
Bayesian Optimization of Structural Properties of Fibre and Material of Matrix for Composite Plate for Maximum Buckling Load - Stage 4 Report

Roshan Suresh Kumar¹

¹ Department of Aerospace Engineering, Texas A&M University

e-mail address of corresponding author: roshan94@tamu.edu

Keywords: Composites, Buckling Analysis, Bayesian Optimization, Constrained Optimization

ABSTRACT

Since the discovery of composites, many industries have raced to determine their applicability and feasibility for different uses. The ability to achieve a wide range of mechanical, electrical and thermal properties has made composites quick favourites for material design specialists in various disciplines. Initially, the only way to determine the material properties of a certain composite was to manufacture a specimen for testing. This is very expensive and time consuming. However, with the advent of machine learning, it is now possible to simulate and predict the behaviour of various composites without the need to consider manufacturing feasibility. Since data driven models can surrogate expensive simulations and experiments, they can be used in an inverse manner to learn the composite composition for an optimal or the required performance. This research is focused on an effort to determine the fibre radius, fibre angle and matrix Young's Modulus for a composite plate with maximum buckling load. A multi-fidelity multi-information source constrained Bayesian Optimization approach was used to achieve this goal. The analysis of the problem from the technological and materials science standpoints is presented. A literature review is conducted to delve deeper into the relationship between the chosen design parameters and the buckling load. The models capturing the forward mapping between the design parameters and the objective are discussed in detail with tests to compare different data-driven surrogate models. The algorithm is described in detail and results on a test problem are presented and interpreted. Although the optimization algorithm could not be implemented on the true goal of this work, the conclusions and future work sections present the various insights gained and potential modifications to the approach employed here.

1 Introduction

The discovery of composites brought with it the possibility of manipulation of its material properties in order to either target or optimize requisite performance metric(s). The advent of artificial intelligence and machine learning techniques meant that materials scientists would no longer have to determine the behaviour of different composites by expensive simulations and experiments. Optimization techniques specifically tailored to the data-driven surrogates of expensive simulations and experiments enabled the efficient computation of the optimal design parameters. In this respect, the current research is aimed at the utilization of the information available from both expensive simulations and data-driven surrogate models to solve the inverse problem of determining the material properties to maximize a performance metric. **The solution neutral problem statement for this project is to determine the optimal structural and material properties for a composite plate with maximum buckling load.** A high-fidelity Finite Element model, a low-fidelity analytical model and a low-fidelity neural network are available for the computation of the buckling load. A constraint on the volume fraction is assumed, along with two lower fidelity models for the computation of the constraint metric. A multi-fidelity multi-information source constrained Bayesian Optimization technique is employed to combine the information of the models with varying fidelities to solve the optimization problem.

The progress of this project so far was recorded in three previous reports, each focused on a different stage of the project. The first report presented the definition of the project and its boundaries of applicability. The problem to be solved and the approach proposed to solve it was analyzed from the technological, materials and Olson' Framework perspective. The technological analysis of the problem described the optimization technique of choice and ex-

amined the deployment of surrogate models and the possible gains of doing so. The materials analysis of the problem looked at the material science interpretation of the solution of the problem. The problem analysis using Olson's Framework illustrated the relationships between the process parameters, structural parameters and the performance metric in question. Since the optimization problem only maps the structural properties to the performance metric, the process parameters were taken from the automated fibre placement process which provided a real world context to the problem being solved.

The second report picked up from where the materials science analysis in the first report left off. A literature survey was conducted to examine the relationship between each of the chosen design parameters (fibre radius, fibre angle and matrix Young's Modulus) and the performance metric (buckling load). Plots and tables from different works of research were presented to show the nature of the mapping between the design parameters and the objective. A discussion on the optimization techniques used for similar problems in previous works shed some light on the complexity of the optimization task. The knowledge of the design-to-objective relationships informed the brief discussion on the balance of exploration and exploitation of the design space knowledge accumulated by the optimization technique in this project.

The third report investigated the forward maps between the design parameters and the buckling load, both the knowledge-based analytical model and the data-driven machine learning based surrogate models. A dimensionality reduction of the model was carried out to examine the relationship between each individual design parameter and the computed buckling load. A qualitative comparison of the performance of the neural network (the second low-fidelity model of choice) against a linear regressor and a gaussian process regressor (GPR) was carried out. Since the GPR is an integral part of the multi-fidelity constrained Bayesian Optimization algorithm, a comparison of different kernels was also conducted to determine the ideal kernel for the mapping between the design parameters and the performance metric. The role of the GPR in the optimization algorithm was also discussed in brief,

This report is a culmination of the work done in the duration of this course and brings together the different elements discussed in the previous reports. In addition to the reiteration of the salient points presented in the previous reports, this report presents and discusses the details of the implementation of the optimization algorithm on a test problem. The attempt at the deployment of the optimization algorithm to solve the main problem of this project is also discussed and the many insights gained are provided.

The rest of the report is structured as follows. Section 2 and its subsections summarize the details of the first report. Section 3 and its subsections describe the findings in the second report. Section 4 and its subsections provide details

of the forward mapping and comparison results shown in the third report. Section 5 presents the details of the optimization algorithm and its implementation on the test problem and the main composite plate problem. Conclusions and Future Work sections summarize the results of the project and the new insights gained, which lead to the potential modifications of the salient aspects of the models and the optimization algorithm. Acknowledgements and References conclude the report.

2 Overarching Analysis of the Problem

The first report provides an overarching analysis of the problem. The bounds of the problem and the chosen design parameters were defined in that report.

The technological analysis of the problem provides a discussion about Bayesian Optimization and its evolution towards the optimization algorithm of choice in this project and presents the needs of the problem that must be addressed by the optimization algorithm of choice. This discussion, along with an illustration of the Bayesian Optimization algorithm, is presented in the next subsection.

2.1 Analysis of the Technological Problem

The inverse problem of solving for structure based on performance requirements is done computationally to allow for fast exploration of various tunable parameters in the material structure. This research also deals with solving the inverse problem computationally using Bayesian Optimization. Bayesian Optimization is one of the most ubiquitous optimization schemes used in literature to solve a deductive problem. Bayesian Optimization uses a bayesian network or representation of a model (functional or black box) to step towards the optimal design. The most common bayesian representation used in literature is the Gaussian Process Regressor. The bayesian network is employed since it allows for a natural representation of the uncertainty in the objective value. The Bayesian Optimization scheme uses a function to determine the next design point to sample. This is called the acquisition function and it represents an information metric of sorts. The design point that maximises this acquisition function is sampled in the next iteration. This process is discussed in detail in Snoek et. al. [1]. Shahriari et. al. [2] is a useful review paper that discusses Bayesian Optimization and its applications in detail.

The previous paragraph discusses the vanilla Bayesian Optimization. This framework has now evolved to incorporate multiple models of different fidelity mapping the design space to the objective space. The Bayesian Optimization framework now focuses on utilizing the information about the objective space available from various sources with lower fidelities to improve their ability to estimate the objective accurately for different regions in the design space using occasional evaluations from the high fidelity model. This framework finds particular use in situations where high

fidelity simulations or experiments take a long time and/or are too expensive to run, which places an implicit limit on the number of times they can be employed. In his tutorial on Bayesian Optimization, Frazier[3] discusses the various types of acquisition functions introduced in literature and transitions to their manner of deployment in the presence of multifidelity, multi-information sources and other specific scenarios.

Multifidelity multi-information source Bayesian Optimization has also found use in Material Design and Engineering following the advent of the Materials as Systems thinking. Ghoreishi et. al. (2018) [4] discusses the employment of Bayesian Optimization through multi-information source fusion for Dual-Phase steel design for maximum strength-normalized strain hardening rate. The work is discussed and presented in the context of ICME (Integrated Computation Materials Engineering). Another example illustrating the utilization multi-information source fusion through Bayesian Optimization as applied to Materials Design and Engineering is Ghoreishi et. al. (2019) [5]. The same problem of optimizing the design of dual-phase steel for maximum strength-normalized strain hardening rate is explored in this paper as well. However, the key differences between Ghoreishi et. al. (2018) [4] and Ghoreishi et. al. (2019) [5] can be summarized as follows. Ghoreishi et. al. (2018) [4] explores the correlations between each of the lower fidelity information sources and between the lower fidelity information sources and the ground truth through different cases of sampling of the design space. It also approaches the optimization problem from the perspective of adding design point evaluations using ground truth experiments and using them to improve the fused Gaussian Process. Ghoreishi et. al. (2019) [5] incorporates more lower fidelity models (six as opposed to three in Ghoreishi et. al. (2018) [4]) and focusses on improving the Gaussian Processes of the individual lower fidelity models by adding design point evaluations through the fused Gaussian Process. The high fidelity ground truth is merely used to construct the fused Gaussian Process in the first iteration.

The multi-information source fusion based Bayesian Optimization framework was updated to account for design constraints and presented in Ghoreishi, Allaire (2019) [6]. This framework is adopted for this research work and is illustrated in Figure 1. For the current design problem, the high fidelity ground truth is taken to be Finite Element Analysis and the low fidelity models are an analytical buckling differential equation based formulation and a Neural Network trained by a dataset generated using high fidelity evaluations.

