

# Robust optimisation of computationally expensive models using adaptive multi-fidelity emulation

M. Ellison<sup>a,\*</sup>, F.A. Diaz De la O<sup>b</sup>, N.Z. Ince<sup>c</sup>, M. Willetts<sup>c</sup>

<sup>a</sup>*Institute for Risk and Uncertainty, School of Engineering, University of Liverpool, Liverpool L69 7ZF, United Kingdom*

<sup>b</sup>*Department of Mathematics, University College London, London WC1H 0AY, United Kingdom*

<sup>c</sup>*General Electric Power, Newbold Road, Rugby CV21 2NH, United Kingdom*

---

## Abstract

Computationally expensive models are increasingly employed in the design process of engineering products and systems. Robust design in particular aims to obtain designs that exhibit near-optimal performance and low variability under uncertainty. Surrogate models are often employed to imitate the behaviour of expensive computational models. Surrogates are trained from a reduced number of samples of the expensive model. A crucial component of the performance of a surrogate is the quality of the training set. Problems occur when sampling fails to obtain points located in an area of interest and/or where the computational budget only allows for a very limited number of runs of the expensive model. This paper employs a Gaussian process emulation approach to perform efficient single-loop robust optimisation of expensive models. The emulator is enhanced to propagate input uncertainty to the emulator output, allowing single-loop robust optimisation. Further, the emulator is trained with multi-fidelity data obtained via adaptive sampling to maximise the quality of the training set for the given computational budget. An illustrative example is presented to highlight how

---

\*Corresponding author.  
Email address: [sgmellis@liv.ac.uk](mailto:sgmellis@liv.ac.uk) (M. Ellison).

the method works, before it is applied to two industrial case studies.

*Keywords:* Robust optimization, Gaussian process emulation, Subset simulation, Multi-fidelity.

---

## 1. Introduction

The goal of engineering design is to create technological systems that satisfy specific performance objectives and constraints over a period of time. Usually, there exist many feasible designs that satisfy the required objectives. For this reason, it is desirable to choose an optimal design according to some criterion. Modern engineering systems are inherently complex. This complexity means that endogenous (geometry, material properties) and exogenous (loads) information is never complete, and often varies throughout the life cycle of the system (e.g. degradation altering geometry, etc). The objective of robust design is to determine a set of designs that exhibit high levels of performance with low variability, whilst taking uncertainties into account. The benefits of robust design include the assurance of high performance regardless of a variety of unknown factors and occurrences throughout the system's life cycle. Robust design is essentially a traditional optimisation task, but with an added constraint relating to the performance variability, or robustness, within some predefined neighbourhood of the input variables. There are various definitions of robustness. A detailed review of which is presented in [1], leading to various methodologies for tackling the robust optimisation problem. The authors of [2] employed a reliability-based optimisation algorithm which utilised Monte Carlo integration to obtain an averaged performance value within the neighbourhood. Similarly, [3] employed a probability distribution estimation method to obtain an approximate distribution of the performance within the neighbourhood. Another approach utilised the Taylor expansion of the expectation and variance of the performance and attempted to minimise both criteria simultaneously. Alternatively, several papers chose to optimise the worst-case scenario rather than any sort of averaged performance [4, 5].

Typically, the behaviour of modern engineering systems is modelled by computationally expensive simulators, which can be seen as mappings from the *d-dimensional* input space to the output space, denoted  $f : \mathbf{x} \in \mathbb{R}^d \rightarrow y \in \mathbb{R}$ .

However, working directly with  $f(\mathbf{x})$  is often infeasible due to computational expense. A popular approach to tackle this problem is to replace  $f(\mathbf{x})$  with a surrogate model, which has been trained using data obtained from a low amount of simulator evaluations. One option is to train a Gaussian Process Emulator (GPE) [6, 7], which is defined by a mean function and a covariance function respectively. The mean function provides an inexpensive approximation to the simulator,  $\eta(\mathbf{x}) \approx f(\mathbf{x})$ , whilst the covariance function provides a measure of output uncertainty at each set of inputs,  $V_x[\eta(\mathbf{x})]$  [8]. The result is that the robust design problem can be interpreted mathematically as

$$\begin{aligned} & \underset{x}{\text{Minimise}} && \left\{ \eta(\mathbf{x}_\epsilon), V_x[\eta(\mathbf{x}_\epsilon)] \right\}, \\ & \text{subject to} && h_j(\mathbf{x}_\epsilon) \geq 0, j = 1, 2, \dots, m, \\ & && w_\nu(\mathbf{x}_\epsilon) = 0, \nu = 1, 2, \dots, p. \end{aligned} \tag{1}$$

Here  $\mathbf{x}_\epsilon$  represents the set of input variables located within the hypercube, or neighbourhood, centered at  $\mathbf{x}$  and bounded by  $\mathbf{x} \pm \epsilon$ . **Consequently,  $\eta(\mathbf{x}_\epsilon)$  represents the emulator output for this neighbourhood, and  $V_x[\eta(\mathbf{x}_\epsilon)]$  a measure of uncertainty associated with the emulator output, whilst taking into account the added input uncertainty. Similarly,  $h_j(\mathbf{x}_\epsilon)$  and  $w_\nu(\mathbf{x}_\epsilon)$  are the respective inequality and equality constraints of this neighbourhood.** In this context, robust design is interpreted as a double-loop optimisation task, with the outer-loop optimising the overall performance, subject to the constraint functions, and the inner-loop optimising for robustness in neighbourhood of the input variables (see algorithm 1).

The ability of the GPE to accurately approximate  $f(\mathbf{x})$  is directly related to the quality of the training set. **This issue can be addressed through measures such as employing adaptive sampling schemes and supplementing the training set with data from multiple levels of fidelity.** Adaptive sampling approaches tend to involve a utility function which measures some form of model improvement to select additional sample points. The most popular choice is expected improvement [9], which has been widely used in reliability [10], optimisation [11] and robust optimisation problems [12], amongst others. Further, the concept can be extended to multiple performance functions by considering the expected improvement of the current Pareto front via hy-

---

**Algorithm 1** Double-loop Optimisation

---

```
1: Outer Loop
2:   Minimise  $\eta(\mathbf{x}_\epsilon), V_x[\eta(\mathbf{x}_\epsilon)]$ 
3:   Inner Loop
4:   Generate  $\mathbf{x}_i \in \mathbf{x}_\epsilon, i = 1, \dots, N$ 
5:    $\eta(\mathbf{x}_\epsilon) = \frac{1}{N} \sum_{i=1}^N \eta(\mathbf{x}_i)$ 
6:    $V_x[\eta(\mathbf{x}_\epsilon)] = \frac{1}{N} \sum_{i=1}^N V_x[\eta(\mathbf{x}_i)]$ 
7:    $h_j(\mathbf{x}_\epsilon) =$  Total Individual Violations of  $h_j(\mathbf{x}_i)$ 
8:    $w_\nu(\mathbf{x}_\epsilon) =$  Total Individual Violations of  $w_\nu(\mathbf{x}_i)$ 
9:   End
10: End
```

---

pervolume expected improvement [13]. Other schemes include maximising the probability of improvement [14] or selecting samples with high uncertainty [15]. Multi-fidelity (MF) approaches are applicable when more than one potential simulator exists for the system under study. Lower-fidelity (LF) samples are defined by a lower-computational cost, but lower accuracy, than higher-fidelity (HF) samples. Multi-fidelity surrogate approaches exploit LF samples to gain information of the behaviour of the underlying system, and HF samples to maintain the desired accuracy. Most multi-fidelity approaches utilise LF data and adaptive sampling to attempt to sample the HF points in regions of interest and maximise the effectiveness of the surrogate [16, 17, 18]. Employing a surrogate model reduces the computational cost involved in robust design problems considerably. However, when there are a large number of performance functions and/or input variables, the double-loop approach becomes increasingly inefficient. **This is due to the fact that double-loop approaches utilise an averaged objective value, which is found by averaging the objective value of a set of  $N$  points within the neighbourhood of the original input configuration  $\mathbf{x}$ . The exact number of neighbourhood points is usually selected in relation to the number of input dimensions  $d$ , with  $N = 10d$  used in [19]. Consequently, the computational expense associated with double-loop approaches is often proportional to the number of input variables within a given problem.** A solution is to collapse the problem into a single-loop approach as done for a single-fidelity surrogate in [20]. In that paper, a GPE was enhanced to provide exact values of output uncertainty in the presence of uncertain inputs.

