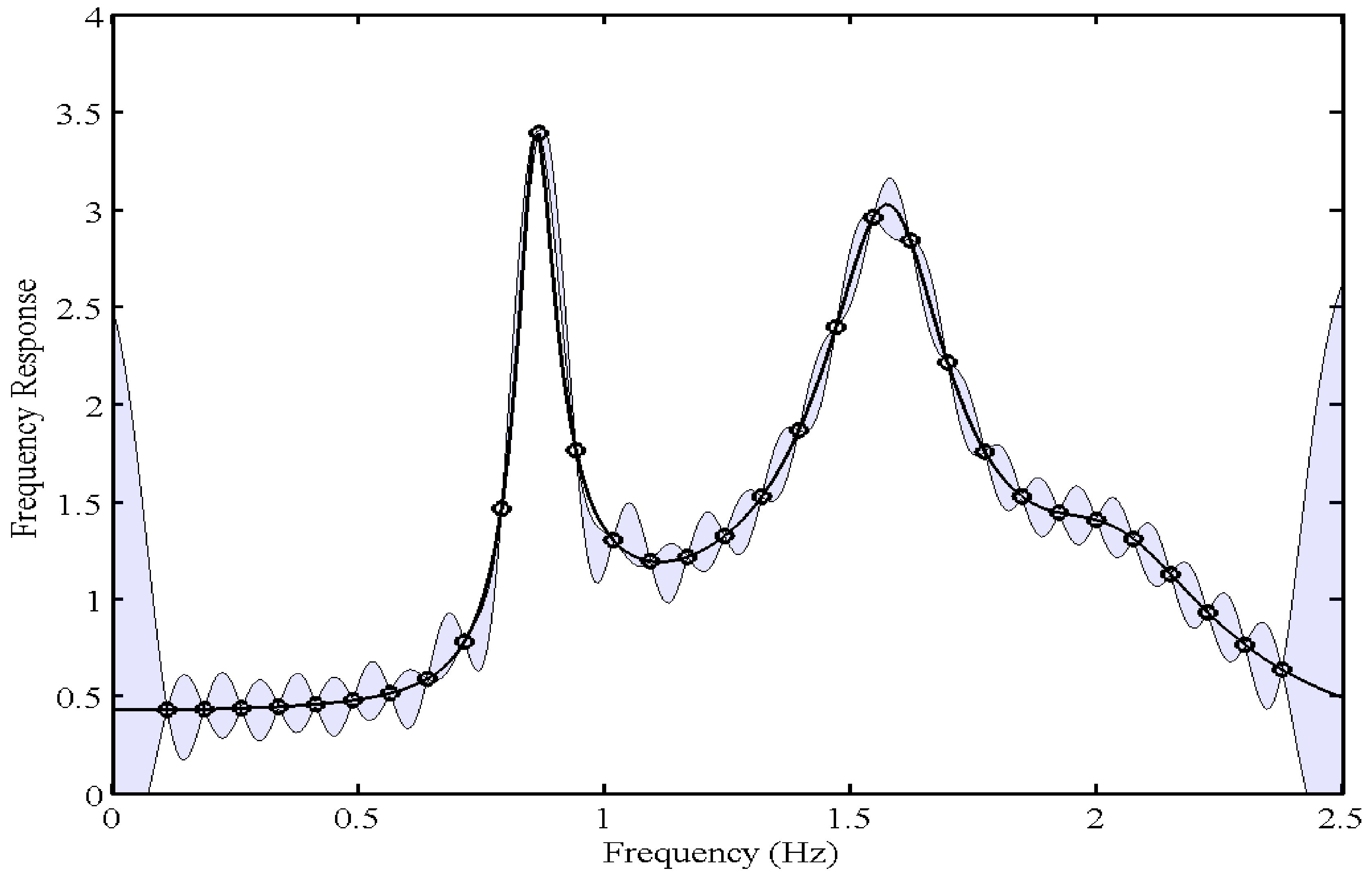


Figure 5: Uncertainty bounds using 31 training runs (o).

Linear dynamical systems

- The system above is non-proportionally damped. For such systems, the computation of the FRF is numerically more expensive than in the idealized case of proportional damping.
- The FRF of a much bigger system, with 1200 degrees of freedom, divided in three frequency ranges (0 - 1.0 kHz as the low-frequency range, 1.0 - 2.5 kHz as the medium-frequency range, and 2.5 - 4.0 kHz as the high-frequency range), was emulated.
- The system corresponds to an experiment involving a cantilever plate excited by a force whose the frequency response was measured at six different locations.

Linear dynamical systems

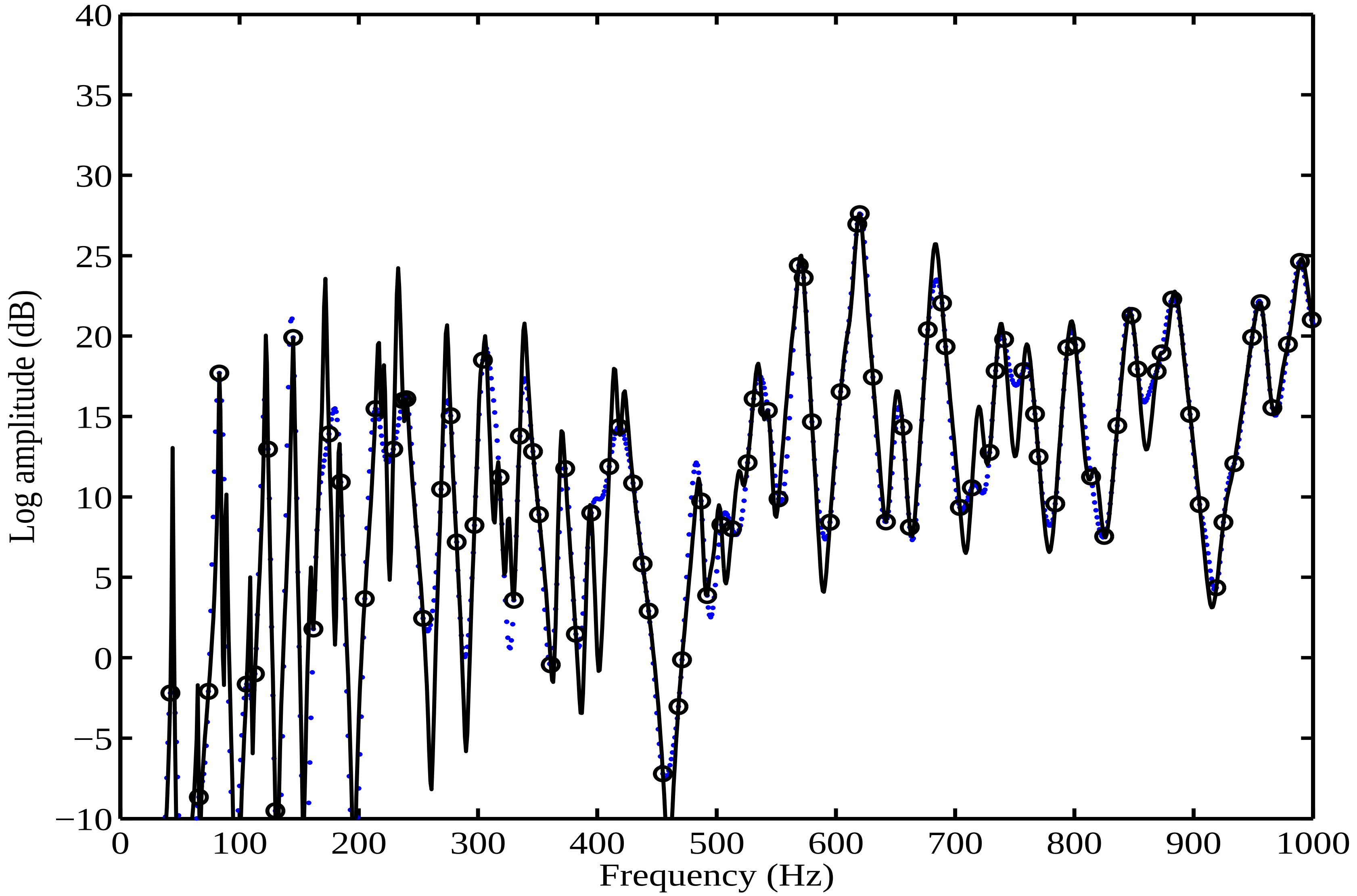


Figure 6: Predictive mean (· · ·) using 100 training runs (o), low frequency range.

