

# Multi-view Bayesian optimisation in reduced dimension for engineering design

Thomas A. Archbold\*, Ieva Kazlauskaitė and Fehmi Cirak

*Department of Engineering, University of Cambridge, Trumpington Street, Cambridge CB2 1PZ, U.K.*

## SUMMARY

Bayesian optimisation is an adaptive sampling strategy for constructing a Gaussian process surrogate to emulate a black-box computational model with the aim of efficiently searching for the global minimum. However, Gaussian processes have limited applicability for engineering problems with many design variables. Their scalability can be significantly improved by identifying a low-dimensional vector of latent variables that serve as inputs to the Gaussian process. In this paper, we introduce a multi-view learning strategy that considers both the input design variables and output data representing the objective or constraint functions, to identify a low-dimensional space of latent variables. Adopting a fully probabilistic viewpoint, we use probabilistic partial least squares (PPLS) to learn an orthogonal mapping from the design variables to the latent variables using training data consisting of inputs and outputs of the black-box computational model. The latent variables and posterior probability densities of the probabilistic partial least squares and Gaussian process models are determined sequentially and iteratively, with retraining occurring at each adaptive sampling iteration. We compare the proposed probabilistic partial least squares Bayesian optimisation (PPLS-BO) strategy to its deterministic counterpart, partial least squares Bayesian optimisation (PLS-BO), and classical Bayesian optimisation, demonstrating significant improvements in convergence to the global minimum.

KEY WORDS: Multi-view learning, Probabilistic partial least squares, Dimension reduction, Surrogate modelling, Bayesian optimisation

## 1 INTRODUCTION

Conventional approaches to design optimisation require the gradients of objective and constraint functions such as the compliance or maximum stress, with respect to design variables often defined in terms of computer aided design (CAD) model parameters [1, 2, 3, 4, 5, 6]. In practice such gradient information may not be available and could quickly become costly to evaluate, especially for multiphysics problems with complex geometry. Consequently, it is expedient to interpret the computational model as a black-box function where output quantities of interest (QOIs), such as objective and constraint function values, are observed for a finite set of prescribed training inputs (design variables). Output QOIs for arbitrary inputs are inferred using a quick-to-evaluate surrogate that emulates the black-box function. Clearly, a surrogate cannot precisely reproduce a black-box function and uncertainty about the inferred QOI is unavoidable. Statistical surrogates such as Gaussian processes (GPs) [7, 8, 9] yield a probability density to quantify the uncertainty in inferring unobserved QOIs. Although GPs have been used effectively for engineering design [10, 11, 12], they do not scale to problems involving a large number of input variables, i.e. they are susceptible to the “curse of dimensionality” [13]. GPs can be applied to design optimisation using adaptive sampling methods such as Bayesian optimisation (BO) [14, 15].

---

\*Correspondence to: taa42@cantab.ac.uk

In this paper, we emulate each QOI defined via the black-box computational model, using a GP surrogate  $f(\mathbf{s})$ . The Gaussian prior probability density characterised by its mean and covariance, is updated using a training data set  $\mathcal{D}$  and Bayes' rule to yield the posterior probability density. The training data set  $\mathcal{D} = \{(\mathbf{s}_i, y_i)\}_{i=1}^n$  collects the  $n$  pairs of design variables  $\mathbf{s}_i \in \mathbb{R}^{d_s}$  and the QOI  $y_i \in \mathbb{R}$  from the computational model. The likelihood function follows from the posited statistical model  $y = f(\mathbf{s}) + \epsilon_y$ , with an additive noise (error) term  $\epsilon_y$ . We use BO to perform global optimisation to find optimum design variables that minimise the objective function and satisfy the constraints [16]. BO is an adaptive sampling methodology that iteratively updates the GP using samples proposed by an acquisition function [17]. The acquisition function balances the exploration of the design space with the exploitation of regions where the GP posterior predicts the global minimum [18]. The training data set  $\mathcal{D}$  expands with each proposed sample, and the GP is updated to fit the new data. Like standard GPs, BO is prone to the "curse of dimensionality" and is typically limited to engineering design problems with approximately 10 or fewer design parameters, see e.g. [19, 20] and [21].

For computational tractability when using GP surrogates, the dimension  $d_s$  of the design variable vector  $\mathbf{s}$  must be restricted. To achieve this, we assume that the GP  $f(\mathbf{z})$  depends on a low-dimensional latent variable vector  $\mathbf{z}$  rather than  $\mathbf{s}$ . The latent variables are obtained from a linear mapping  $\mathbf{z} = \mathbf{W}^\top \mathbf{s}$ , with a non-square (tall) orthogonal matrix  $\mathbf{W}$ , i.e.  $\mathbf{W}^\top \mathbf{W} = \mathbf{I}$ . The orthonormal rows and columns of the matrix  $\mathbf{W}$  represent the low-dimensional latent subspace corresponding to  $\mathbf{z}$ . Engineering design problems usually consist of multiple QOIs, such as the compliance objective function and stress constraints; both these must be considered in constructing the orthogonal matrix  $\mathbf{W}$ . We adopt a multi-view learning (MVL) strategy to discover the low-dimensional orthonormal basis [22]. Although the suite of MVL methods is extensive [23], sensible models are not deterministic but have epistemic uncertainties when predicting unobserved data (as is the case when computing unobserved latent variables) [24]. Epistemic uncertainties decrease with more training data and converge to zero in the limit of infinite data.

In the proposed approach, we adopt a fully probabilistic viewpoint by combining BO with probabilistic partial least squares (PPLS). PPLS [25] postulates a probabilistic model given by  $\mathbf{s} = \mathbf{W}\mathbf{z} + \hat{\epsilon}_s$  and  $\mathbf{y} = \mathbf{Q}\mathbf{z} + \hat{\epsilon}_y$ , such that design variable and QOI vector pairs  $(\mathbf{s}, \mathbf{y})$  are generated from an unobserved latent variable vector  $\mathbf{z}$ . The error incurred by reconstructing  $\mathbf{s}$  and  $\mathbf{y}$  from  $\mathbf{z}$  is assumed to have Gaussian probability densities, i.e.  $\hat{\epsilon}_s \sim \mathcal{N}(\mathbf{0}, \hat{\Sigma}_s)$  and  $\hat{\epsilon}_y \sim \mathcal{N}(\mathbf{0}, \hat{\Sigma}_y)$ .

The entries of the orthogonal matrices  $\mathbf{W}$  and  $\mathbf{Q}$ , and covariance matrices  $\hat{\Sigma}_s$  and  $\hat{\Sigma}_y$  are treated as PPLS model hyperparameters. The hyperparameters and the approximate posterior probability density of the latent variable vector  $\mathbf{z}$  are determined using variational Bayes [26]. We propose probabilistic partial least squares Bayesian optimisation (PPLS-BO) that combines PPLS with adaptive sampling using BO. The probabilistic model ensures that each latent variable proposed by the BO acquisition function yields a density of corresponding design variables. By sampling from this density, we show that exploration is enhanced, leading to improved convergence towards the global minimum and increased robustness to misspecification in the dimension of the low-dimensional latent space.

BO approaches for high-dimensional problems have received substantial attention in the machine learning and engineering communities. A low-dimensional subspace can be discovered using principal component analysis (PCA) [27, 28] or random embedding [29]. Alternatively, gradient information about the QOI can be used to discover a low-dimensional "active" subspace [30, 31, 32, 33]. Where gradient information is unavailable, the PCA covariance can be weighted using output data [34], or the PCA components can be interrogated when combined with GPs using a penalised log-likelihood formulation [35]. The subspace basis can be optimised jointly with the GP hyperparameters using two-step algorithms based on coordinate descent [36, 37]. Partial least squares (PLS) has been combined with GPs [38, 39, 40, 41] and extended to BO for adaptive sampling with multiple QOIs [42, 43]. We compare this approach, which we abbreviate as PLS-BO, to the PPLS-BO algorithm proposed in this paper.

This paper is structured as follows. In Section 2, we review PPLS for dimensionality reduction and BO, consisting of GPs and acquisition functions for adaptive sampling. In Section 3, we

present the proposed PPLS-BO algorithm for adaptive sampling in reduced dimension. We detail the formulations used to compute the posterior probability density of the GP in reduced dimension, and provide the pseudocode of the PPLS-BO algorithm. Three examples are given in Section 4, demonstrating the improved convergence of PPLS-BO when compared to PLS-BO and classical BO. We demonstrate the algorithms' versatility through design optimisation of a complex manufacturing example. Finally, Section 5 concludes the paper and discusses promising directions for further research.

## 2 BACKGROUND

In this section, we provide the relevant background on design optimisation, PPLS, and BO that are used in the proposed approach.

### 2.1 Design optimisation

In structural optimisation, the objective and constraint functions are determined by solving a FE model. The objective  $J : \mathbb{R}^{d_s} \rightarrow \mathbb{R}$  and constraint  $H^{(j)} : \mathbb{R}^{d_s} \rightarrow \mathbb{R}$  are functions of the design variables  $s \in \mathbb{R}^{d_s}$ . A total of  $d_y$  objective and constraint functions, consisting (without loss of generality) of a single objective and  $d_y - 1$  constraint functions, are considered. The design optimisation problem may be stated as

$$\begin{aligned} \min_{s \in D_s} \quad & J(s), \\ \text{s.t.} \quad & H^{(j)}(s) \leq 0, \quad j \in \{1, 2, \dots, d_y - 1\}, \\ & D_s = \{s \in \mathbb{R}^{d_s} \mid \bar{s}^{(l)} \leq s \leq \bar{s}^{(u)}\}, \end{aligned} \quad (1)$$

where  $D_s \subset \mathbb{R}^{d_s}$  is the design domain, bounded between a lower limit  $\bar{s}^{(l)}$  and upper limit  $\bar{s}^{(u)}$ . We collect the objective and constraint function evaluations in an output vector  $y \in \mathbb{R}^{d_y}$  such that  $y = (J(s) \ H^{(1)}(s) \ H^{(2)}(s) \cdots \ H^{(d_y-1)}(s))^T$ .

### 2.2 Probabilistic partial least squares

PPLS extends deterministic PLS (Appendix A) by treating the low-dimensional latent space as uncertain [25]. In Bayesian analysis, the uncertain latent variables are treated as random variables. PPLS is based on the following statistical observation (or, generative) model,

$$s = Wz + \hat{\epsilon}_s, \quad \hat{\epsilon}_s \sim \mathcal{N}\left(\mathbf{0}, \hat{\Sigma}_s\right), \quad (2a)$$

$$y = Qz + \hat{\epsilon}_y, \quad \hat{\epsilon}_y \sim \mathcal{N}\left(\mathbf{0}, \hat{\Sigma}_y\right). \quad (2b)$$

The unobserved latent variable vector  $z \in \mathbb{R}^{d_z}$  is defined in a linear subspace given by the columns of the orthogonal matrices  $W \in \mathbb{R}^{d_s \times d_z}$  and  $Q \in \mathbb{R}^{d_y \times d_z}$ , with  $W^T W = Q^T Q = I$ . Reconstructing the design variables  $s$  and outputs  $y$  incur reconstruction errors modelled by zero-mean Gaussian random vectors  $\hat{\epsilon}_s$  and  $\hat{\epsilon}_y$  with diagonal covariance matrices  $\hat{\Sigma}_s \in \mathbb{R}^{d_s \times d_s}$  and  $\hat{\Sigma}_y \in \mathbb{R}^{d_y \times d_y}$  respectively. The joint probability density

$$p_\Phi(y, s, z) = p_v(y|z)p_\zeta(s|z)p(z), \quad (3)$$

of all observed and unobserved variables is factorised using the posited conditional independence structure underlying (2), as also visualised in Figure 1. According to (2a) and (2b) the prior

probability densities for  $s$  and  $y$  are given by

$$p_\zeta(s|z) = \mathcal{N}\left(\mathbf{W}z, \hat{\Sigma}_s\right), \quad (4a)$$

$$p_v(y|z) = \mathcal{N}\left(\mathbf{Q}z, \hat{\Sigma}_y\right). \quad (4b)$$

The trainable PPLS model hyperparameters are collected in the two sets  $\zeta = \{\mathbf{W}, \hat{\Sigma}_s\}$  and  $v = \{\mathbf{Q}, \hat{\Sigma}_y\}$ . The prior probability density of the latent variable vector  $z$  is assumed as

$$p(z) = \mathcal{N}(\mathbf{0}, \mathbf{I}). \quad (5)$$

Using the likelihood  $p_\Phi(y, s|z) = p_v(y|z)p_\zeta(s|z)$ , the PPLS model hyperparameters  $\Phi = \zeta \cup v$  can be determined by maximising the marginal likelihood

$$p_\Phi(y, s) = \int p_\Phi(y, s|z)p(z) dz := \mathcal{N}(\mathbf{0}, \hat{\Sigma}_{ys}), \quad (6)$$

which is a zero-mean Gaussian with the covariance

$$\hat{\Sigma}_{ys} = \begin{pmatrix} \mathbf{Q}\mathbf{Q}^\top + \hat{\Sigma}_y & \mathbf{Q}\mathbf{W}^\top \\ \mathbf{W}\mathbf{Q}^\top & \mathbf{W}\mathbf{W}^\top + \hat{\Sigma}_s \end{pmatrix}; \quad (7)$$

see the derivation in Appendix B. The posterior probability density of the unobserved latent variables is given by Bayes' rule

$$p_\Phi(z|y, s) = \frac{p_\Phi(y, s|z)p(z)}{p_\Phi(y, s)}, \quad (8)$$

which takes the form of a Gaussian probability density

$$p_\Phi(z|y, s) := \mathcal{N}(\hat{\mu}_z, \hat{\Sigma}_z), \quad (9)$$

where the mean and covariance are given by

$$\hat{\mu}_z = \mathbf{B}^\top \hat{\Sigma}_{ys}^{-1} \mathbf{d}, \quad (10a)$$

$$\hat{\Sigma}_z = \mathbf{I} - \mathbf{B}^\top \hat{\Sigma}_{ys}^{-1} \mathbf{B}, \quad (10b)$$

and

$$\mathbf{B} := \begin{pmatrix} \mathbf{Q} \\ \mathbf{W} \end{pmatrix}, \quad \mathbf{d} := \begin{pmatrix} \mathbf{y} \\ \mathbf{s} \end{pmatrix}. \quad (11)$$

The posterior probability density still depends on unknown PPLS model hyperparameters  $\Phi$ . Although these could be obtained directly by maximising the marginal likelihood  $p_\Phi(y, s)$  given in (6) using gradient descent, the computational complexity is proportional to the input dimension  $d_s$ . To improve computational tractability, we use variational Bayes to approximate the posterior with a Gaussian trial density  $q(z)$  where the Kullback-Leibler (KL) divergence between the two densities is defined as

$$\begin{aligned} D_{KL}(q(z) || p_\Phi(z|y, s)) &= \int q(z) \ln \left( \frac{q(z)}{p_\Phi(z|y, s)} \right) dz \\ &= \int q(z) \ln \left( \frac{q(z)}{p_\Phi(y, s|z)p(z)} \right) dz + \ln p_\Phi(y, s). \end{aligned} \quad (12)$$

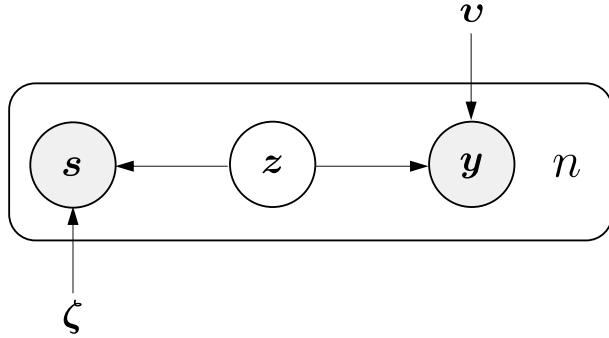


Figure 1. Graphical Model of PPLS where  $n$  is the size of the training data set  $\mathcal{D}$ ,  $s$  are the design variables,  $z$  are the low-dimensional latent variables,  $y$  are the observations, and  $\zeta$  and  $v$  are two vectors containing the PPLS model hyperparameters.