This paper provides a framework to perform efficient robust design on computationally expensive models. The framework adapts the single-loop approach discussed above to factor in multiple levels of fidelity, and supplements it with a hybrid adaptive sampling scheme. The paper is organised as follows. Section 2 provides an overview of various forms of Gaussian process emulation. The proposed approach is introduced in Section 3 and discusses the main components. An illustrative example and two industrial CFD case studies are presented in Section 4. The final section provides relevant conclusions and highlights future work.

## 2. Methodology Overview

This section provides an overview of various forms of Gaussian process emulation. The main steps involved in the training of a single-fidelity (SF) GPE are described. Two extensions of the GPE framework are then discussed: training a GPE with MF data and factoring in input uncertainty for a SF GPE.

### 2.1. Single-Fidelity Gaussian Process Emulation

Computationally expensive models are deterministic mappings from some input  $\mathbf{x} \in \chi$  to an output  $y = f(\mathbf{x})$ . Due to the computational expense, often only a limited number of input samples can be evaluated. Under the Bayesian paradigm,  $f(\mathbf{x})$  can be regarded as a random variable as the output is unknown until it is computed (and thus observed) by the modeller. Gaussian process emulation follows a Bayesian framework to provide a statistical approximation of  $\eta(\mathbf{x}) \approx f(\mathbf{x})$ . Initially, a Gaussian Process prior is placed on the output, in the form [6]

$$\eta(\mathbf{x}) = \mathbf{h}(\mathbf{x})^T \beta + \mathbf{Z}(\mathbf{x}). \quad (2)$$

The first term is the mean of the emulator and provides the general trend;  $\mathbf{h}(\mathbf{x})$  is a vector of  $q$  known regression functions,  $\beta$  is a vector of  $q$  unknown coefficients. The second term controls local behaviour;  $\mathbf{Z}(\mathbf{x})$  represents a Gaussian Process, with mean zero and covariance  $\sigma^2 c(\mathbf{x}, \mathbf{x}'; \theta)$ . Here  $\sigma^2$  is

a scalar parameter, and  $\theta$  is a vector which specifies the smoothness of the inputs and ultimately dictates the behaviour of the correlation function

$$c(\mathbf{x}, \mathbf{x}'; \theta) = \exp(-(\mathbf{x} - \mathbf{x}')^T \theta (\mathbf{x} - \mathbf{x}')). \quad (3)$$

Given a set of single-fidelity (i.e. data from only one model/simulator) training data,  $D = [(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_n, y_n)]$ , the emulator distribution, conditional on training data and the unknown hyperparameters  $\beta, \sigma, \theta$ , is defined as

$$\eta(\mathbf{x}) | \mathbf{D}, \beta, \sigma^2, \theta \sim \mathcal{GP}(\mathbf{M}(\cdot), \mathbf{C}(\cdot, \cdot)), \quad (4)$$

where  $M(\cdot)$  and  $C(\cdot, \cdot)$  are the mean and covariance functions of the GP, respectively. Both  $\beta$  and  $\sigma^2$  are assigned a prior and estimated via their respective maximum a posteriori estimates,  $\hat{\beta}$  and  $\hat{\sigma}^2$ , while  $\hat{\theta}$  is often estimated via maximising a likelihood function [21]. Ultimately, the posterior distribution of at some unobserved input  $x^*$ , conditional on the observed data, is given by [7]

$$\eta(\mathbf{x}^*) | \mathbf{D} \sim \mathcal{GP}(\mathbf{M}^*(\mathbf{x}^*), \mathbf{C}^*(\mathbf{x}^*, \mathbf{x}')), \quad (5)$$

with posterior predictive mean function

$$M^*(\mathbf{x}^*) = \mathbf{h}(\mathbf{x}^*)^T \beta + \mathbf{t}(\mathbf{x}^*)^T \Sigma^{-1} (\mathbf{y} - \mathbf{H}\beta), \quad (6)$$

and posterior predictive covariance function

$$\begin{aligned} C^*(\mathbf{x}^*, \mathbf{x}'^*) &= \hat{\sigma}^2 [c(\mathbf{x}^*, \mathbf{x}'^*) + \mathbf{t}(\mathbf{x}^*)^T \Sigma^{-1} \mathbf{t}(\mathbf{x}'^*) \\ &\quad + (\mathbf{h}(\mathbf{x}^*)^T - \mathbf{t}(\mathbf{x}^*)^T \Sigma^{-1} \mathbf{H})(\mathbf{H}^T \Sigma^{-1} \mathbf{H})^{-1} \\ &\quad \times (\mathbf{h}(\mathbf{x}'^*)^T - \mathbf{t}(\mathbf{x}'^*)^T \Sigma^{-1} \mathbf{H})^T]. \end{aligned} \quad (7)$$

Here  $\Sigma$  is a matrix containing the correlation between each training point,  $t(x^*)$  is a vector containing correlation between  $x^*$  and the training points,

and  $H$  is a vector containing the training points evaluated at the regression function  $\mathbf{h}(\mathbf{x})$ . For the work in this paper, the Gaussian process prior is assumed to have mean zero, i.e.  $\mathbf{h}(\mathbf{x})^T = 0$ , which will be adopted from here onward.

## 2.2. Multi-Fidelity Gaussian Process Emulation

Computationally expensive models are designed to capture the behaviour of an underlying physical system or product. It is often the case that more than one computational model is available, with each model corresponding to a varying degree of computational cost and accuracy. The models can usually be organised in levels of fidelity; a model with lower computational costs but less accuracy is considered to be of a lower fidelity than a more expensive and accurate model. For example, a 2-dimensional versus a 3-dimensional model, or a model with a coarse mesh versus one with a fine mesh. The concept of Gaussian Process Emulation can be extended to incorporate multiple levels of fidelity within the training process [22], allowing for improved emulator performance or lower training costs. In the case where there are two levels of fidelity, denoted low-fidelity (LF) and high-fidelity (HF), this paper adopts the recursive multi-fidelity approach from [23], to approximate the output from the HF model as:

$$\eta_{HF}(\mathbf{x}) = \rho_{LF}(\mathbf{x})\eta_{LF}(\mathbf{x}) + \delta_{HF}(\mathbf{x}). \quad (8)$$

Here  $\eta_{LF}$  represents a GPE trained using data from the LF model,  $\rho_{LF}(\mathbf{x})$  represents a regression function, and  $\delta_{HF}$  represents a Gaussian Process Emulator which models the discrepancy between the HF estimation of  $\rho_{LF}(\mathbf{x})\eta_{LF}(\mathbf{x})$  and the true HF simulator realisations. Both emulators are trained via the steps described in section (2.1). This can be generalised for  $t$  levels of fidelity in a recursive fashion:

$$\eta_t(\mathbf{x}) = \rho_{t-1}(\mathbf{x})\eta_{t-1}(\mathbf{x}) + \delta_t(\mathbf{x}). \quad (9)$$

### 2.3. Single-Fidelity Robust Gaussian Process Emulation

The two-looped approach to solving the robust optimisation problem (1) works by first attempting to minimise the objective functions  $\eta(\mathbf{x})$  and  $V_x[\eta(\mathbf{x})]$  in the outer-loop. Once a potential solution is found, the inner-loop measures the robustness over the input distribution. As a result, the predictive distribution of the emulator given input uncertainty is found by marginalising over the input distribution:

$$p(\eta(\mathbf{x}_*)|\mathbf{u}, \mathbf{S}, D) = \int p(\eta(\mathbf{x}_*)|D)p(\mathbf{x}_*|\mathbf{u}, \mathbf{S})d\mathbf{x}_*. \quad (10)$$

This marginalisation is basically the aforementioned inner-loop and often achieved via Monte Carlo sampling. In the case where the uncertainty within the inputs is normally distributed, i.e. for an unknown point  $\mathbf{x}_* \sim \mathcal{N}(\mathbf{u}, \mathbf{S})$ , it is possible to extract the first and second moments of (10) via methods described in [24, 25]. These moments provide analytical expressions for the mean,  $m(\mathbf{u}, \mathbf{S})$ , and variance,  $v(\mathbf{u}, \mathbf{S})$ , of  $p(\eta(\mathbf{x}_*)|\mathbf{u}, \mathbf{S}, D)$ . Ultimately, having direct access to the mean and variance of the emulator conditional on the input uncertainty collapses the robust optimisation problem down to a single-loop:

$$\begin{aligned} & \underset{u}{\text{Minimise}} && \left\{ m(\mathbf{u}, \mathbf{S}), v(\mathbf{u}, \mathbf{S}) \right\}, \\ & \text{subject to} && h_j(\mathbf{u}, \mathbf{S}) \geq 0, j = 1, 2, \dots, m, \\ & && w_\nu(\mathbf{u}, \mathbf{S}) = 0, \nu = 1, 2, \dots, p. \end{aligned} \quad (11)$$

The full details and steps involved are discussed further in [20]. The resulting mean and variance functions are fundamentally equations (6) and (7) ‘corrected’ to factor in the input uncertainty. The added input uncertainty essentially flattens the output, with a decreased vertical amplitude and increased correlation.