It must be noted that the earlier intention of the project was to solve for the structural and material properties of a composite column for maximum buckling load. This was quickly substituted for a composite plate due to the faster computations in the Abaqus FEA software for composite plate as compared to a composite column, which meant

faster generation of the training data for the neural network. The details of the composite plate structure are shown here. The dimensions of the composite plate (10ft x 10ft), the number of fibres (5) and the relative positions of these fibres, shown in Figure 2. The thickness of the plate is 254mm.

However, the design parameters as described in the first report still remain the same. They are given in Equation 1

$$\mathbf{x} = [r_f \quad \theta_f \quad E_m]^T \quad (1)$$

where r_f is the fibre radius, θ_f is the fibre angle and E_m is the matrix Young's Modulus. It was decided that to limit the design search space of the problem to something reasonable, the Young's Modulus of the matrix (E_m) would have the following bounds (in GPa)

$$10 \leq E_m \leq 800$$

These bounds roughly encompass the Young's Modulus region for metals as shown in the Ashby Plot in Figure 3, taken from [7].

The number of fibres is fixed to five for simplicity, however theoretically the number of fibres can also be a candidate parameter for optimization. The objective to maximize is the buckling load, formally presented in Equation 2.

$$f(\mathbf{x}) = P_{cr} \quad (2)$$

In order to ensure that a reasonable solution is obtained, a constraint is incorporated into the optimization problem. These are shown in Equation 3 where v_f is the volume fraction. Volume fraction is defined as the ratio between the volume occupied by the fibre and the total volume.

$$v_f < 0.5 \quad (3)$$

In this case, the limit of the volume fraction was set to be 0.5. However this was subject to change, as shall be seen in Section 5.

A brief summary of the multifidelity multi-information source constrained Bayesian Optimization approach is provided here. As illustrated in Figure 1, the optimization algorithm starts with the generation of an initial set of design vectors. These design vectors can be generated using many different Design of Experiments methods like Orthogonal Arrays. The most common method employed for this purpose is Latin Hypercube Sampling. It provides a number of design vectors that cover as much of the design space as possible. These design vectors are now evaluated using the ground truth and the low fidelity models. The idea is to construct a Bayesian representation/surrogate of the low fidelity models (Gaussian Process in this case). The user-defined parameters in Figure 1 refer to the constants in the problem, namely the length, width and thickness of the composite plate (l , w and t respectively), relative positions of the fibres, fibre material E_f and the number of fibres. The prior distributions of the Gaussian Processes of the low fidelity

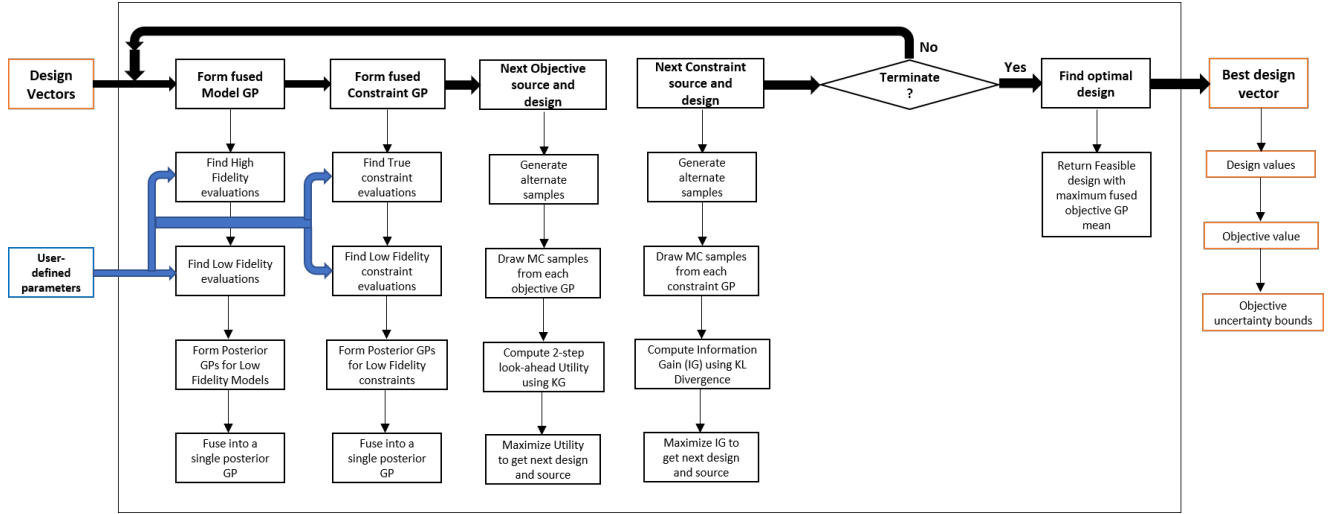


Figure 1: Flowchart illustrating the computational process that represents the approach to solve the design problem

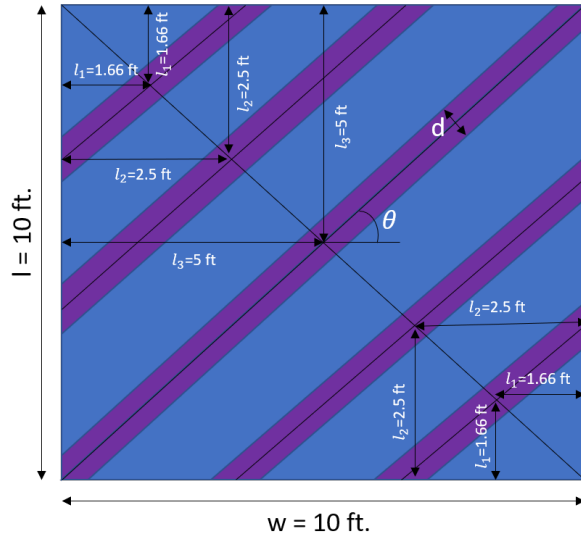


Figure 2: Front view of the composite plate showing matrix in blue and fibres in purple. The user defined parameters are plate length l , plate width w , relative positions of the fibres l_1, l_2 and l_3 and the physical design inputs are fibre diameter d and fibre orientation angle θ

models are assumed to be zero mean with a squared exponential kernel $k_i(\mathbf{x}, \mathbf{x})$.

$$f_{GP,i}(\mathbf{x}) = \mathcal{GP}(\mathbf{0}, k_i(\mathbf{x}, \mathbf{x}))$$

The choice of squared exponential kernel to capture the objective uncertainty is an interesting one and will be discussed further in the next section. Given the evaluations of the initial design points by each low fidelity model the posterior distributions of the Gaussian Processes can be es-

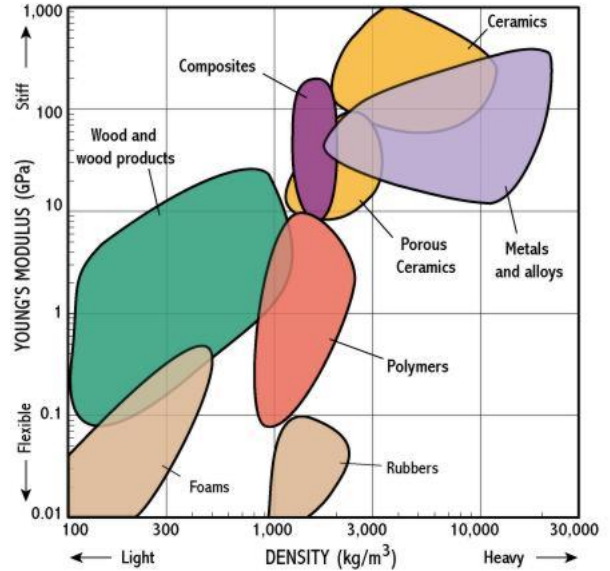


Figure 3: Ashby Plot comparing Young's Modulus of Elasticity and Mass Density of various materials

timated. An interesting feature of note in the framework provided by Ghoreishi, Allaire (2019) [6] is the separation of the effect on Gaussian uncertainty due to the surrogate modelling and due to the inaccuracy of the underlying low fidelity model. The latter effect is captured as an additional variance term in the variance of the posterior Gaussian distribution and requires the evaluation of the design points using the high fidelity ground truth.

Once the posterior distributions for the Gaussian Processes for the low fidelity models have been estimated, the information from each of these models is fused into a single Gaussian Process using Winkler (1981) [8]. This fused Gaussian Process is used in the acquisition function to de-

termine both the next information source to query and the next design point to sample. For brevity, the method of fusing multiple Gaussian Process distributions is not discussed in detail here and interested readers are encouraged to go through the previously referenced work.

A similar approach is applied to the constraint, specifically Equation 3. It is assumed that there are two other analytical approximations of the constraint. The idea of using Gaussian Process surrogates for constraints is to quantify uncertainty of using low fidelity constraint models, but this may not be warranted for linear or simple nonlinear constraints.

After the fused processes for both objectives and constraints are formed, acquisition functions are used for both objectives and constraints to find next design point to sample and information source to query. The acquisition function used for objective query is a two step look-ahead Utility function. Since the exact computation of this function is intractable, Monte Carlo simulations are run and the Utility Function is approximated as the average of the Utility function evaluated for each run. The Utility (or Utility to Cost ratio) is maximised to determine the next design and information source. A similar approach is utilized for the constraints, however the acquisition function used in this case is the Information Gain based on Knowledge Gradient. Once the next design points and information sources for the objectives and constraints are determined, the Gaussian Processes are updated.