Since the KL divergence is non-negative  $D_{KL}(\cdot||\cdot) \geq 0$ , the marginal likelihood is upper bounded by

$$\ln p_\Phi(y, s) \geq \int q(z) \ln \left( \frac{p_\Phi(y, s|z)p(z)}{q(z)} \right) dz := \mathcal{F}(y, s). \quad (13)$$

Hence, the KL divergence is minimised by maximising the evidence lower bound (ELBO)  $\mathcal{F}(y, s)$ , which may be written more compactly as

$$\mathcal{F}(y, s) = \mathbb{E}_{q(z)} (\ln p_\Phi(y, s|z)) + \mathbb{E}_{q(z)} (\ln p(z)) - \mathbb{E}_{q(z)} (\ln q(z)). \quad (14)$$

Both the trial density  $q(z)$  and PPLS model hyperparameters  $\Phi$  are obtained by maximising the ELBO using alternating coordinate descent with the expectation maximisation (EM) algorithm [44] (see Appendix C). Alternatively, the PPLS model hyperparameters may also be obtained using stochastic gradient descent.

### 2.3 Bayesian optimisation

BO is composed of two components, a GP surrogate to emulate the QOI and an adaptive sampling algorithm to update the GP while balancing design space exploration with exploitation of belief in where the global minimum is located. In the proposed approach, the GP is constructed over the low-dimensional latent space.

**2.3.1 Gaussian process regression** A single observation of the objective  $J(\mathbf{W}z)$  or constraint  $H^{(j)}(\mathbf{W}z)$  functions generically denoted by  $y \in \mathbb{R}$ , obeys the statistical data generating model

$$y = f(z) + \epsilon_y, \quad \epsilon_y \sim \mathcal{N}(0, \sigma_y^2), \quad (15)$$

where  $\sigma_y \in \mathbb{R}^+$  denotes the noise or error standard deviation (such as the FE modelling or discretisation error), see also Figure 2. The GP  $f : \mathbb{R}^{d_z} \rightarrow \mathbb{R}$  maps the latent variables  $z$  to the observation  $y$  and has the prior probability density

$$f(z) \sim \mathcal{GP}(0, c(z, z')), \quad (16)$$

Without loss of generality, we choose the squared-exponential covariance function  $c : \mathbb{R}^{d_z} \times \mathbb{R}^{d_z} \rightarrow \mathbb{R}^+$  given by

$$c(z, z') = \sigma_f^2 \exp \left( - \sum_{i=1}^{d_z} \frac{(z_i - z'_i)^2}{2\ell_i} \right), \quad (17)$$

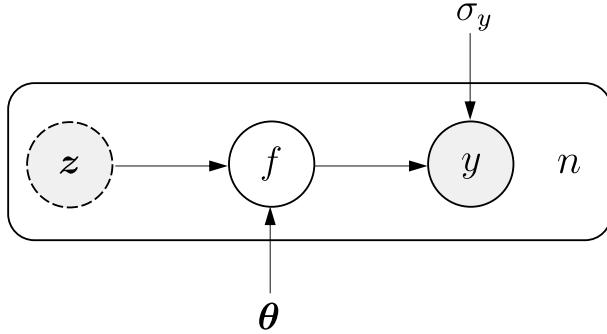


Figure 2. Graphical Model of GP regression where  $n$  is the size of the training data set  $\mathcal{D}$ ,  $s$  are the design variables,  $f$  is the target output variable,  $y$  is the observation, and  $\theta \cup \{\sigma_y\}$  are the model hyperparameters.

where  $\sigma_f \in \mathbb{R}^+$  is a scaling parameter,  $\ell_i \in \mathbb{R}^+$  is a length scale parameter, and  $z_i$  denotes the  $i^{\text{th}}$  component of latent variable vector  $\mathbf{z}$ . The hyperparameters of the GP are collected in the set  $\Theta = \{\sigma_f, \ell_1, \ell_2, \dots, \ell_{d_z}\}$ .

Let  $\mathcal{D} = \{(z_i, y_i) \mid i = 1, 2, \dots, n\}$  denote the training data, which is collected into the latent variable matrix  $\mathbf{Z} = (z_1 \ z_2 \ \dots \ z_n)^T \in \mathbb{R}^{n \times d_z}$  and output vector  $\mathbf{y} = (y_1 \ y_2 \ \dots \ y_n)^T \in \mathbb{R}^n$ . For any test output variables  $\mathbf{f}_* \in \mathbb{R}^{n_*}$  corresponding to test latent variables  $\mathbf{Z}_* \in \mathbb{R}^{n_* \times d_z}$ , the joint probability density with the outputs is given by

$$p_\Theta(\mathbf{f}_*, \mathbf{y}) = \mathcal{N} \left( \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \end{pmatrix}, \begin{pmatrix} \mathbf{C}_{Z_* Z_*} & \mathbf{C}_{Z_* Z} \\ \mathbf{C}_{Z Z_*} & \mathbf{C}_{Z Z} + \sigma_y^2 \mathbf{I} \end{pmatrix} \right), \quad (18)$$

where the entries of the covariance matrix  $\mathbf{C}_{ZZ}$  consist of the covariance function  $c(z, z')$  evaluated at  $z$  and  $z'$  stored as rows of  $\mathbf{Z}$ , and the hyperparameters are  $\Theta = \theta \cup \{\sigma_y\}$ . Similarly, the entries of the covariance matrix  $\mathbf{C}_{Z Z_*}$  consist of the covariance function  $c(z, z_*)$ , where  $z_*$  is a row of  $\mathbf{Z}_*$ . The posterior predictive probability density is obtained by conditioning the joint probability density  $p_\Theta(\mathbf{f}_*, \mathbf{y})$  on the output vector  $\mathbf{y}$  and is given by

$$p_\Theta(\mathbf{f}_* | \mathbf{y}) = \mathcal{N} \left( \mathbf{C}_{Z_* Z} (\mathbf{C}_{ZZ} + \sigma_y^2 \mathbf{I})^{-1} \mathbf{y}, \mathbf{C}_{Z_* Z_*} - \mathbf{C}_{Z_* Z} (\mathbf{C}_{ZZ} + \sigma_y^2 \mathbf{I})^{-1} \mathbf{C}_{Z Z_*} \right). \quad (19)$$

Furthermore, the optimal GP hyperparameters  $\Theta$  can be computed by maximising the log marginal likelihood

$$\Theta^* = \arg \max_{\Theta} \ln p_\Theta(\mathbf{y}), \quad (20)$$

where

$$p_\Theta(\mathbf{y}) = \mathcal{N} (\mathbf{0}, \mathbf{C}_{ZZ} + \sigma_y^2 \mathbf{I}). \quad (21)$$

**2.3.2 Adaptive sampling** For any test input  $z^*$ , the GP posterior probability density  $p_{\Theta_k}(f_* | y)$  given by (19) yields a predictive mean  $\mu : \mathbb{R}^{d_z} \rightarrow \mathbb{R}$  and standard deviation  $\sigma : \mathbb{R}^{d_z} \rightarrow \mathbb{R}^+$ , which are functions of the latent variables  $\mathbf{z}$ . In the adaptive sampling iteration  $k$ , a new sampling point

$$z_{k+1} = \arg \max_{z \in D_z} a(\mu_k(z), \sigma_k(z)), \quad (22)$$

is proposed by maximising an acquisition function  $a : \mathbb{R} \rightarrow \mathbb{R}$ , which balances exploration and exploitation. Although there are many kinds of acquisition functions [45], upper confidence bound (UCB) [46, 47] and expected improvement (EI) [14] are used more extensively. The UCB acquisition function

$$a_{ucb}(\mu(z), \sigma(z)) = -\mu(z) + \gamma \sigma(z), \quad (23)$$

uses a weighted sum of the negative mean and standard deviation, where  $\gamma \in \mathbb{R}^+$  is an exploration parameter. Similarly, the EI acquisition function is defined in terms of the GP mean and standard

deviation and is given by

$$a_{ei}(\mu(z), \sigma(z)) = (y^+ - \mu(z) - \xi) \Phi(u) + \sigma(z)\phi(u), \quad (24)$$

where  $\Phi : \mathbb{R} \rightarrow \mathbb{R}^+$  is the standard cumulative density function,  $\phi : \mathbb{R} \rightarrow \mathbb{R}^+$  is the standard probability density function,  $\xi \in \mathbb{R}$  is a jitter parameter, and

$$u = \frac{y^+ - \mu(z) - \xi}{\sigma(z)}. \quad (25)$$

The acquisition function can be extended to prohibit solutions that do not satisfy the constraints from being proposed [48]. By denoting  $f_*^{(i)} \in \mathbb{R}$  as the test output variable for the  $i^{\text{th}}$  GP corresponding to the  $i^{\text{th}}$  objective or constraint function with output vector  $\mathbf{y}^{(i)} \in \mathbb{R}^n$ , the constrained acquisition function is given by

$$a_c(\mu^{(i)}(z), \sigma^{(i)}(z)) = a(\mu^{(1)}(z), \sigma^{(1)}(z)) \prod_{i=2}^{d_y} \left( 1 + \rho^{(i)} \int_0^\infty p_{\Theta^{(i)}}(f_*^{(i)} | \mathbf{y}^{(i)}) df_*^{(i)} \right), \quad (26)$$

where  $\rho^{(i)} \in \mathbb{R}$  is a penalty constant to prevent infeasible solutions being proposed. Let  $\mu^{(i)}(\cdot)$  and  $\sigma^{(i)}(\cdot)$  denote the mean and standard deviation of the  $i^{\text{th}}$  GP posterior predictive probability density respectively, where

$$p_{\Theta^{(i)}}(f_*^{(i)} | \mathbf{y}^{(i)}) = \mathcal{N}\left(\mu^{(i)}(z_*), \left(\sigma^{(i)}(z_*)\right)^2\right). \quad (27)$$

Since the constrained acquisition function may be dominated by sharp peaks, genetic algorithms such as NSGA-II [49] can be used to maximise the acquisition function. At each adaptive sampling iteration, the training data  $\mathcal{D}$  is updated with the proposed sample and the GP is subsequently updated, as summarised in Algorithm 1.

### 3 MULTI-VIEW BAYESIAN OPTIMISATION

In this section, the posterior probability density of the latent variables is computed using PPLS and embedded within the GP. We derive a posterior predictive probability density, which makes it possible to obtain empirical estimates. Finally, we summarise our approach culminating in the probabilistic partial least squares Bayesian optimisation (PPLS-BO) algorithm.

---

#### Algorithm 1 Adaptive sampling with BO

---

**Data:**  $\mathcal{D}_0^{(i)} = \{(z_j, y_j^{(i)}) | j = 1, 2, \dots, n\}$   
**Input:** adaptive sampling budget  $n_k$   
**for**  $k \in \{0, 1, \dots, n_k - 1\}$  **do**  
    Compute  $\mu_k^{(i)}(z_*)$  and  $\sigma_k^{(i)}(z_*)$  using  $\mathcal{D}_k^{(i)}$  ▷ posterior probability density (19)  
    Propose sample  $z_{k+1} \leftarrow \arg \max_{z \in D_z} a_c(\mu_k^{(i)}(z), \sigma_k^{(i)}(z))$  ▷ maximise acquisition function (26)  
    Compute design variables  $s_{k+1} = \mathbf{W} z_{k+1}$  ▷ linear mapping  
    Collect observations  $y_{k+1}^{(i)}$  ▷ solve computational model  
    Augment data  $\mathcal{D}_{k+1}^{(i)} \leftarrow \{\mathcal{D}_k^{(i)}, (z_{k+1}, y_{k+1}^{(i)})\}$   
**Result:** global minimum  $y^{(1)*} \leftarrow \min\{y_1^{(1)}, y_2^{(1)}, \dots, y_{n+n_k}^{(1)}\}$

---

### 3.1 Gaussian processes in reduced dimension

The inputs of a standard GP, introduced in section 2.3.1, are deterministic. However, the latent variables determined with PPLS, introduced in Section 2.2, are random. We denote the posterior density as  $p_{\Theta^{(i)}}(f_*^{(i)} | \mathbf{y}^{(i)}, \mathbf{Z}, z_*)$ . For each random realisation of the latent variables  $\mathbf{Z}$  and test latent variables  $z_*$ , the density  $p_{\Theta^{(i)}}(f_*^{(i)} | \mathbf{y}^{(i)}, \mathbf{Z}, z_*)$  may be computed as the posterior predictive probability density of a standard GP (19), as shown in Figure 3. The marginal posterior predictive probability density is obtained by marginalising over latent variables and test latent variables, that is

$$p_{\Theta^{(i)}, \Phi}(f_*^{(i)} | \mathbf{Y}, \mathbf{S}) = \int p_{\Theta^{(i)}}(f_*^{(i)} | \mathbf{y}^{(i)}, \mathbf{Z}, z_*) p_{\Phi}(\mathbf{Z} | \mathbf{Y}, \mathbf{S}) p_{\Phi}(z_* | \mathbf{Y}, \mathbf{S}) d\mathbf{Z} dz_* . \quad (28)$$

The latent variables and test latent variables are given by

$$p_{\Phi}(\mathbf{Z} | \mathbf{Y}, \mathbf{S}) = \prod_{i=1}^n p_{\Phi}(z_i | \mathbf{y}_i, s_i) , \quad (29)$$

and

$$p_{\Phi}(z_* | \mathbf{Y}, \mathbf{S}) = \mathcal{N}(\bar{z}_*, \hat{\Sigma}_z) . \quad (30)$$

Here, we assumed independently and identically distributed samples. The design variable matrix  $\mathbf{S} = (s_1 s_2 \dots s_n)^T \in \mathbb{R}^{n \times d_s}$  collects the  $n$  design variables and output vectors  $\mathbf{y}^{(i)}$  consisting of observations corresponding to the  $i^{\text{th}}$  objective or constraint function are collected into the matrix  $\mathbf{Y} = (\mathbf{y}^{(1)} \mathbf{y}^{(2)} \dots \mathbf{y}^{(n)}) \in \mathbb{R}^{n \times d_y}$ . The latent variable posterior probability density  $p_{\Phi}(z_i | \mathbf{y}_i, s_i)$  is computed using the PPLS model (9), and the test latent variable posterior density has the same covariance  $\hat{\Sigma}_z$  given by (10b). The mean test latent variables  $\bar{z}_*$  now become the independent variables for optimisation. Test output variables  $f_*^{(i)}$  used for surrogate-based design optimisation can be sampled from the marginal posterior probability density  $p_{\Theta^{(i)}, \Phi}(f_*^{(i)} | \mathbf{Y}, \mathbf{S})$  yielding an expectation (by the law of total expectation)

$$\begin{aligned} \mathbb{E}(f^{(i)}(z_*)) &= \mathbb{E}_{p_{\Phi}(z_* | \mathbf{Y}, \mathbf{S})} \left( \mathbb{E}_{p_{\Phi}(\mathbf{Z} | \mathbf{Y}, \mathbf{S})} \left( \mathbb{E}_{p_{\Theta^{(i)}}(f_*^{(i)} | \mathbf{y}^{(i)}, \mathbf{Z}, z_*)} (f_*^{(i)}) \right) \right) \\ &= \mathbb{E}_{p_{\Phi}(z_* | \mathbf{Y}, \mathbf{S})} \left( \mathbb{E}_{p_{\Phi}(\mathbf{Z} | \mathbf{Y}, \mathbf{S})} (\mu^{(i)}(z_*)) \right) \\ &= \mathbb{E}(\mu^{(i)}(z_*)) , \end{aligned} \quad (31)$$

and variance (by law of total variance)

$$\begin{aligned} \text{var}(f^{(i)}(z_*)) &= \text{var}_{p_{\Phi}(z_* | \mathbf{Y}, \mathbf{S})} \left( \text{var}_{p_{\Phi}(\mathbf{Z} | \mathbf{Y}, \mathbf{S})} \left( \mathbb{E}_{p_{\Theta^{(i)}}(f_*^{(i)} | \mathbf{y}^{(i)}, \mathbf{Z}, z_*)} (f_*^{(i)}) \right) \right) + \\ &\quad \mathbb{E}_{p_{\Phi}(z_* | \mathbf{Y}, \mathbf{S})} \left( \mathbb{E}_{p_{\Phi}(\mathbf{Z} | \mathbf{Y}, \mathbf{S})} \left( \text{var}_{p_{\Theta^{(i)}}(f_*^{(i)} | \mathbf{y}^{(i)}, \mathbf{Z}, z_*)} (f_*^{(i)}) \right) \right) \\ &= \text{var}_{p_{\Phi}(z_* | \mathbf{Y}, \mathbf{S})} \left( \text{var}_{p_{\Phi}(\mathbf{Z} | \mathbf{Y}, \mathbf{S})} (\mu^{(i)}(z_*)) \right) + \mathbb{E}_{p_{\Phi}(z_* | \mathbf{Y}, \mathbf{S})} \left( \mathbb{E}_{p_{\Phi}(\mathbf{Z} | \mathbf{Y}, \mathbf{S})} \left( (\sigma^{(i)}(z_*))^2 \right) \right) \\ &= \text{var}(\mu^{(i)}(z_*)) + \mathbb{E}((\sigma^{(i)}(z_*))^2) . \end{aligned} \quad (32)$$