### 3. Proposed Approach

The goal of the proposed approach is to perform efficient robust optimisation of computationally expensive models. The method is a combination of the various forms of Gaussian process emulation discussed in the previous section, and is termed Multi-Fidelity Robust Gaussian Process Emulator (MF-RGPE). When employing a GPE for the purposes of robust optimisation, the two main considerations are the ability of the GPE to accurately portray the behaviour of the underlying expensive model, and the efficiency of the robust optimisation process. To address the former, the proposed approach utilises training data from multiple levels of fidelity obtained via an extension of the Expected Improvement (EI) criterion [9] to maximise the quality of the training set. To increase the efficiency, the proposed approach extends the robust GPE detailed in section 2.3 to the multi-fidelity case. Further details of the steps are discussed in the following subsections.

#### *3.1. Generating Training Samples*

The framework begins with the design of experiment (DoE) of the LF model. Latin hypercube sampling (LHS) [26] is used as the space-filling algorithm to generate the initial samples. These samples are then evaluated on both the LF model and any relevant constraints functions, and are referred to as LF samples. To generate the initial HF samples, the LF samples are first sorted according to their objective and constraint values. A proportion of the top performing samples are selected to be part of the initial HF samples. The remaining initial HF samples are selected by filling the remaining space using a space-filling algorithm. This is done to encourage sampling of high-interest areas, whilst not neglecting the general performance of the GPE elsewhere. The proportion used in this work was 20% of initial samples from the top performing LF samples and 80% resulting from the space-filling algorithm. The HF samples were then evaluated on both the HF model and any relevant constraint functions.

#### *3.2. Constructing the MF RGPE*

The MF RGPE provides an approximation of the HF output whilst considering input uncertainty, and is constructed in a similar fashion to the standard

MF GPE described in (8):

$$\eta_{RHF}(\mathbf{x}) = \rho_{RLF}(\mathbf{x})\eta_{RLF}(\mathbf{x}) + \delta_{HF}(\mathbf{x}). \quad (12)$$

Here,  $\eta_{RLF}(\mathbf{x})$  represents a robust GPE trained using data from the LF model via the steps described in section 2.3 and  $\rho_{RLF}(\mathbf{x})$  represents a regression function. The last term,  $\delta_{HF}(\mathbf{x})$ , represents a GPE which models the discrepancy between the estimation of the output of the HF training data, without accounting for input uncertainty, i.e.  $\rho_{LF}(\mathbf{x})\eta_{LF}(\mathbf{x})$ , and the actual HF simulator output. In an industrial context, there will usually be a predetermined computational budget and the stopping criterion will be met once this budget is exhausted. Other stopping criterion may include reaching a certain threshold of performance, such as obtaining a suitable design or reducing overall GPE uncertainty below some required value.

### 3.3. Adaptive Sampling

For a SF GPE, the expected improvement (EI) [9] at some point  $\mathbf{x}$  is defined as

$$\mathbb{E}[I(\mathbf{x})] = |y_{min} - \eta(\mathbf{x})| \Phi\left(\frac{y_{min} - \eta(\mathbf{x})}{s}\right) + s\phi\left(\frac{y_{min} - \eta(\mathbf{x})}{s}\right). \quad (13)$$

Here  $y_{min}$  represents the current best performing objective value amongst the training data,  $s$  denotes the standard deviation of the GPE and  $\Phi(\cdot)$  and  $\phi(\cdot)$  represent the cumulative and probability density functions of a standard Gaussian random variable, respectively. EI attempts to locate samples that offer improved nominal performance against the current best sample. The method balances higher probability of a relatively small improvement (exploitation) versus a lower probability of high improvement (exploration). The concept of EI can also be applied to cases with more than one objective function by considering a hypervolume of improvement. Following the steps described in [27], given some reference point  $\mathbf{r}$ , the HVEI at some point  $\mathbf{x}$  against a set of solutions  $\mathbf{U}$  is defined as

$$HVE\mathbb{E}[I(\mathbf{x})] = \sum_{i=1}^{N_U} \left[ \prod_{j=1}^{N_{Obj}} \mathbb{E}[I(\mathbf{x}, U_{i,j})] \right], \quad (14)$$

where  $N_U$  is the number of points in the set  $U$  and  $N_{Obj}$  is the number of objective functions. Both EI and HVEI are designed for optimising nominal performance. The proposed approach extends them for the purposes of robust design by employing the mean and standard deviation GPE output from the SF RGPE and MF RGPE within the EI process. Consequently, the robust EI and robust HVEI can therefore be defined as

$$\mathbb{E}[I_R(\mathbf{x})] = |y_{Rmin} - \eta_{RMF}(\mathbf{x})| \Phi\left(\frac{y_{Rmin} - \eta_{RMF}(\mathbf{x})}{s_{RMF}}\right) + s\phi\left(\frac{y_{Rmin} - \eta_{RMF}(\mathbf{x})}{s_{RMF}}\right), \quad (15)$$

$$HVE\mathbb{E}[I_R(\mathbf{x})] = \sum_{i=1}^{N_U} \left[ \prod_{j=1}^{N_{Obj}} \mathbb{E}[I_R(\mathbf{x}, U_{i,j})] \right],$$

where  $y_{Rmin}$  is the current best performing objective value amongst the training data whilst also taking input uncertainty into account, **and within any subscripts,  $R$  denotes robust and  $MF$  multi-fidelity**. Figure 1 illustrates the concept of robust HVEI in the case of two objective functions. The set of solutions,  $\mathbf{P}$ , represents the robust Pareto solutions taken from the current training data. Utilising these solutions and some reference point  $\mathbf{r}$ , a set of local upper bounds  $\mathbf{U}$  can be constructed such that  $U_i$  lies at the intercept of  $P_i$  and  $P_{i+1}$ . The robust HVEI is thus the summation of the robust EI against each local upper bound, to give an overall value of improvement. To obtain promising adaptive samples, an algorithm known as subset simulation is used to explore the input space and locate samples with high values of robust HVEI.

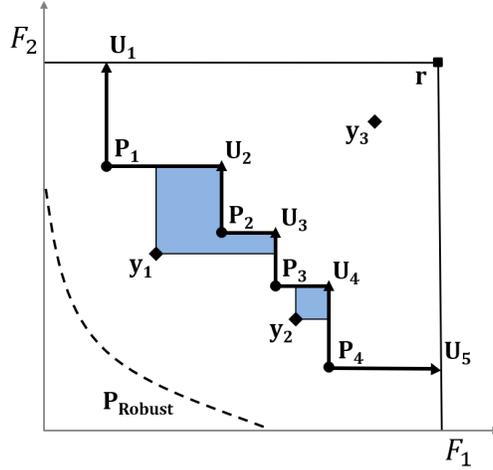


Figure 1: **Robust Hypervolume Expected Improvement:** The blue areas represent the hypervolume of improvement of a point  $y_i$  against the set  $U$ . The point  $y_1$ , offers the most improvement,  $y_2$  offers some improvement, while  $y_3$  offers no improvement at all.

Subset simulation is an efficient Monte Carlo technique that employs Markov Chain Monte Carlo (MCMC) and a simple evolutionary strategy to converge to and sample from extremely small, or rare, regions of the input domain [28]. Samples within these rare regions are associated with superior performance according to some user-defined criteria of interest. Originally developed and applied to reliability problems with great success [29, 30], the authors of [31] made the analogy between sampling from a small failure region and sampling from a small region exhibiting high performance. Consequently, the algorithm was adapted for the purposes of optimisation [31, 32, 33], allowing it to be used to locate promising samples even in the case where the areas offering improvement are extremely rare. Moreover, it is adept at dealing with high-dimensional problems, such that the curse of dimensionality has a limited impact. To increase efficiency, several samples are adaptively sampled in one optimisation iteration. An influence function [34], denoted  $\tau(\mathbf{x})$ , is employed to discourage the adaptive samples clustering in one area, by scaling the robust EI values after each new adaptive sample is taken, i.e.

$$\begin{aligned}\mathbb{E}[I_R(\mathbf{x})] &= \mathbb{E}[I_R(\mathbf{x})]\tau(\mathbf{x}), \\ \tau(\mathbf{x}) &= 1 - c(\mathbf{x}, \mathbf{x}_{AS}),\end{aligned}\tag{16}$$

where  $\mathbf{x}_{AS}$  represents the latest adaptive sample and  $c(\cdot, \cdot)$  is the correlation function from Eq. 3.