Linear dynamical systems

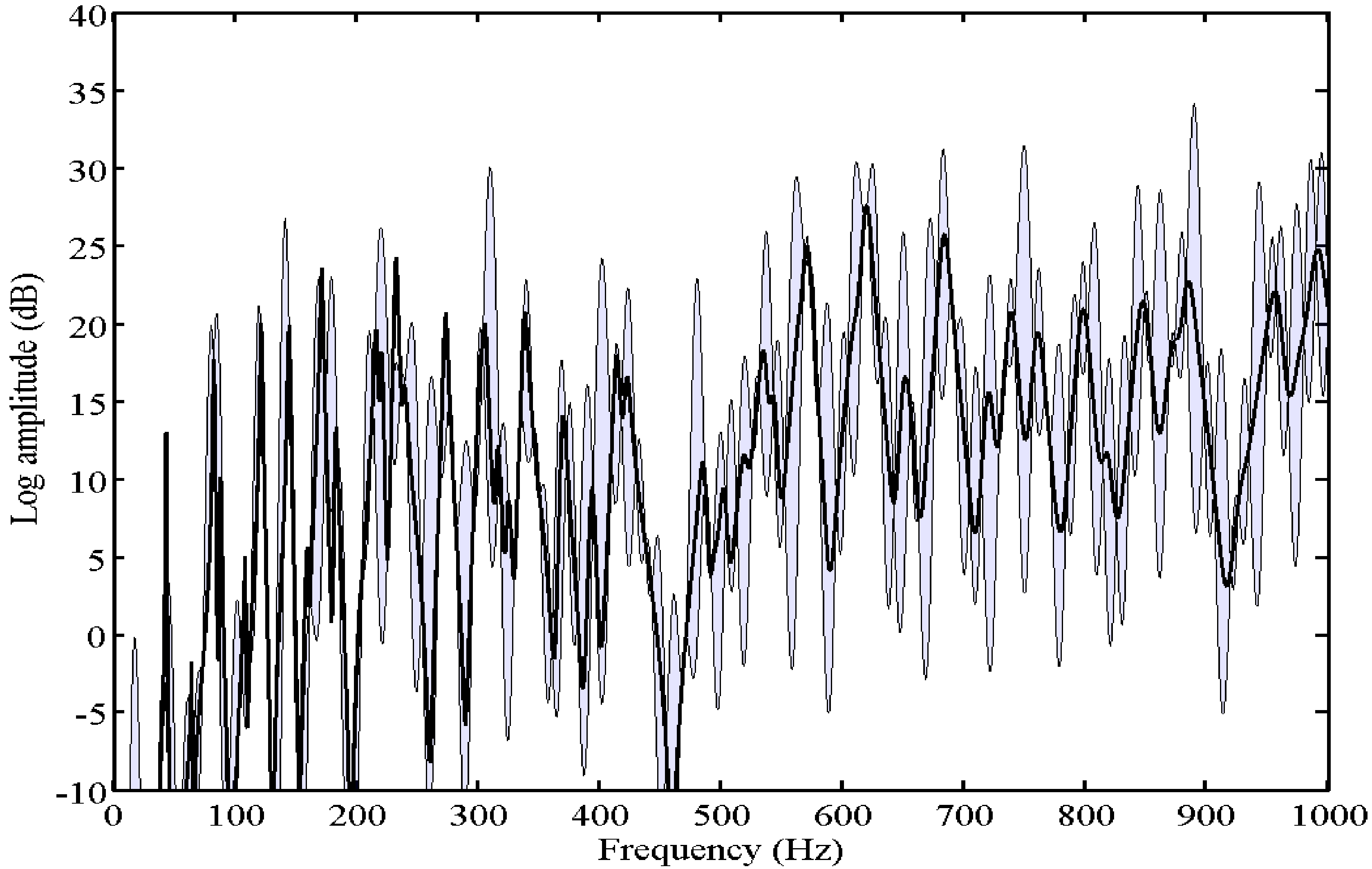


Figure 7: Uncertainty bounds using 100 training runs, low frequency range.

Linear dynamical systems

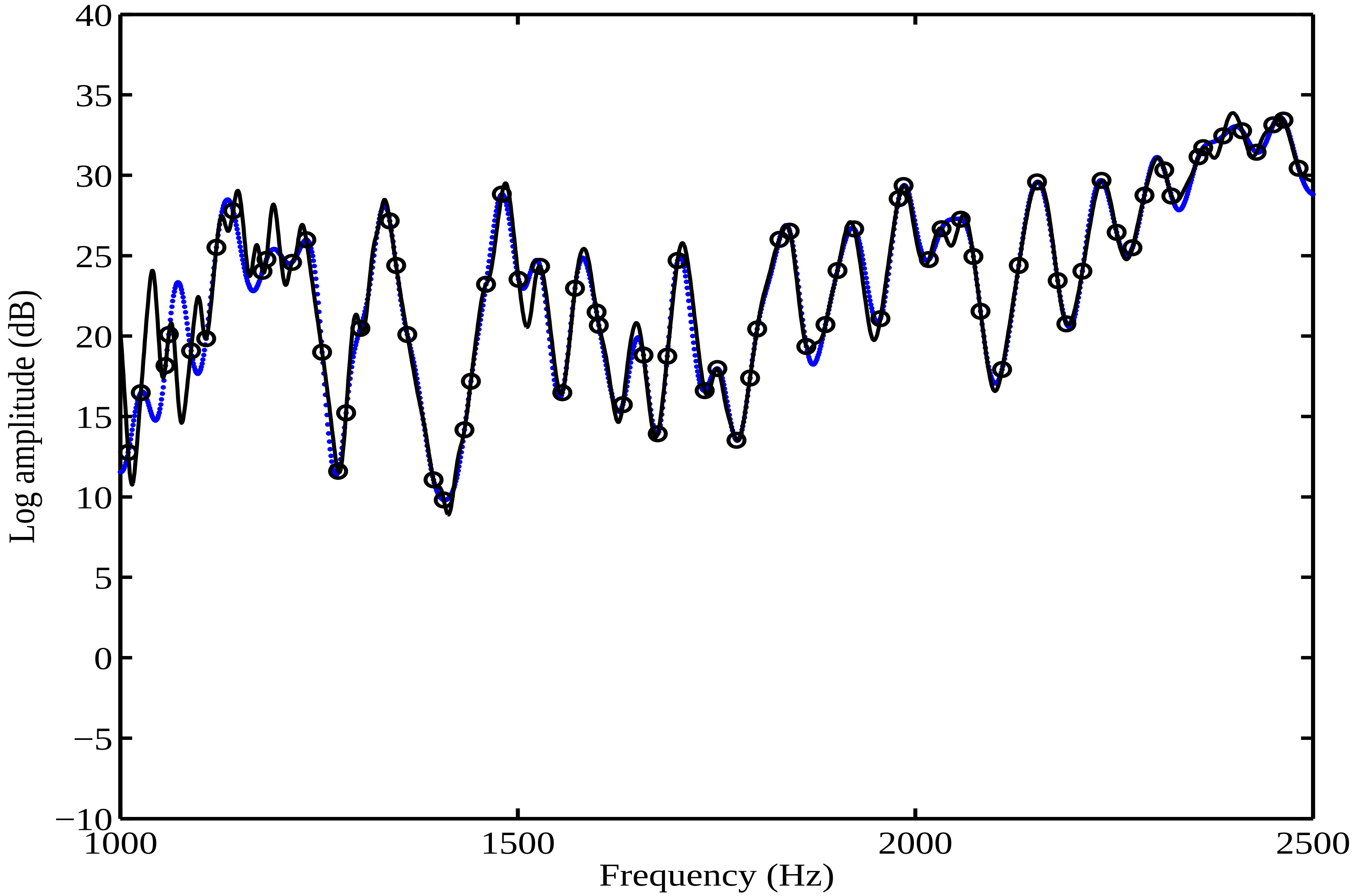


Figure 8: Predictive mean (· · ·) using 75 training runs (o), medium frequency range.

Linear dynamical systems

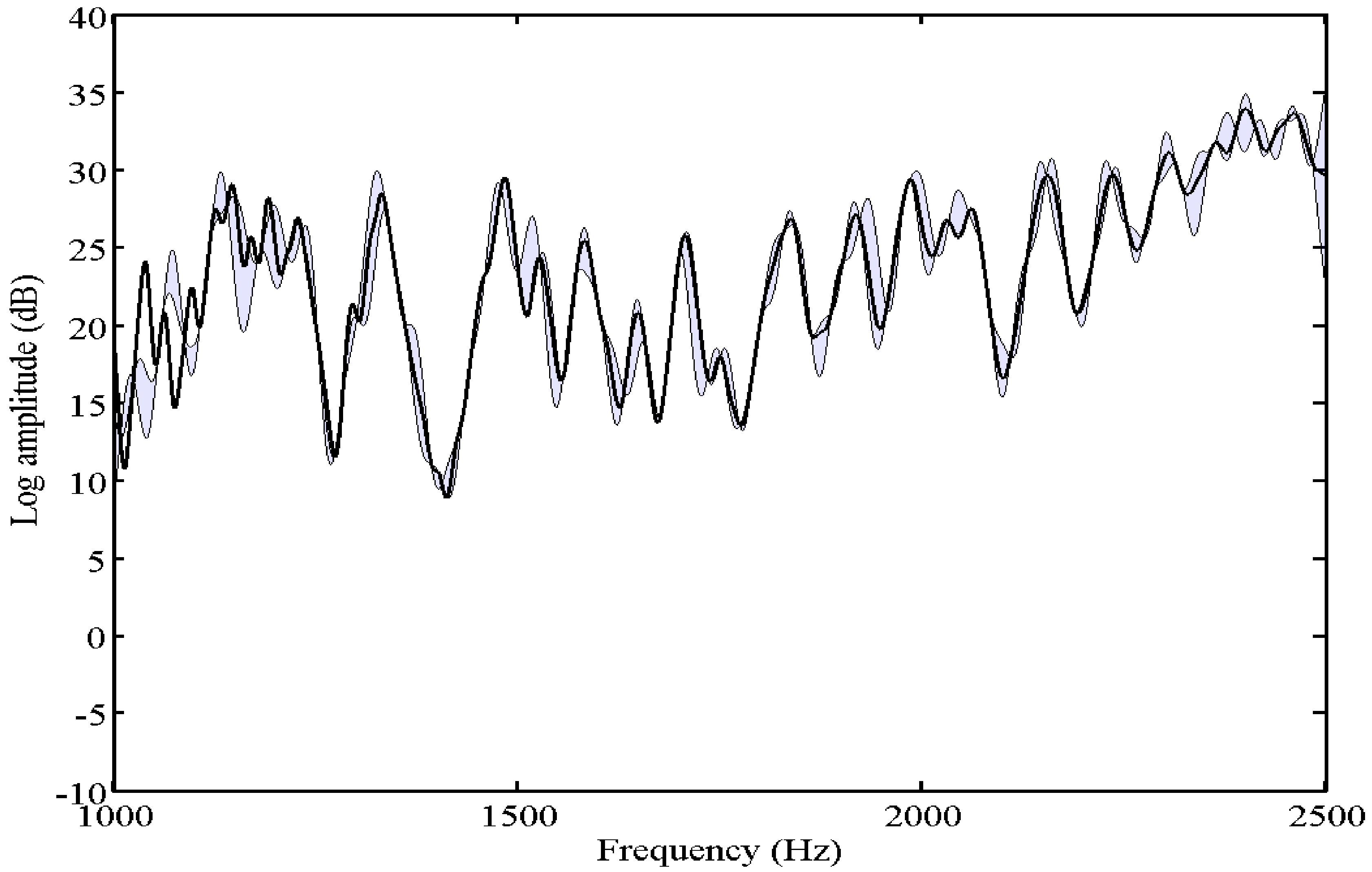


Figure 9: Uncertainty bounds using 75 training runs, medium frequency range.