Both the expectation and variance can be estimated numerically using Monte Carlo (MC) sampling, yielding an approximate marginal posterior probability density

$$p_{\Theta^{(i)}, \Phi}(f_*^{(i)} | \mathbf{Y}, \mathbf{S}) \approx \hat{p}_{\Theta^{(i)}, \Phi}(f_*^{(i)} | \mathbf{Y}, \mathbf{S}) = \mathcal{N}\left(\hat{\mu}^{(i)}(\bar{z}_*), (\hat{\sigma}^{(i)}(\bar{z}_*))^2\right) , \quad (33)$$

where

$$\hat{\mu}^{(i)}(\bar{z}_*) = \mathbb{E}(\mu^{(i)}(z_*)) , \quad (34a)$$

$$\hat{\sigma}^{(i)}(\bar{z}_*) = \sqrt{\text{var}(\mu^{(i)}(z_*)) + \mathbb{E}((\sigma^{(i)}(z_*))^2)} . \quad (34b)$$

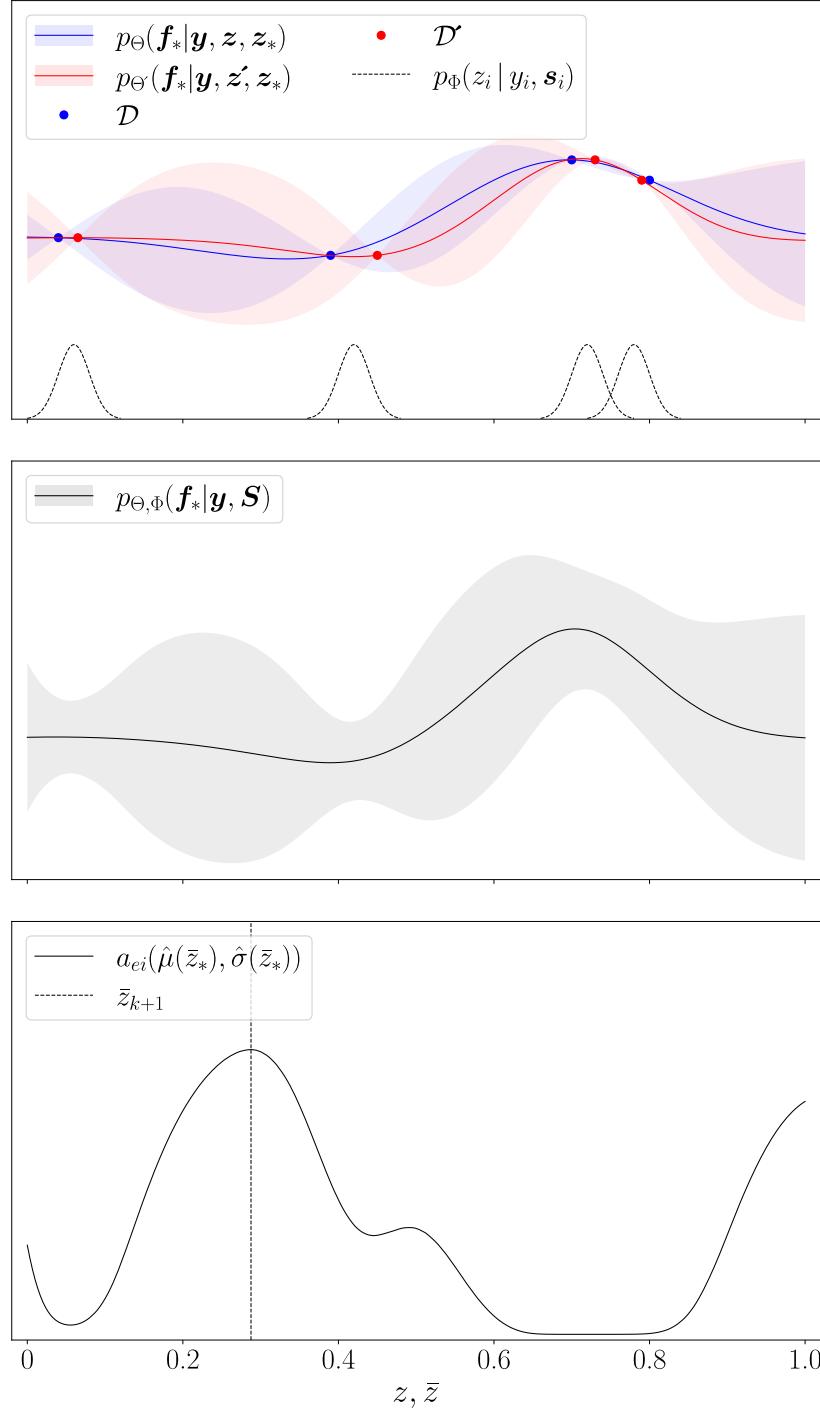


Figure 3. Illustrative one-dimensional example for the GP regression in reduced dimension. Two realisations of the GP posterior density  $p_{\Theta}(f_* | \mathbf{y}, \mathbf{z}, \mathbf{z}_*)$  and  $p_{\Theta'}(f_* | \mathbf{y}, \mathbf{z}', \mathbf{z}_*)$  as in (19) are fitted to training data  $\mathcal{D}$  and  $\mathcal{D}'$  composed of the same observation vector  $\mathbf{y}$  but different (iid) latent variables sampled from their posterior densities  $\mathbf{z}, \mathbf{z}' \sim \prod_{i=1}^n p_{\Phi}(z_i | y_i, s_i)$  given by (29). The marginal posterior predictive density  $p_{\Theta,\Phi}(f_* | \mathbf{y}, \mathbf{S})$  from (28) is trained by marginalising over the training  $\mathbf{z}$  and test  $\mathbf{z}_*$  latent variables. In PPLS-BO, the acquisition function  $a_{ei}(\cdot)$  is fitted to the mean  $\hat{\mu}(\bar{z}_*)$  and standard deviation  $\hat{\sigma}(\bar{z}_*)$  obtained from the marginal predictive posterior, and a new sample  $\bar{z}_{k+1}$  is proposed.

### 3.2 PPLS-BO algorithm

We present the PPLS-BO algorithm, which sequentially combines the PPLS model and GP surrogate for adaptive sampling in reduced dimension. At each adaptive sampling iteration, the PPLS model is fitted to obtain the posterior probability densities  $p_{\Phi}(Z|Y, S)$  and  $p_{\Phi}(z_*|Y, S)$  given by (29) and (30) respectively. For each pair of latent variables  $Z$  and test latent variables  $z_*$  drawn from their respective densities, a GP posterior predictive probability density  $p_{\Theta^{(i)}}(f_*^{(i)}|y^{(i)}, Z, z_*)$  given by (19) is computed. The hyperparameters of the PPLS model  $\Phi$  and each GP surrogate  $\Theta^{(i)}$  are learned independently by minimising the KL divergence using the EM algorithm (Appendix C) and maximising the log marginal likelihood (20) respectively (Figure 4).

The mean (31) and variance (32) of the marginal posterior predictive density  $p_{\Theta^{(i)}, \Phi}(f_*^{(i)}|Y, S)$  given by (28), is computed by MC sampling from the latent variable posterior densities and the GP posterior predictive density. This predictive mean and variance are inputs to the acquisition function (22), which is subsequently maximised to propose mean test latent variables  $\bar{z}$ . The mean test latent variables exist in the bounded domain

$$D_{\bar{z}}^f = \left\{ \bar{z} \in \mathbb{R}^{d_z} \mid W\bar{z} \in D_s \right\}, \quad (35)$$

such that the mean of the probability density  $p_{\zeta}(s|\bar{z})$  given by (4a) is contained within the original design variable domain  $D_s$  given in (1). Since the latent variables are a linear combination of the design variables, the feasible subdomain  $D_{\bar{z}}^f$  represents a convex hull. This may be contained within a minimum bounding box  $D_{\bar{z}}$  that represents the union of the feasible subdomain  $D_{\bar{z}}^f$  and an infeasible region  $D_{\bar{z}}^{nf}$  that exists outside the original design variable domain, see Figure 5.

Each mean latent variable  $\bar{z}$  proposed at each adaptive sampling iteration is mapped to a corresponding conditional density  $p_{\zeta}(s|\bar{z})$  from which design variables  $s$  are sampled. The conditional density has a diagonal covariance matrix  $\hat{\Sigma}_s$  with each entry being the variance, which represents the error in reconstructing each design variable in the vector  $s$  from the low-rank approximation  $s \approx W\bar{z}$ . Entries of the covariance matrix  $\hat{\Sigma}_s$  become small when the linear subspace basis  $W$  is aligned with the axes of the original design space, i.e. where the reconstruction error is low. Conversely, the entries of the covariance matrix become large when the reconstruction error is large. Instead of neglecting exploration along axes of the original design space misaligned with the linear subspace basis, we sample randomly along these axes. Where the latent variable dimension  $d_z$  is specified incorrectly (for example, is too small), locations that do not exist within

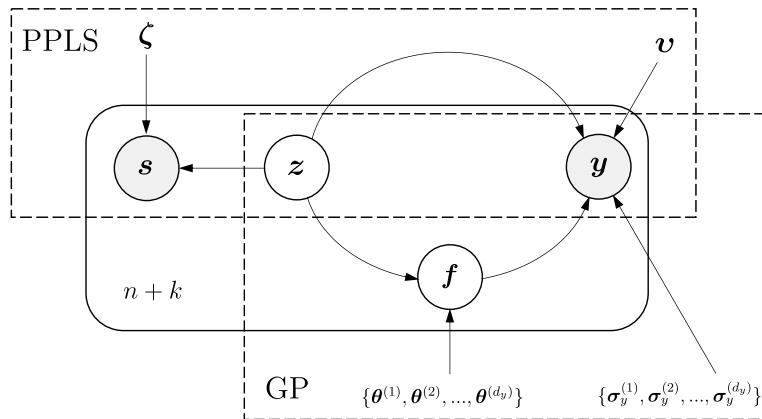


Figure 4. Graphical model for the  $k^{\text{th}}$  adaptive sampling iteration of PPLS-BO with a training data set  $\mathcal{D}$  of size  $n + k$  with (random) design variables  $s$  and observations  $y = (y^{(1)} \ y^{(2)} \ \dots \ y^{(d_y)})$ . Unobserved latent variables  $z$  and hyperparameters  $\zeta = \{W, \hat{\Sigma}_s\}$  and  $v = \{Q, \hat{\Sigma}_y\}$  are learned using PPLS independent from the GP surrogates with target output variables  $f = (f^{(1)} \ f^{(2)} \ \dots \ f^{(d_y)})$  and GP hyperparameters  $\{\theta^{(1)}, \theta^{(2)}, \dots, \theta^{(d_y)}\}$  and  $\{\sigma_y^{(1)}, \sigma_y^{(2)}, \dots, \sigma_y^{(d_y)}\}$ .

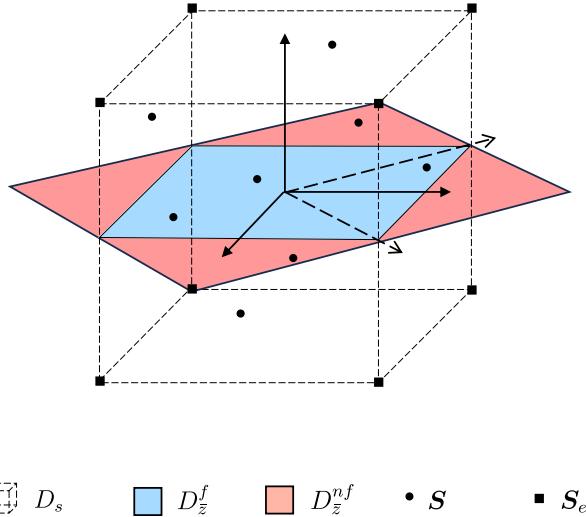


Figure 5. Schematic of the mean latent variable domain  $D_{\bar{z}}$  composed of the union between the feasible subdomain  $D_{\bar{z}}^f$  contained within the design variable domain  $D_s$ , and the infeasible subdomain  $D_{\bar{z}}^{nf}$  which extends beyond the design variable domain, where  $S$  are the design variables and  $S_e$  are the extreme design variables found at the vertices of the hyperplane describing the design variable domain.

the linear subspace are still explored by sampling randomly from the conditional density  $p_{\zeta}(s|\bar{z})$ . For accurate low-rank approximations, sampling from the conditional density does not adversely affect the ability for adaptive sampling to converge towards the global minimum.

For each design variable vector  $s$  proposed at each adaptive sampling iteration, the expensive-to-evaluate computational model (for example, a PDE) is solved to obtain an output vector  $y$  consisting of objective and constraint function evaluations pertaining to each QOI. Design variable  $S$  and output  $Y$  matrices are augmented with the proposed design variable vector  $s$  and output vector  $y$  respectively. The adaptive sampling iterations repeat until a computational budget  $n_k$  is exceeded or convergence criteria is satisfied, as summarised in Algorithm 2.

## 4 EXAMPLES

We demonstrate the computational advantages of the proposed PPLS-BO algorithm on three numerical examples of increasing complexity. We compare our PPLS-BO algorithm with its deterministic counterpart PLS-BO, see [42] and Appendix A, and classical BO. To begin, an illustrative example visually demonstrates the robustness of our approach to latent variable dimension misspecification. Next, a cantilever beam example shows the computational advantages of using PPLS-BO. Finally, we demonstrate the use of PPLS-BO on a complex manufacturing example that is otherwise difficult to optimise using classical methods. In all examples, the acquisition function is maximised using genetic algorithms (i.e. NSGA-II) with tuned optimisation parameters.