#### 3.4. Robust Design

Once the computational budget is exhausted and the final batch of adaptive sampling completed, the final MF RGPEs can be utilised for robust design. Subset simulation is employed to locate the input regions corresponding to samples with high performance according to the MF RGPEs. These samples should be insensitive to perturbation in the values of the input variables, and given the computational resources, be validated on the HF model. The steps involved in the proposed method are outlined in a flowchart provided in figure 2. Steps 1-4 involve generating the LF and initial HF training data, and are described in section 3.1. This provides the foundation for the construction of the initial MF RGPE in step 5, which is detailed in section 3.2. Provided the stopping criterion (step 6) has not been met, this MF RGPE is then used as a tool to attempt to locate samples with improved performance in step 7, using the adaptive sampling process from section 3.3. This procedure is repeated on a loop, with an improved MF RGPE constructed at each generation until the stopping criterion is met. Optimisation of the MF RGPE(s) takes place in step 8.

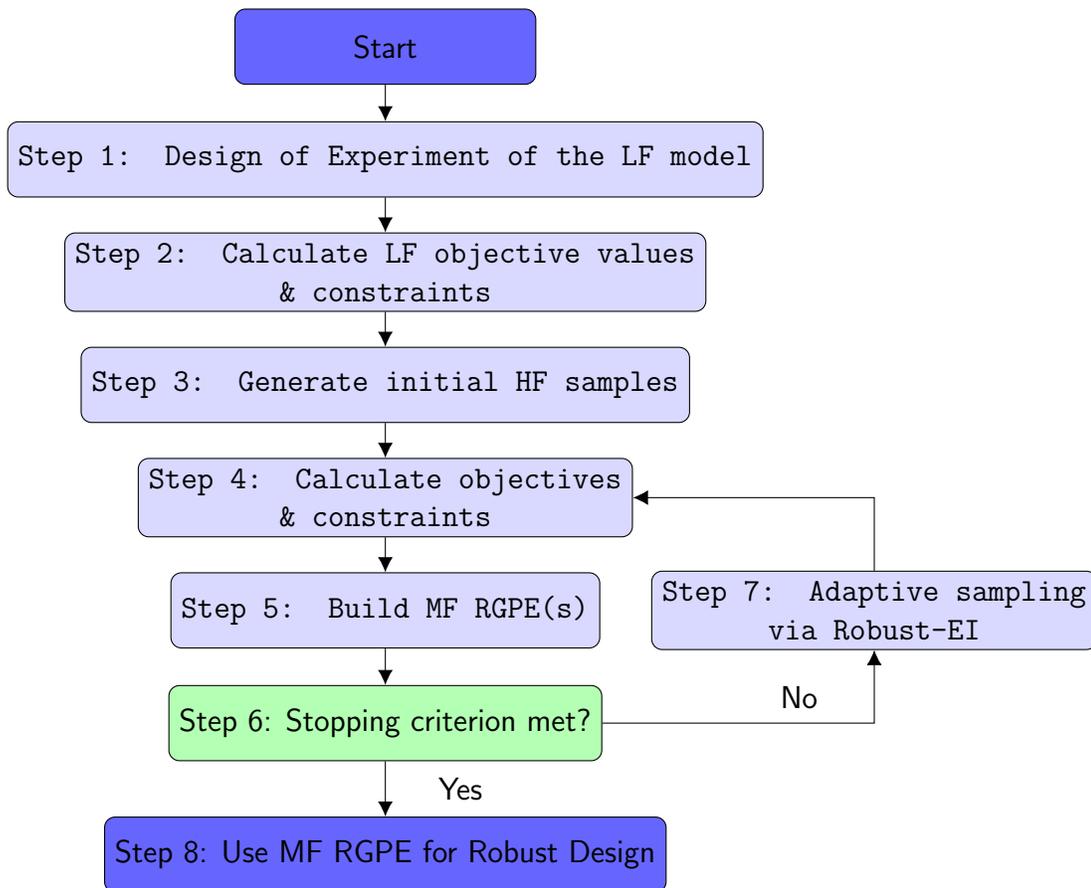


Figure 2: Framework for Robust Design via multi-fidelity robust Gaussian process emulation.

#### 4. Numerical Examples

This section provides three examples showcasing the MF RGPE approach discussed in the previous section. A synthetic example is first presented to showcase the concept of the approach before it is applied to two industrially-relevant test cases. In all examples, the regression function  $\rho_{RLF}$  from equation 12) is set to one, as in each example there is no assumed prior knowledge regarding the relationship between the LF simulator output and HF simulator output.

#### 4.1. Synthetic Example 1

The motivation behind this synthetic example was to illustrate the main concepts of the proposed approach. The HF and LF functions are defined as

$$\begin{aligned}
 f_{HF}(x_1, x_2) &= \sin(x_1^2)\sin(x_2^2) + \frac{2(x_1 + x_2)}{25}, \\
 f_{LF}(x_1, x_2) &= \left(1 + \frac{x_1}{5}\right)\sin\left(\frac{9x_1^2}{10}\right)\left(1 + \frac{x_2}{5}\right)\sin\left(\frac{9x_2^2}{10}\right).
 \end{aligned}
 \tag{17}$$

The functions were constructed such that the LF function exhibited similar behaviour to the HF function, and as such could be used to infer regions of high interest. Additionally, both were designed to possess two maxima; a global maximum that was more sensitive to input uncertainty, and a local more robust maximum. The goal is to maximise  $f_{HF}$  in the face of some input uncertainty, with the intention to favour the more robust local maximum. An initial batch of 50 LF samples were selected via LHS. The 4 samples with the highest objective values were then selected, alongside 16 further samples from LHS, to populate the HF training set. A MF RGPE was then constructed with training data normalised between 0 and 1, such that an untested input configuration  $\mathbf{x}_*$  is defined by the probability distribution  $\mathbf{x}_* \sim \mathcal{N}(\mathbf{u}, \text{diag}[0.01, 0.01])$ . Here,  $\mathbf{u}$  is the mean vector of  $\mathbf{x}_*$ , which corresponds to the input values if there was no uncertainty present, while  $\text{diag}[0.01, 0.01]$  is a diagonal matrix containing the variance with respect to each input variable. A further 3 samples were obtained via the robust EI adaptive sampling algorithm, and the retrained MF RGPE employed for robust optimisation. Finally, the inputs were transformed back to their original domains.

Figure 3 displays the contours of  $f_{LF}$  and  $f_{HF}$ , as well as all samples for the MF RGPE. On inspection, the function possesses a global optimum (maximum) located in the top right at approximately  $\mathbf{x} \approx (\sqrt{\frac{3\pi}{2}}, \sqrt{\frac{3\pi}{2}})$  and a local, more robust, optimum around  $\mathbf{x} \approx (\sqrt{\frac{\pi}{2}}, \sqrt{\frac{\pi}{2}})$  in the bottom left. The adaptive samples all lie within these regions of interest, with a preference to the local, more robust optimum. The local optimum in the bottom left is considered more robust as it has a wider base, meaning there is a

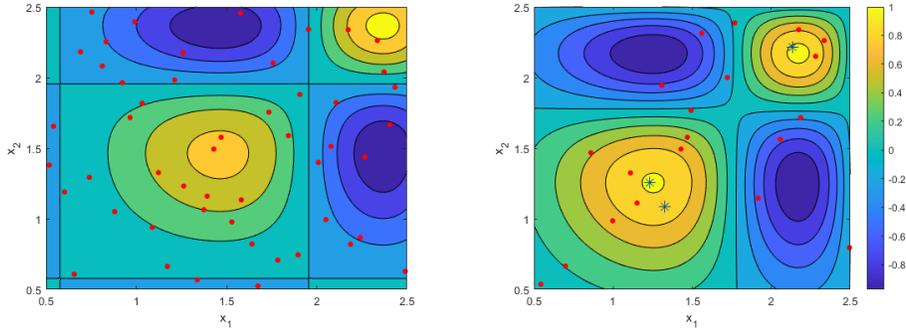


Figure 3: Left: Contour plot of LF function, LF samples denoted by the red dots. Right: Contour plot of HF function. The red dots are the initial 20 HF samples, whilst the blue stars represent those obtained via adaptive sampling.

lower drop in performance given any perturbation in the inputs. Note that several of the initial batch of HF samples were already in proximity to the two optima, highlighting the importance of utilising the best performing LF samples. Further, the LF data provided valuable information in the regions where HF samples were sparse (e.g. top left), saving an adaptive sample being wasted in an area of low interest. The illustrative example was repeated 10,000 times, and the normalised error from the true robust optimum presented in Figure 4. The error was normalised to illustrate the discrepancy between the true robust optimum and the actual values more clearly. The goal of the study was to showcase the individual steps described in Figure 2 and illustrate the merits of the approach. Overall, the majority of cases were within 1% of the true robust optimal input values.