There are many possible termination criteria that can be used for example number of function evaluations or increment in objective value. The most common criterion used is the number of function evaluations. Once the termination criterion is met, the design corresponding to the maximum fused mean is returned subject to the fact that the constraint value for that design point must lie within the 3σ bounds of the fused constraint process. The advantage of using Gaussian Process surrogates is the fact that in addition to the optimal design point, the maximum objective and the uncertainty bounds on the objective are also obtained.

As shall be discussed in Section 5, a few modifications were made to the optimization approach discussed above. However, it is worth noting that those modifications do not change the nature of the working of the optimization algorithm but merely some of the constituents of the algorithm. The identified needs of the technological problem are presented in Table 1. The importance values range from 0 to 5 with 0 being least important and 5 being of paramount importance. Smooth variation in matrix material within the prescribed bounds is a key assumption in the usage of the squared exponential kernel so its importance is very high. In the initial stages, the fibre material is assumed to be a user-defined constant (eg. 228 GPa for Carbon Fibre) but in the later stages the fibre material can also be made a parameter for optimization (with prescribed bounds). It is not necessary that the solution for matrix material from

Bayesian Optimization correspond to a physically available material. However, an additional process can be added in the end where the closest possible physical material can be found. The upper bound on the volumetric fraction can be increased however the bound is present to guard against obtaining the trivial solution of $v_f = 1$. The first stage of the optimization framework will be time consuming because of the high fidelity evaluations. However, after that the Finite Element Analysis is no longer consulted which speeds up computation. Hence, near real-time efficiency is kept at a low priority because of the initial computation.

Table 1: Identified needs of the problem and their importance

Need	Importance
Variation in matrix material (within bounds)	5
Variation in fibre material (within bounds)	2
Real-life solution for matrix material	3
Variation in volumetric fraction	2
Near real-time efficiency	2

Since the attempt to accomplish the true goal of this project was not successful, the extent of the satisfaction of the above mentioned needs cannot be warranted at this time. However, these needs are useful to keep in mind while devising a better approach to solve this problem. The next subsection presents the salient discussion points from the materials analysis of the problem.

2.2 Analysis of the Materials Problem

The materials science based analysis of the problem, as presented in the first report, focuses on the analytical model used to compute the buckling load for a composite column. Though this model was changed to a model applicable to composite plates, the requirement for the computation of an effective Young's Modulus still persists. The Effective Young's Modulus E is computed using the rule of mixtures. The rule of mixtures is represented in Equation 4, where E_f is the fibre Young's Modulus. It is to be noted that Chawla [9] warns that Rule-of-mixtures as applied conventionally to strength properties of metal-matrix composites is not valid for certain reasons. However, the corrected version of the Rule-of-Mixtures could not be found so the general version is utilized here with a note to apply the corrected version in future formulations as and when it becomes available.

$$E = E_f v_f + E_m (1 - v_f) \quad (4)$$

To be able to utilize Euler's Buckling formula effectively, the underlying assumption among the other standard ones is that the mapping between v_f and E is a smooth one. There is precedence for this. Many composite materials are present, especially biocompatible ones that exhibit variable mechanical properties in response to

a variety of stimuli. Shan et. al. [10] talks of an elastomer composite whose elastic rigidity is electrically tunable. Similarly, Cheng et. al. [11] presents work on a novel open-cell foam coated in wax that is able to achieve high ranges in stiffness and strength in response to temperature. More relevantly, composites exist in literature wherein changing the proportion of one of the constituents changes the Young's Modulus, among other properties. Bayer et. al. [12] discusses a biocompatible Poly(vinylidene fluoride) (PVDF) and Cyanoacrylate (CA) composite whose Young's Modulus decreases almost monotonically with the PVDF to CA ratio. Similarly, Visentin et. al. [13] talks about a Poly(Ethylene Glycol) Diacrylate (PEGDA) and ethyl-3-methylimidazolium bis(trifluoromethanesulfonyl)imide (EMI TFSI) based ionogel whose elastic modulus shows significant variation when the weight percentage of PEGDA is changed. Zhang et. al. [14] investigates a piezoelectric polymer composite for the possibility of harvesting energy from liquid fluctuations. The composite is comprised of Poly(vinylidene fluoride) (PVDF) and Poly(dimethylsiloxane) (PDMS) elastomer. Different stress-strain curves are found when the volume fraction of PVDF is varied. Also, the compressive modulus seems to increase monotonically with PVDF volume fraction.

In general, this section deals with the rationality of applying the Rule of Mixtures, given by Equation 4, to the analytical model through literature review. The explanation of the Euler Buckling Formulation for composite columns present in the first report is omitted here and the equivalent model for composite plates is discussed in the next section. The next subsection analyses the problem using Olson's Framework.

2.3 Analysis of the Problem using Olson's Framework

The Olson Framework representation of the design problem is shown in Figure 4. This representation enables us to see the various relationships between the manufacturing process, composite material structure and the property or properties that lead to the performance requirements we are optimizing for.

Here, the processing factors are taken from the Automated Fibre Placement (AFP) process which is extremely relevant to this design problem. The matrix melt pool is created first on which the fibres will be aligned. In the conventional process, the fibre feed is made to pass at the required alignment while depositing fibre in that alignment. A roller presses the fibre onto the matrix melt pool. This is called one tow. Multiple tows are required to set the fibre in the required alignment. Behind the fibre feed there is a temperature control rod that controls the temperature of the deposited fibre.

In light of the summary of the AFP process above, the factors under processing for the chart in Figure 4 are listed. They include the Matrix Melt Pool, Fibre Tows, Compres-

sion Force of the rollers, Direction of travel, Speed of travel and Temperature Control. These are the identified factors that affect the composite material structure. The factors within this category include fibre properties like cross-section radius and orientation, matrix properties like the material (which we solve for as a candidate for optimization) and certain defects like gaps in the matrix and fibre overlaps. These defects are identified and considered in Arian Nik et. al. [15] in their multiobjective optimization formulation pertaining to Automated Fibre Placement. The matrix material is directly related to the matrix melt pool factor under the processing category. It is rather obvious to see the relationship between the processing factors and the fibre properties. The number of fibre tows, the compression force and travel speed will affect the fibre cross-section radius whereas the previous factors plus the travel direction will determine the fibre orientation. Temperature of the composite after the fibre has been laid will also affect the cross-section dimensions and alignment of the fibres. There may also be possibly unforeseen matrix fibre interactions that may affect the fibre alignment by producing kinks. Finally, it is important to make note of the possible defects in the manufactured composite so that future frameworks can perform optimization in the presence of these defects. In any case, if the defects are small enough their effects on the material properties will be negligible but the effect of their consideration could also be explored in future work.

Now that the problem has been defined, an optimization algorithm is chosen and the problem analysis from the technological and Olson's Framework perspectives is performed, the next section summarizes the findings from the second report.

3 Materials Science Analysis of the Problem

The second report essentially extends the materials science analysis of the problem in the first report with an in-depth literature survey exploring the relationship between the fibre radius, fibre angle and matrix Young's Modulus. In addition, the aspect of balance between the exploration and exploitation of the design space is explained with reference to the acquisition functions for the optimization algorithm. The first subsection presents the findings from the literature survey exploring the relationships between the design parameters and the buckling load.

3.1 Composite Material Properties affecting Buckling Load

The choice of input parameters in the optimization process and their relationship with the buckling load of the composite plate is examined in this section. To reiterate, the objective of this research is to determine the optimal fibre radius, fibre orientation and matrix material for a composite plate to maximize the buckling load. The number of fibres and their relative location within the matrix is kept constant

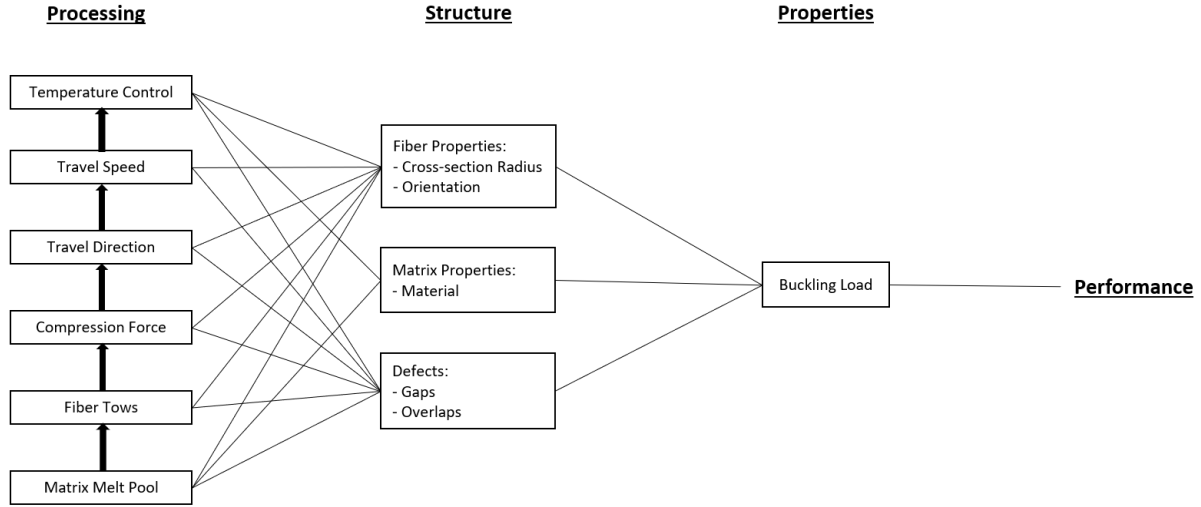


Figure 4: Material Systems Design chart showing the significant Process-Structure-Property relationships

and all fibres are straight and have the same orientation with respect to the global coordinate system.