### 4.1 Illustrative example

For the design optimisation problem (1), we consider the multi-modal objective function given by

$$J(s) = \frac{1}{9} (6s_1^2 + 3) \sin(9s_1^2 + 1) \cos(6s_2^2 + 2) + \frac{1}{1000} \sum_{i=3}^{20} s_i, \quad (36)$$

**Algorithm 2** PPLS-BO

---

**Data:**  $S, Y$  ▷ mean centred and normalised

**Input:** adaptive sampling budget  $n_k$ , EM budget  $n_t$ , MC sampling budget  $n_l$ ,  
linear subspace dimension  $d_z$

Initialise  $S_0 \leftarrow S, Y_0 \leftarrow Y$

**for**  $k \in \{0, 1, \dots, n_k - 1\}$  **do**

- Fit PPLS model  $\Phi_k \leftarrow \text{EM}(S_k, Y_k, n_t, d_z)$  ▷ Algorithm 3
- Compute  $p_{\Phi_k}(Z_k|Y_k, S_k)$  and  $p_{\Phi_k}(z_*|Y, S)$  ▷ posterior densities (29),(30)
- for**  $l \in \{0, 1, \dots, n_l - 1\}$  **do**

  - Sample  $Z_k \sim p_{\Phi_k}(Z_k|Y_k, S_k)$  and  $z_* \sim p_{\Phi_k}(z_*|Y_k, S_k)$
  - Sample  $(f_*^{(i)})_l \sim p_{\Theta_k^{(i)}}(f_*^{(i)}|y_k^{(i)}, Z_k, z_*)$  ▷ GP posterior density (19)
  - Compute  $\hat{\mu}^{(i)}(\bar{z}_*)$  and  $\hat{\sigma}^{(i)}(\bar{z}_*)$  using samples  $\{(f_*^{(i)})_1, (f_*^{(i)})_2, \dots, (f_*^{(i)})_{n_l}\}$  ▷ marginal density (33)
  - Define latent variable domain  $D_{\bar{z}}^f$  ▷ approximated using (35)
  - Propose latent variable  $\bar{z}_{k+1} \leftarrow \arg \max_{\bar{z} \in D_{\bar{z}}^f} a_c(\hat{\mu}_k^{(i)}(\bar{z}), \hat{\sigma}_k^{(i)}(\bar{z}))$  ▷ maximise acquisition function (26)
  - Sample design parameter  $s_{k+1} \sim p_{\zeta_k}(s_{k+1}|\bar{z}_{k+1})$  ▷ posterior density (4a)
  - Solve for observation  $y_{k+1}^{(i)}, y_{k+1} \leftarrow (y_{k+1}^{(1)}, y_{k+1}^{(2)}, \dots, y_{k+1}^{(d_y)})^\top$
  - Augment data  $S_{k+1} \leftarrow (S_k^\top, s_{k+1})^\top, Y_{k+1} \leftarrow (Y_k^\top, y_{k+1})^\top$

**Result:** global minimum  $(y^{(1)})^* \leftarrow \min \{y_1^{(1)}, y_2^{(1)}, \dots, y_{n+n_k}^{(1)}\}$

---

with the constraint

$$H(s) = \frac{3}{4} - s_1 - s_2 - \frac{1}{1000} \sum_{i=3}^{20} s_i, \quad (37)$$

which are both functions of 20 design variables  $s$  ( $d_s = 20$ ) with an intrinsic dimension of two, i.e.  $d_z = 2$ . The goal is to compute the optimum design variables  $s^*$  that minimise the objective  $J(s)$  subject to the constraint  $H(s) \leq 0$ . The design variables are sampled from the domain  $D_s = \{s \in \mathbb{R}^{20} | 0 \leq s_i \leq 1, i = 1, 2, \dots, 20\}$ . The training data set  $\mathcal{D}$  is initialised with  $n = 24$  samples, comparing samples initialised randomly using Latin hypercube sampling (LHS) with orthogonal samples generated using a Plackett–Burman design (PBD). For a two-level PBD, low and high levels of 0 and 1 for each design variable are sampled, respectively.

For training data initialised using PBD, both PPLS-BO and PLS-BO correctly discover the intrinsic dimensionality, as shown by the large components in the first and second rows of the initial basis  $W_0$  demonstrating that  $s_1$  and  $s_2$  are the most influential design variables (Table I). The first and second diagonal entries of the initial design variable covariance matrix  $(\Sigma_s)_0$  computed from the PPLS model show that the reconstruction error for the first  $s_1$  and second  $s_2$  design variable is relatively small. However, when initialising the training data using LHS, both PPLS-BO and PLS-BO algorithms are unable to correctly discover the intrinsic dimensionality. Consequently, this incurs a larger reconstruction error estimate in the PPLS model. Since LHS samples are not orthogonal, confounding between the effects of each design variable on the objective and constraint function variance is observed. Using the basis  $W_0$  computed from LHS samples would restrict exploration in the domain of the most influential design variables  $s_1$  and  $s_2$ . Consequently, we use orthogonal samples generated from a PBD.

To test the robustness of the PPLS-BO algorithm, we consider the case where the latent variables are misspecified to be of a single dimension ( $d_z = 1$ ). The optimum design variables  $s^*$  are computed using the constrained acquisition function (26) paired with the EI acquisition function (24) (with jitter parameter  $\xi = 0$ ) fitted to the objective function GP. The training data set  $\mathcal{D}$  is initialised using a PBD, with three additional space-filling data points sampled using LHS (to initialise the GP surrogate). An adaptive sampling budget of  $n_k = 10$  is used, with an EM budget  $n_t = 100$  and MC sampling budget  $n_l = 10^3$ .

Initially (at adaptive sampling iteration  $k = 0$ ), the basis  $\mathbf{W}_k$  of the linear subspace computed using PLS-BO and PPLS-BO are approximately identical (Figure 6a). However, the GP marginal posterior probability density  $\hat{p}(\mathbf{J}_* | \mathbf{Y}_k, \mathbf{S}_k)$  computed using PPLS-BO has a similar mean, but a significantly larger variance when compared to the GP posterior probability density  $p(\mathbf{J}_* | \mathbf{y}_k^{(1)})$  fitted using PLS-BO. This is due to the latent variable  $z$  being uncertain in PPLS-BO, which increases the total uncertainty when inferring the objective function  $J(\mathbf{s})$  (36). The proposed posterior probability density  $\hat{p}(J_{k+1} | \mathbf{Y}_k, \mathbf{S}_k)$  shows small and large variance in the prediction of the first  $s_1$  and second  $s_2$  design variables respectively, with this being correlated to the basis vectors (columns of  $\mathbf{W}_k$ ).

At the third adaptive sampling iteration ( $k = 2$ ), the basis  $\mathbf{W}_k$  slightly differs between the PPLS-BO and PLS-BO algorithms. The GP marginal posterior probability density  $\hat{p}(\mathbf{J}_* | \mathbf{Y}_k, \mathbf{S}_k)$  computed using PPLS-BO differs substantially from the GP posterior probability density  $p(\mathbf{J}_* | \mathbf{y}_k^{(1)})$  fitted using PLS-BO, both in the mean and variance prediction (Figure 6b). However, samples  $\bar{z}_{k+1}$  (PPLS-BO) and  $z_{k+1}$  (PLS-BO) are proposed at approximately the same location. Due to the change in basis, the variance has both increased and decreased in the first  $s_1$  and second  $s_2$  design variables respectively, when compared to the initialised models (at adaptive sampling iteration  $k = 0$ ). Contrary to PLS-BO, PPLS-BO proposes a probability density  $\hat{p}(J_{k+1} | \mathbf{Y}_k, \mathbf{S}_k)$  which estimates reconstruction error and permits further exploration. Consequently, the design variables  $s_{k+1}$  proposed using the PPLS-BO algorithm are close to the global minimum of  $J(\mathbf{s}^*) = -0.817$  at  $\mathbf{s}^* = (0.642 \ 0.858 \ 0 \ 0 \cdots 0)^\top$ , a location that would otherwise not be proposed if using the deterministic counterpart PLS-BO.

PPLS-BO and PLS-BO algorithms both converge to different solutions, as demonstrated at the ninth adaptive sampling iteration ( $k = 8$ ) (Figure 6c). The estimate of reconstruction error using PPLS-BO enhances exploration, enabling the global minimum to be located. The posterior probability densities computed using the PPLS-BO and PLS-BO algorithms differ substantially. For PPLS-BO, similar values of the mean latent variable  $\bar{z}$  yield substantially different values of the objective function  $J(\mathbf{s})$ , resulting in a large estimate of the observation standard deviation, where  $\sigma_y = 0.153$ . Conversely, the latent variable  $z$  is deterministic in the PLS-BO model, resulting in a minimal observation standard deviation, where  $\sigma_y = 0.002$ .

As illustrated in Figure 7, if using the correct latent variable dimension  $d_z = 2$ , both the PPLS-BO and PLS-BO algorithms propose solutions in close proximity to the global minimum, which when using adaptive sampling budget of  $n_k = 10$ , is not located when using classical BO due to its practical limitations. PPLS-BO predicts a solution of  $J(\mathbf{s}^*) = -0.777$  at  $\mathbf{s}^* = (0.633 \ 0.882 \ 0.500 \ 0.500 \cdots 0.500)^\top$  and PLS-BO predicts a solution of  $J(\mathbf{s}^*) = -0.790$  at  $\mathbf{s}^* = (0.653 \ 0.842 \ 0.500 \ 0.500 \cdots 0.500)^\top$ . The variance in the first  $s_1$  and second  $s_2$  design

Table I. Illustrative example showing for LHS and PBD data initialisation methods, the linear subspace basis  $\mathbf{W}_0$  and PPLS design variable covariance matrix  $(\hat{\Sigma}_s)_0$  computed for the first adaptive sampling iteration ( $k = 0$ ) of the PPLS-BO and PLS-BO algorithms.

Data Initialisation	PLS-BO		PPLS-BO
	$\mathbf{W}_0$	$\mathbf{W}_0$	$(\hat{\Sigma}_s)_0$
LHS	$\begin{pmatrix} 0.567 & 0.021 \\ 0.518 & 0.294 \\ \vdots & \vdots \end{pmatrix}$	$\begin{pmatrix} 0.564 & 0.093 \\ 0.516 & 0.302 \\ \vdots & \vdots \end{pmatrix}$	$\begin{pmatrix} 0.051 & 0 & \dots \\ 0 & 0.056 & \dots \\ \vdots & \vdots & \ddots \end{pmatrix}$
PBD	$\begin{pmatrix} 0.925 & -0.380 \\ 0.381 & 0.925 \\ \vdots & \vdots \end{pmatrix}$	$\begin{pmatrix} 0.925 & -0.379 \\ 0.381 & 0.926 \\ \vdots & \vdots \end{pmatrix}$	$\begin{pmatrix} 0.013 & 0 & \dots \\ 0 & 0.037 & \dots \\ \vdots & \vdots & \ddots \end{pmatrix}$

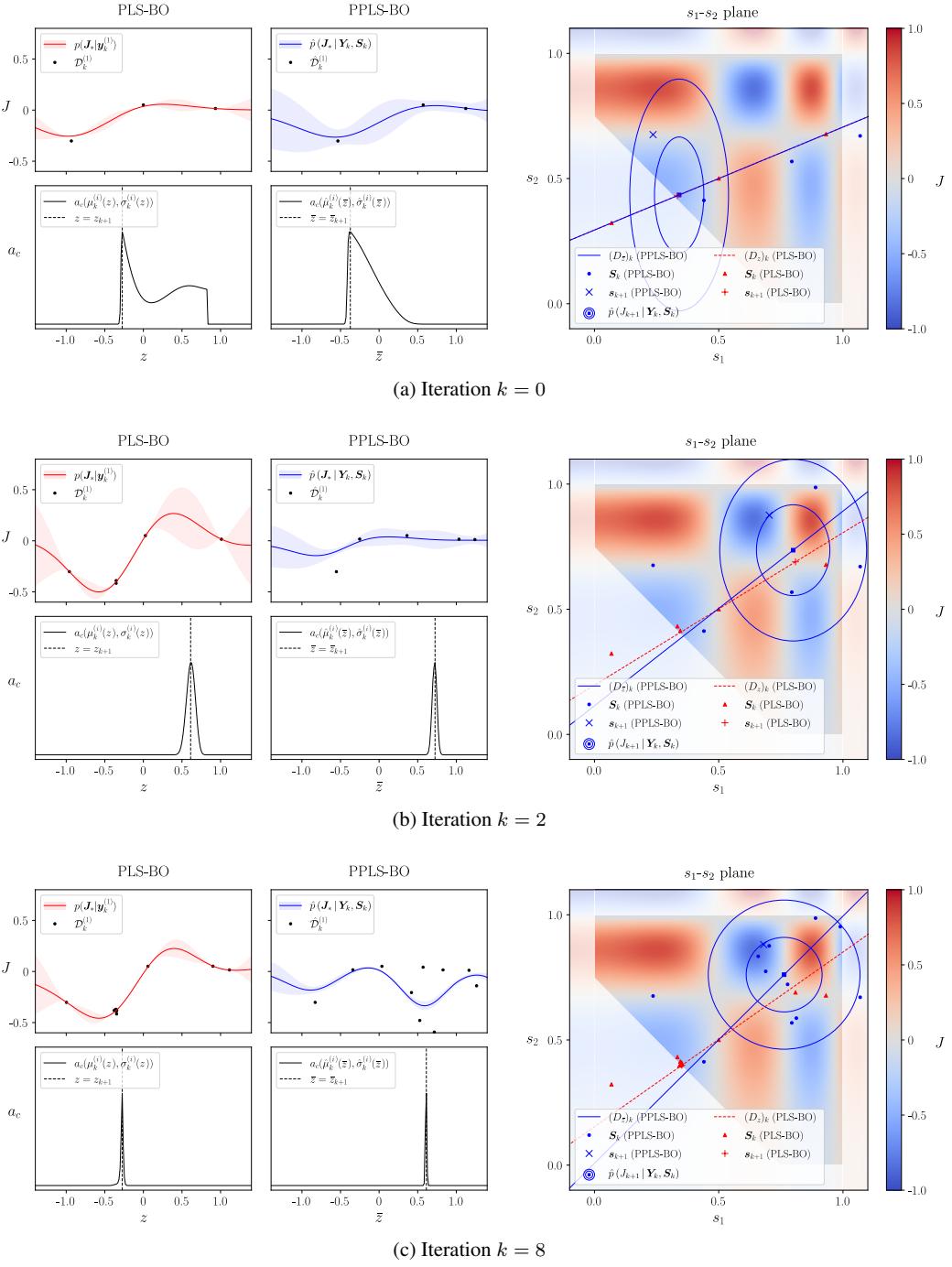


Figure 6. Illustrative example. Solution field of the objective function  $J(\mathbf{s})$  showing the PLS-BO posterior predictive density  $p(\mathbf{J}_*|\mathbf{y}_k^{(1)})$  fitted to data  $\mathcal{D}_k^{(1)} = \{(z_i, y_i^{(1)}) | i = 1, 2, \dots, n+k\}$  and acquisition function  $a_c(\mu_k^{(i)}(z), \sigma_k^{(i)}(z))$  with proposed sample  $z_{k+1}$ . This is compared to the PPLS-BO marginal posterior predictive density  $\hat{p}(\mathbf{J}_*|\mathbf{Y}_k, \mathbf{S}_k)$  showing the mean training data points  $\hat{\mathcal{D}}_k^{(1)} = \{((\hat{\mu}_z)_i, y_i^{(1)}) | i = 1, 2, \dots, n+k\}$  with acquisition function  $a_c(\hat{\mu}_k^{(i)}(\bar{z}), \hat{\sigma}_k^{(i)}(\bar{z}))$  and proposed sample  $\bar{z}_{k+1}$ . Both models are projected onto the  $s_1 s_2$  plane showing the proposed design variable  $s_{k+1}$  and corresponding marginal posterior probability density  $\hat{p}(J_{k+1}|\mathbf{Y}_k, \mathbf{S}_k)$  for adaptive sampling iterations (a)  $k = 0$ , (b)  $k = 2$ , and (c)  $k = 8$ .