#### 4.2. Industrial Examples

Design engineers often utilise computationally expensive models in their design process. It is often desirable to factor input uncertainty into this process. The proposed approach has been designed to assist design engineers in this task, within a reasonable computational budget. Computational Fluid Dynamics (CFD) [35] models are a common tool in engineering design. They are usually computationally expensive, which limits their ability to be used directly in practical applications, but makes them a prime candidate for the MF RGPE approach. **A key challenge in both industrial examples presented is the presence of significant computational constraints. This is useful to**

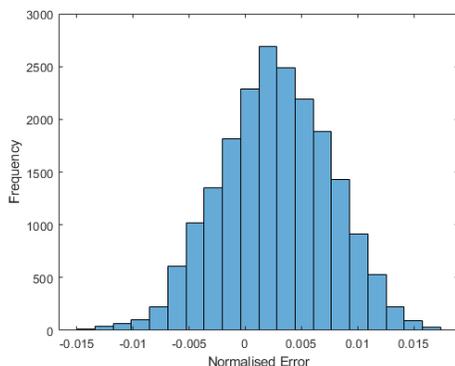


Figure 4: Histogram of normalised error: A comparison between the estimated robust optimal input values and the true robust optimal input values.

highlight the validity of the approach to industrial-scale problems, but does limit each of the examples to one iteration.

#### 4.2.1. Turbulated Duct Case Study

A frequent feature in turbine blades is the presence of turbulated internal cooling ducts. The presence of rib turbulators repeatedly perturb the boundary layer, which can result in significant heat transfer by promoting convective mixing with the core cooling flow. A downside is that this heat transfer comes at the cost of higher-pressure drop [36]. However, due to manufacturing constraints and degradation during the life cycle, the duct will likely diverge from the initial design configuration at some point. The shape and size of the rib turbulators is dictated by four geometric parameters that control its cross-sectional profile and angle to the duct. This motivates a robust optimisation task, whereby the goal is to maximize the heat transfer across the duct, represented by the Nusselt number, whilst minimising the drop in pressure. Moreover, this optimisation task should be done whilst considering uncertainty in the input configuration, to factor in the aforementioned manufacturing constraints and degradation. To address this challenge, a model of the turbulated duct was constructed using ANSYS software according to four geometric parameters, as shown in figure 5. The range of parameter values are shown in Table 1 in the appendix. Within the ANSYS software, each combination of these four parameters would result in a unique turbu-

lated duct geometry. This geometry was then meshed using an unstructured tetrahedral grid and solved using Reynolds-averaged Navier–Stokes (RANS) [35] to output the Nusselt number and pressure coefficient for that particular design.

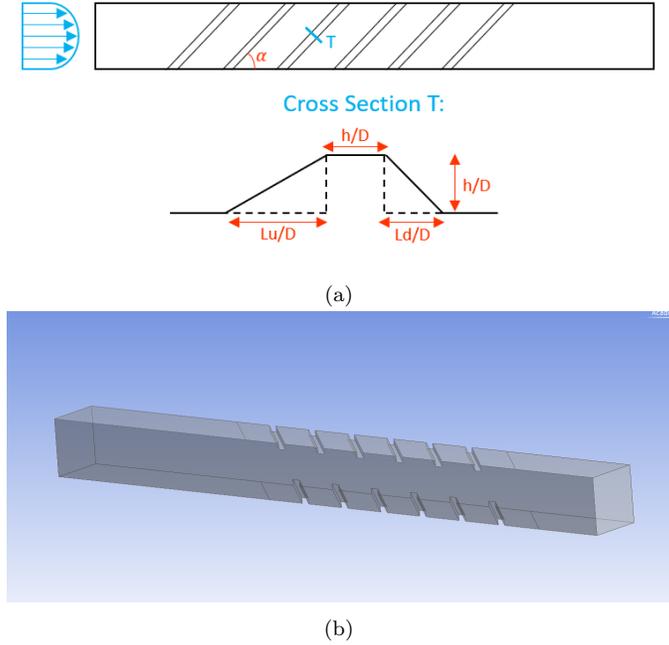


Figure 5: **Turbulated Duct Geometry:** (a) Illustrates the influence the four geometric parameters have on the design. Additionally, the blue arrows to the left represent a mass flux profile mapped to the inlet to represent the operating conditions. (b) provides an example geometry taken from ANSYS.

For the MF RGPE approach, the overall computational budget assigned was equivalent to 44 HF samples. The LF model consisted of a mesh of approximately one million elements and solved using  $k - \omega$  SST RANS on ANSYS, whilst the HF model consisted of a mesh of approximately five million elements and solved using  $k - \omega$  SST RANS. The approximate computational cost was 20 LF samples  $\approx$  1 HF sample. Consequently, two separate MF RGPEs were trained for the Nusselt number and pressure coefficient respectively, with the initial MF RGPEs trained using 80 LF samples and 20 HF samples. A further 20 HF samples were adaptively selected in two batches of 10 samples to supplement the training set. The training data was nor-

malised between 0 and 1, with input uncertainty represented via a diagonal matrix  $S = \text{diag}[0.025, 0.025, 0.025, 0.025]$ ). The final MF RGPEs were then optimised using a multi-objective subset simulation algorithm. For comparison, the case study was repeated with the same computational budget, but using only HF samples to construct two SF RGPEs. The initial SF RGPEs were trained using 40 HF samples. A further 4 HF samples were adaptively selected in four batches of a single sample to supplement the training set.

Figure 6 demonstrates the adaptive sampling process and the final Pareto front for the MF RGPE approach. On inspection, several of the initial HF samples (green dots, top left plot) were located in close proximity to the eventual Pareto front, again highlighting the advantages of incorporating the LF training data to locate regions of interest. Indeed, the general performance of the adaptive points (blue stars) is significantly better than that of the randomly sampled points, showcasing the benefits of adaptive sampling. Furthermore, by comparing the two batches of adaptive sampling, it is clear how the adaptive sampling process attempts to converge towards the true Pareto front. It should be noted that the adaptive sampling procedure was assisted by the LF data to discard areas of low interest. The optimisation process placed constraints on the output variance of the respective GPEs to ensure a certain level of performance. This is highlighted in the close proximity between the Pareto front and the best performing training samples, placing further emphasis on the importance of quality training data. A single training point lies above the Pareto front, however this point was deemed to lack the necessary robustness according to the final MF RGPEs.

