Safety, Reliability and Risk of Structures, Infrastructures and Engineering Systems – Furuta, Frangopol & Shinozuka (eds)
© 2010 Taylor & Francis Group, London, ISBN 978-0-415-47557-0

Gaussian Process Emulators for Dynamical Systems with Random Parameters

F. A. DiazDelaO & S. Adhikari
Swansea University, *Swansea, United Kingdom*

ABSTRACT. This paper compares two strategies that reduce the cost of running a simulator of the frequency response of a dynamical system. To this end, the input domain of the simulator is divided in two subdomains, one corresponding to the frequency domain and one associated with the random nature of the response. The mean frequency response is then approximated using Gaussian process emulators, which take inputs from either subdomain depending on the strategy. The results obtained confirm that Gaussian process emulation is less a computationally expensive approach than Monte Carlo simulation. They also show that selecting different sets of inputs upon which to build an emulator may improve the accuracy and computation time of this kind of surrogate models.

1 INTRODUCTION

Despite the spectacular advance in computational technology and the development of increasingly efficient algorithms over the recent decades, complex engineering systems are studied using computer codes (also known as *simulators*) which might still be very expensive to run. For instance, the computational cost associated with running a high-resolution finite element model can be a serious impediment, even for obtaining the dynamical response at few frequency points. For this reason, an increasingly common solution is to employ a less expensive surrogate model to investigate complex systems. There exists a vast amount of methods of surrogate modeling, such as Taylor series approximations with multipoint approximations, black-box modeling, response surface methods, and sparse approximation techniques (Faravelli, 1989; Fadel et al., 1990; Chatfield, 1995; Vapnik, 1998). For a detailed account of such techniques, see Keane and Nair (2005). Yet another method of surrogate modeling is to construct a statistical approximation to the simulator, known as *Gaussian process emulator*. Such technology is based on the analysis and design of computer experiments (Sacks et al., 1989; Satner et al., 2003), and on concepts of Bayesian statistics. Using this approach, it is possible to efficiently make inference about unknown values of a computer code by evaluating a fairly limited number of carefully selected points in the input domain.

Gaussian process emulators have been implemented in various scientific fields. Challenor et al. (2006) emulated what they consider to be a moderately complex climate model. Rougier (2007) presented another application to a climate model. Haylock and O'Hagan (1996) emulated a model of doses to organs of the body after ingestion of a radioactive substance. Kolachalama et al. (2007) applied emulation to analyze the influence of model parameters applied to human hemodynamics. The simulators involved in these studies can be considered deterministic, as they always produce the same output given the same set of inputs.

The aim of this paper is to compare two strategies for selecting points to run a simulator of the mean frequency response of a dynamical system with a random parameter, taking Monte Carlo simulation as a benchmark. The input domain of the simulator is divided in two well differentiated subdomains: the one corresponding to the frequency domain and the one associated with the random nature of the response. The results obtained verify that emulation is a less computationally expensive alternative than pure Monte Carlo simulation. More importantly, the comparison of selection strategies helps elucidate whether the accuracy and computation time of Gaussian process emulators can be further improved.

2 GAUSSIAN PROCESS EMULATORS

2.1 *Mathematical Background*

Let $\eta(\cdot)$ be a computationally expensive simulator that for every input $\mathbf{x}$, it returns the output $\eta(\mathbf{x})$. The cost of running $\eta(\cdot)$ implies that it can only be evaluated

at a very limited number of inputs. This allows $\eta(\cdot)$ to be regarded as a random variable, in the sense that the output is unknown until the simulator is actually run. Assume that it admits the following stochastic representation

$$\eta(\cdot) = \mathbf{h}(\cdot)^T\boldsymbol{\beta} + Z(\cdot) \tag{1}$$

where $\mathbf{h}(\cdot)$ is a vector of known functions, $\boldsymbol{\beta}$ is a vector of unknown coefficients, and $Z(\cdot)$ is a stochastic process with mean zero and covariance function $Cov(\cdot,\cdot)$.