As seen in the Olson Framework analysis above, the structural properties of interest for composites depend on many factors of the AFP process. The process parameters listed in the Material Systems Design Chart include the compression force of the fibre tow drum, fibre tow travel speed and direction, matrix melt pool and temperature control. The listed process parameters directly affect the consistency of the physical parameters of the fibre throughout the composite plate. The AFP process is susceptible to defects like gaps and overlaps which would influence the local stress resistance characteristics and hence the overall buckling resistance. However, the effect of these defects on the buckling load is not considered in this project. It is worth noting however that this effect can be captured by accounting for uncertainty in the design parameters. The manufacturing defects, although not considered in this project, can affect the buckling resistance of the manufactured composite and must be kept in mind for future work. For example, Matveev et. al.[16] discusses the effects of tow steering on defects like wrinkle formation and an analytical framework based parametric study is carried out to explore the effects of different processing parameters on wrinkle formation. New technology developed for manufacturing of composite structures could provide advantages in terms of material efficiency, production efficiency and/or reduction in mass for the same material behaviour and must also be actively explored. Dodwell et. al.[17] compares the established AFP process of manufacturing Variable Angle Tow (VAT) composites with a new process called continuous Tow Shearing (CTS). The paper shows the superiority of the CTS process over AFP in terms of weight when buckling and manufacturing constraints are imposed.

Assuming that the composite plate is free of local and global defects and the fibre orientation and radius are kept

consistent throughout the physical space of the composite plate, let us analyze the impact that the fibre radius, orientation and matrix Young's Modulus have on the buckling load.

3.1.1 Relationship between Buckling Load and Fibre Radius

Literature specifically investigating the effects of change in fibre radius on the buckling load of a composite structure could not be found. However, the result of variation in the fibre radius can be studied indirectly by studying the relationship between the volume fraction and the buckling load. It is worth noting that since the number of fibres is kept constant and the orientation of each fibre is the same, the variation in fibre radius results in a more significant change in the volume fraction than variation in fibre orientation. Wu et. al.[18] studies the buckling and post-buckling behaviour of functionally graded rectangular plates in the presence of mechanical and thermal loads. First order shear deformation theory informs the mathematical formulation used to model the buckling behaviour and the influence of the volume fraction on the effective Young's Modulus is in the form of a power law. This power law is presented in Equation 5.

$$E(z) = E_m + (E_c - E_m) \left(\frac{2z + h}{2h} \right)^k \quad (5)$$

Here, m and c correspond to metal and ceramic constituents respectively, h is the thickness of the plate and z is the thickness coordinate. The power index k represents the volume fraction influence, since if $k = 0$ the material is purely ceramic. Chebyshev polynomials are used to explicitly compute the buckling and post-buckling response of the composite plate. An example functionally graded mate-

rial (FGM) composite plate used in the paper is Aluminium-Alumina.

Buckling load for square plate with all edges clamped (CCCC)

Type	a/h	Volume fraction exponent (k)	Buckling loads	
			Dimensional $(N_x)_{cr}$	Non-dimensional $\lambda_{cr} \left(= \frac{(N_x)_{cr} b^2}{E_c h^3} \right)$
Ceramic (alumina)	100	—	348.0	9.1579
FGM	100	1	175.4	4.6158
FGM	100	2	136.0	3.5789
FGM	100	5	115.3	3.0342
Metal (aluminum)	40	—	996.7	1.6786
FGM	40	1	2720.4	4.5817
FGM	40	2	2120.0	3.5705
FGM	40	5	1781.0	2.9996
FGM	40	10	1615	2.72
Ceramic (alumina)	40	—	5380.8	9.0624
Ceramic (alumina)	20	—	4150.6	0.8738
FGM	20	2	1636.9	0.3446

Figure 5: Table showing the variation in the explicit and non-dimensionalized buckling loads for a square clamped plate with different aspect ratios (length to height ratio) and different power indices. Taken from Wu et. al.[18]

Figure 5 shows a table from the paper that lists the buckling loads (explicit and non-dimensionalized) for a square clamped plate for a variety of aspect ratios and power indices. The table clearly shows a variation in the buckling load with power index (indirectly a variation in the volume fraction). This variation would of course depend on the material of the matrix and fibre.

3.1.2 Relationship between Buckling Load and Fibre Orientation

Literature analyzing the effect of fibre orientation on the buckling load of a composite plate is in abundance. In fact, Zhu et. al. [19] introduces the Variable Angle Tow (VAT) placement technique that provides the ability to locally reinforce composites with holes to optimize for variable stiffness laminates. Hamani et. al.[20] specifically studies the effect of fibre orientation on critical buckling load of symmetric composite laminated plates of carbon/BMI with single and double circular notch using the finite element method. The variation of the critical buckling load with fibre orientation for different ratios of notch diameter to plate height is shown in Figure 6. The orientation angle θ is 0° when the fibres are perpendicular to the applied load, hence higher critical buckling load is observed at higher values of θ . Other works of literature also investigate the relationship between fibre orientation and buckling load for composite laminated plates. Raju et. al. [21] utilizes the Differential Quadrature Method (DQM) to perform buckling analysis of

VAT composite panels. A study is conducted on a square VAT plate and the non-dimensional buckling coefficient is plotted against the non-dimensional stiffness for varying angles of fibre orientation. The DQM method was compared with the finite element method for both cases and it was found to be in reasonable agreement. More importantly the curves for different fibre orientation angles were different, supporting the fact that fibre orientation influences buckling load among other factors. Almroth [22] is one of the oldest reports on designing of composite structures for buckling. The report conducts an extensive evaluation of the existing theory and verifies them with experiments.

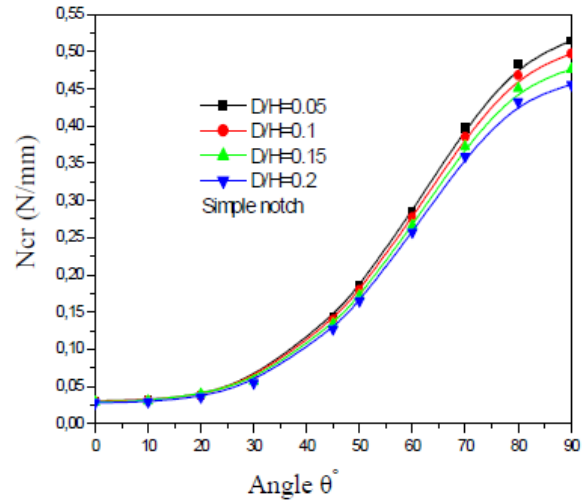


Figure 6: Plot showing variation of critical buckling load (N_{cr}) with variation in fibre orientation (θ) for different ratios of notch diameter D to plate height H . Taken from Hamani et. al.[20]

3.1.3 Relationship between Buckling Load and Matrix Young's Modulus

It is rather obvious that the choice of matrix material will heavily influence the buckling resistance of the resulting composite. The mapping between various material properties and the buckling load for composite plates has been studied in various works in literature. For example, Singh et. al.[23] models the material properties as random variables to simulate the effect of structural and/or manufacturing defects in the material behaviour. Higher Order Shear Deformation Theory was used to derive the governing equation for buckling in composite laminated rectangular plates. In two cases, dispersion (ratio of standard deviation to the mean) of the critical buckling load was plotted against the dispersion in longitudinal elastic modulus E_{11} and transverse elastic modulus E_{22} respectively. All other material properties were kept constant in each case. The plots taken

from the paper are shown in Figure 7.

The variation in the dispersion of the buckling load shows the effect of modelling manufacturing defects on the estimation of the buckling load using deterministic models. Also, as is expected, a higher matrix Young's Modulus would imply a higher buckling load.

Having explored the relationship between the chosen design parameters and the buckling load through literature, the next subsection presents a discussion on the various optimization techniques used for composite design in previous works of literature.