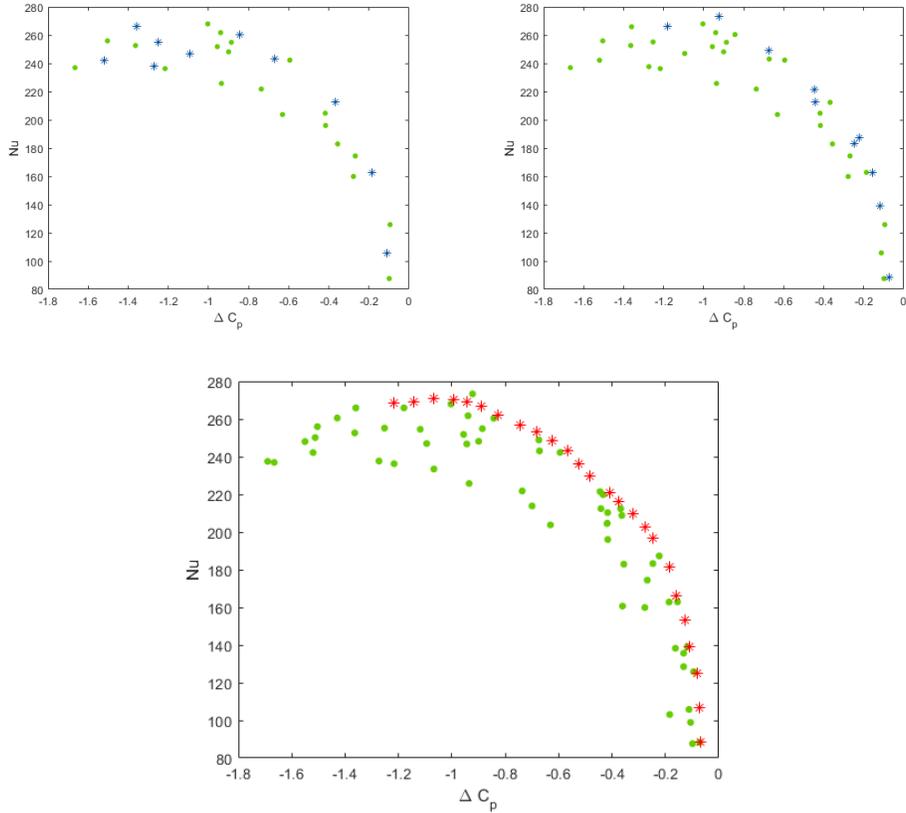


Figure 6: **HF Training Points and Pareto Front:** The top two figures showcase the adaptive sampling process, with the first batch top left and the second batch top right. In each, the green dots are the HF points that have previously been obtained, and the blue stars represent the newly evaluated adaptive samples. The plot on the bottom contains the full set of HF samples (green dots) and the Pareto front (red stars) obtained via optimising the MF RGPEs for each objective.

Figure 7 contains the Pareto fronts from the MF RGPE approach (red stars) and the SF RGPE approach (blue dots). In general, the two Pareto fronts possess similar behaviour, although the MF RGPE Pareto solutions exhibit superior performance than the SF RGPE Pareto solutions. A potential contributor to this discrepancy is the fact that the SF approach was made up of a higher proportion of randomly sampled training data. However, it is standard practice to use a budget of at least ten training samples per input dimension [37] in order to have sufficient confidence in the output of the underlying surrogate. The MF RGPE approach circumvents this issue by

utilising LF data to make up for any loss of information. As such, it is reasonable that increasing the proportion of adaptively sampled data in the SF RGPE case would not necessarily improve the performance, due to surrogate inaccuracy and added uncertainty disrupting the sampling process. A second contributor is the fact that the MF RGPE approach was able to infer regions of high interest from the LF training data to aid in the adaptive sampling procedure. Overall, the MF RGPE offered superior performance than the SF alternative for the same computational budget.

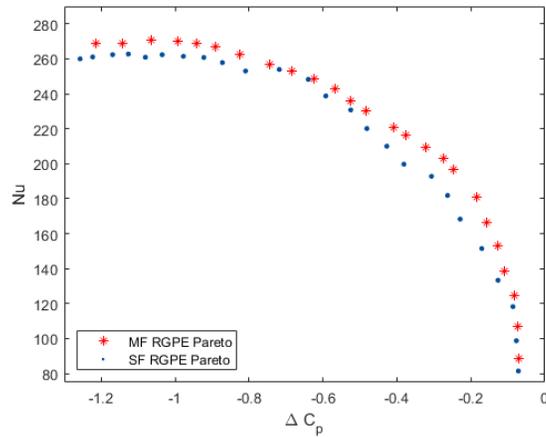


Figure 7: **Pareto Front Comparison:** The red stars represent the Pareto solutions obtained using the MF RGPE approach. The blue dots are the Pareto solutions obtained using the SF RGPE approach.

Figure 8 displays 20 validation samples for 4 designs taken from the MF RGPE Pareto front. A validation sample was generated from the distribution  $\mathcal{N}(\mathbf{x}_P, \mathbf{S})$ , where  $\mathbf{x}_P$  denotes the original Pareto solution. The validation sample was then evaluated on the HF model to measure its performance. Given the computational cost involved, 4 Pareto solutions were chosen to ensure a reasonable amount of validation samples per Pareto solution could be evaluated, whilst making certain of validation across various areas of the Pareto front. It should be noted that the second and third Pareto solutions were located further from HF training samples than the other two solutions, hence their larger uncertainty bounds. Nevertheless, each validation sample was still within the  $2\sigma$  uncertainty bounds of the original Pareto solution.

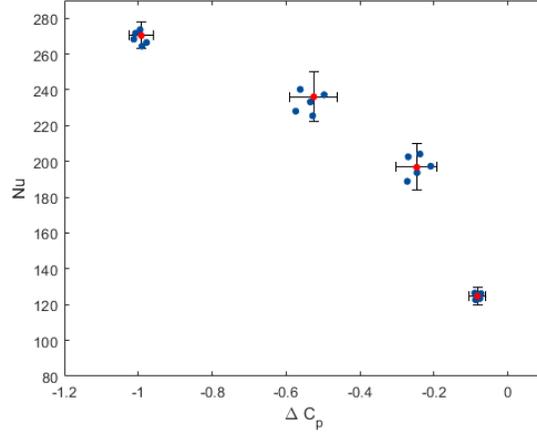


Figure 8: **Validation Points:** The red dots represent the original Pareto solutions and the black lines their respective uncertainty bounds. The blue dots are the simulator realisations at some perturbed design of the original Pareto solution.

#### 4.2.2. Aerofoil Case Study

The aerofoil test case involved obtaining a set of aerofoil solutions that maximise lift-to-drag (L/D) ratio whilst minimising maximum blade thickness of a turbine blade. **As in the turbulated duct example, turbine blades are subject to manufacturing constraints and degradation, motivating the need to factor in potential perturbation of input values caused by this uncertainty.** A prospective aerofoil geometry was defined using the Class-Shape Transformation (CST) method [38]. In particular the Au and Al parameters are the weighting coefficients that help prescribe the thickness/shape at various locations along the upper and lower surfaces respectively. The parameters and their respective ranges are shown in Table 2 in the appendix. The LF model consisted of the aerofoil being solved over a range of angles of attack in XFOIL software, which performed a potential flow calculation without taking into account viscosity or a boundary layer. The HF model consisted of the aerofoil being solved via  $k-\omega$  RANS in ANSYS. Unlike the turbulated duct case study, where the level of fidelity was solely due to mesh resolution, the fidelity in this case is dictated by two separate methods of varying accuracy and cost. It should be noted that the the definition of varying levels of fidelity is problem specific, with the only requirement that they exhibit similar behaviour in attempting to model the same underlying phenomena. The

computational budget for the test case was approximately 240 HF samples. The comparative computational costs were approximately 20 LF samples per HF sample. Two separate MF RGPEs were trained for the lift-to-drag ratio and maximum thickness respectively, with the initial MF RGPEs trained using 600 LF samples and 120 HF samples. A further 80 HF samples were adaptively sampled in four batches of 20 to supplement the training set. The training data was normalised between 0 and 1, with input uncertainty represented via a  $20 \times 20$  diagonal matrix  $S$  with each of the entries equal to 0.025. The final MF RGPEs were then optimised using a multi-objective subset simulation algorithm. As in the previous example, the case study was repeated using the same computational budget comprising of only HF samples. The initial SF RGPEs were trained using 200 HF samples, with a further four batches of a 10 samples added via adaptive sampling.

Figure 9 presents the adaptive sampling process and the final MF RGPE Pareto front. As in the turbulated duct study, several of the initial HF samples (green dots, top left plot) exhibited high performance and there was a clear convergence towards the suspected true Pareto front as the number of adaptive samples increased. The Pareto front closely followed the path of the best performing training samples. The training sample with the lowest maximum thickness was omitted from the Pareto front, as the performance of this point was particularly sensitive to input perturbations.