Linear dynamical systems

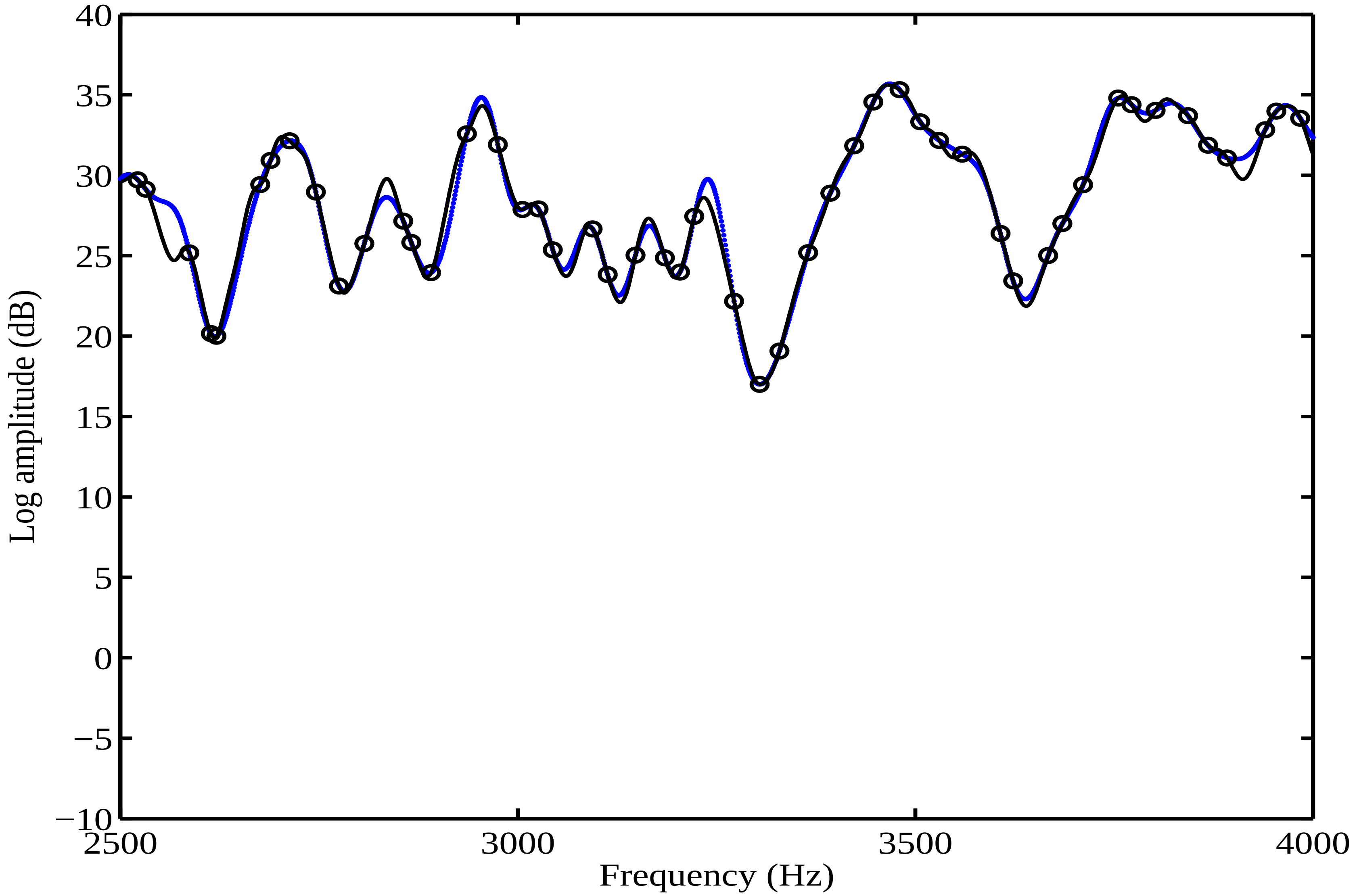


Figure 10: Predictive mean (· · ·) using 50 training runs (o), high frequency range.

Linear dynamical systems

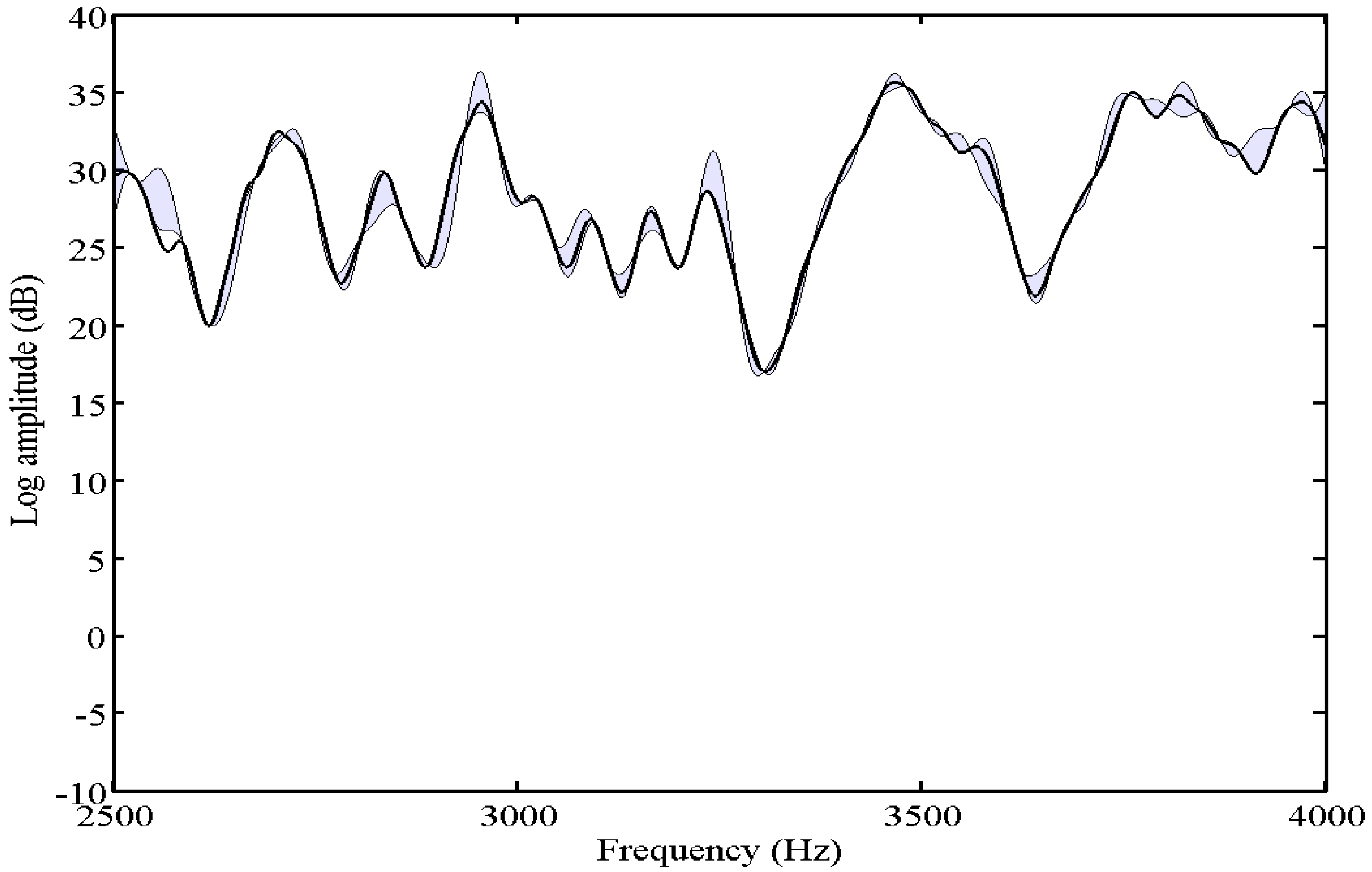


Figure 11: Uncertainty bounds using 50 training runs, high frequency range.