3.2 Composite Material Optimization Techniques

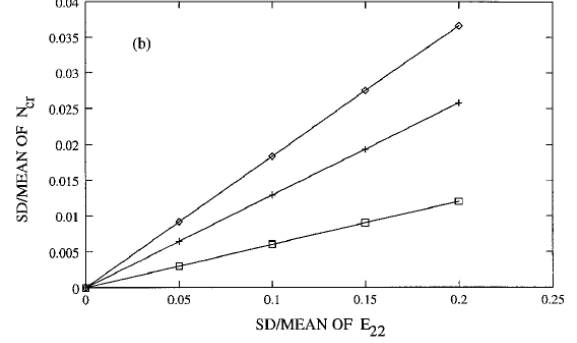
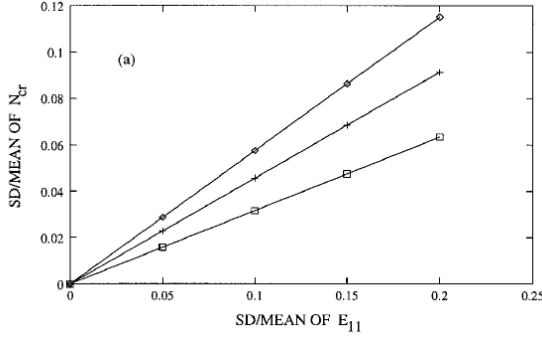
Viewing materials as a system enabled the formulation of the inverse deductive problem of determining process or structural parameters for the required performance characteristics as an optimization problem. For particularly expensive functional mappings between the process and performance parameter space, surrogate based optimization strategies were preferred. One of the surrogate models commonly used in place of the expensive physics based functions is the neural network. Zhang et. al.[24] is a review publication that explores the myriad of ways neural networks have been used in the design of polymer composites specifically. The key takeaways from the paper are the usefulness of neural networks in the prediction of material properties, the ability of the neural networks to capture complex relationships between material parameters and performance metrics and the need for optimization of the neural network architecture for computational efficiency. Hidayat et. al.[25] discusses the idea of adding multiplicative factors to the squared loss and regularization terms and optimizing over those factors using Bayesian Learning techniques in order to favour smaller and simpler architectures. The neural network is used for fatigue life prediction of composite structures. Hu et. al.[26] discusses the design optimization of the blades of a 5MW composite wind turbine for tortuous lightning strike and non-proportional multi-axial fatigue damage resistance. A Sequential Quadratic Programming (SQP) optimization approach was compared with a Bayesian Optimization approach on the same framework and was found to recommend a superior design. Araujo et. al.[27] utilizes experimental vibrational data to determine the material parameters for composite plates. The function to be minimized is the weighted sum of the normalized errors between the numerically computed and experimentally obtained eigenmodes of vibration. Huang et. al. [28] introduces a novel semi-analytical method for the buckling analysis of stiffened composite plates which involves the use of the Heaviside function and solving the characteristic stiffened plate buckling equation using the Galerkin method. For cheap or computationally efficient functions relating the process parameters to the performance metrics, evolutionary algorithms were used. Grosset et. al.[29] introduce the Double-Distribution Optimization Algorithm, a member of

the family of Estimation of Distribution Algorithms (EDA). It uses the distribution of the design variables and the distribution of physically meaningful auxiliary variables to generate more accurate distributions of the promising regions of the design space at lower computational cost.

Bayesian Optimization is a surrogate based optimization procedure best utilized when the physics based model is computationally expensive. It involves the formation of a Bayesian surrogate (Bayesian Network or Gaussian Process) which is cheaper but less accurate to the physics based model and maximizing an acquisition function to determine the next design point to query till a computational budget is exhausted. One of the earliest publications investigating the determination of material properties of a composite laminated plate is Papazoglou et. al.[30]. In the two step methodology proposed by the paper, first some of the natural frequencies of the plate are experimentally obtained and then an iterative Bayesian parameter estimation method is used to vary the plate rigidities to match the computational vibrational frequencies with the experimentally observed frequencies. Milani et. al.[31] compares a response surface method, an adaptive one-factor-at-a-time method and a Bayesian Design-of-Experiments based method through a simulation based layout optimization problem for composite laminate under fibre misalignment fabrication error. The research found that while the overall optimality, robustness and accuracy of the Bayesian method is superior to the other two, its global predictability can be quite poor.

The above Bayesian Optimization methods are useful when a single function mapping the composite material parameters to the material behaviour is available. However, there are other lower fidelity physics based models available that do not capture the relationships between the design parameters and the performance characteristics completely but still provide some information about the mapping between the two. The multi-fidelity multi-information source Bayesian Optimization algorithm, discussed in Ghoreishi et. al.[4] and utilized in a material design problem, can be used to fuse the information available from different information sources and perform the necessary optimization. In the presence of design constraints, the multi-fidelity multi-information source constrained Bayesian Optimization algorithm, introduced in Ghoreishi-Allaire[6] can be employed. This method is particularly suitable for this current research, given the availability of multiple lower fidelity models and the presence of design constraints and hence this method is utilized. The next few subsections discuss the features of this optimization procedure.

In light of the mapping between the design parameters and the buckling load as seen from the literature review conducted earlier, it is imperative that the design space search strategy accommodate the ability to improve the low-fidelity models while simultaneously moving towards the optimum. Since a Bayesian Optimization approach is employed in this research, a short discussion is presented below on the acqui-



(a) Dispersion in critical buckling load (N_{cr}) vs Dispersion in longitudinal elastic modulus (E_{11}) for different aspect ratios.

(b) Dispersion in critical buckling load (N_{cr}) vs Dispersion in transverse elastic modulus (E_{22}) for different aspect ratios.

Figure 7: Plots showing the variation in dispersion in critical buckling load with the dispersion in different elastic moduli for different aspect ratios ($\diamond = 1$, $+$ = 2, \square = 3). Both plots indirectly show the relationship between the elastic modulus and buckling load. Taken from Singh et. al.[23].

sition functions and balance between design space exploration and exploitation they achieve.

3.3 Design Space Exploration vs. Exploitation - The Acquisition Functions

The multi-fidelity constrained Bayesian Optimization algorithm taken from Ghoreishi-Allaire[6] uses different acquisition functions: the two-step look ahead Utility function for the objectives and the information gain for the constraints. Let us first look at the two step Utility function used in the case of the objectives. This is given in Equation 6.

$$U_{x,i} = \mathbb{E} \left[\max_{x' \in X_{feas}} \mu^{fused}(x') + \max_{x'' \in X_{feas}} EI_{x,i}(x'') \mid x_{1:N}, y_{1:N}, x_{N+1} = x, i_{N+1} = i \right] \quad (6)$$

where $EI_{x,i}(x'')$ is the one step look ahead expected improvement given that $x_{N+1} = x$ and $i_{N+1} = i$, given by Equation 7.

$$EI_{x,i}(x'') = \mathbb{E} \left[\max_{x' \in X_{feas}} \mu^{fused}(x') \mid x_{N+2} = x'' \right] - \max_{x' \in X_{feas}} \mu^{fused}(x') \quad (7)$$

Maximizing this utility divided by the querying cost $C_{x,i}$ determines the next design point x_{N+1} to query and next information source to evaluate x_{N+1}, i_{N+1} .

$$(x_{N+1}, i_{N+1}) = \underset{i \in [1, \dots, S], x \in X_{feas}}{\operatorname{argmax}} \frac{U_{x,i}}{C_{x,i}}$$

The Utility function $U_{x,i}$ is computed as a sum of the maximum mean $\mu^{fused}(x')$ of the fused GP obtained by incorporating x' and the expected improvement in the max

fused mean from choosing the $(N + 2)$ th design point as x'' . The maximization of the fused mean makes sense since the objective $y = f(x)$ must be maximized according to the theoretical development. The Expected Improvement is computed using the Knowledge Gradient metric, maximization of this Knowledge Gradient determines the design point to sample whose uncertainty bound on the corresponding output makes it a possible candidate for the optimal design. The only way to find out if this is indeed true is to sample this point. If the sampled design point does not turn out to be the optimal design, it still increases the fidelity of that corresponding GP in the design space. In essence, the idea is to find design points that improve the fidelity of the corresponding GPs and hence improve their capability to find the true optimal design.

For reasons that will be discussed further in Section 5, the two-step look ahead Utility function is switched to a one-step Expected Improvement function for this research. However the fundamentals behind the two functions are the same and they balance design space exploration and exploitation in the same manner.

The Information Gain metric used for the constraints is shown in Equation 16.

$$IG_{x,i}^j = \frac{1}{N_r} \sum_{r=1}^{N_r} \sum_{s=1}^{N_s} D_{KL} \left(\hat{g}_{x,i,j}^{fused,r}(x'_s) \parallel \hat{g}_j^{fused}(x'_s) \right) \quad (8)$$

D_{KL} is the KL-Divergence. It is defined between two probability distributions P and Q , with probability densities $p(x)$ and $q(x)$ as

$$D_{KL}(P \parallel Q) = \int_{-\infty}^{\infty} p(x) \log \frac{p(x)}{q(x)} dx$$

Maximizing the Information Gain divided by the querying cost of the constraint $C_{x,i}^j$ determines the next design

point \mathbf{x}_{N+1}^j to query and next information source to evaluate $\mathbf{x}_{N+1}^j, i_{N+1}^j$.

$$(\mathbf{x}_{N+1}^j, i_{N+1}^j) = \underset{i \in [1, \dots, G_j], \mathbf{x} \in X_g}{\operatorname{argmax}} \frac{IG_{\mathbf{x}, i}^j}{C_{\mathbf{x}, i}^j}$$

The fused constraint distribution density $\hat{g}_{\mathbf{x}, i, j}^{fused, r}$ is formed using the additional N_r independent designs sampled from the normal distribution denoted by $g_{i, j}^r(\mathbf{x})$ and augmenting each sample one at a time to \hat{g}_j^{fused} . The average information gain is found using N_s Monte Carlo samples.

In this case, the idea is to determine the design point and constraint information source that changes the resulting constraint source distribution the most. Similar to the maximization of the utility function for the objectives, the next design point to be sampled improves the information about the constraint variance in the design space captured by the corresponding constraint information source. Hence, the information gain is computed using KL-Divergence.