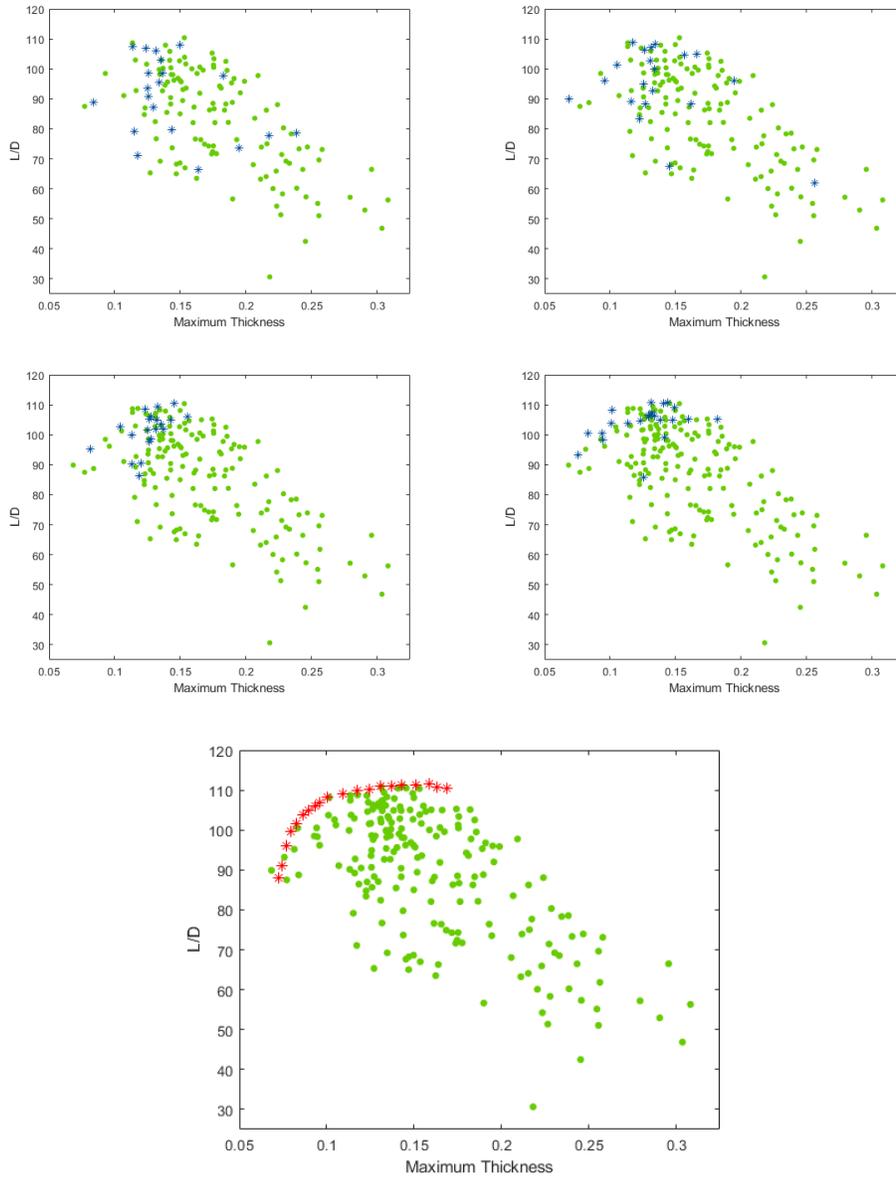


Figure 9: **HF Training Points and Pareto Front:** The top four figures contain the respective adaptive sampling batches. In order, top left, top right, bottom left, bottom right. The green dots are the HF points that have previously been obtained, and the blue stars represent the newly evaluated adaptive samples. The plot on the bottom contains the full set of HF samples (green dots) and the Pareto front (red stars) obtained via optimising the MF RGPEs for each objective.

Figure 10 contains the Pareto fronts from the MF RGPE approach (red stars) and the SF RGPE approach (blue dots). Both Pareto fronts initially rise relatively sharply before reaching a plateau with respect to the lift-to-drag coefficient. However, there is a significant discrepancy between the respective performance of the two Pareto fronts, with that of the MF RGPE completely dominating the SF RGPE counterpart. Whilst the SF RGPE approach wasted a number of samples searching in uncertain but ultimately low interest areas, the MF RGPE approach was able to discard these areas and target more promising locations due to the information provided by the LF data.

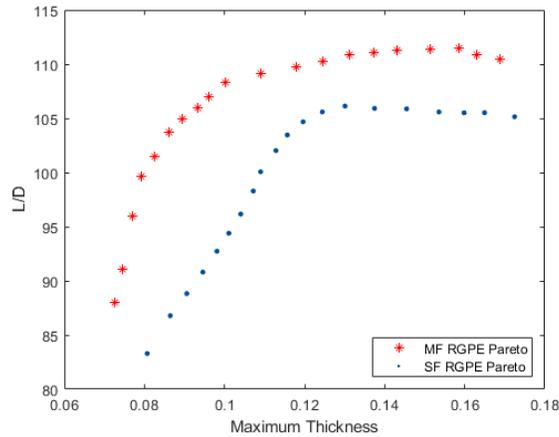


Figure 10: **Pareto Front Comparison:** The red stars represent the Pareto solutions obtained using the MF RGPE approach. The blue dots are the Pareto solutions obtained using the SF RGPE approach.

Figure 11 displays 20 validation samples for 4 designs taken from the MF RGPE Pareto front. As in the turbulated duct case study, the number of validation samples were limited due to the computational costs involved. The validation samples were selected to verify the performance of the MF RGPE approach across the Pareto front. It should be noted that there was zero discrepancy between the LF simulator and HF simulator output for the maximum thickness. As a result, there was significantly less GPE uncertainty for this objective, and the majority of the uncertainty bounds with respect to the maximum thickness is due to input uncertainty. Each validation sample was within the  $2\sigma$  uncertainty bounds of the original Pareto solution.

Moreover, the aerofoils corresponding to the validation points for the third (in ascending order of L/D ratio) Pareto solution were plotted against the original for a visual depiction of input uncertainty.

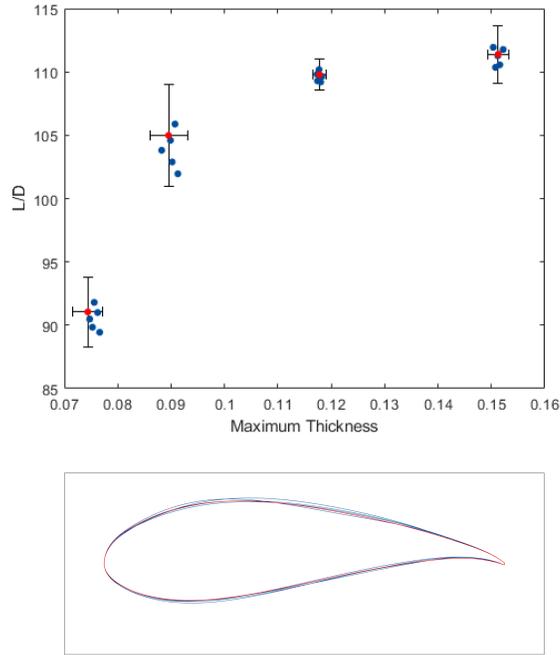


Figure 11: **Validation Points** Top: The red dots represent the original Pareto solutions and the black lines their respective uncertainty bounds. The blue dots are the simulator realisations at some perturbed design of the original Pareto solution. Bottom: Aerofoils for the validation samples (blue) and Pareto solution (red).

## 5. Conclusion

A Gaussian process emulation approach to perform efficient single-loop robust optimisation of expensive models, denoted MF RGPE, was presented. The approach combines various enhancements of the Gaussian process emulation approach, utilising MF training data and factoring input uncertainty into the output of the emulator. MF RGPE addresses the two main issues found when employing emulation-based approaches for robust optimisation, namely the quality of the emulator training set, and the efficiency of the

robust optimisation process itself. Provided lower-fidelity simulators exhibit similar behaviour to their higher-fidelity counterparts (as expected), the approach offers improvement over single-fidelity methods. This is due to the increased information available in both the adaptive sampling phase and in estimating the performance in areas without HF training data. In the situation where the LF simulator exhibits dissimilar, or even misleading behaviour, the approach essentially reverts to a SF GPE, albeit with a slightly reduced training set. Compared to other MF methods, MF RGPE offers increased efficiency in performing robust optimisation, by collapsing the problem down to a single-loop. An illustrative example highlighted some of the key concepts of the approach before two industrial test cases demonstrated its ability to outperform produce quality results. Future work involves augmenting the adaptive sampling regime with the ability to choose both the location and fidelity of an adaptive sample, as well as applying the approach to cases with more than two levels of fidelity.

## Acknowledgements

M. Ellison gratefully acknowledges General Electric Power for supporting this research through an EPSRC industrial case (Reference number: 1794913).

## Appendix A: Parameters of the CFD case studies and their respective bounds.

Turbulated Duct Input Variables			
Input Variable	Vari-	Lower Bound	Upper Bound
$\alpha$		35	55
Lu/D		0	0.5
Ld/D		0	0.5
h/D		0.05	0.15

Table 1: Input parameter ranges for turbulated duct case study.