An emulator is a statistical approximation to the simulator. Thus, it provides both an approximation and a probability distribution for $\eta(\cdot)$. Emulation works by carefully selecting a small set of *training runs* $\{\mathbf{x}_\ell, \eta(\mathbf{x}_\ell)\}_{\ell=1}^n$, which are used to update a prior distribution of the simulator. Following Haylock and O'Hagan (1996), an analytically convenient choice for such prior is a Gaussian process distribution of the form

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

where the correlation function $C(\cdot,\cdot)$ is such that

$$C(\mathbf{x}, \mathbf{x}') = e^{-(\mathbf{x}-\mathbf{x}')^T\mathbf{B}(\mathbf{x}-\mathbf{x}')} \tag{3}$$

and $\mathbf{B}$ is a diagonal positive definite matrix of smoothness parameters.

Definition. Gaussian stochastic process: Let $\mathbf{x} \in \mathbb{R}^d$. Then $Z(\cdot)$ is a Gaussian stochastic process if for any $J \geq 1$ and any choice $\{\mathbf{x}_1, \ldots, \mathbf{x}_J\}$, the vector $[Z(\mathbf{x}_1), \ldots, Z(\mathbf{x}_J)]^T$ has a multivariate normal distribution.

After conditioning on the training runs and updating the prior distribution (2), the mean of the resulting posterior distribution approximates the output of the simulator at any untried input, whereas it reproduces the known output of the simulator at each initial input. The corresponding variance quantifies the uncertainty that arises from having only a limited number of evaluations of $\eta(\cdot)$. Very conveniently, the posterior is also a Gaussian process distribution, obtained as follows. Let $\mathbf{y} = [\mathbf{y}_1 = \eta(\mathbf{x}_1), \ldots, \mathbf{y}_n = \eta(\mathbf{x}_n)]^T$, where $\mathbf{x}_1, \ldots, \mathbf{x}_n$ are called *design points*. Define $\mathbf{H} = [\mathbf{h}(\mathbf{x}_1), \ldots, \mathbf{h}(\mathbf{x}_n)]^T$ and $\mathbf{A} \in \mathbb{R}^{n\times n}$ with $\mathbf{A}_{\ell j} = C(x_\ell, x_j)\ \forall \ell, j \in \{1, \ldots, n\}$. Thus,

$$\mathbf{y}|\boldsymbol{\beta}, \sigma^2 \sim N(\mathbf{H}\boldsymbol{\beta}, \sigma^2\mathbf{A}) \tag{4}$$

To incorporate the objective information $\mathbf{y}$ and obtain the distribution of $\eta(\cdot)|\mathbf{y}$, use the following result, given in Krzanowski (2000).

Theorem. Let $\mathbf{z} \in \mathbb{R}^d$ be a random vector such that $\mathbf{z} \sim N(\boldsymbol{\mu}, \Sigma)$. Partition $\mathbf{z}$ as $(\mathbf{z}_1, \mathbf{z}_2)^T$, where $\mathbf{z}_1 \in \mathbb{R}^p$ and $\mathbf{z}_2 \in \mathbb{R}^{d-p}$. Consequently, partition $\boldsymbol{\mu} = (\boldsymbol{\mu}_1, \boldsymbol{\mu}_2)^T$ and $\Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix}$, so that $\mathbf{E}[\mathbf{z}_j] = \boldsymbol{\mu}_j$ and $Cov(\mathbf{z}_j, \mathbf{z}_k) = \Sigma_{jk}$. Then, $\mathbf{z}_1|\mathbf{z}_2 \sim N(\widetilde{\boldsymbol{\mu}}, \widetilde{\Sigma})$, where $\widetilde{\boldsymbol{\mu}} = \boldsymbol{\mu}_1 + \Sigma_{12}\Sigma_{22}^{-1}(\mathbf{z}_2 - \boldsymbol{\mu}_2)$ and $\widetilde{\Sigma} = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}$.