With an idea about the relationship between the design parameters and the buckling load along with some knowledge of the manner of design space traversal conducted by the optimization algorithm of choice, we now delve deeper into the forward maps between these design parameters and the performance metric (both functional and data-driven) in the next section.

4 Forward Mapping between Designs and Objective

This section presents the investigations of the forward mappings between the design parameters and buckling load as were carried out in the third report. First, the high fidelity numerical Finite Element Analysis model (as computed through the Abaqus CAE software) is discussed below. It must be noted that at the time, the Abaqus python script was not in working condition. The discussion relates to the final working version of the Abaqus python script.

4.1 Numerical Prediction of Buckling Load - Abaqus FEA

the high-fidelity model for this project is considered to be the Finite Element Model (using the Abaqus CAE software). Abaqus CAE has a Graphical User Interface (GUI) version as well as inbuilt compatibility with python. This compatibility with python is exploited in order to conduct Finite Element Analyses for different design parameter combinations. The python script titled "abaqus_fea_macro.py" is created for this purpose. The idea is that the python script can be run through the command window with the design parameters as arguments. An additional job number argument is provided to avoid the overwriting of job files for multiple evaluations and the resultant issues. The python script runs the job for the give nset of

parameters, extracts the eigen value information from the output database (.odb) file and outputs them as a string. It is important to note that linear buckling analysis will be carried out in Abaqus for this project and hence further nonlinear post-buckling behavior will not be considered. Figure 8 shows the results of an example buckling job run in Abaqus CAE. Five buckling modes were requested in the buckling analysis. The training dataset will have three eigenmodes for each datapoint and the Neural Network will be trained to regress three eigenmodes for each design point. The details of the generation of the training dataset will be discussed in the next section.

The meshing of the part is done solely using linear tetrahedral elements. The coarseness of the mesh is governed by three factors: the seed size, the deviation factor and the minimum size factor. The python script has the ability to re-mesh the part if the job does not go through due to the generation of zero volume elements. In every re-meshing cycle, the seed size is reduced to create a finer mesh while keeping the deviation factor and minimum size factor the same. A limit of three re-mesh cycles was kept in order to be able to efficiently evaluate multiple designs and get results for most of them as quickly as possible and to avoid the mesh node limit of 250,000 (for the academic teaching version). If the design cannot be evaluated successfully within the three re-meshing cycles, an error is thrown.

Now, let us look at the forward mapping between the fibre radius, fibre angle and matrix Young's Modulus and the buckling load individually as captured by the analytical model. First, the analytical model is explained and then the forward mapping is investigated.

4.2 Analytical Prediction of Buckling Load

A brief derivation of the analytical model, taken from Brush et. al.[32] is presented.

The analytical model is based on a simplification of the buckling differential equation for rectangular plates. Given that a simply supported rectangular plate is subject to a uniform compressive load shown in Figure 9 and the rectangular plate is only subject to in-plane loads and has membrane pre-buckled shape (no out-of-plane deflections initially), the full buckling differential equation can be simplified to Equation 9.

$$D\nabla^4(\delta w) + N_x \delta w_{,xx} = 0 \quad (9)$$

Here, $D = \frac{Et^3}{12(1-\nu^2)}$ is the Flexural Rigidity of the rectangular plate, $E = E_f v_f + E_m(1 - v_f)$ is the Effective Young's Modulus computed using the Rule of Mixtures, E_f is the fibre Young's Modulus, E_m is the matrix Young's Modulus, v_f is the volume fraction, ν is the Poisson's ratio (taken to be a constant = 0.3 for both matrix and fibre), $\nabla^4(u) = \frac{\partial^4 u}{\partial x^4} + \frac{\partial^4 u}{\partial y^4}$, δw is the deflection due to buckling and $u_{,xx} = \frac{\partial^2 u}{\partial x^2}$.

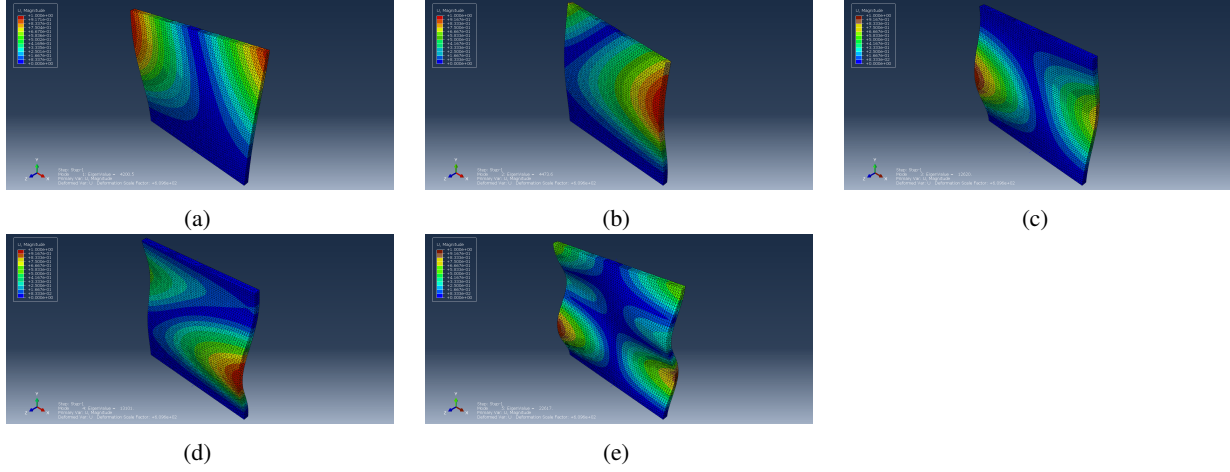


Figure 8: Abaqus FEM Visualization of the displacement field for five requested buckling eigenmodes for a test composite rectangular plate. r_f was chosen to be 95.25mm, θ_f was chosen to be 45° , the matrix was chosen to be Iron ($E_m = 1.406 \times 10^6 \text{ N/mm}^2$, $\nu_m = 0.29$) and fibre was chosen to be Copper ($E_f = 0.121 \times 10^6 \text{ N/mm}^2$, $\nu_f = 0.34$). (a) First eigenmode, (b) Second eigenmode, (c) Third eigenmode, (d) Fourth eigenmode, (e) Fifth eigenmode

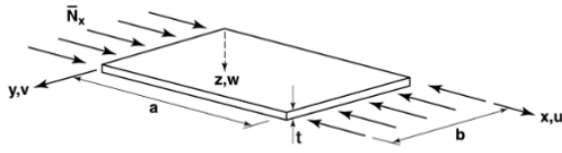


Figure 9: Plate Buckling setup. a and b are the plate length and width (referred to in this report as l and w respectively), t is the plate thickness and \bar{N}_x is the critical buckling load

Taking into consideration the form of the buckling equation given by Equation 9, the solution δw is assumed to have a form given by Equation 10, where \tilde{w} is the amplitude of the deflection and m and n are the buckling mode numbers in the x and y directions respectively.

$$\delta w = \tilde{w} \sin\left(\frac{m\pi x}{a}\right) \sin\left(\frac{n\pi y}{b}\right) \quad (10)$$

Using the assumed form of δw , the solution for the buckling load for the described conditions is found to be Equation 11.

$$N_x = \frac{\pi^2 a^2 D}{m^2} \left[\left[\frac{m}{a} \right]^2 + \left[\frac{n}{b} \right]^2 \right]^2 \quad (11)$$

This is the analytical model used to evaluate the designs generated by LHS for this report and it will be used as a low fidelity model in the multifidelity constrained Bayesian Optimization algorithm.

The volume fraction is computed by computing the length of the fibres, which in turn is a function of the position and angle of the fibres. Figure 10 shows a fibre located at a distance x ft. from the left face of the plate and at an angle θ_f

from the horizontal. Given these measurements, the length of the fibre l_f can be computed as

$$l_f = \frac{2x}{\cos(\theta_f)}$$

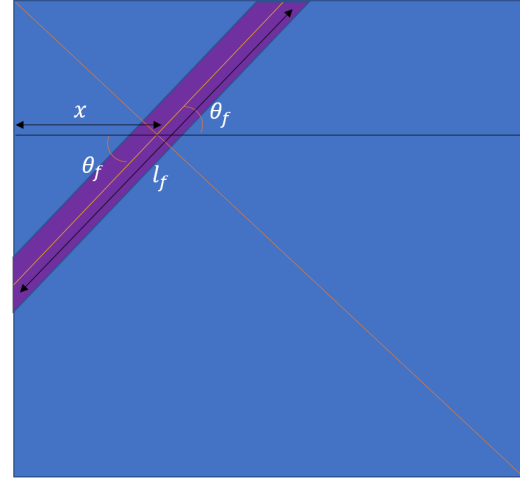


Figure 10: Illustration of a fibre (shown in purple) in the matrix (shown in blue) located at a distance x ft. from the left face and inclined at an angle θ_f from the horizontal. Length of the fibre is denoted by l_f . The centre of each fibre lies on the diagonal shown in orange.

The number of fibres is kept at 5 and are spaced equally from each other. The length and width of the composite plate is taken to be 10 ft. and thickness is 254mm. The positions of the fibres is given in Table 2.