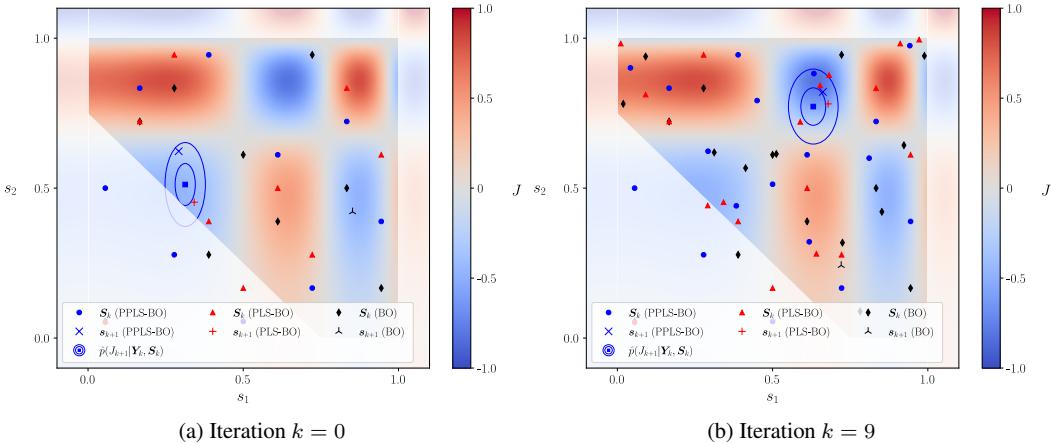


Figure 7. Illustrative example. Solution field of the objective function  $J(s)$  with design variable data  $S_k$  and the sample  $s_{k+1}$  proposed using PPLS-BO, PLS-BO, and classical BO with  $d_2 = 2$ . The PPLS-BO model yields a probability density  $\hat{p}(J_{k+1}|Y_k, S_k)$  conditioned on design variables  $S_k$  and observations  $Y_k$ , from which to propose a subsequent sample  $s_{k+1}$ , as demonstrated using adaptive sampling at (a) iteration  $k = 0$  and (b) iteration  $k = 9$ .

variable as deduced from the proposed probability density  $\hat{p}(J_{k+1} | \mathbf{Y}_k, S_k)$ , is reduced when compared to the case where the latent variable dimension is misspecified, i.e. when  $d_z = 1$ .

This illustrative example demonstrates that PPLS-BO can be used to solve global minimisation problems that are impractical to solve when using classical BO. Furthermore, PPLS-BO adds additional flexibility such that the adaptive sampling is more robust to misspecification in the latent variable dimension  $d_z$  when compared to the deterministic PLS-BO.

## 4.2 Cantilever beam

Consider the cantilever beam depicted in Figure 8 with a domain  $\Omega$  discretised over spatial coordinates  $x \in \mathbb{R}^3$ . The deformation measured by the displacement vector field,  $\mathbf{g}(x, s) \in \mathbb{R}^3$  is governed by the equations of (linear) elasticity, using the plane stress deformation assumption. The cantilever beam has an outer boundary  $\partial\Omega_o$  and an end boundary  $\partial\Omega_e$  where the load  $P$  is applied. The Dirichlet boundary condition is applied along the left edge  $\partial\Omega_D$ , with the Neumann boundary condition applied along the remaining boundary  $\partial\Omega_N$ , where  $\partial\Omega_N = \partial\Omega_o \cup \partial\Omega_e$ .

The geometry is parameterised by five design variables  $\mathbf{s} = (s_1 \ s_2 \ \dots \ s_5)^\top$ , where  $s_1$  and  $s_2$  control the length of each segment of the beam which has a total length  $L = 500$ , and  $s_3$ ,  $s_4$ , and  $s_5$  control the depth of each segment. The body force equates to self-weight, which is constant over the geometry domain. The surface traction is zero along the outer edge  $\partial\Omega_o$  and the resultant load  $P$  is applied vertically along the end boundary  $\partial\Omega_e$ . The first two design variables  $s_i \in \{s_i \in \mathbb{R} \mid 100 \leq s_i \leq 200\}$  for  $i \in \{1, 2\}$  and the remaining design variables  $s_i \in \{s_i \in \mathbb{R} \mid 20 \leq s_i \leq 70\}$  for  $i \in \{3, 4, 5\}$  are sampled from the domain  $D_s$ . In this example, we consider two "toy" objective functions  $J^{(j)}(\mathbf{s})$ ,  $j \in \{1, 2\}$  each with different behavior. The goal is to compute the design variables that minimise the objective function

$$J^{(j)}(\mathbf{s}) = J^{(0)}(\mathbf{s}) + \sum_{i=3}^5 J^{(j)}(s_i), \quad (38)$$

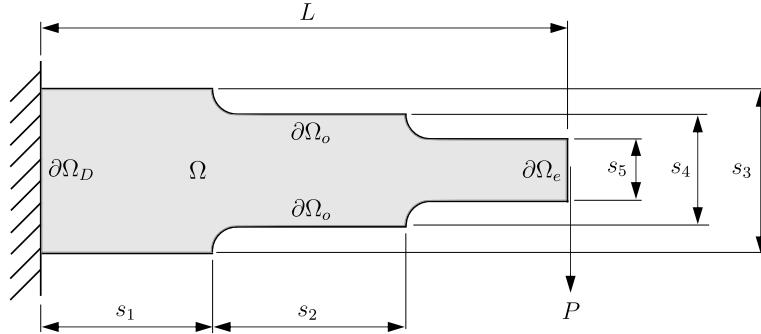


Figure 8. Cantilever beam schematic showing the domain  $\Omega$  with Dirichlet boundary condition  $\partial\Omega_D$  applied on the left edge, outer boundary  $\partial\Omega_o$ , the end boundary  $\partial\Omega_e$  where the load  $P$  is applied, and design variables

$s = (s_1 s_2 \cdots s_5)^\top$  of the cantilever beam with total length  $L$ .

where

$$J^{(0)}(s) = 0.000108(s_1s_3 + s_2s_4 + s_5(500 - s_1 - s_2)), \quad (39a)$$

$$J^{(1)}(s_i) = s_i(0.0963 - 0.0450f_l(s_i - 30) + 0.0662f_l(s_i - 40) + 0.0313f_l(s_i - 50)), \quad (39b)$$

$$J^{(2)}(s_i) = 0.0513s_i + 1.38\cos^2(0.15s_i), \quad (39c)$$

and

$$f_l(s_i) = \frac{1}{1 + e^{-100s_i}}, \quad (40)$$

subject to the **displacement constraint**

$$H(s) = \max_{x \in \Omega} (\|u(x, s)\|_2) - u_0, \quad (41)$$

where  $u_0 = 2$  is the displacement limit. The function  $J^{(0)}(s)$  is approximately proportional to the volume of the domain  $\Omega$  and  $J^{(j)}(s_i)$  ( $j \in \{1, 2\}$ ) is an additional contribution inducing both step for  $j = 1$  and periodic for  $j = 2$  behaviours (Figure 9). Although this is a "toy" example, these types of objective functions could represent the cost of different manufacturing processes where for example, the step function would represent certain supply chain constraints where it is relatively inexpensive to source components of specific sizes.

The **constraint function**  $H(s)$  is sampled using the FE model shown in Figure 10. The cantilever beam is modelled as a thin plate with a thickness of 5, using the plane stress assumption. A Young's modulus of  $2 \times 10^5$  and Poisson's ratio of 0.3 are used.

The adaptive sampling convergence to target solutions of 7.18 and 7.44 (as defined by the number of iterations required to propose a solution less than the target solution) for the step  $J^{(1)}(s)$  and periodic  $J^{(2)}(s)$  objective functions respectively for PPLS-BO is compared to PLS-BO and classical BO. The training data is initialised using  $n = 8$  orthogonal (PBD) samples. The two-level PBD uses design variables  $s$  with levels at their lower and upper limits. Convergence rates for both the UCB (23) and EI (24) constrained acquisition functions are compared. The convergence rates are computed across 10 runs for a mean and standard deviation. For the UCB acquisition function, an adaptive exploration parameter of  $\gamma = 0.2d_z \ln(2(k+1))$  is used, and for the EI acquisition function, a jitter parameter of  $\xi = 0$  is used. An adaptive sampling budget of  $n_k = 100$  is used in all algorithms, with an EM budget of  $n_t = 100$  and MC sampling budget  $n_l = 10^3$  used in the PPLS-BO algorithm. A latent variable dimension of three ( $d_z = 3$ ) is selected using cross-validation.

For the step objective function  $J^{(1)}(s)$ , the convergence rates of PPLS-BO are on average between 2.3 and 3.2 times faster than classical BO. Convergence rates decrease for PLS-BO, which is between 1.5 and 1.7 times faster than classical BO. The convergence rates also exhibit less variance for PPLS-BO than for PLS-BO. The relative improvement is consistent across all

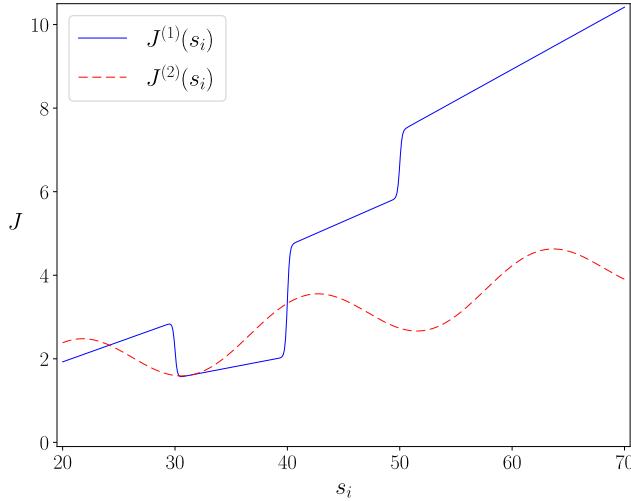


Figure 9. Cantilever beam. Step and periodic objective functions  $J^{(1)}(s_i)$  and periodic  $J^{(2)}(s_i)$ , respectively.

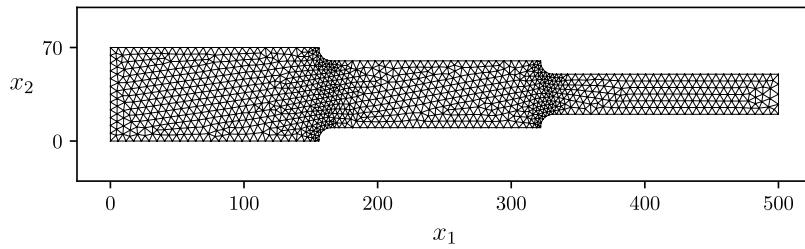


Figure 10. Cantilever beam. FE mesh of an example geometry containing 2212 linear triangle elements, discretised over spatial coordinates  $\mathbf{x} = (x_1 \ x_2 \ x_3)^\top$  according to the plane stress assumption.

data initialisation methods (where convergence is achieved) and acquisition functions. Although faster convergence is achieved on average when using PPLS-BO compared to PLS-BO, the variance in the convergence rates is high in all cases since the adaptive sampling is statistical in nature (Table II). An optimum solution of  $J^{(1)}(\mathbf{s}^*) = 6.8$  is obtained with design variables  $\mathbf{s}^* = (129 \ 200 \ 32.0 \ 32.1 \ 32.8)^\top$ .

For the periodic objective function  $J^{(2)}(\mathbf{s})$ , convergence rates of PPLS-BO are on average 5.7 to 6.2 times faster than classical BO. Convergence rates decrease for PLS-BO, which is between 4.4 and 5.3 times faster than classical BO. An optimum solution of  $J^{(2)}(\mathbf{s}^*) = 6.6$  is obtained with design variables  $\mathbf{s}^* = (133 \ 166 \ 31.6 \ 32.3 \ 29.6)^\top$ .

Significantly more improvement in convergence rates between PPLS-BO, PLS-BO, and classical BO is observed for the periodic objective function  $J^{(2)}(\mathbf{s})$  when compared to the step objective function  $J^{(1)}(\mathbf{s})$ . The gradient of the objective function within close proximity of the global minimum, is much smaller over a larger region of the design variable domain  $D_s$  for the step objective function than for the periodic objective function. In this case, the global minimum can be more easily located using classical BO. Furthermore, GPs make smoothness assumptions based on the covariance function (17), which may poorly emulate the step objective function.

Since PPLS-BO proposes a probability density that design variables  $\mathbf{s}$  are sampled from at each adaptive sampling iteration (as opposed to proposing a deterministic value with PLS-BO), this facilitates additional exploration, which leads to improvement in the rates of convergence. Furthermore, the basis  $\mathbf{W}$  computed using PPLS-BO has a higher degree of sparsity in the linear

combination of design variables  $\mathbf{s}$ , which further improves convergence by focusing exploration on the most influential design variables.

The properties of the PPLS-BO algorithm enhance exploration of the design variable domain  $D_s$  when compared to PLS-BO, which is advantageous when the latent variable dimension  $d_z$  is incorrectly specified. In the case of PLS-BO, the optimum solution for both the step  $J^{(1)}(\mathbf{s})$  and periodic  $J^{(2)}(\mathbf{s})$  objective functions is well approximated with a linear subspace dimension greater than three. However for PPLS-BO, the optimum solution is well approximated but with a linear subspace dimension of two. Consequently, PPLS-BO is more robust to misspecification of the linear subspace dimension  $d_z$  when compared to PLS-BO (Figure 11).

#### 4.3 Welded assembly

Consider the welded assembly shown in Figure 12 with its deformation governed by the elasticity equations. The geometry represents a complex engineering component for a manufacturing application. The geometry domain  $\Omega$  of the welded assembly is composed of a thin-walled beam  $\Omega_b$  and two geometrically identical plates  $\Omega_{p_1}$  and  $\Omega_{p_2}$ , such that  $\Omega = \Omega_b \cup \Omega_{p_1} \cup \Omega_{p_2}$ . All boundary faces of the thin-walled beam are denoted  $\partial\Omega_b$  excluding the left boundary face  $\partial\Omega_D$ , which is where the Dirichlet boundary condition is applied. The geometry of the forward-most plate  $\Omega_{p_1}$  for example, consists of through-thickness face boundaries  $\partial\Omega_{e_1}$ , flat-area face boundaries  $\partial\Omega_{p_1}$ , and the circular hole face boundaries  $\partial\Omega_{c_1}$  where half of the total load  $P = 10^4$  is applied. The second plate is defined similarly to the first. The boundary  $\partial\Omega_N$  where the Neumann boundary condition is

Table II. Cantilever beam. Mean and standard deviation (in parenthesis) of the number of iterations  $k$  required to satisfy convergence criteria averaged over 10 runs using PBD initialised data with UCB and EI constrained acquisition functions for the cost models  $J^{(j)}(\mathbf{s})$ .

$J^{(j)}(\mathbf{s})$	Acquisition Function	PPLS-BO	PLS-BO	BO
$J^{(1)}(\mathbf{s})$	UCB	11.4 (6.8)	18.5 (8.3)	31.9 (14.2)
	EI	13.4 (7.3)	20.4 (11.9)	30.2 (13.0)
$J^{(2)}(\mathbf{s})$	UCB	9.5 (4.7)	13.4 (7.7)	58.7 (26.0)
	EI	10.1 (2.7)	10.7 (6.5)	57.7 (21.5)

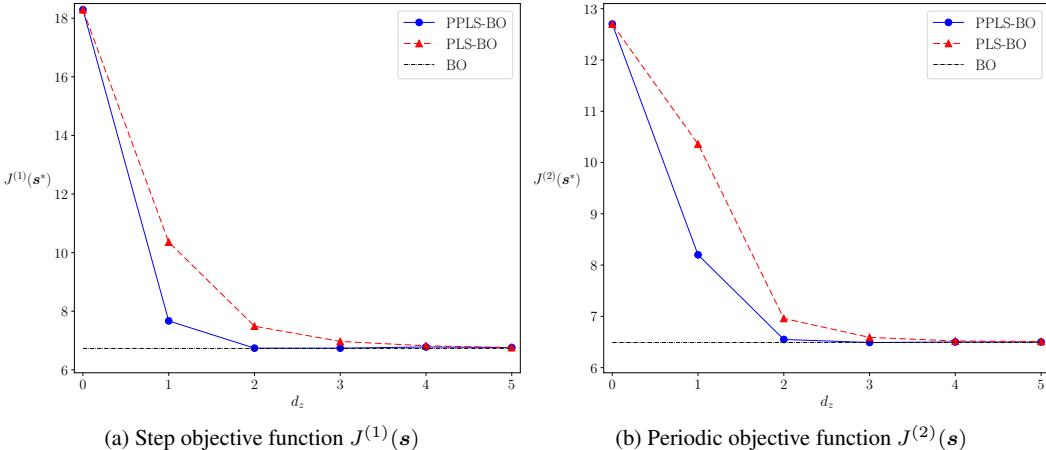


Figure 11. Cantilever beam. Optimum solution  $J^{(j)}(\mathbf{s})$  ( $j \in \{1, 2\}$ ) as a function of linear subspace dimension  $d_z$  comparing PPLS-BO, PLS-BO and the classical BO benchmark solutions for the (a) step objective function  $J^{(1)}(\mathbf{s})$ , and (b) periodic objective function  $J^{(2)}(\mathbf{s})$ .

applied, is defined as the union of all faces of the geometry excluding the left face boundary of the beam  $\partial\Omega_D$ .