From this, it follows that

$$\eta(\cdot)|\mathbf{y}, \boldsymbol{\beta}, \sigma^2 \sim N(m^*(\cdot), \sigma^2 C^*(\cdot,\cdot)) \tag{5}$$

where

$$m^*(x) = \mathbf{h}(x)^T\boldsymbol{\beta} + \mathbf{t}(x)\mathbf{A}^{-1}(\mathbf{y} - \mathbf{H}\boldsymbol{\beta}) \tag{6}$$

$$C^*(\mathbf{x}, \mathbf{x}') = C(\mathbf{x}, \mathbf{x}') - \mathbf{t}(\mathbf{x})^T A^{-1}\mathbf{t}(\mathbf{x}') \tag{7}$$

$$\mathbf{t}(\mathbf{x}) = [C(\mathbf{x}, \mathbf{x}_1), \ldots, C(\mathbf{x}, \mathbf{x}_n)]^T \tag{8}$$

Removing the conditioning on $\boldsymbol{\beta}$ using standard integration techniques (Haylock and O'Hagan, 1996), obtain the posterior distribution

$$\eta(\cdot)|\mathbf{y}, \sigma^2 \sim N(m^{**}(\cdot), \sigma^2 C^{**}(\cdot,\cdot)) \tag{9}$$

where

$$m^{**}(\mathbf{x}) = \mathbf{h}(\mathbf{x})^T\widehat{\boldsymbol{\beta}} + \mathbf{t}(\mathbf{x})\mathbf{A}^{-1}(\mathbf{y} - \mathbf{H}\widehat{\boldsymbol{\beta}}) \tag{10}$$

$$C^{**}(\mathbf{x}, \mathbf{x}') = \tag{11}$$

$$\begin{aligned} &C^*(\mathbf{x}, \mathbf{x}') + (\mathbf{h}(\mathbf{x})^T \\ &-\mathbf{t}(\mathbf{x})^T\mathbf{A}^{-1}\mathbf{H})(\mathbf{H}^T\mathbf{A}^{-1}\mathbf{H})^{-1}(\mathbf{h}(\mathbf{x}')^T \\ &-\mathbf{t}(\mathbf{x}')^T\mathbf{A}^{-1}\mathbf{H})^T \end{aligned}$$

$$\widehat{\boldsymbol{\beta}} = (\mathbf{H}^T\mathbf{A}^{-1}\mathbf{H})^{-1}\mathbf{H}^T\mathbf{A}^{-1}\mathbf{y} \tag{12}$$

To estimate σ in (9), let q be the rank of $\mathbf{H}$. Then

$$\widehat{\sigma}^2 = \frac{\mathbf{y}^T(\mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{H}(\mathbf{H}^T\mathbf{A}^{-1}\mathbf{H})\mathbf{H}^T\mathbf{A}^{-1})\mathbf{y}}{n - q - 2} \tag{13}$$

The complete process is summarized below.

Emulation Algorithm

1. Select the design points $\mathbf{x}_1, \ldots, \mathbf{x}_n$.
2. Obtain the vector of observations $\mathbf{y} = [\mathbf{y}_1 = \eta(\mathbf{x}_1), \ldots, \mathbf{y}_n = \eta(\mathbf{x}_n)]^T$.
3. Update the prior distribution (2), which contains subjective information, by adding the objective information $\mathbf{y}$ and thus obtaining the posterior distribution (9). This enables the calculation of the predictive mean $m^{**}(\cdot)$, given the data $\mathbf{y}$. As already mentioned, such mean is a fast approximation of $\eta(\mathbf{x})$ for any $\mathbf{x}$.

2.2 *Example*

For illustration of the above, consider the nonproportionally damped three-degree-of-freedom spring-mass system shown in Figure 1. The simulator of the corresponding frequency response function (FRF) is regarded as if it were computer intensive. To obtain such FRF, begin with the equation of motion of a damped N-degree-of-freedom linear structural system, given by

$$\mathbf{M}\ddot{\mathbf{u}}(t) + \mathbf{C}\dot{\mathbf{u}}(t) + \mathbf{K}\mathbf{u}(t) = \mathbf{f}(t) \tag{14}$$

where $\mathbf{f}(t) \in \mathbb{R}^N$ is the forcing vector, $\mathbf{u}(t) \in \mathbb{R}^N$ is the response vector and $\mathbf{M} \in \mathbb{R}^{N\times N}$, $\mathbf{C} \in \mathbb{R}^{N\times N}$ and $\mathbf{K} \in \mathbb{R}^{N\times N}$ are respectively the mass, damping and stiffness matrices.