Linear dynamical systems

- With a resolution of 1 Hz, the number of design points in the figures above represents 10%, 5% and 3% the size of the input domain.
- The results were validated using adequacy measures, such as the standardized prediction error

$$\delta_j^{se}(\mathbf{y}^*) = \frac{y_j^* - m^{**}(\eta(\omega_j^*))}{\widehat{\sigma}\sqrt{C^{**}(\eta(\omega_j^*), \eta(\omega_j^*))}} \tag{7}$$

where the set of validation data $\mathbf{y}^* = [y_1^* = \eta(\omega_1^*), \ldots,$ $y_n^* = \eta(\omega_v^*)]^T$ should be such that $|\delta_j^{se}(\mathbf{y}^*)| \leq 2$.

Swansea University
Prifysgol Abertawe

Linear dynamical systems

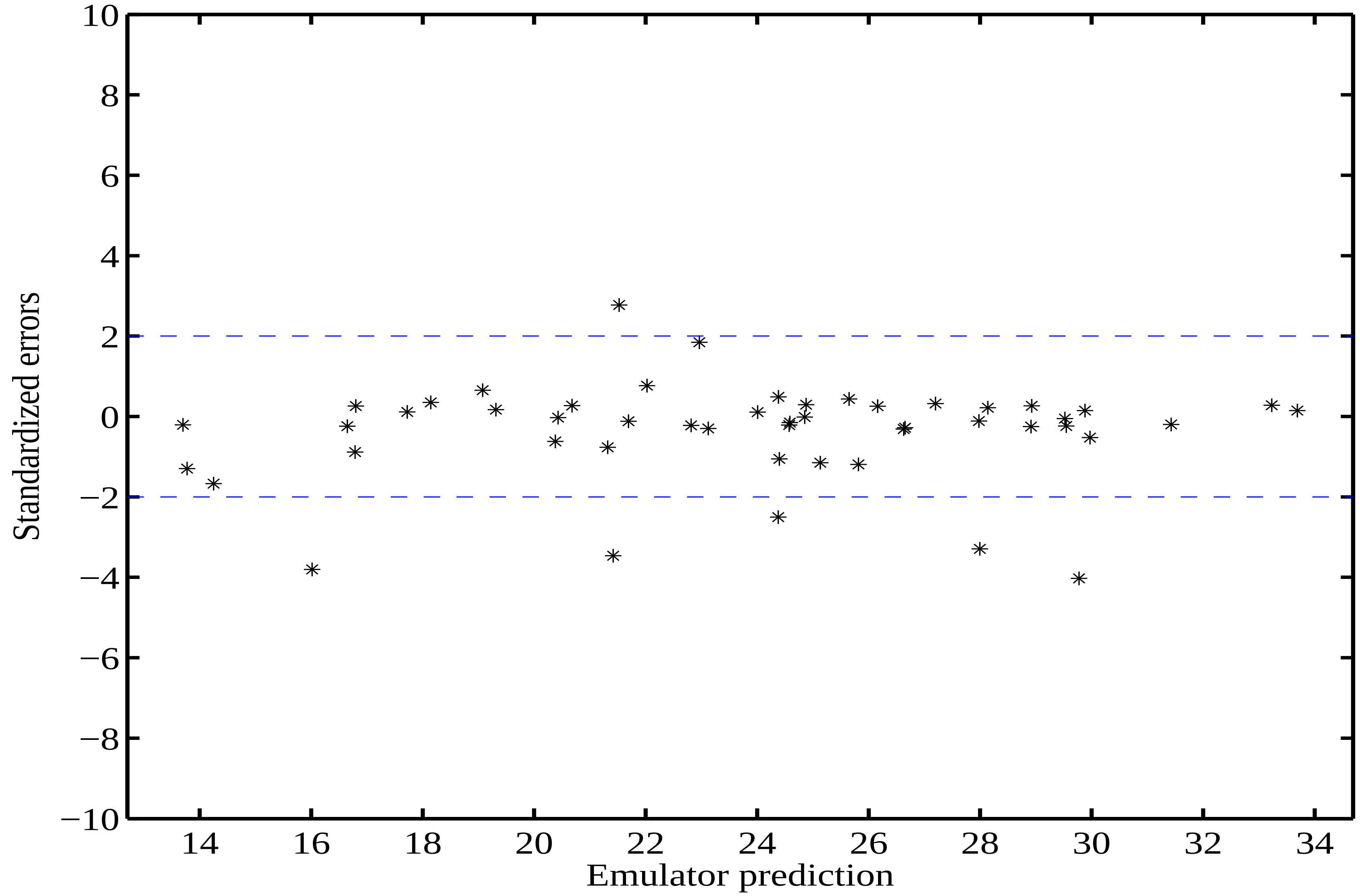


Figure 12: Predictive mean vs standardized errors for 50 validation points.

Linear dynamical systems

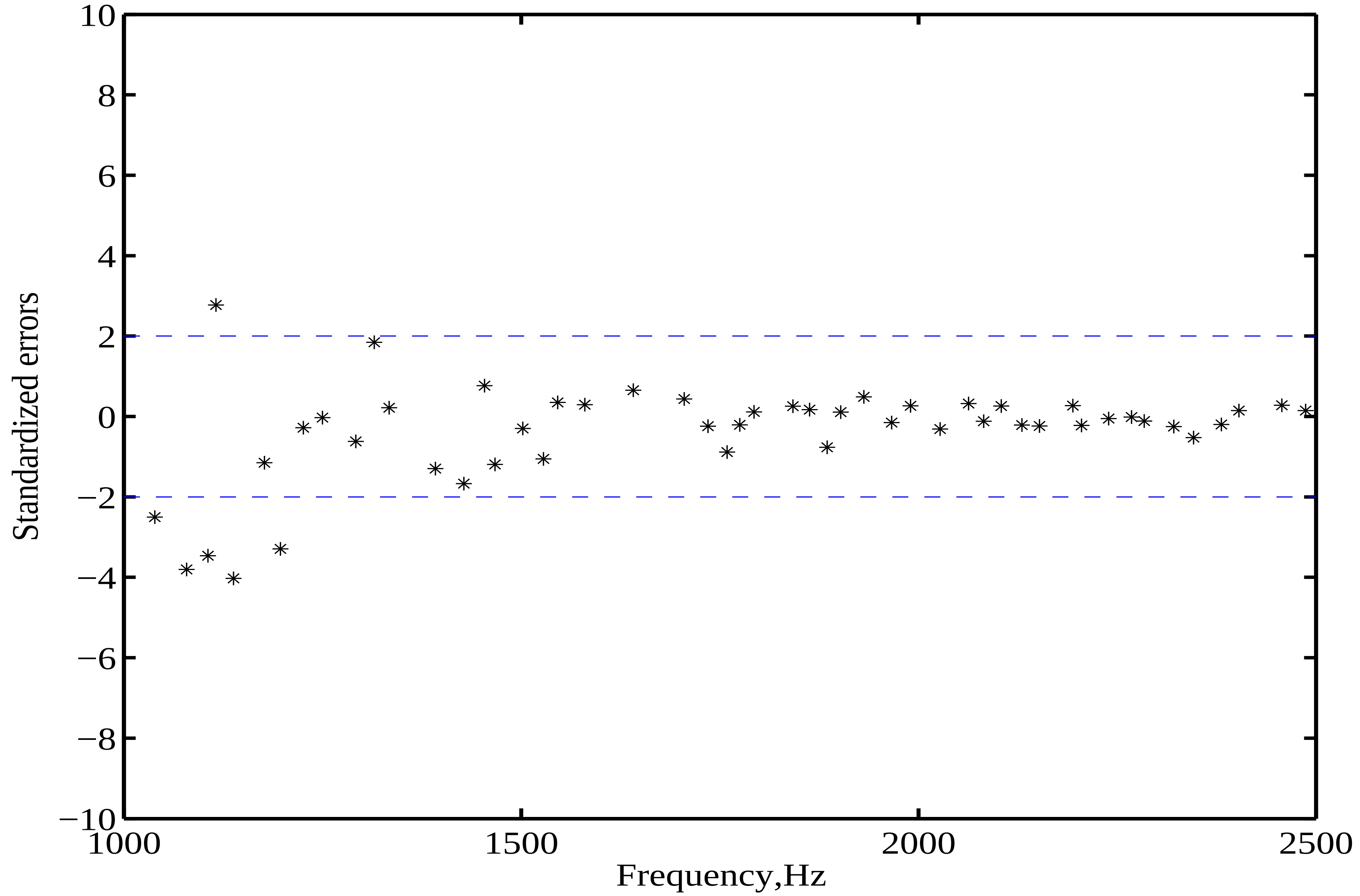


Figure 13: Validation points vs standardized errors for 50 validation points.

Initial Design

- **Title:** Gaussian Process Emulators for Dynamical Systems with Random Parameters
- **Conference:** Tenth International Conference on Structural Safety and Reliability (ICOSSAR 09); Osaka, Japan.
- **Rationale:** Compare two strategies for selecting design points for an emulator of the mean frequency response of a dynamical system with random parameters, taking Monte Carlo simulation as a benchmark. The input domain the simulator can be divided into two subdomains: the one corresponding to the frequency domain and the one associated with the random nature of the response. The comparison of strategies can help elucidate whether the accuracy and computation time of Gaussian process emulators can be improved.

Swansea University
Prifysgol Abertawe

Initial Design

- Let $S \simeq 10^4$ and $\underline{\theta} = [\theta_1, \ldots, \theta_S]^T$. Suppose that, for $\{\omega_\ell\}_{\ell=1}^M$ in the input domain of $\eta(\cdot)$, the mean FRF

$$\overline{\eta}(\omega_\ell, \underline{\theta}) = \frac{1}{S} \sum_{s=1}^{S} \eta(\omega_\ell, \theta_s) \tag{8}$$

  is to be estimated. Note that if $\eta(\cdot)$ is computationally expensive, M should be fairly small with respect to the size of the input domain.