Aerofoil Input Variables			
Input Variable	Vari-	Lower Bound	Upper Bound
Au1		0.1	0.3
Au2		0.1	0.5
Au3		0.0	0.6
Au4		0.0	0.7
Au5		-0.1	0.7
Au6		-0.1	0.7
Au7		-0.1	0.7
Au8		0.0	0.6
Au9		0.0	0.5
Au10		0.1	0.4
Al1		0.1	0.3
Al2		0.1	0.4
Al3		0.1	0.5
Al4		0.0	0.6
Al5		0.0	0.5
Al6		0.0	0.4
Al7		-0.1	0.4
Al8		-0.2	0.4
Al9		-0.2	0.3
Al10		-0.2	0.3

Table 2: Input parameter ranges for aerofoil case study.

## References

- [1] V. Gabrel, C. Murat, A. Thiele, Recent advances in robust optimization: An overview, *Eur. J. Oper. Res.* 235 (3) (2014) 471–483.
- [2] L. Wang, S. Kodiyalam, An efficient method for probabilistic and robust design with non-normal distributions, *AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Denver, 2002, Paper No.AIAA-2002-1754.
- [3] K.M. Ryan, Robust multi-objective optimization of hypersonic vehicles under asymmetric roughness induced boundary-layer transition, University of Maryland, 2014, Ph.D thesis.
- [4] J. Marzat, E. Walter, H. Piet-Lahanier, Worst-case global optimization of black-box functions through Kriging and relaxation, *J. Global. Opt.*, 55 (4) (2013) 707-727.
- [5] S. ur Rehman, M. Langelaar, F. van Keulen, Efficient Kriging-based robust optimization of unconstrained problems, *J. Comput. Sci.* 5 (6) (2014) 872-881.
- [6] M. Kennedy, A. O’Hagan, Bayesian Calibration of Computer Models, *J. Royal Stat. Soc.* 63 (3) (2001) 425-464.
- [7] J. Oakley, Eliciting Gaussian Process Priors for Complex Computer Codes, *J. Royal Stat. Soc.* 51 (1) (2002) 81-97.
- [8] A. O’Hagan, Bayesian analysis of computer code outputs: A tutorial, *Reliab. Eng. Syst. Saf.* 91 (10) (2006) 1290-1300.
- [9] D.R. Jones, M. Schonlau, W.J. Welch, Efficient Global Optimization of Expensive Black-Box Functions, *J. Global. Opt.* 13 (4) (1998) 455-492.
- [10] P.O. Hristov, F.A. Diaz De la O, U. Farooq, K.J. Kubiak, Adaptive Gaussian process emulators for efficient reliability analysis, *Appl. Math. Model.* 71 (2019) 138–151.
- [11] Q. Xu, E. Wehrle, H. Baier Adaptive surrogate-based design optimization with expected improvement used as infill criterion, *Optimization* 61 (6) (2012) 661-684.

- [12] P.D. Arendt, D.W. Apley, W. Chen, Objective-Oriented Sequential Sampling for Simulation Based Robust Design Considering Multiple Sources of Uncertainty, *J. Mech. Des.* 135 (5) (2013).
- [13] Y. Ling et al, An intelligent sampling framework for multi-objective optimization in high dimensional design space, *AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Kissimmee, 2018, doi: 10.2514/6.2018-0912.
- [14] C. Song, X. Yang, W. Song, Multi-infill strategy for kriging models used in variable fidelity optimization, *Chin. J. Aeronaut.* 31 (3) (2018) 448–456.
- [15] S. Ghosh et al, A strategy for adaptive sampling of multi-fidelity Gaussian processes to reduce predictive uncertainty, *Proceedings of the ASME Design Engineering Technical Conference*, Anaheim, 2019, doi: 10.1115/DETC2019-98418.
- [16] H. Zhong-Hua, S. Gortz, Hierarchical kriging model for variable-fidelity surrogate modeling, *AIAA J.* 50 (9) (2012) 1885–1955.
- [17] D.J.J. Toal, A.J. Keane, Efficient multipoint aerodynamic design optimization via cokriging, *J. Aircr.* 48 (5) (2011) 1685–1695.
- [18] J. Bailly, D. Bailly, Multifidelity Aerodynamic Optimization of a Helicopter Rotor Blade, *AIAA J.* 57 (8) (2019) 3132–3144.
- [19] Deb, K., and Gupta, H. (2006) ‘Introducing Robustness in Multi-Objective Optimization’, *Evolutionary Computation*, Vol. 14, No. 4, pp. 463–494.
- [20] K. M. Ryan et al, A Gaussian process modeling approach for fast robust design with uncertain inputs, *Proceedings of the ASME Turbo Expo*, Oslo, 2018, doi: 10.1115/GT201877007.
- [21] C.M. Bishop, *Pattern recognition and machine learning*, 1st Edition, New York: Springer, 2006.
- [22] M. Kennedy, A. O’Hagan, Predicting the output from a complex computer code when fast approximations are available, *Biometrika*, 87 (1) (2000) pp. 1–13.

- [23] L. Le Gratiet, J. Garnier, Recursive co-kriging model for design of computer experiments with multiple levels of fidelity, *Int. J. Uncertain. Quantif.* 4 (5) (2014) 365–386.
- [24] J. Quinonero-Candela, A. Girard, C.E. Rasmussen, Prediction at an uncertain input for Gaussian processes and relevance vector machines application to multiple-step ahead time-series forecasting, Tech. rep., Technical University of Denmark, 2003.
- [25] J. Quinonero-Candela, A. Girard, J. Larsen, C.E. Rasmussen, Propagation of uncertainty in bayesian kernels models - application to multiple-step ahead forecasting, *International Conference on Acoustics, Speech and Signal Processing*, Hong Kong, 2003, doi: 10.1109/ICASSP.2003.1202463.
- [26] M.D. McKay, R.J. Beckman, W.J. Conover, A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code, *Technometrics*, 21 (2) (1979) 239-245.
- [27] Z. Li, X. Wang, S. Ruan, Z. Li, C. Shen, Y. Zeng, A modified hypervolume based expected improvement for multi-objective efficient global optimization method, *Struct. Multidiscipl. Optim.* 58 (5) (2018) 1961-1979.
- [28] S.K. Au, J.L. Beck, Estimation of small failure probabilities in high dimensions by subset simulation. *Probab. Eng. Mech.* 16 (4) (2001) 263-277.
- [29] S.K. Au, J. Ching, J.L. Beck, Application of subset simulation methods to reliability benchmark problems, *Struct. Saf.* 29 (3) (2007) 183–193. doi: 10.1016/j.strusafe.2006.07.008.
- [30] J.-M. Bourinet, F. Deheeger, M. Lemaire, Assessing small failure probabilities by combined subset simulation and Support Vector Machines, *Struct. Saf.* 33 (6) (2011) 343–353. doi: 10.1016/j.strusafe.2011.06.001.
- [31] H. Li, Subset simulation for unconstrained global optimization, *Appl. Math. Model.* 35 (10) (2011) 5108-5120.
- [32] H. Li, S.K. Au, Design optimization using Subset Simulation algorithm, *Struct. Saf.* 32 (6) (2010) 384-392.

- [33] X. Suo, X. Yu, H. Li, Subset simulation for multi-objective optimization, *Appl. Math. Model.* Vol. 44 (2017) 425-445, doi: 10.1016/j.apm.2017.02.005
- [34] D. Zhan, J. Qian, Y. Cheng, Pseudo expected improvement criterion for parallel EGO algorithm, *J. Global. Opt.* 68 (3) (2016) 641-662.
- [35] H.K. Versteeg, W. Malalasekera, *An Introduction to Computational Fluid Dynamics: The Finite Volume Method*, 2nd Edition, Harlow: Pearson, 2007.
- [36] M. Zlatinov, G. Laskowski, Hybrid Large-Eddy Simulation Optimization of a Fundamental Turbine Blade Turbulated Cooling Passage, *J. Propuls. Power* 31 (5) (2015).
- [37] J. Loepky, J. Sacks, W. Welch, Choosing the Sample Size of a Computer Experiment: A Practical Guide, *Technometrics* 51 (4) (2009) 366-376.
- [38] E. Olsen, Three-Dimensional Piecewise-Continuous Class-Shape Transformation of Wings, 16th AIAA/ ISSMO Multidisciplinary Analysis and Optimization Conference, Dallas, 2015, <https://doi.org/10.2514/6.2015-3238>.