Since the fibres are equidistant from each other, fibres 1 & 5 and fibres 2 & 4 will have the same length. Thus the total

Fibre Number	x in ft.
1	5/3
2	10/3
3	5
4	20/3
5	25/3

Table 2: Table showing the positions of the different fibres in the composite plate

volume occupied by the fibres is roughly given by Equation 12, where $l_{f,1}$ is the length of fibre 1, $l_{f,2}$ is the length of fibre 2 and $l_{f,3}$ is the length of fibre 3.

$$Vol_f = \pi r_f^2 (2l_{f,1} + 2l_{f,2} + l_{f,3}) \quad (12)$$

The volume fraction is then computed by dividing Vol_f by the total volume.

$$v_f = \frac{Vol_f}{abt}$$

The subsequent subsections explore the mapping between each design parameter and the buckling load as captured by the analytical model given by Equation 11 one at a time. For this mapping study, the length and width of the composite plate are taken as 10 ft. (3048 mm) each, thickness is taken as 254 mm and fibre Young's Modulus is taken as 40 GPa. For the purpose of ease in visualization, only one eigen mode is queried and visualized for each case in this study (i.e. $m = n = 1$).

4.2.1 Buckling Load vs. Fibre Radius

To visualize the variation in buckling load as a result of variation in fibre radius exclusively, the fibre angle is kept at a constant 45° and the matrix Young's Modulus is kept constant at 150 GPa. The fibre radius is varied from 25 mm to 125 mm and the resulting variation in the buckling load is illustrated in Figure 11. An almost quadratic decrease in the buckling load is observed with increase in the fibre radius. This can be attributed to the fact that increase in the fibre radius implies an increase in the volume fraction v_f . This in turn decreases the Effective Young's Modulus E which decreases the Flexural Rigidity D .

4.2.2 Buckling Load vs. Fibre Angle

To visualize the variation in buckling load as a result of variation in fibre angle exclusively, the fibre radius is kept constant at 75 mm and the matrix Young's Modulus is kept constant at 150 GPa (as in the previous study). The fibre angle is varied from -89° to 89° (since the fibre lengths will go to infinity as the fibre angle approaches 90°) and the resulting variation in the buckling load is shown in Figure 12. There is not much variation in the buckling load seen in the range $-80 \leq \theta_f \leq 80$ however there is a sharp decrease in the

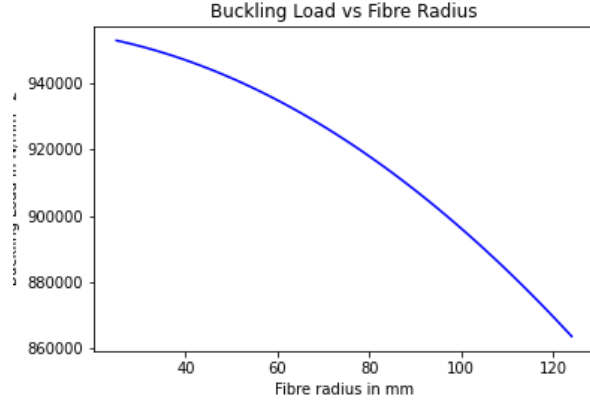


Figure 11: Plot showing the variation in buckling load (in N/mm^2) with variation in fibre radius. A monotonic decrease in the buckling load is observed with increase in the fibre radius.

buckling load at the extreme ends of the fibre angle range. A change in fibre angle constitutes a very negligible change in the volume fraction since it can be imagined that for any increase in fibre material within the composite in any of the off diagonal fibres, there is a commensurate decrease in fibre material in the diagonal fibre. As a result, the Effective Young's Modulus E and Flexural Rigidity D do not change much. However, there is clear evidence that placing fibres in an inclined manner within the matrix constitutes an increase in the buckling load.

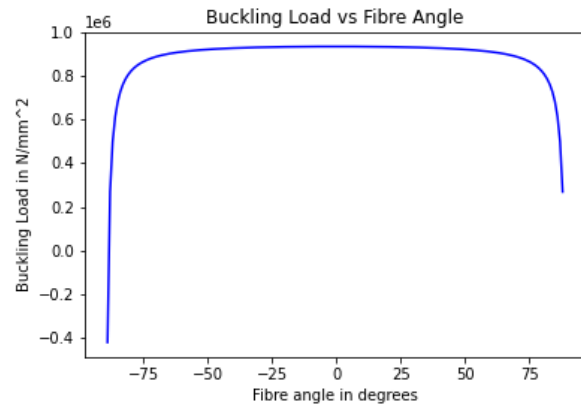


Figure 12: Plot illustrating the variation in buckling load (in N/mm^2) with fibre angle. The major variation in buckling load is seen at the extreme ends of the fibre angle range wherein the buckling load shows a sharp drop.

4.2.3 Buckling Load vs. Matrix Young's Modulus

To visualize the variation in buckling load as a consequence of variation in the matrix Young's Modulus exclusively, the

fibre radius is kept constant at 75 mm and the fibre angle is kept constant at 45° . The matrix Young's Modulus is varied from 10 GPa to 800 GPa and the resulting variation in the buckling load is presented in Figure 13. A linear increase in the buckling load is observed with increase in the matrix Young's Modulus. This is expected, since a higher matrix Young's Modulus implies a higher Effective Young's Modulus E . This in turn means a higher Flexural Rigidity D . It can also be seen, upon substitution of the formula for D into Equation 11 that the buckling load is linearly related to the matrix Young's Modulus.

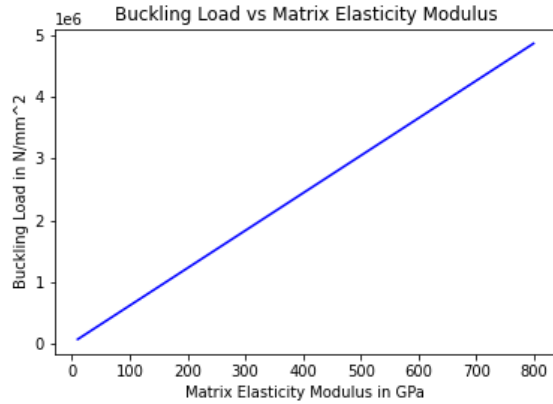


Figure 13: Plot illustrating the variation in the buckling load (in N/mm^2) with variation in the matrix Young's Modulus. A linear increase in the buckling load is observed with increase in the matrix Young's Modulus.

From the above parametric studies it can be concluded that the analytical model contains at least some information about the mapping from the design space to the objective space. It must be noted that although the variation of the buckling load with each individual parameter may be trivial but the final solution will depend on the values set for the problem constants, like fibre Young's Modulus E_f . Although placing the fibres in an inclined manner will lead to a higher buckling load, the variation of the buckling load within a huge portion the fibre angle range is not very significant. The code and plots generated for this study can be found in the IPython notebook "Design_Relationships_Analytical.ipynb".

Having learnt these lessons, the next subsection compares the Neural Network (the data-driven surrogate model of choice for the optimization routine) with a GPR and a Linear Regressor.

4.3 Comparison of Data-Driven Regressors

The data-driven surrogate of choice which will be used as the second low fidelity model in the multifidelity constrained Bayesian Optimization algorithm is the Neural Network due to its ability to capture complex nonlinear mappings between multi-dimensional spaces. In this subsection,

the Neural Network is compared with a GPR and a Linear Regressor in terms of uncertainty in prediction. The design dataset of 100 datapoints used to train the regressors is generated using Latin Hypercube Sampling (LHS), making sure they satisfy the volume fraction constraint given by Equation 3. The LHS method does not have the facility to handle design constraint functions so a looped approach is followed wherein new design sample sets are generated until the generated designs satisfy the design constraint functions.

It is important to acknowledge the fact that the LHS function in pyDOE2 does not have a random seed option and hence will generate a different design sample set every time its run.

The generated design datapoints are evaluated using the low fidelity physics-based analytical model. This dataset is then subject to pre-processing wherein the data is centred using the mean and scaled to unit variance using the standard deviation. This processed dataset is then split into a training dataset and a testing dataset in 80-20 proportion. The training dataset is used to train the regressors whereas the testing dataset is used to compare the trained regressors.

For the regressor comparison study, the testing datapoints are inputted to the regressors and their predictions are stored. Since the Neural Network and Linear Regressor do not provide uncertainty metrics inherently as part of the framework, the uncertainty metric is computed by subtracting the analytical evaluations from their predictions. The uncertainty metric for the GPR is the standard deviation. The statistics for the uncertainty metrics for the three regressors are shown as a boxplot in Figure 14. It can be seen that the GPR uncertainty metric does not have an inter-quartile range since the standard deviation for all the test datapoints was found to be 1. The figure implies that the Linear Regressor has a better predictive capability than the Neural Network or the GPR.

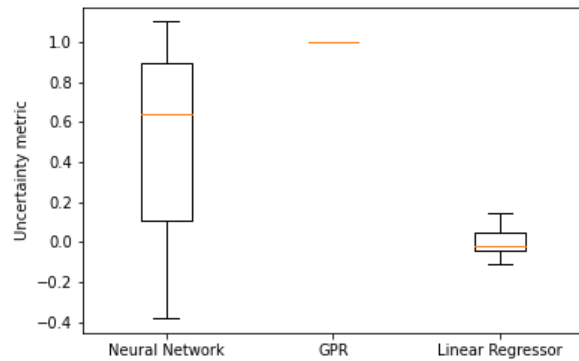


Figure 14: Boxplot showing the comparison of the different uncertainty metrics for the three regressors. GPR does not have an inter-quartile range because the standard deviation for all the datapoints is 1.