- A straightforward approach is Monte Carlo simulation: For $\ell = 1, \ldots, M$, simulate $\{\eta(\omega_\ell, \theta_s)\}_{s=1}^S$ and evaluate Eq.(8) directly.

Swansea University
Prifysgol Abertawe

Initial Design

- The number of evaluations of $\eta(\cdot)$ using Monte Carlo simulation would be $SM \simeq 10^4 M$. The theoretical advantage of this approach is that the results are evaluated directly. Unfortunately, since $\eta(\cdot)$ is assumed to be computationally expensive, the method can quickly become intractable.
- **Emulation strategy 1:** Simulate a much smaller number of means and use that information as the training runs necessary to emulate the M means of interest.
- Select the design points $\omega_1^*, \ldots, \omega_m^*$, with $m \ll M$. Then obtain $\overline{\eta}(\omega_1^*, \underline{\theta}), \ldots, \overline{\eta}(\omega_m^*, \underline{\theta})$ as in Eq.(8). Use the resulting training runs $\{\omega_i^*, \overline{\eta}(\omega_i^*, \theta)\}_{i=1}^m$ to emulate the M means corresponding to $\{\omega_\ell\}_{\ell=1}^M$.

Initial Design

- The number of evaluations of $\eta(\cdot)$ with strategy 1 would be $Sm \simeq 10^4 m$. Theoretically, the computational burden should be reduced as $10^4 m \ll 10^4 M$.
- **Emulation strategy 2:** Select design points along the "θ-dimension" of the input domain of $\eta(\cdot)$, rather than frequency design points.
- First, generate $\theta_1^*, \ldots, \theta_L^*$ for $L \ll 10^4$. Then, for each of the M frequency points in $\{\omega_\ell\}_{\ell=1}^M$ compute the training runs $\{\theta_j^*, \eta(\omega_\ell, \theta_j^*)\}_{j=1}^L$ and emulate N_E points $\{\widetilde{\theta}_k, \eta(\omega_\ell, \widetilde{\theta}_k)\}_{k=1}^{N_E}$ to calculate the mean

$$\overline{\eta}(\omega_\ell, \underline{\widetilde{\theta}}) = \frac{1}{N_E} \sum_{p=1}^{N_E} \eta(\omega_\ell, \widetilde{\theta}_p) \tag{9}$$

Swansea University
Prifysgol Abertawe

Initial Design

where $\underline{\widetilde{\theta}} = [\widetilde{\theta}_1, \ldots, \widetilde{\theta}_S]^T$.

- The number of evaluations of $\eta(\cdot)$ would be $L \cdot M$. The computational burden should also be reduced compared to Monte Carlo simulation, as $L \cdot M \ll 10^4 M$.
- Both strategies above involve less evaluations of $\eta(\cdot)$ than pure Monte Carlo simulation. However, in order for strategy 2 to outperform strategy 1, it has to be true that

$$L \cdot M < S \cdot m \tag{10}$$

- Suppose $S = 10^4$ and $m = M/10$, which might be reasonable, since m is small. Then, the inequality is true if and only if $L < 10^3$, which is also plausible, since by assumption $L \ll 10^4$.

Swansea University
Prifysgol Abertawe

Initial Design

- Numerical experiments were performed to test the efficiency of both strategies, taking Monte Carlo simulation as benchmark. For that purpose, the stiffness matrix $\mathbf{K}(\theta)$ was modeled using a random k of the form

$$k = k_0(1 + \epsilon\theta) \tag{11}$$

  where k, $\epsilon \in \mathbb{R}$ and $\theta \sim U[0, 1]$.

- The result of implementing strategy 1 is shown in Figure 14. For $S = 10^4$ simulations, the total number of values to be inferred was $M = 180$. The number of design points was $m = 18$. In order for these design points to be evenly spread in the input domain $[0, 3]$, they were generated using Latin hypercube sampling.

Initial Design

- The rest of the parameters were $k_0 = 1$ and $\epsilon = 0.3$. The stars represent the Monte Carlo benchmark means $\overline{\eta}(\omega_1, \theta), \ldots, \overline{\eta}(\omega_{180}, \theta)$. The circles represent the emulated means. The shaded area is the superposition of the 10^4 simulations of $\eta(\omega, \theta)$.
- As predicted, strategy 1 resulted a less expensive approximation, taking 9.28 seconds whereas Monte Carlo simulation took 84.77 seconds.

Initial Design

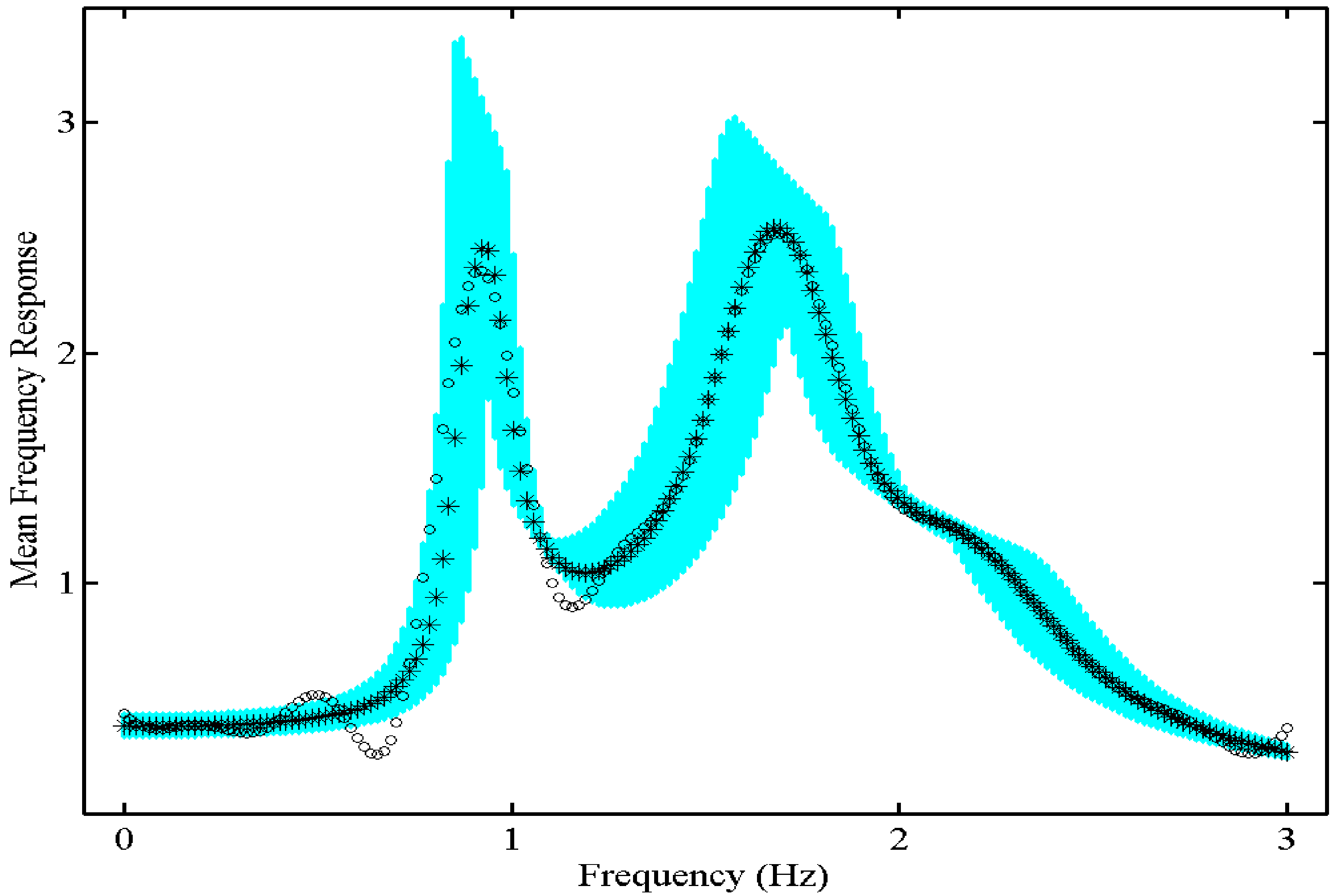


Figure 14: Emulation taking design points in the ω-dimension of $\eta(\cdot)$.

Initial Design

- For strategy 2, the mean for every $\omega_\ell \in \{\omega_\ell\}_{\ell=1}^{180}$ was emulated selecting an initial design $\theta_1^*, \ldots, \theta_L^*$, with $L = 10$. Latin hypercube sampling was used to evenly spread the design points in the corresponding input domain. Such input domain was $[k_0, k_0 + \epsilon]$, with $k_0 = 1$ and $\epsilon = 0.3$. Once the training runs were obtained, $N_E = 20$ values $\eta(\omega_\ell, \theta_1), \ldots, \eta(\omega_\ell, \theta_{20})$ were emulated and their mean was computed.
- An improvement in the accuracy of the approximation was observed. This strategy took 1.98 seconds, outperforming both Monte Carlo simulation and strategy 1. In Figure 15, the diamonds are the emulated means. The stars are the same Monte Carlo benchmark means and the shaded area is the superposition of the 10^4 simulations of $\eta(\omega, \theta)$.