The domain of the welded assembly  $\Omega$  is parameterised by 20 design variables  $s = (s_1 \ s_2 \ \dots \ s_{20})^\top$ . Design variables  $s_i$  for  $i \in \{1, 2, \dots, 7\}$  control the shape of the plates, design variables  $s_i$  for  $i \in \{8, 9, 10\}$  control the shape of the thin-walled beam, which has a fixed length  $L = 500$ , design variables  $s_i$  for  $i \in \{11, 12, \dots, 15\}$  control the position of the welds, and design variables  $s_i$  for  $i \in \{16, 17, \dots, 20\}$  control the length of the welds, which have a total combined length  $l_w$ . All welds are positioned along the through-thickness faces  $\partial\Omega_{e_1}$  and  $\partial\Omega_{e_2}$  of each plate and intersect the beam faces  $\partial\Omega_b$ , excluding the weld of length  $s_{16}$  which intersects the flat-area faces  $\partial\Omega_{p_1}$  and  $\partial\Omega_{p_2}$  and the beam face  $\partial\Omega_b$  (Figure 13). The body force is constant over the geometry domain and is equal to self-weight. The surface traction is zero on all boundaries  $\partial\Omega_N \setminus (\partial\Omega_{c_1} \cup \partial\Omega_{c_2})$  and yields half the resultant applied load along each of the circular hole boundaries. The design domain  $D_s$  is defined by box-constraints for each design variable  $s_i \in \{s_i \in \mathbb{R} \mid \bar{s}_i^{(l)} \leq s_i \leq \bar{s}_i^{(u)}\}$  where  $i \in \{1, 2, \dots, 20\}$ , and  $\bar{s}_i^{(l)}$  and  $\bar{s}_i^{(u)}$  are lower and upper limits on the design variables respectively (Table III). In this example, the goal is to minimise manufacturing cost

$$J(s) = J^{(0)}(s) + J^{(1)}(s) + J^{(2)}(s), \quad (42)$$

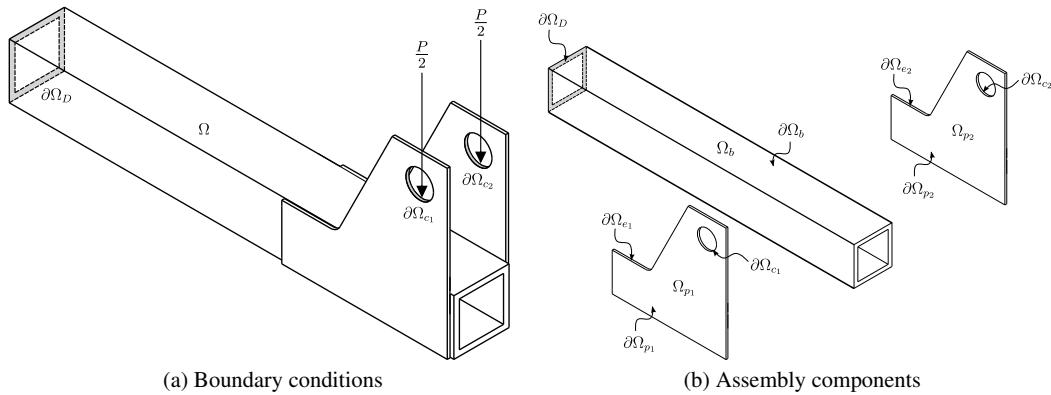


Figure 12. Welded assembly. Schematic showing (a) the geometry domain  $\Omega$  with left face boundary of the beam  $\partial\Omega_D$  where the Dirichlet boundary condition is applied, and the circular hole face boundaries  $\partial\Omega_{c_1}$  and  $\partial\Omega_{c_2}$  where the load  $P$  is applied. The (b) assembly components show the beam geometry  $\Omega_b$  with boundaries  $\partial\Omega_b$  and plates  $\Omega_{p_1}$  and  $\Omega_{p_2}$  with flat-area face boundaries  $\partial\Omega_{p_1}$  and  $\partial\Omega_{p_2}$ , through-thickness edge face boundaries  $\partial\Omega_{e_1}$  and  $\partial\Omega_{e_2}$ , and circular hole face boundaries  $\partial\Omega_{c_1}$  and  $\partial\Omega_{c_2}$ .

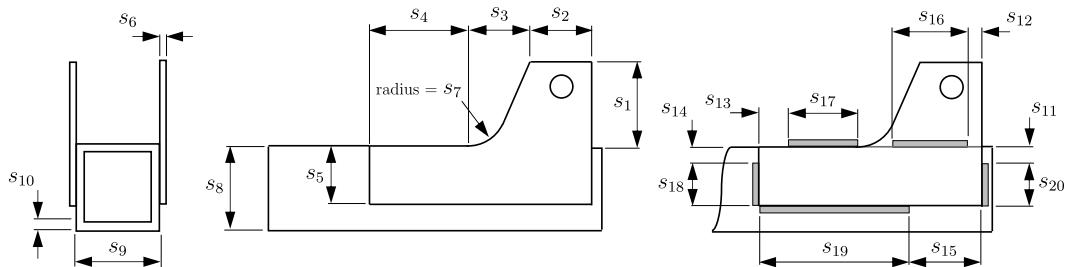


Figure 13. Welded assembly. Schematic showing the design variables  $s_i$ , where  $i \in \{1, 2, \dots, 20\}$ .

Table III. Welded assembly. Upper and lower limits for the box-constrained design variables  $s_i \in \{s_i \in \mathbb{R} \mid \bar{s}_i^{(l)} \leq s_i \leq \bar{s}_i^{(u)}\}$  where  $i \in \{1, 2, \dots, 20\}$ .

	$s_1$	$s_2$	$s_3$	$s_4$	$s_5$	$s_6$	$s_7$	$s_8$	$s_9$	$s_{10}$	$s_{11}$	$s_{12}$	$s_{13}$	$s_{14}$	$s_{15}$	$s_{16}$	$s_{17}$	$s_{18}$	$s_{19}$	$s_{20}$
$\bar{s}_i^{(l)}$	90	45	25	20	30	3	5	75	50	3	0	0	0	0	0	0	0	0	0	0
$\bar{s}_i^{(u)}$	150	80	75	100	75	6	25	150	125	8	75	155	100	75	255	155	100	75	255	75

which may be formulated in terms of material  $J^{(0)}(\mathbf{s})$ , welding  $J^{(1)}(\mathbf{s})$ , and cutting  $J^{(2)}(\mathbf{s})$  costs [50]. The material cost

$$J^{(0)}(\mathbf{s}) = c_m \rho \int_{\Omega} d\mathbf{x}, \quad (43)$$

is computed based on a material cost per unit mass  $c_m \in \mathbb{R}^+$ , material density  $\rho \in \mathbb{R}^+$ , and the volume of material used. The welding operation cost

$$J^{(1)}(\mathbf{s}) = c_w (t_w p_w(\mathbf{s}) l_w(\mathbf{s}) + t_{ws}), \quad (44)$$

is computed using a welding cost per unit time  $c_w \in \mathbb{R}^+$ , welding time per unit length  $t_w \in \mathbb{R}^+$ , welding complexity coefficient  $p_w(\mathbf{s})$  which accounts for additional time for more complex welds, total weld length  $l_w(\mathbf{s})$ , and welding setup time  $t_{ws} \in \mathbb{R}^+$ . For this example, the weld parameterised by length  $s_{19}$  is more complex as it is an overhead weld so is assumed to take twice the welding time as the other welds, where

$$p_w(\mathbf{s}) = 1 + \frac{s_{19}}{l_w(\mathbf{s})}, \quad l_w(\mathbf{s}) = \sum_{i=16}^{20} s_i. \quad (45)$$

The cutting operation cost

$$J^{(2)}(\mathbf{s}) = c_c (p_c t_c(\mathbf{s}) l_c(\mathbf{s}) + t_{cs}), \quad (46)$$

is computed using the cutting cost per unit time  $c_c \in \mathbb{R}^+$ , a cutting factor  $p_c \in \mathbb{R}^+$  which scales for complexity (assumed constant in this example), cutting time per unit length of cut  $t_c(\mathbf{s})$ , total length of the cut  $l_c(\mathbf{s})$ , and cutting setup time  $t_{cs} \in \mathbb{R}^+$ . The cutting time per unit length  $t_c(\mathbf{s})$  can be assumed to be proportional to the thickness [51], and the cutting length is based on the length of the outer edge of the identical plates, where

$$t_c(\mathbf{s}) = \frac{1}{100} s_6, \quad l_c(\mathbf{s}) = \frac{2}{s_6} \int_{\partial\Omega_{e_1} \cup \partial\Omega_{c_1}} d\mathbf{x}. \quad (47)$$

We assume a material cost per unit mass of  $c_m = 1.5$ , a welding cost per unit time of  $c_w = 5.5 \times 10^{-3}$ , a welding time per unit length of  $t_w = 0.32$ , a welding setup time of  $t_{ws} = 18$ , a cutting cost per unit time of  $c_c = 5 \times 10^{-3}$ , a cutting factor of  $p_c = 1$ , and a cutting setup time of  $t_{cs} = 180$ . The manufacturing cost  $J(\mathbf{s})$  is minimised subject to the displacement constraint

$$H^{(1)}(\mathbf{s}) = \max_{x \in \Omega} (||\mathbf{u}(\mathbf{x}, \mathbf{s})||_2) - u_0, \quad (48)$$

where  $u_0 = 2$  is the limit on the total displacement, and the stress constraint

$$H^{(2)}(\mathbf{s}) = \max_{x \in \Omega_b'} (\sigma_b(\mathbf{x}, \mathbf{s})) - \sigma_{b0}, \quad (49)$$

where  $\sigma_b(\mathbf{x}, \mathbf{s})$  denotes the bending stress in the beam, and  $\sigma_{b0} = 125$  is the limit on the bending stress. An additional constraint is also imposed on the weld length

$$H^{(3)}(\mathbf{s}) = l_{w0} - l_w(\mathbf{s}), \quad (50)$$

such that the total length of weld  $l_w(s)$  (45) must exceed the minimum weld length  $l_{w0} = 100$ .

The constraint functions  $H^{(1)}(s)$  and  $H^{(2)}(s)$  are sampled using an FE model, and are evaluated from any location in the beam geometry domain  $\Omega_b$  excluding any location within four elements of the left boundary  $\partial\Omega_D$  (this revised domain is denoted  $\Omega'_b$ ). A Young's modulus of  $2 \times 10^5$  and Poisson's ratio of 0.3 are used. Since the geometry exhibits symmetry, only half of the geometry domain  $\Omega$  is used to generate the FE model (reducing computational expense).

This is an example of a complex engineering component that is difficult to optimise with classical BO, due to the number of design variables  $s$ . We compare the optimum manufacturing cost  $J(s^*)$  obtained using PPLS-BO and PLS-BO, within a fixed adaptive sampling budget  $n_k$  for various latent variable dimensions  $d_z$ . The data is initialised using PBD with  $n = 24$  samples, and adaptive sampling is performed using the EI acquisition function (24) with constraints (26) and a jitter parameter  $\xi = 0$ . The two-level PDB uses the design variables  $s$  with levels at their lower and upper limits (Table III). An adaptive sampling budget of  $n_k = 100$  is used with an EM budget of  $n_t = 100$  and MC sampling budget  $n_l = 10^3$  used in the PPLS-BO algorithm.

Both the PPLS-BO and PLS-BO algorithms obtain a more optimum manufacturing cost  $J(s^*)$  when compared to classical BO since classical BO scales poorly to problems with more than 10 design variables (Figure 14). The welded assembly objective  $J(s)$  and constraint functions  $H^{(1)}(s)$ ,  $H^{(2)}(s)$ , and  $H^{(3)}(s)$ , have a low intrinsic dimension since the material cost, displacement, and stress are predominantly influenced by the beam geometry. Consequently, a significant improvement from the mean geometry (computed when  $d_z = 0$ ) is obtained when using PPLS-BO and PLS-BO algorithms with three-dimensional latent variables where  $d_z = 3$ . Additional improvement in the optimum manufacturing cost  $J(s^*)$  is obtained by further increasing the latent variable dimension  $d_z$ . The optimum welded assembly geometry exhibits a large flexural rigidity to satisfy the displacement and stress constraints while minimising manufacturing cost. For the optimum design variables  $s^*$ , only the stress constraint is active. The weld lengths and positions minimally affect the manufacturing cost  $J(s)$ , displacement  $H^{(1)}(s)$ , and stress  $H^{(2)}(s)$  constraint; consequently, the basis  $W$  promotes minimal exploration of the design variables relating to the welds, and their optimum values are approximately equal to the mean. The optimised component and its deflection and von Mises stress isocontours are shown in Figure 15.

An optimum latent variable dimension is obtained when the optimum manufacturing cost begins to diverge for any given adaptive sampling budget  $n_k$ . The optimum manufacturing cost begins to diverge with a latent variable dimension greater than six, i.e.  $d_z > 6$ . As more dimensions are added for the same adaptive sampling budget  $n_k$ , the degree of exploration along the basis vectors

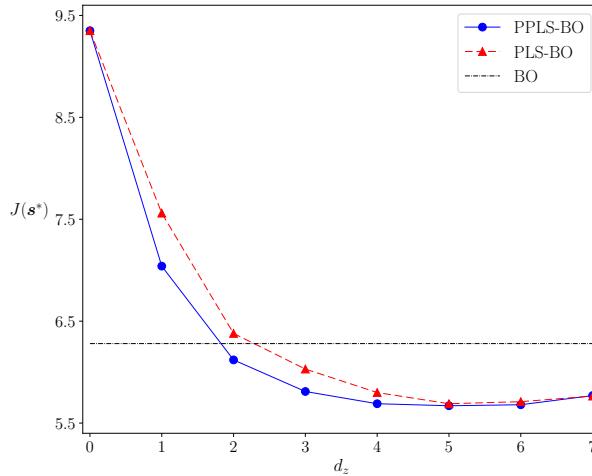


Figure 14. Welded assembly. Optimum manufacturing cost  $J(s^*)$  as a function of the latent variable dimension  $d_z$  comparing PPLS-BO, PLS-BO and the classical BO benchmark solutions with a predetermined adaptive sampling budget of  $n_k = 100$ .