After calculating the Fourier transform, Eq. (14) is expressed in terms of the excitation frequency level, $\omega \in [0, ..., \infty)$, as

$$\mathbf{D}(\omega)\bar{\mathbf{u}}(\omega) = \bar{\mathbf{f}}(\omega) \tag{15}$$

where $\bar{\mathbf{u}}(\omega)$ and $\bar{\mathbf{f}}(\omega)$ are the Fourier transforms of $\mathbf{u}$ and $\mathbf{f}$ respectively. The complex symmetric matrix $\mathbf{D}(\omega)$, known as *dynamic stiffness matrix*, is given by

$$\mathbf{D}(\omega) = -\omega^2\mathbf{M} + i\omega\mathbf{C} + \mathbf{K} \tag{16}$$

If $\mathbf{D}(\omega)^{-1}$ exists, the response vector is $\bar{\mathbf{u}}(\omega) = \mathbf{D}(\omega)^{-1}\bar{\mathbf{f}}(\omega)$. Since $\mathbf{D} \in \mathbb{C}^{N\times N}$, only the moduli of the response vector components are relevant in practice. Hence the simulator is

$$\eta(\omega) = \left|[-\omega^2\mathbf{M} + i\omega\mathbf{C} + \mathbf{K}]^{-1}\bar{\mathbf{f}}(\omega)\right| \tag{17}$$

Note that, in terms of the emulation algorithm, the

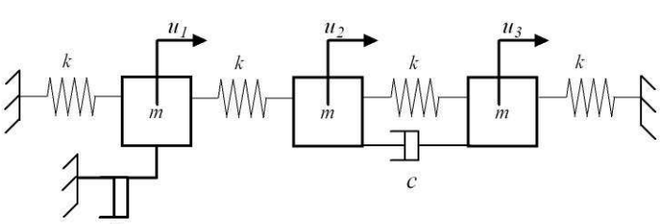

Figure 1: Three-degree-of-freedom damped spring-mass system; $m = 0.8$ kg, $k = 1$ N/m, $c = 0.2$ Ns/m.

design points will now be denoted by $\omega_1, \ldots, \omega_n$. Let the mass of each block be 0.8 kg, the stiffness of each spring be 1 N/m, and the viscous damping constant associated with each block be 0.2 Ns/m. The mass, stiffness and damping matrices of this simple system can be computed as

$$\mathbf{M} = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & m \end{bmatrix}, \quad \mathbf{K} = \begin{bmatrix} 2k & -k & 0 \\ -k & 2k & -k \\ 0 & -k & 2k \end{bmatrix}$$

$$\text{and} \quad \mathbf{C} = \begin{bmatrix} c & -c & 0 \\ -c & c & 0 \\ 0 & 0 & c \end{bmatrix} \tag{18}$$

where $m = 0.8$, $k = 1$ and $c = 0.2$. Suppose the dynamic response is to be obtained for the forcing vector $\mathbf{f} = [0, 1, 0]^T$. Figure 2 shows the case when $k = 3$ and 23 training runs (the circles) are used. The predictive mean of the emulator (the dots) approximates the values of the simulator (the solid line) at several untried inputs throughout the input domain. On the other hand, it returns the exact value of the simulator at each one of the training runs. Additionally, Figure 3