Initial Design

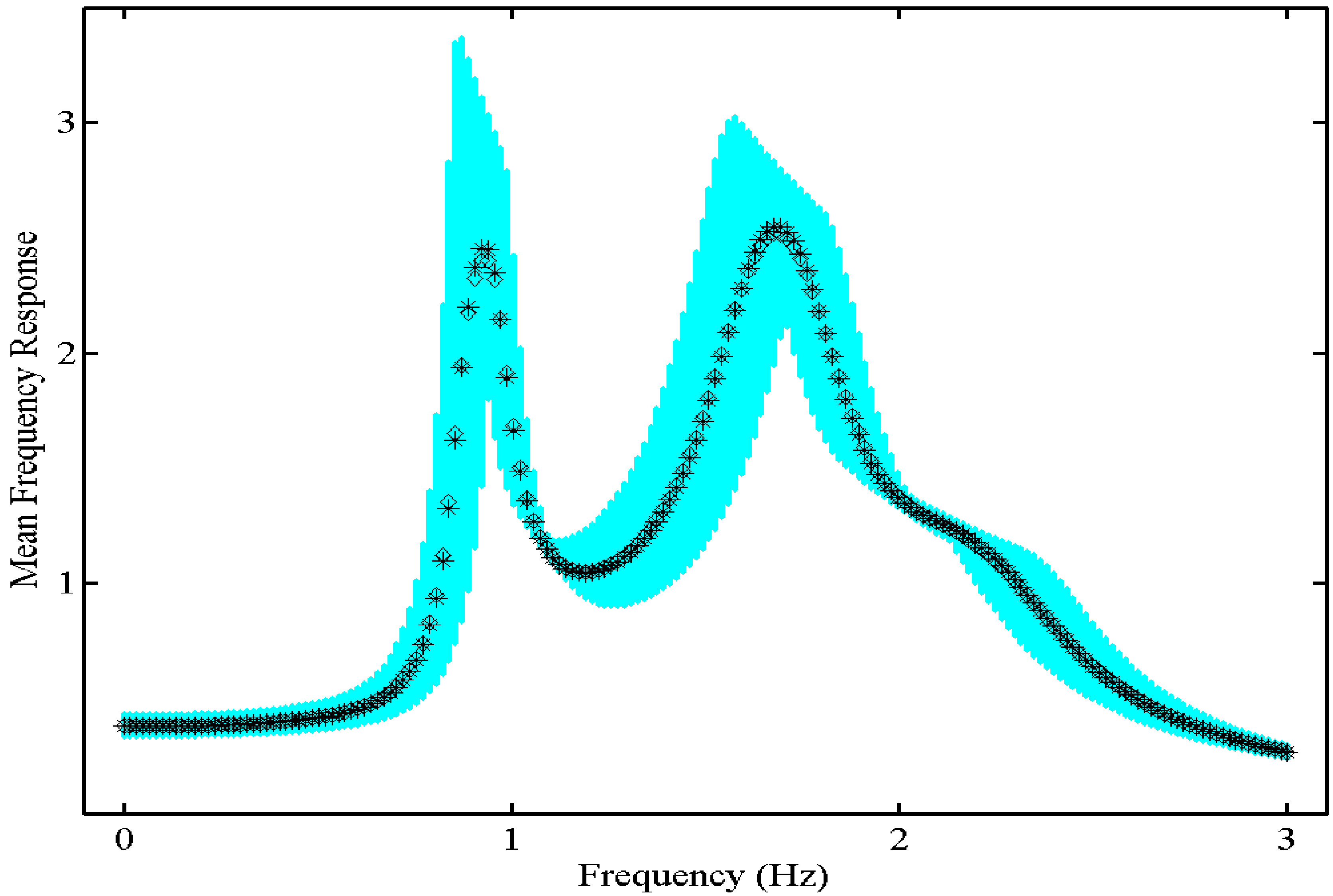


Figure 15: Emulation taking design points in the θ-dimension of $\eta(\cdot)$.

Random field discretisation

- **Title:** Bayesian Emulators and the Stochastic Finite Element Method
- **Conference:** The Sixth International Conference on Engineering Computational Technology; Athens, Greece.
- **Rationale:** The efficient evaluation of discretised random fields is necessary for calculating the response of complex engineering systems with uncertain parameters. In such cases, few training runs of an expensive simulator that computes the corresponding Karhunen-Loève expansion can be employed to build an emulator that approximates the random field involved.

Random field discretisation

- The Stochastic Finite Element Method (SFEM) models the elastic, mass and damping properties with the following equilibrium equation

$$\mathbf{D}(\mathbf{x},\theta)\mathbf{u}(\mathbf{x},\theta) = \mathbf{f}(\mathbf{x},\theta) \tag{12}$$

  where $\mathbf{D}(\mathbf{x},\theta) \in \mathbb{R}^{d\times d}$ is a random matrix, $\mathbf{u}(\mathbf{x},\theta) \in \mathbb{R}^{d}$ is the random response vector, $\mathbf{f}(\mathbf{x},\theta) \in \mathbb{R}^{d}$ is the (possibly) random forcing vector, and d is the number of degrees of freedom (dof) in the finite element mesh.

- The random matrix $\mathbf{D}(\mathbf{x},\theta)$ is modeled with a random field as:

$$\mathbf{D}(\mathbf{x},\theta) = \mathcal{H}(\mathbf{x},\theta)\mathbf{D}_{\mathbf{det}} \tag{13}$$

  where $\mathbf{D}_{\mathbf{det}}$ is a deterministic matrix.

Random field discretisation

- A convenient discretisation of the random field $\mathcal{H}(\mathbf{x}, \theta)$ is the Karhunen-Loève expansion (KLE), for which

$$\mathcal{H}(\mathbf{x}, \theta) = \mu(\mathbf{x}) + \sum_{i=1}^{\infty} \sqrt{\lambda_i} \xi_i(\theta) \phi_i(\mathbf{x}) \qquad (14)$$

where $\mu(\mathbf{x})$ is the mean of the process, $\{\xi_i(\theta)\}_{i=0}^{\infty}$ are random variables, $\{\lambda_i\}_{i=0}^{\infty}$ and $\{\phi_i(\mathbf{x})\}_{i=0}^{\infty}$ are the eigenvalues and eigenfunctions of the covariance kernel $\mathbf{K}(\cdot, \cdot)$, obtained by solving the integral equation

$$\int_{\Omega} \mathbf{K}(\mathbf{x}_i, \mathbf{x}_j) \phi_n(\mathbf{x}) d\mathbf{x}_i = \lambda_n \phi_n(\mathbf{x}_j) \qquad (15)$$

Random field discretisation

- A Gaussian homogeneous two-dimensional random field with mean μ, variance σ^2 and exponential covariance kernel was considered, that is

$$\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j) = \sigma^2 e^{-(|\mathbf{x}_i^{(1)} - \mathbf{x}_j^{(1)}|/\gamma_1 + |\mathbf{x}_i^{(2)} - \mathbf{x}_j^{(2)}|/\gamma_2)} \qquad (16)$$

  where the parameters γ_1, γ_1 are known as correlation lengths.

- If the order of the KLE is M, then the relevant simulator is

$$\eta(\mathbf{x}, \theta) = \mu + \sum_{i=1}^{M} \sqrt{\lambda_i}\xi_i(\theta)\phi_i(\mathbf{x}) \qquad (17)$$

Swansea University
Prifysgol Abertawe

Random field discretisation

- The spatial domain of the simulator was chosen to be the region $\mathcal{R}_L = [0, L] \times [0, L]$. Hence, the number of dof in the finite element mesh was $d = (L + 1)^2$.
- The initial spatial design $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ was selected using Latin hypercube sampling, such that n was 10% of d.
- Figure 16 shows the values of the emulator’s predictive mean compared against the simulator’s realization of the random field, as well as the initial design.