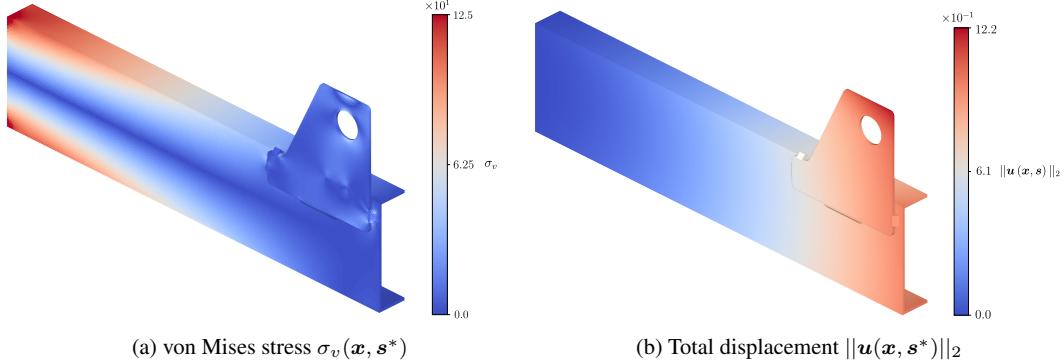


Figure 15. Welded beam. (a) von Mises stress  $\sigma_v(\mathbf{x}, \mathbf{s}^*)$  and (b) total displacement  $\|\mathbf{u}(\mathbf{x}, \mathbf{s}^*)\|_2$ , as computed using the FE method using symmetry. The geometry is meshed using second-order tetrahedron elements, with a minimum of two elements through-thickness with the optimum design variables  $\mathbf{s}^*$  computed using PPLS-BO with a linear subspace dimension of six  $d_z = 6$ .

(columns of the basis  $\mathbf{W}$ ) composed of a linear combination of the most influential design variables  $\mathbf{s}$  is reduced. Consequently, a trade-off exists between the latent variable dimension  $d_z$  and the adaptive sampling budget  $n_k$ . As with the other examples considered in this paper, PPLS-BO can obtain similar optimum solutions but with a smaller latent variable dimension  $d_z$  when compared to PLS-BO.

The welded assembly example demonstrates how MVL is used to extend the application of BO to complex engineering components or systems that would otherwise be impractical to optimise with classical BO due to the large number of design variables. These complex engineering systems may have multiple constraints that must be considered when forming a low-dimensional basis, i.e.  $\mathbf{W}$ , composed of linear combinations of the most influential design variables  $\mathbf{s}$ , describing an intrinsic dimensionality  $d_z$  pertaining to the entire problem, not just the objective function  $J(\mathbf{s})$ . This ensures the linear subspace crosses the boundary between active and inactive regions of the constraint function, where the optimum solution exists.

## 5 CONCLUSION

We introduced the novel PPLS-BO algorithm, which combines probabilistic partial least squares with Gaussian process surrogates to perform Bayesian optimisation in reduced dimension. Both the PPLS model and the GP surrogate are trained sequentially. PPLS assumes a generative model for generating output data from unobserved low-dimensional random latent variables. The posterior probability density of the latent variables is computed by minimising the KL divergence using expectation maximisation. We marginalise the GP posterior predictive density over the random latent variables to obtain a mean and variance as inputs to a BO acquisition function. The acquisition function is then maximised to obtain a more optimal mean latent variable vector and subsequently sample the corresponding design variable from a conditional density, augmenting the training data. Overall, the PPLS-BO algorithm improves convergence rates compared to its deterministic counterparts, PLS-BO and classical BO. Furthermore, optimum solutions comparable to those computed using PLS-BO can be obtained with PPLS-BO but with fewer latent variable dimensions. This makes PPLS-BO more robust to incorrectly chosen latent variable dimensions and improves computational tractability in high-dimensional problems.

However, when emulating functions that lack smoothness, the advantages of using PPLS-BO over PLS-BO and classical BO diminish. This is caused by shortcomings of the smoothness assumptions of GP covariance functions; therefore, exploring other surrogates such as random forests [52] for problems involving discrete data or discontinuities is a promising direction for further research.

Furthermore, other novel surrogates derived using variational Bayes [53] could be further extended to adaptive sampling problems using a similar framework introduced in this paper. The expressivity of the GP surrogate may also be further enhanced by parameterising its mean and covariance with neural networks [54, 55]. The independent learning of the low-dimensional subspace and the GP surrogate could enable further extensions, such as the learning of Riemannian manifolds, to account for more complex geometric constraints [56].

## APPENDIX A. PARTIAL LEAST SQUARES

PLS is a family of MVL methods that attempt to obtain an orthonormal basis from one set of data that is highly correlated to an orthonormal basis obtained from the other data set [57]. PLS maximises the covariance between pairs of latent variables  $\mathbf{z} \in \mathbb{R}^{d_z}$  and  $\mathbf{v} \in \mathbb{R}^{d_v}$ . The latent variables

$$\mathbf{z} = \mathbf{W}^\top \mathbf{s}, \quad (\text{A.1a})$$

$$\mathbf{v} = \mathbf{Q}^\top \mathbf{y}, \quad (\text{A.1b})$$

are low-dimensional linear combinations of the design variable vector  $\mathbf{s}$  and output vector  $\mathbf{y}$  respectively. The bases  $\mathbf{W}$  and  $\mathbf{Q}$  are elements of the Stiefel manifolds  $V_k(\mathbb{R}^{d_s}) = \{\mathbf{W} \in \mathbb{R}^{d_s \times d_z} \mid \mathbf{W}^\top \mathbf{W} = \mathbf{I}\}$  and  $V_k(\mathbb{R}^{d_y}) = \{\mathbf{Q} \in \mathbb{R}^{d_y \times d_v} \mid \mathbf{Q}^\top \mathbf{Q} = \mathbf{I}\}$  respectively. PLS maximised the covariance of design variable  $\mathbf{S}$  and output data  $\mathbf{Y}$  mean-centred matrices projected onto their respective bases, where

$$\begin{aligned} & \max_{\mathbf{W}, \mathbf{Q}} \quad \text{Tr}(\mathbf{W}^\top \Sigma_{sy} \mathbf{Q}) \\ & \text{s.t.} \quad \mathbf{W}^\top \mathbf{W} = \mathbf{Q}^\top \mathbf{Q} = \mathbf{I}, \end{aligned} \quad (\text{A.2})$$

and  $\Sigma_{sy} = \mathbf{S}^\top \mathbf{Y}$ . This can be solved using the non-linear iterative partial least-squares (NIPALS) algorithm, which iteratively solves an eigenvalue problem for each basis vector [58]. For each latent variable vector  $\mathbf{z}$ , the corresponding design variables

$$\mathbf{s} = (\mathbf{W}^\dagger)^\top \mathbf{z} + (\mathbf{I} - (\mathbf{W}^\dagger)^\top \mathbf{W}^\top) \mathbf{e} \quad (\text{A.3})$$

can be computed, where  $\mathbf{W}^\dagger \in \mathbb{R}^{d_z \times d_s}$  is the Moore-Penrose inverse of the basis  $\mathbf{W}$ , and  $\mathbf{e} \in \mathbb{R}^{d_s}$  is the reconstruction error incurred by the low-rank approximation.

The PLS-BO algorithm for adaptive sampling is similar to classical BO (Algorithm 1), except the PLS model is solved using NIPALS at each iteration, and the low-rank approximation is computed before solving the computational model to collect the corresponding observation vector.

## APPENDIX B. DERIVATION OF MARGINAL LIKELIHOOD COVARIANCE

The entries of the marginal likelihood covariance matrix  $\hat{\Sigma}_{ys}$  given by (7), can be derived using the properties of covariance. The input covariance is given by

$$\text{cov}(\mathbf{s}, \mathbf{s}) = \text{cov}(\mathbf{W}\mathbf{z} + \hat{\epsilon}_s, \mathbf{W}\mathbf{z} + \hat{\epsilon}_s) = \text{cov}(\hat{\epsilon}_s, \hat{\epsilon}_s) + 2\mathbf{W} \text{cov}(\mathbf{z}, \hat{\epsilon}_s) + \mathbf{W} \text{cov}(\mathbf{z}, \mathbf{z}) \mathbf{W}^\top = \hat{\Sigma}_s + \mathbf{W}\mathbf{W}^\top, \quad (\text{B.1})$$

the output covariance is given by

$$\text{cov}(\mathbf{y}, \mathbf{y}) = \text{cov}(\mathbf{Q}\mathbf{z} + \hat{\epsilon}_y, \mathbf{Q}\mathbf{z} + \hat{\epsilon}_y) = \text{cov}(\hat{\epsilon}_y, \hat{\epsilon}_y) + 2\mathbf{Q} \text{cov}(\mathbf{z}, \hat{\epsilon}_y) + \mathbf{Q} \text{cov}(\mathbf{z}, \mathbf{z}) \mathbf{Q}^\top = \hat{\Sigma}_y + \mathbf{Q}\mathbf{Q}^\top, \quad (\text{B.2})$$

and the cross covariance is given by

$$\text{cov}(\mathbf{s}, \mathbf{y}) = \text{cov}(\mathbf{W}\mathbf{z} + \hat{\epsilon}_s, \mathbf{Q}\mathbf{z} + \hat{\epsilon}_y) = \mathbf{W} \text{cov}(\mathbf{z}, \mathbf{z}) \mathbf{Q}^\top + \mathbf{W} \text{cov}(\mathbf{z}, \hat{\epsilon}_s) + \text{cov}(\mathbf{z}, \hat{\epsilon}_y) \mathbf{Q}^\top + \text{cov}(\hat{\epsilon}_s, \hat{\epsilon}_y) = \mathbf{W}\mathbf{Q}^\top. \quad (\text{B.3})$$

### APPENDIX C. EXPECTATION MAXIMISATION ALGORITHM

The EM algorithm is an iterative procedure based on alternating coordinate descent used to compute the optimum model hyperparameters  $\Phi^*$  [44]. First we maximise with respect to the trial density  $q(\mathbf{z})$  and then the PPLS hyperparameters  $\Phi$ . Following from (8) and (13), for fixed PPLS hyperparameters  $\Phi_t$  the optimum trial density is given by

$$\begin{aligned} q_{t+1}(\mathbf{z}) &= \arg \max_{q(\mathbf{z})} \int q(\mathbf{z}) \ln \left( \frac{p_{\Phi_t}(\mathbf{z}|\mathbf{y}, \mathbf{s}) p_{\Phi_t}(\mathbf{y}, \mathbf{s})}{q(\mathbf{z})} \right) d\mathbf{z} \\ &= \arg \max_{q(\mathbf{z})} \left( -D_{KL}(q(\mathbf{z}) || p_{\Phi_t}(\mathbf{z}|\mathbf{y}, \mathbf{s})) + \ln p_{\Phi_t}(\mathbf{y}, \mathbf{s}) \right) \\ &= p_{\Phi_t}(\mathbf{z}|\mathbf{y}, \mathbf{s}). \end{aligned} \quad (\text{C.1})$$

By fixing the trial density, the PPLS model hyperparameters can be computed by maximising

$$\Phi_{t+1} = \arg \max_{\Phi} \mathbb{E}_{q_{t+1}(\mathbf{z})} (\ln p_{\Phi}(\mathbf{y}, \mathbf{s}|\mathbf{z})). \quad (\text{C.2})$$

The first expectation term in the ELBO (14) given by

$$\begin{aligned} \mathbb{E}_{p_{\Phi_t}(\mathbf{z}|\mathbf{y}, \mathbf{s})} (\ln p_{\Phi_{t+1}}(\mathbf{y}, \mathbf{s}|\mathbf{z})) &= -\frac{1}{2} \left( d_s \ln(2\pi) + \ln |(\hat{\Sigma}_s)_{t+1}| + \mathbf{s}^\top (\hat{\Sigma}_s)_{t+1}^{-1} \mathbf{s} - 2 \mathbb{E}(\mathbf{z})^\top \mathbf{W}_{t+1}^\top (\hat{\Sigma}_s)_{t+1}^{-1} \mathbf{s} + \right. \\ &\quad \text{Tr} \left( \mathbb{E}(\mathbf{z}\mathbf{z}^\top) \mathbf{W}_{t+1}^\top (\hat{\Sigma}_s)_{t+1}^{-1} \mathbf{W}_{t+1} \right) + d_y \ln(2\pi) + \ln |(\hat{\Sigma}_y)_{t+1}| + \mathbf{y}^\top (\hat{\Sigma}_y)_{t+1}^{-1} \mathbf{y} - \\ &\quad \left. 2 \mathbb{E}(\mathbf{z})^\top \mathbf{Q}_{t+1}^\top (\hat{\Sigma}_y)_{t+1}^{-1} \mathbf{y} + \text{Tr} \left( \mathbb{E}(\mathbf{z}\mathbf{z}^\top) \mathbf{Q}_{t+1}^\top (\hat{\Sigma}_y)_{t+1}^{-1} \mathbf{Q}_{t+1} \right) \right). \end{aligned} \quad (\text{C.3})$$

is expanded using the likelihood  $p_{\Phi_{t+1}}(\mathbf{y}, \mathbf{s}|\mathbf{z}) = p_{v_{t+1}}(\mathbf{y}|\mathbf{z}) p_{\zeta_{t+1}}(\mathbf{s}|\mathbf{z})$ . In the E-step of the EM algorithm, the expectation terms

$$\mathbb{E}(\mathbf{z}) = (\hat{\mu}_z)_t, \quad \mathbb{E}(\mathbf{z}\mathbf{z}^\top) = \left( \hat{\Sigma}_z \right)_t + \mathbb{E}(\mathbf{z}) \mathbb{E}(\mathbf{z})^\top, \quad (\text{C.4})$$

are computed using the posterior probability density  $p_{\Phi_t}(\mathbf{z}|\mathbf{y}, \mathbf{s})$  for a known  $\Phi_t$  and laws of total variance. The expectation terms are pre-determined using PPLS model hyperparameters  $\Phi_t$  at the current time-step  $t$ .

In the M-step, the expectation term (C.3) is maximised to determine the PPLS model hyperparameters for the subsequent iteration-step  $\Phi_{t+1}$ , subject to bases  $\mathbf{W}_{t+1}$  and  $\mathbf{Q}_{t+1}$  being orthogonal. This can be written as an unconstrained optimisation problem with the Lagrangian

$$\begin{aligned} \mathcal{L}(\Phi_{t+1}) &= -\frac{1}{2} \left( d_s \ln(2\pi) + \ln |(\hat{\Sigma}_s)_{t+1}| + \mathbf{s}^\top (\hat{\Sigma}_s)_{t+1}^{-1} \mathbf{s} - 2 \mathbb{E}(\mathbf{z})^\top \mathbf{W}_{t+1}^\top (\hat{\Sigma}_s)_{t+1}^{-1} \mathbf{s} + \right. \\ &\quad \text{Tr} \left( \mathbb{E}(\mathbf{z}\mathbf{z}^\top) \mathbf{W}_{t+1}^\top (\hat{\Sigma}_s)_{t+1}^{-1} \mathbf{W}_{t+1} \right) + d_y \ln(2\pi) + \ln |(\hat{\Sigma}_y)_{t+1}| + \mathbf{y}^\top (\hat{\Sigma}_y)_{t+1}^{-1} \mathbf{y} - \\ &\quad \left. 2 \mathbb{E}(\mathbf{z})^\top \mathbf{Q}_{t+1}^\top (\hat{\Sigma}_y)_{t+1}^{-1} \mathbf{y} + \text{Tr} \left( \mathbb{E}(\mathbf{z}\mathbf{z}^\top) \mathbf{Q}_{t+1}^\top (\hat{\Sigma}_y)_{t+1}^{-1} \mathbf{Q}_{t+1} \right) \right) - \\ &\quad \text{Tr} ((\mathbf{W}_{t+1}^\top \mathbf{W}_{t+1} - \mathbf{I}) \boldsymbol{\Lambda}_s) - \text{Tr} ((\mathbf{Q}_{t+1}^\top \mathbf{Q}_{t+1} - \mathbf{I}) \boldsymbol{\Lambda}_y), \end{aligned} \quad (\text{C.5})$$

where  $\boldsymbol{\Lambda}_s \in \mathbb{R}^{d_s \times d_s}$  and  $\boldsymbol{\Lambda}_y \in \mathbb{R}^{d_y \times d_y}$  are diagonal matrices of Lagrange multiplier constants. Taking the derivative of the Lagrangian  $\mathcal{L}(\mathbf{W}_{t+1}, \mathbf{Q}_{t+1}, (\hat{\Sigma}_s)_{t+1}, (\hat{\Sigma}_y)_{t+1})$  with respect to  $\mathbf{W}_{t+1}$  and  $\mathbf{Q}_{t+1}$ , and equating to zero yields

$$\mathbf{W}_{t+1} = \mathbf{s} \mathbb{E}(\mathbf{z})^\top (\mathbb{E}(\mathbf{z}\mathbf{z}^\top) + \boldsymbol{\Lambda}_s)^{-1}, \quad (\text{C.6a})$$

$$\mathbf{Q}_{t+1} = \mathbf{y} \mathbb{E}(\mathbf{z})^\top (\mathbb{E}(\mathbf{z}\mathbf{z}^\top) + \boldsymbol{\Lambda}_y)^{-1}. \quad (\text{C.6b})$$