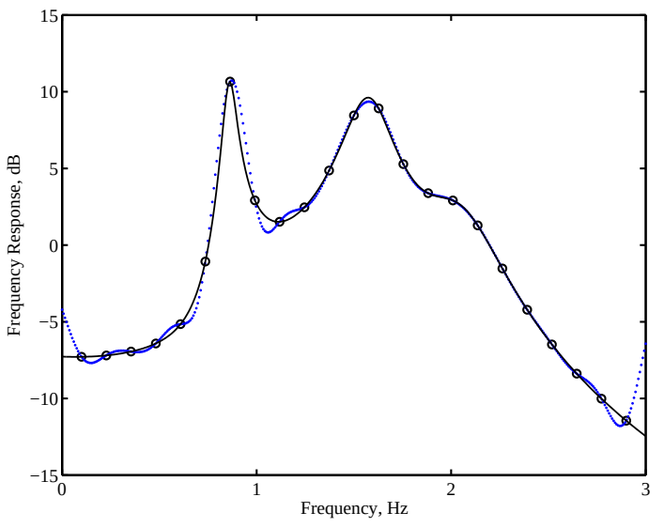

Figure 2: Emulation of $\eta(\omega)$ for a non proportionally damped spring-mass system with 3 degrees of freedom. The forcing vector is $[0, 1, 0]^T$; (-): output of the simulator, (o): training runs, $(\cdots)$: emulator's predictive mean.

shows upper and lower bounds of two standard deviations for the predictive mean. Note how the uncertainty is equal to zero in each of the training runs, as it would be expected, since the emulator reproduces the simulator's output at these points.

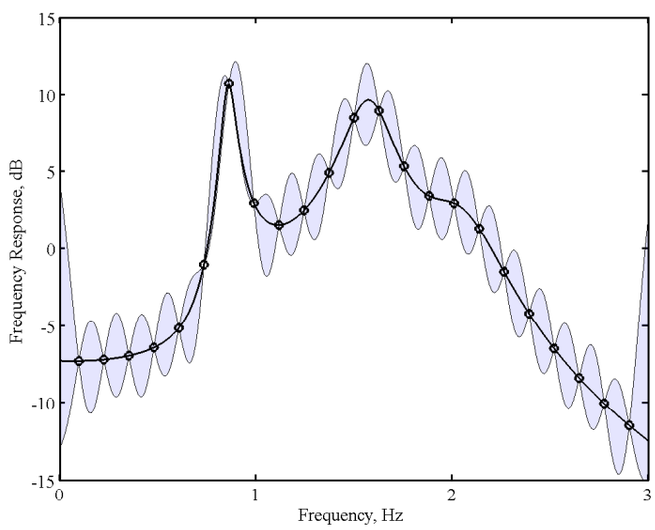

Figure 3: Probability bounds for the emulator's predictive mean. The forcing vector is $[0, 1, 0]^T$; (-): output of the simulator, (o): training runs.

3 PARAMETRIC UNCERTAINTY

A more realistic model of the spring-mass system shown in Figure 1 would incorporate uncertainty in the equation that governs the system's response. Suppose that any of the parameters m, c or k can be modeled as a random field. Then Eq. (17) becomes

$$\eta(\omega,\theta) = \left|[-\omega^2\mathbf{M}(\theta) + i\omega\mathbf{C}(\theta) + \mathbf{K}(\theta)]^{-1}\bar{\mathbf{f}}(\omega)\right| \tag{19}$$

where $\theta \in \Theta$ denotes an elementary random event. Note that for this simulator, the pseudo-random number seed can be made an input, as suggested by O'Hagan (2006). For notational purposes, the seed does not appear explicitly although it should be understood that it remains fixed.

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

$$\overline{\eta}(\omega_\ell, \underline{\theta}) = \frac{1}{S}\sum_{s=1}^{S}\eta(\omega_\ell, \theta_s) \tag{20}$$

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 to solve this problem is Monte Carlo simulation: For $\ell = 1, \ldots, M$, simulate $\{\eta(\omega_\ell, \theta_s)\}_{s=1}^S$ and evaluate Eq. (20) directly. The number of evaluations of $\eta(\cdot)$ 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. To circumvent this problem, two strategies using Gaussian process emulators are explored.