Swansea University
Prifysgol Abertawe

Random field discretisation

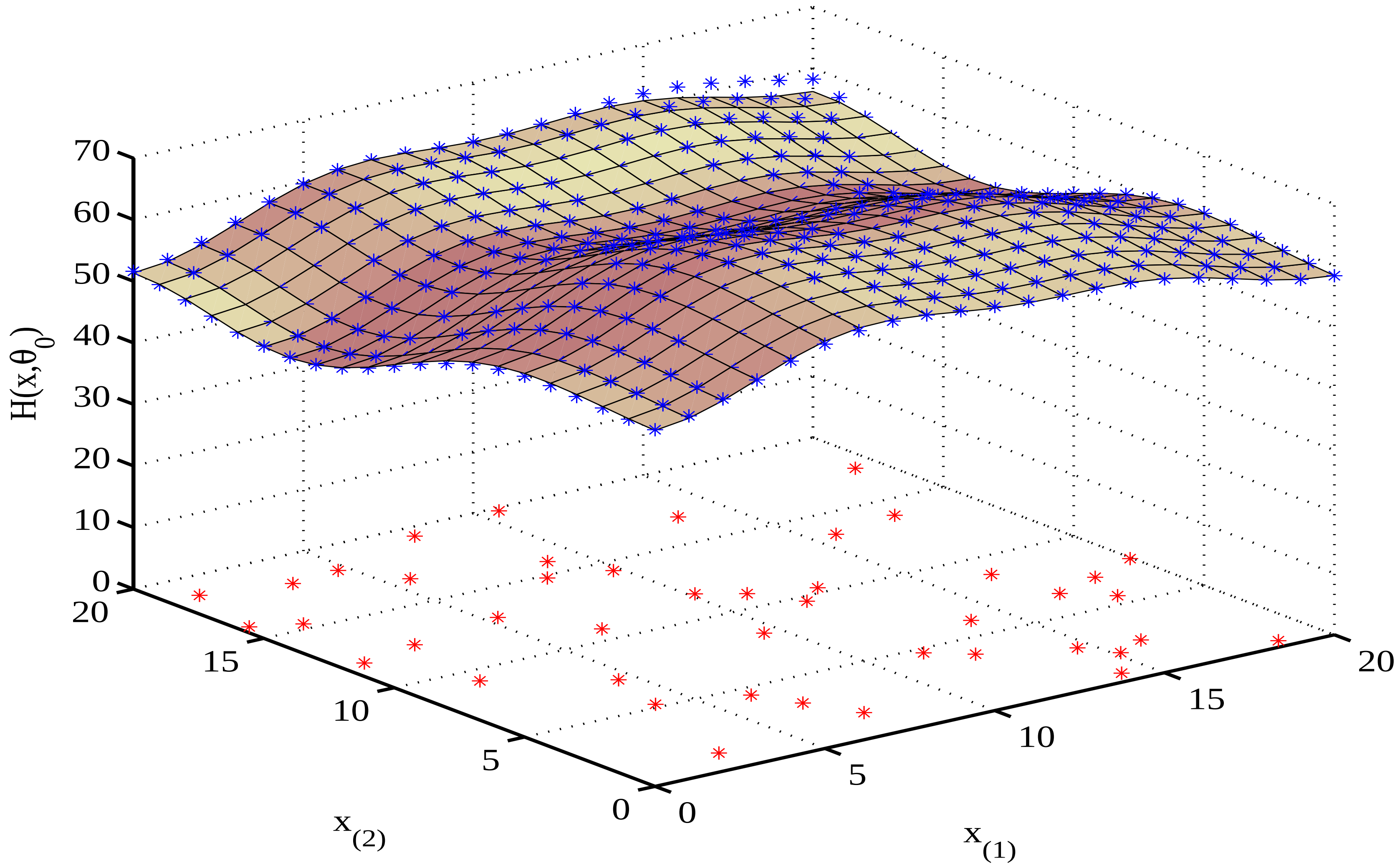


Figure 16: Emulation of the values of $\mathcal{H}(\mathbf{x}, \theta)$ at the nodal points and initial design.

Random field discretisation

- The time employed to produce one realization of $\mathcal{H}(\mathbf{x}, \theta)$ for an increasing number of nodes is shown in Table 1.

No. Nodes	Time (sec.) Direct	Time (sec.) Emulator
121	9.56	0.07
256	19.92	0.24
441	34.43	0.75
961	76.23	6.05
1681	131.29	17.76
2601	273.18	59.66

Table 1: Number of nodes vs. CPU time employed

Swansea University
Prifysgol Abertawe

Polynomial chaos expansion

- **Title:** Coupling Polynomial Chaos Expansions with Gaussian Process Emulators: An Introduction
- **Conference:** 27^{th} International Modal Analysis Conference (IMAC-XXVII); Orlando, Florida, USA.
- **Rationale:** Several methods to solve linear systems of stochastic partial differential equations are available in the literature. Despite their theoretical appeal, their implementation presents at least one of the following disadvantages: a) lack of the geometrical appeal; b) limited applicability due to restrictive analytical constraints; c) non-guaranteed convergence of the series expansions involved; d) potentially high computational cost. The Polynomial Chaos Expansion Method (PCEM) has gained attention for addressing a) - c). However, this method's applicability is still bounded by computational cost.

Polynomial chaos expansion

- Aiming to solve the system in Eq.(12), the PCEM expresses each component of the random displacement vector $\mathbf{u}(\mathbf{x}, \theta)$ as a series of orthogonal polynomials (e.g. Hermite) $\{\Psi_j(\theta)\}_{j=0}^{\infty}$, such that

$$\mathbf{u}(\mathbf{x}, \theta) = \sum_{j=0}^{P-1} \mathbf{c}_j(\mathbf{x})\Psi_j(\theta) \qquad (18)$$

  where each $\mathbf{c}_j$ is a deterministic vector.

- Simultaneously, the random field defining the random matrix $\mathbf{D}(\mathbf{x}, \theta)$ in Eq.(12) is discretised through the truncated Karhunen-Loève expansion such that

Swansea University
Prifysgol Abertawe

Polynomial chaos expansion

$$\mathbf{D}(\mathbf{x},\theta)=\sum_{i=0}^{M}\mathbf{D}_i(\mathbf{x})\xi_i(\theta) \tag{19}$$

where each $\mathbf{D}_i$ is a deterministic matrix.

- The approximate solution is obtained by minimizing the underlying residual

$$\mathcal{R}(\mathbf{x},\theta)=\sum_{i=0}^{M}\sum_{j=0}^{P-1}\mathbf{D}_i\mathbf{c}_j(\mathbf{x})\xi_i(\theta)\Psi_j(\theta)-\mathbf{f}(\mathbf{x},\theta) \tag{20}$$

by means of a projection onto the space spanned by $\{\Psi_j(\theta)\}_{j=0}^{P-1}$.

Swansea University
Prifysgol Abertawe

Polynomial chaos expansion

- It can be shown that the PCEM eventually yields a system of the form
$$\mathcal{K} \cdot \mathcal{U} = \mathcal{F} \tag{21}$$
where $\mathcal{K}$ is a global stiffness matrix of size $NP \times NP$, where N is the number of dof.
- Unfortunately, the number of polynomials P in the expansion grows very rapidly for small increases in the order of the polynomials p, and in the number of terms in the Karhunen-Loève expansion M, namely
$$P(p, M) = \sum_{k=0}^{p} \binom{M + k - 1}{k} \tag{22}$$

Polynomial chaos expansion

- A polynomial chaos simulator was used to solve a problem involving a cantilever thin square plate subject to a uniform tension along the free edge.
- The modulus of elasticity was modeled as a Gaussian random process with known mean and known exponential covariance structure.
- The aim was to calculate the norm of the displacement of each node in the finite element model of the plate. The plate was assumed to coincide with the set $[0, 1] \times [0, 1]$.
- The following table illustrates how quickly can the method become computationally intractable, even for the seemingly simple case of 25 nodes.

Polynomial chaos expansion

- For 25 nodes, the table shows the time employed to assemble the global stiffness matrix $\mathcal{K}$ and its size for a given order of the polynomial chaos expansion and order of the Karhunen-Loève expansion equal to 4. The parentheses indicate that Matlab returned an out-of-memory message.

No. nodes	p	Secs. Global	Size Global
25	2	0.73	750^2
25	4	14.2	$3{,}500^2$
25	6	301.44	$10{,}500^2$
25	8	1,722.73	$24{,}750^2$
25	10	7,420.65	$(50{,}050^2)$

Table 2: Computation time of the simulator, 25 of nodes.

Swansea University
Prifysgol Abertawe

Polynomial chaos expansion

- For 64 nodes, the table shows the time employed to assemble the global stiffness matrix $\mathcal{K}$ and its size for a given order of the polynomial chaos expansion and order of the Karhunen-Loève expansion equal to 4. The parentheses indicate that Matlab returned an out-of-memory message.

No. nodes	p	Secs. Global	Size Global
64	2	4.57	$1{,}920^2$
64	4	294.28	$8{,}960^2$
64	6	2,937.46	$26{,}880^2$
64	8	16,409.48	$(63{,}360^2)$

Table 3: Computation time of the simulator, 64 of nodes.

Swansea University
Prifysgol Abertawe

Polynomial chaos expansion

- For 121 nodes, the table shows the time employed to assemble the global stiffness matrix $\mathcal{K}$ and its size for a given order of the polynomial chaos expansion and order of the Karhunen-Loève expansion equal to 4. The parentheses indicate that Matlab returned an out-of-memory message.

No. nodes	p	Secs. Global	Size Global
121	2	30.27	3,630^2
121	4	1,289.50	16,940^2
121	6	12,464.53	(50,820^2)

Table 4: Computation time of the simulator, 121 of nodes.

Swansea University
Prifysgol Abertawe

Polynomial chaos expansion

- Solving for a small number of nodes, and computing the norms of their displacements, an emulator was used to infer the corresponding norms in the rest of the nodes.
- The nodes were selected generating a Latin hypercube in $[0,1] \times [0,1]$ and taking the closest node to each point generated, aiming to spread evenly the set of "design nodes" across the plate. The number of design nodes was chosen to be around 15% of the total.
- Figure 13 shows the norms of the node displacements as a function of the nodes' location, for the case of $p = 6$, $M = 4$ and 121 nodes. The circles correspond to the displacements of the design nodes, whereas the stars are emulated norms of displacements for the rest of the nodes.