However, since the bases are orthogonal, the following identity can be derived

$$\mathbf{W}_{t+1}^T \mathbf{W}_{t+1} = \mathbf{I} = \left( (\mathbb{E}(zz^T) + \Lambda_s)^{-1} \right)^T \mathbb{E}(z)s^T s \mathbb{E}(z)^T (\mathbb{E}(zz^T) + \Lambda_s)^{-1}, \quad (\text{C.7a})$$

$$\mathbf{Q}_{t+1}^T \mathbf{Q}_{t+1} = \mathbf{I} = \left( (\mathbb{E}(zz^T) + \Lambda_y)^{-1} \right)^T \mathbb{E}(z)y^T y \mathbb{E}(z)^T (\mathbb{E}(zz^T) + \Lambda_y)^{-1}, \quad (\text{C.7b})$$

which can be decomposed using the Cholesky decomposition

$$\mathbf{L}_W \mathbf{L}_W^T = (\mathbb{E}(zz^T) + \Lambda_s)^T (\mathbb{E}(zz^T) + \Lambda_s) = \mathbb{E}(z)s^T s \mathbb{E}(z)^T, \quad (\text{C.8a})$$

$$\mathbf{L}_Q \mathbf{L}_Q^T = (\mathbb{E}(zz^T) + \Lambda_y)^T (\mathbb{E}(zz^T) + \Lambda_y) = \mathbb{E}(z)y^T y \mathbb{E}(z)^T. \quad (\text{C.8b})$$

The bases

$$\mathbf{W}_{t+1} = s \mathbb{E}(z)^T \mathbf{L}_W^{-1}, \quad (\text{C.9a})$$

$$\mathbf{Q}_{t+1} = y \mathbb{E}(z)^T \mathbf{L}_Q^{-1}, \quad (\text{C.9b})$$

can therefore be computed using the Cholesky decomposition (C.8a) and (C.8b) respectively.

Similarly, the PPLS covariance matrix hyperparameters  $(\hat{\Sigma}_s)_{t+1}$  and  $(\hat{\Sigma}_y)_{t+1}$  can be computed by taking the derivative of the Lagrangian  $\mathcal{L}(\mathbf{W}_{t+1}, \mathbf{Q}_{t+1}, (\hat{\Sigma}_s)_{t+1}, (\hat{\Sigma}_y)_{t+1})$  with respect to the covariance matrix hyperparameters  $(\hat{\Sigma}_s)_{t+1}$  and  $(\hat{\Sigma}_y)_{t+1}$ , and equating to zero, which yields

$$(\hat{\Sigma}_s)_{t+1} = \text{diag} (ss^T - 2\mathbf{W}_{t+1} \mathbb{E}(z)s^T + \mathbf{W}_{t+1} \mathbb{E}(zz^T) \mathbf{W}_{t+1}^T), \quad (\text{C.10a})$$

$$(\hat{\Sigma}_y)_{t+1} = \text{diag} (yy^T - 2\mathbf{Q}_{t+1} \mathbb{E}(z)y^T + \mathbf{Q}_{t+1} \mathbb{E}(zz^T) \mathbf{Q}_{t+1}^T). \quad (\text{C.10b})$$

The E and M steps can be repeated iteratively until the EM budget  $n_t$  (which is chosen such that the PPLS model hyperparameters  $\Phi$  converge) is exceeded. The initial bases  $\mathbf{W}_0$  and  $\mathbf{Q}_0$  can be set using QR decomposition [59], with the PPLS covariance matrix hyperparameters  $(\hat{\Sigma}_s)_0$  and  $(\hat{\Sigma}_y)_0$  set equal to the identity matrices (Algorithm 3).

---

### Algorithm 3 EM

---

**Data:**  $S, Y$  ▷ mean centred and normalised  
**Input:** budget  $n_t$ , linear subspace dimension  $d_z$   
 Initialise  $\mathbf{W}_0$  and  $\mathbf{Q}_0$  with QR decomposition,  $(\hat{\Sigma}_s)_0 \leftarrow \mathbf{I}, (\hat{\Sigma}_y)_0 \leftarrow \mathbf{I}$   
**for**  $t \in \{0, 1, \dots, n_t - 1\}$  **do**  
   Compute expectations  $\mathbb{E}(z), \mathbb{E}(zz^T)$  ▷ E-step (C.4)  
   Compute bases  $\mathbf{W}_{t+1}, \mathbf{Q}_{t+1}$ , and covariance  $(\hat{\Sigma}_s)_{t+1}, (\hat{\Sigma}_y)_{t+1}$  ▷ M-step (C.9a),(C.9b),(C.10a),(C.10b)  
**Result:**  $\Phi \leftarrow \{\mathbf{W}_{t+1}, \mathbf{Q}_{t+1}, (\hat{\Sigma}_s)_{t+1}, (\hat{\Sigma}_y)_{t+1}\}$

---

### REFERENCES

1. Cirak F., Scott M. J., Antonsson E. K., Ortiz M., Schröder P. Integrated modeling, finite-element analysis, and engineering design for thin-shell structures using subdivision *Computer-Aided Design*. 2002;34:137–148.
2. Hughes T. J. R., Cottrell J. A., Bazilevs Y. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement *Comput. Methods Appl. Mech. Eng.*. 2005;194:4135–4195.
3. Bandara K., Rüberg T., Cirak F. Shape optimisation with multiresolution subdivision surfaces and immersed finite elements *Comput. Methods Appl. Mech. Eng.*. 2016;300:510–539.
4. Lieu Q X, Lee J. Multiresolution topology optimization using isogeometric analysis *Int. J. Numer. Methods Eng.*. 2017;112:2025–2047.
5. Yin G, Xiao X, Cirak F. Topologically robust CAD model generation for structural optimisation *Comput. Methods Appl. Mech. Eng.*. 2020;369:113102.

6. Xiao X., Cirak F. Infill topology and shape optimization of lattice-skin structures *Int. J. Numer. Methods Eng.*. 2022;123:664–682.
7. Williams C K I, Rasmussen C E. *Gaussian processes for machine learning*. MIT Press 2006.
8. Forrester A, Sobester A, Keane A. *Engineering design via surrogate modelling: a practical guide*. John Wiley & Sons 2008.
9. Diaz De la O F A, Adhikari S. Gaussian process emulators for the stochastic finite element method *Int. J. Numer. Methods Eng.*. 2011;87:521–540.
10. Sakata S, Ashida F, Zako M. Structural optimization using Kriging approximation *Comput. Methods Appl. Mech. Engrg.*. 2003;192:923–939.
11. Xiong Y, Chen W, Apley D, Ding X. A non-stationary covariance-based Kriging method for metamodeling in engineering design *Int. J. Numer. Methods Eng.*. 2007;71:733–756.
12. Bessa M A, Pellegrino S. Design of ultra-thin shell structures in the stochastic post-buckling range using Bayesian machine learning and optimization *Int. J. Solids Struct.*. 2018;139:174–188.
13. Bengio Y, Delalleau O, Roux N. The Curse of Highly Variable Functions for Local Kernel Machines Advances in neural information processing systems 2005.
14. Jones D R, Schonlau M, Welch W J. Efficient global optimization of expensive black-box functions *J. Glob. Optim.*. 1998;13:455–492.
15. Fuhr J N, Fau A, Nackenhorst U. State-of-the-art and comparative review of adaptive sampling methods for kriging *Arch. Comput. Methods Eng.*. 2021;28:2689–2747.
16. Horst R, Pardalos P M. *Handbook of global optimization*;2. Springer 2013.
17. Frazier P I. A tutorial on Bayesian optimization *arXiv preprint arXiv:1807.02811*. 2018.
18. Wilson J, Hutter F, Deisenroth M. Maximizing acquisition functions for Bayesian optimization *Adv. Neural Inf. Process. Syst.*. 2018;31.
19. Mathern A, Steinoltz O S, Sjöberg A, et al. Multi-objective constrained Bayesian optimization for structural design *Struct. Multidiscip. Optim.*. 2021;63:689–701.
20. Morita Y, Rezaeiravesh S, Tabatabaei N, Vinuesa R, Fukagata K, Schlatter P. Applying Bayesian optimization with Gaussian process regression to computational fluid dynamics problems *J. of Comput. Phys.*. 2022;449:110788.
21. Sheikh H M, Callan T A, Hennessy K J, Marcus P S. Optimization of the shape of a hydrokinetic turbine’s draft tube and hub assembly using Design-by-Morphing with Bayesian optimization *Comput. Methods Appl. Mech. Engrg.*. 2022;401:115654.
22. Zhao J, Xie X, Xu X, Sun S. Multi-view learning overview: recent progress and new challenges *Inf. Fusion*. 2017;38:43–54.
23. Xu C, Tao D, Xu C. A survey on multi-view learning *arXiv preprint arXiv:1304.5634*. 2013.
24. Ghahramani Z. Probabilistic machine learning and artificial intelligence *Nature*. 2015;521:452–459.
25. el Bouhaddani S, Uh H-W, Hayward C, Jongbloed G, Houwing-Duistermaat J. Probabilistic partial least squares model: identifiability, estimation and application *J. Multivar. Anal.*. 2018;167:331–346.
26. Jordan M I, Ghahramani Z, Jaakkola T S, Saul L K. An introduction to variational methods for graphical models *Mach. Learn.*. 1999;37:183–233.
27. Berkooz G, Holmes P, Lumley J L. The proper orthogonal decomposition in the analysis of turbulent flows *Annu. Rev. Fluid. Mech.*. 1993;25:539–575.
28. Dulong J-L, Druesne F, Villon P. A model reduction approach for real-time part deformation with nonlinear mechanical behavior *Int. J. Interact. Des. Manuf.*. 2007;1:229–238.
29. Wang Z, Zoghi M, Hutter F, Matheson D, De Freitas N. Bayesian optimization in high dimensions via random embeddings Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI):1778–1784 2013.
30. Constantine P G, Dow E, Wang Q. Active subspace methods in theory and practice: applications to kriging surfaces *SIAM J. Sci. Comput.*. 2014;36:A1500–A1524.
31. Li J, Cai J, Qu K. Surrogate-based aerodynamic shape optimization with the active subspace method *Struct. Multidiscip. Optim.*. 2019;59:403–419.
32. Lam R R, Zahm O, Marzouk Y M, Willcox K E. Multifidelity dimension reduction via active subspaces *SIAM J. Sci. Comput.*. 2020;42:A929–A956.
33. Romor F, Tezzelle M, Mrosek M, Othmer C, Rozza G. Multi-fidelity data fusion through parameter space reduction with applications to automotive engineering *Int. J. Numer. Methods Eng.*. 2021.
34. Raponi E, Wang H, Bujny M, Boria S, Doerr C. High dimensional Bayesian optimization assisted by principal component analysis International Conference on Parallel Problem Solving from Nature:169–183 2020.
35. Gaudrie D, Le Riche R, Picheny V, Enaux B, Herbert V. Modeling and optimization with Gaussian processes in reduced eigenbases *Struct. Multidiscip. Optim.*. 2020;61:2343–2361.
36. Tripathy R, Bilionis I, Gonzalez M. Gaussian processes with built-in dimensionality reduction: applications to high-dimensional uncertainty propagation *J. Comput. Phys.*. 2016;321:191–223.
37. Tsilifis P, Pandita P, Ghosh S, Andreoli V, Vandepitte T, Wang L. Bayesian learning of orthogonal embeddings for multi-fidelity Gaussian processes *Comput. Methods Appl. Mech. Engrg.*. 2021;386:114147.
38. Song K, Tong T, Wu F, Zhang Z. A novel partial least squares weighting Gaussian process algorithm and its application to near infrared spectroscopy data mining problems *Anal. Methods*. 2012;4:1395–1400.
39. Bouhlel M A, Bartoli N, Otsmane A, Morlier J. Improving kriging surrogates of high-dimensional design models by partial least squares dimension reduction *Struct. Multidiscip. Optim.*. 2016;53:935–952.
40. Zhu L, Chen J. Energy efficiency evaluation and prediction of large-scale chemical plants using partial least squares analysis integrated with Gaussian process models *Energy Convers. Manag.*. 2019;195:690–700.
41. Zuhal L R, Faza G A, Palar P S, Liem R P. On dimensionality reduction via partial least squares for Kriging-based reliability analysis with active learning *Reliab. Eng. Syst. Saf.*. 2021;215:107848.

42. Bouhlel M A, Bartoli N, Regis R G, Otsmane A, Morlier J. Efficient global optimization for high-dimensional constrained problems by using the kriging models combined with the partial least squares method *Eng. Optim.*. 2018;50:2038–2053.
43. Bartoli N, Lefebvre T, Dubreuil S, et al. Adaptive modeling strategy for constrained global optimization with application to aerodynamic wing design *Aerosp. Sci. Technol.*. 2019;90:85–102.
44. Moon T K. The expectation-maximization algorithm *IEEE Signal Process. Mag.*. 1996;13:47–60.
45. Wilson J, Hutter F, Deisenroth M. Maximizing acquisition functions for Bayesian optimization Advances in Neural Information Processing Systems:9884–9895 2018.
46. Cox D D, John S. A statistical method for global optimization 1992 IEEE International Conference on Systems, Man, and Cybernetics:1241–1246 1992.
47. Auer P. Using confidence bounds for exploitation-exploration trade-offs *J. Mach. Learn. Res.*. 2002;3:397–422.
48. Gardner J, Kusner M, Zhixiang X, Weinberger K, Cunningham J. Bayesian optimization with inequality constraints International Conference on Machine Learning:937–945 2014.
49. Deb K, Pratap A, Agarwal S, Meyarivan T. A fast and elitist multiobjective genetic algorithm: NSGA-II *IEEE Trans. Evol. Comput.*. 2002;6:182–197.
50. Pavlović L, Krajnc A, Beg D. Cost function analysis in the structural optimization of steel frames *Struct. Multidiscip. Optim.*. 2004;28:286–295.
51. Madić M, Radovanović M, Nedic B, Gostimirović M. CO2 laser cutting cost estimation: mathematical model and application *Int. J. Laser. Sci.*. 2018;1:169–183.
52. Breiman L. Random forests *Mach. Learn.*. 2001;45:5–32.
53. Archbold T. A., Kazlauskaitė I., Cirak F.. Variational Bayesian surrogate modelling with application to robust design optimisation *Comput. Methods Appl. Mech. Eng.*. 2024;432:117423.
54. Rixner M, Koutsourelakis P-S. A probabilistic generative model for semi-supervised training of coarse-grained surrogates and enforcing physical constraints through virtual observables *J. Comput. Phys.*. 2021;434:110218.
55. Vadeboncoeur A, Akyıldız Ö D, Kazlauskaitė I, Girolami M, Cirak F. Fully probabilistic deep models for forward and inverse problems in parametric PDEs *J. Comput. Phys.*. 2023;491:112369.
56. Jaquier N, Rozo L. High-dimensional Bayesian optimization via nested Riemannian manifolds *Adv. Neural. Inf. Process. Syst.*. 2020;33:20939–20951.
57. Höskuldsson A. PLS regression methods *J. Chemom.*. 1988;2:211–228.
58. Wold H. Soft modelling by latent variables: the non-linear iterative partial least squares (NIPALS) approach *J. Appl. Probab.*. 1975;12:117–142.
59. Golub G H, Van Loan C F. *Matrix computations*. JHU Press 2013.