3.1 *Strategy 1*

The principle behind this approach is to 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. (20). Use the resulting training runs $\{\omega_i^*, \overline{\eta}(\omega_i^*, \theta)\}_{i=1}^m$ to emulate the M means corresponding to $\{\omega_\ell\}_{\ell=1}^M$. The number of evaluations of $\eta(\cdot)$ would be $Sm \simeq 10^4 m$. Theoretically, the computational burden should be reduced as $10^4 m \ll 10^4 M$.

3.2 *Strategy 2*

This approach operates by selecting 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

$$\overline{\eta}(\omega_\ell, \underline{\widetilde{\theta}}) = \frac{1}{N_E}\sum_{p=1}^{N_E}\eta(\omega_\ell, \widetilde{\theta}_p) \tag{21}$$

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 $L \cdot M \ll 10^4 M$.

4 NUMERICAL INVESTIGATION

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

$$L \cdot M < S \cdot m \tag{22}$$

Suppose $S = 10^4$ and $m = M/10$, which in the context of strategy 1 might be reasonable, since m is related to a moderate number of design points. Then, inequality (22) is true if and only if $L < 10^3$, which is also plausible, since by assumption $L \ll 10^4$.

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

$$k = k_0(1 + \epsilon\theta) \tag{23}$$

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

The result of implementing strategy 1 is shown in Figure 4. 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 (McKay et al., 1979). 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. For strategy 2, the mean for every $\omega_\ell \in \{\omega_\ell\}_{\ell=1}^{180}$ was emulated as follows. An initial design $\theta_1^*, \ldots, \theta_L^*$, with $L = 10$ was selected. Latin hypercube sampling was used once more to evenly spread the design points in the corresponding input domain. In this case 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. The result of doing so is shown in Figure 5, where the diamonds are the emulated means. The stars are the same Monte Carlo benchmark means previously shown, and the shaded area is the superposition of the 10^4 simulations of $\eta(\omega, \theta)$.

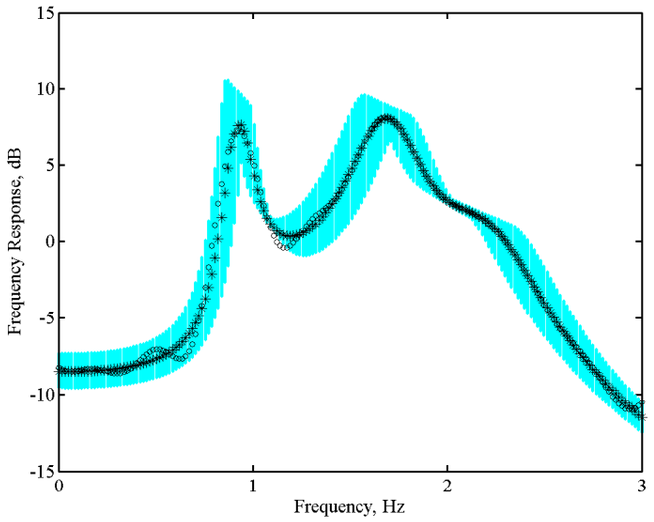

Figure 4: Emulation of the mean FRF for 10^4 simulations, taking design points in the ω-dimension of $\eta(\cdot)$; ($\star$): Monte Carlo benchmark means, (o):emulated means.

An improvement in the accuracy of the approximation was observed. More importantly, this strategy took 1.98 seconds, outperforming both Monte Carlo simulation and strategy 1.