Swansea University
Prifysgol Abertawe

Polynomial chaos expansion

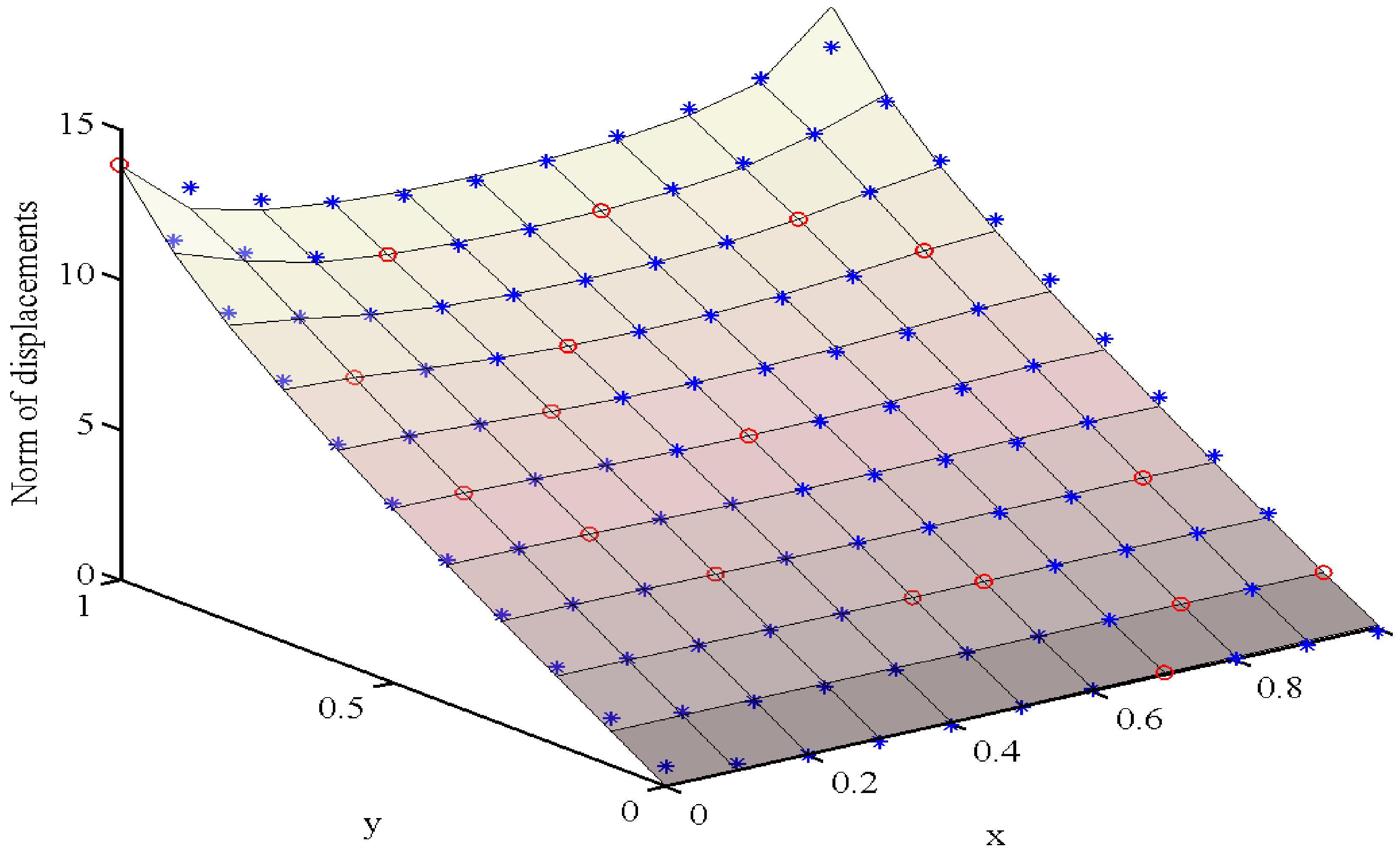
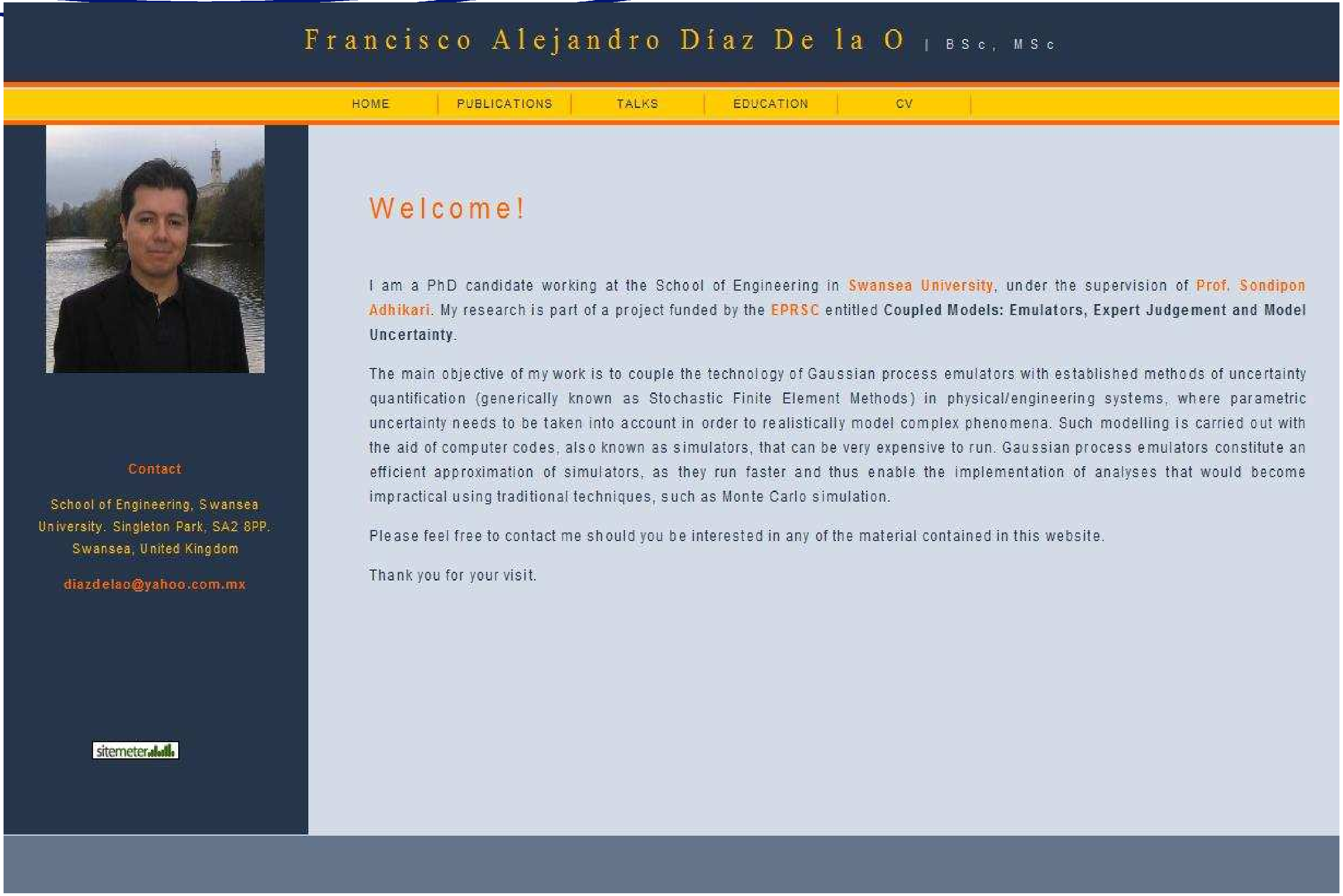


Figure 17: Norms of the displacements.

Webpage


Francisco Alejandro Díaz De la O | BSc, MSc

Welcome!

I am a PhD candidate working at the School of Engineering in Swansea University, under the supervision of Prof. Sondipon Adhikari. My research is part of a project funded by the EPRSC entitled **Coupled Models: Emulators, Expert Judgement and Model Uncertainty**.

The main objective of my work is to couple the technology of Gaussian process emulators with established methods of uncertainty quantification (generically known as Stochastic Finite Element Methods) in physical/engineering systems, where parametric uncertainty needs to be taken into account in order to realistically model complex phenomena. Such modelling is carried out with the aid of computer codes, also known as simulators, that can be very expensive to run. Gaussian process emulators constitute an efficient approximation of simulators, as they run faster and thus enable the implementation of analyses that would become impractical using traditional techniques, such as Monte Carlo simulation.

Please feel free to contact me should you be interested in any of the material contained in this website.

Thank you for your visit.

Contact

School of Engineering, Swansea University. Singleton Park, SA2 8PP. Swansea, United Kingdom

diazdelao@yahoo.com.mx

Figure 18: Visit us at engweb.swan.ac.uk/∼diazdelao.

Future Works

- Validation measures for stochastic emulators.
- Emulation of more general random fields, such as Gaussian with a variety of correlation structures and non-Gaussian.
- Comparison of emulators applied to the PCEM compared against emulators applied to the spectral representation method.
- Emulators for the solution of the Fredholm integral equation compared against emulators for the Karhunen-Loève expansion.
- Exploration of regions of the input domain via efficient genetic algorithms such as particle swarm optimization.

Swansea University
Prifysgol Abertawe

References

- O'Hagan, A. (2006) Bayesian Analysis of Computer Code Outputs: A tutorial, *Reliability Engineering & System Safety*, **91**, 1290-1300.
- Goldstein, M. (2007) Uncertainty Analysis for Complex Physical Models, *XXX Research Student's Conference in Probability and Statistics*, University of Durham, U.K.
- Thomke, S., Holzner, M., and Gholami, T. (1999) The Crash in the Machine, *Scientific American*, **280**, 72-77.
- Challenor, P. G., Hankin, R. K. S., Marsh, R. (2006) Towards the Probability of Rapid Climate Change, in Scellnhuber, H. J., Cramer, W., Nakicenovic, N., Wigley, T., Yohe, G. (eds), *Avoiding Dangerous Climate Change*, Cambridge University Press, Cambridge.
- Haylock, R. G. and O'Hagan, A. (1996) On Inference for Outputs of Computationally Expensive Algorithms with Uncertainty on the Inputs, *Bayesian Statistics 5*, Oxford University Press, Oxford.
- Adhikari, S., Friswell, M. I., and Lonkar, K. (2007) Uncertainty in structural dynamics: Experimental case studies on beams and plates, *Proceedings of the Computational Methods in Structural Dynamics and Earthquake Engineering*, Crete, Greece.

Swansea University
Prifysgol Abertawe