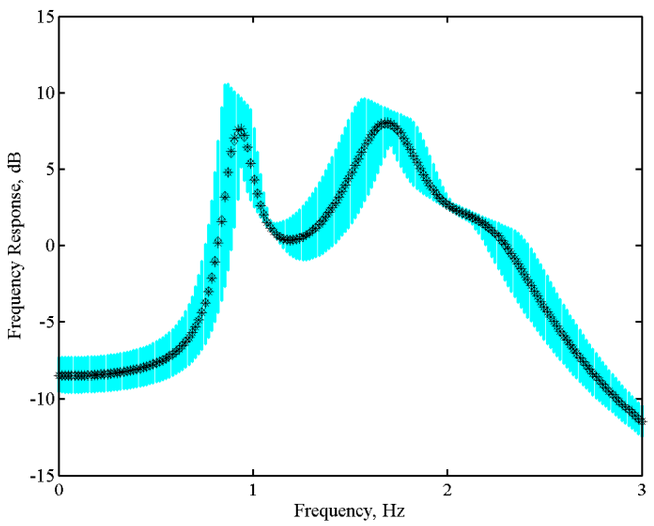

Figure 5: Emulation of the mean FRF for 10^4 simulations, taking design points in the θ-dimension of $\eta(\cdot)$; ($\star$): Monte Carlo benchmark means, ($\diamond$):emulated means.

5 CONCLUSIONS

Two strategies for the selection of design points in different input subdomains of a simulator were compared, to investigate whether the accuracy and computation time of a Gaussian process emulator can be improved. The two strategies were tested emulating the mean FRF of a simple dynamical system, with encouraging results.

There are some questions that need to be addressed in order to test the proposed approach. First, the simulator employed was assumed to be computationally expensive. In practice, a truly expensive simulator might not allow Monte Carlo simulation to be used as a benchmark. Second, the dimension of both input subdomains was equal to 1. It is necessary to apply both strategies to multidimensional simulators, both in the spatial dimension and with several random parameters. Finally, the assumption of the distribution of θ should be modified. A rigorous study on how is the initial design affected by the distribution of the random parameters is necessary.

ACKNOWLEDGEMENTS

FADO gratefully acknowledges the support of the Engineering and Physical Sciences Research Council (EPSRC) for the award of a studentship through an Ideas Factory grant and the Consejo Nacional de Ciencia y Tecnología (CONACYT) for the award of a scholarship from the Mexican Government. SA gratefully acknowledges the support of EPSRC through the award of an Advanced Research Fellowship and The Leverhulme Trust for the award of the Philip Leverhulme Prize.

REFERENCES

Challenor, P., Hankin, R., and Marsh, R. 2006. *Avoiding Dangerous Climate Change*, chapter Achieving robust design from computer simulations. Cambridge University Press, Cambridge, UK.

Chatfield, C. 1995. Model uncertainty, data mining and statistical inference. *Journal of the Royal Statistical Society, Series B*, 158:419–466.

Fadel, G., Riley, M., and Barthelemy, J. 1990. Two-point exponential approximation method for structural optimization. *Structural Optimization*, 2:117–124.

Faravelli, L. 1989. Response-surface approach for reliability analysis. *Journal of Engineering Mechanics*, 115:2763–2781.

Haylock, R. and O'Hagan, A. 1996. *Bayesian Statistics 5*, chapter On inference for outputs of computationally expensive algorithms with uncertainty on the inputs. Oxford University Press, Oxford, UK.

Keane, A. and Nair, P. 2005. *Computational Approaches for Aerospace Design*. John Wiley & Sons, Chichester, UK.

Kolachalama, V., Bressloff, N., and Nair, P. 2007. Mining data from hemodynamic simulations via bayesian emulation. *BioMedical Engineering OnLine*, 6(47).

Krzanowski, W. 2000. *Principles of Multivariate Analysis*. Oxford University Press, Oxford, UK.

McKay, M., Conover, W., and Beckman, R. 1979. A comparison of three methods for selecting values

of input variables in the analysis of output from a computer code. *Technometrics*, 21:239–245.

O'Hagan, A. 2006. Bayesian analysis of computer code outputs: A tutorial. *Reliability Engineering & System Safety*, 91(10-11):1290–1300.

Rougier, J. 2007. Probabilistic inference for future climate using an ensemble of climate model evaluations. *Climatic Change*, 81:247–264.

Sacks, J., Welch, W., Mitchell, T., and Wynn, H. 1989. Design and analysis of computer experiments. *Statistical Science*, 4:409–435.

Satner, T., Williams, B., and Notz, W. 2003. *The Design and Analysis of Computer Experiments*. Springer Series in Statistics, London, UK.

Vapnik, V. 1998. *Statistical Learning Theory*. John Wiley & Sons.