Although the results show that a Linear Regressor is a better data-driven model for this dataset, it must be noted that in the actual optimization routine, three buckling modes will be utilized, which will increase the complexity of the mapping between the design parameters and the performance metrics. The small size of the training and testing datasets are also contributing factors towards the results, the Neural Network is most effective when provided with a large training dataset. Also, the manner of implementation of the data-driven model will be to store the scaling value from the pre-processor, so the scaled output from the data-driven model can be converted to proper buckling load. The comparison of those scaled values was not done here.

The code and results shown above can be seen in the IPython notebook "Regressor_Comparison.ipynb". The next subsection focuses on the GPR since it plays a major role in the multifidelity Bayesian Optimization algorithm. A few different covariance kernels are compared to visualize their effect on the predictive capability of the GPR.

4.4 Comparison of Kernels for Gaussian Process Regressors

The GPR can employ a variety of kernel functions to characterize the covariance between different design points. The key idea of this study is to compare the different kernels to see how the choice of covariance kernel impacts the predictions of the GPR. Four kernel functions (three distinct ones) are chosen for comparison. A great reference for different kernel function choices for Gaussian Processes in general is Chapter 4 in Rasmussen et. al.[33].

For this study, again the training dataset was generated using LHS keeping the volume fraction constraint satisfaction in consideration. The generated designs are evaluated using the analytical model. Then a meshgrid of the fibre radius and fibre angle ranges is created and the predicted means and standard deviations for the trained GPR are found at 3 constant values of matrix Young's Modulus E_m (10, 405 and 800 GPa) and shown for ease of visualization. The ranges for the fibre radius (in mm) and fibre angle (in degrees) are given below, where t is the previously mentioned thickness of the composite plate.

$$25 \leq r_f \leq t/2$$

$$-90 \leq \theta_f \leq 90$$

The subsequent sub-subsections discuss each kernel function one at a time along with the results of the prediction study for each.

4.4.1 Squared Exponential Kernel (No Length Scale)

The squared exponential function is the most commonly used stationary covariance function. It has the form given in Equation 13 where r is the Euclidean distance of a point from the mean and l is a characteristic length scale that is assigned a-priori.

$$k_{SE}(r) = \exp\left(-\frac{r^2}{2l^2}\right) \quad (13)$$

In the scikit learn python library, the squared exponential kernel is represented by the "rbf" function. In this case, no characteristic length scale is provided so it assumes a length scale of 1 in all dimensions. Figures 15 and 16 show the predicted mean and standard deviation respectively for the uniformly distributed meshgrid points for the fibre radius and fibre angle values and at the three constant values of the matrix Young's Modulus.

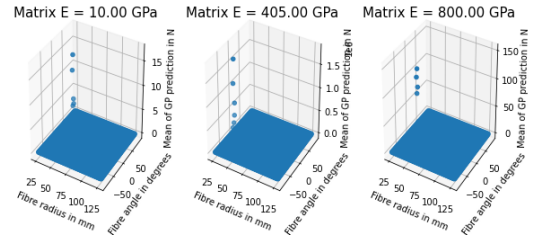


Figure 15: The mean plots of the predictions of the GPR with squared exponential kernel (length scale = 1 in all dimensions) for $E_m = 10GPa$, $E_m = 405GPa$ and $E_m = 800GPa$

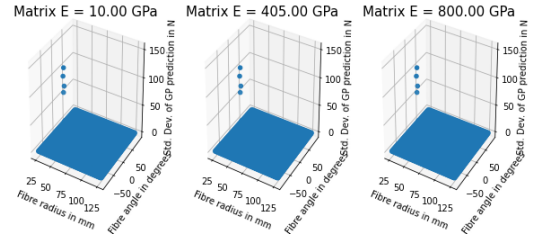


Figure 16: The standard deviation plots of the predictions of the GPR with squared exponential kernel (length scale = 1 in all dimensions) for $E_m = 10GPa$, $E_m = 405GPa$ and $E_m = 800GPa$

It can be clearly seen that the squared exponential kernel is not very good in prediction since it basically predicts zero mean and zero standard deviation at all three values of the matrix Young's Modulus (save for some outliers). In any case, it is understood that the length scale will not be the same in different regions of the design space and the requisite length scale must be provided to deal with the outliers. The next sub-subsection discusses the results of providing a length scale to the squared exponential kernel function.

4.4.2 Squared Exponential Kernel (With Length Scale)

In this case, the characteristic length scale provided is the upper limit of the assigned ranges for each design parameter

(i.e. $t/2$ mm for fibre radius, 90° for fibre angle and 800 GPa for matrix Young's Modulus). Figures 17 and 18 show the predicted mean and standard deviation respectively for the uniformly distributed meshgrid points for the fibre radius and fibre angle values and at the three constant values of the matrix Young's Modulus.

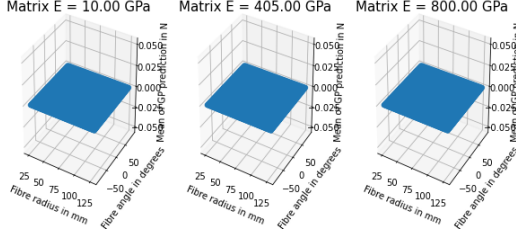


Figure 17: The mean plots of the predictions of the GPR with squared exponential kernel (length scale provided) for $E_m = 10GPa$, $E_m = 405GPa$ and $E_m = 800GPa$

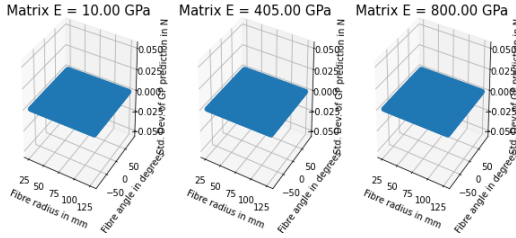


Figure 18: The standard deviation plots of the predictions of the GPR with squared exponential kernel (length scale provided) for $E_m = 10GPa$, $E_m = 405GPa$ and $E_m = 800GPa$

The provision of the length scale values in each dimension gets rid of the outliers but does not do much to improve the predictive capability of the GPR. The next sub-subsection looks at the Matern kernel.

4.4.3 Matern Kernel

The Matern class of functions can be seen as a generalization of the squared exponential kernel and contain an additional parameter ν_m . They have the form shown in Equation 14 where K_{ν_m} is a modified Bessel function, $\Gamma(x)$ is the Gamma function and r and l are as defined in the previous subsection.

$$k_{Matern}(r) = \frac{2^{1-\nu_m}}{\Gamma(\nu_m)} \left(\frac{\sqrt{2\nu_m}r}{l} \right)^{\nu_m} K_{\nu_m} \left(\frac{\sqrt{2\nu_m}r}{l} \right) \quad (14)$$

The Matern kernel function in scikit learn assumes $l = 1$ for all dimensions if not provided with a length scale and $\nu_m = 1.5$ if not provided with a ν_m value. These were

not provided for this study. Figures 19 and 20 show the predicted mean and standard deviation respectively for the uniformly distributed meshgrid points for the fibre radius and fibre angle values and at the three constant values of the matrix Young's Modulus.

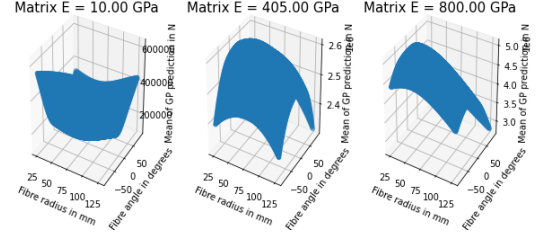


Figure 19: The mean plots of the predictions of the GPR with Matern kernel for $E_m = 10GPa$, $E_m = 405GPa$ and $E_m = 800GPa$

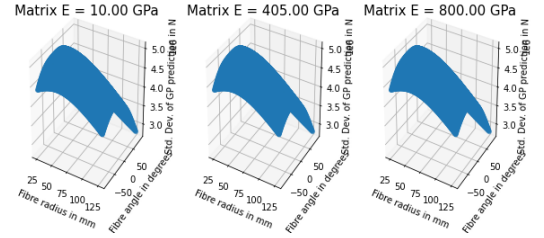


Figure 20: The standard deviation plots of the predictions of the GPR with Matern kernel for $E_m = 10GPa$, $E_m = 405GPa$ and $E_m = 800GPa$

The Matern kernel shows a much better prediction capability as compared to the squared exponential kernel. It displays an ability to capture more nonlinear mappings and perhaps better results could be obtained if the length scale information is provided. The next sub-subsection discusses a combination of the Dotproduct kernel and a White noise kernel.

4.4.4 Dot Product + White Noise Kernel

The DotProduct kernel is a non-stationary kernel which is parameterized by σ_0^2 . It has the form given in Equation 15. For this study, this kernel is appended with a White Noise kernel (which is simply another σ_w^2 term) to overcome training issues.

$$k_{DP}(\mathbf{x}, \mathbf{x}') = \sigma_0^2 + \mathbf{x} \cdot \mathbf{x}' \quad (15)$$

Figures 21 and 22 show the predicted mean and standard deviation respectively for the uniformly distributed meshgrid points for the fibre radius and fibre angle values and at the three constant values of the matrix Young's Modulus.

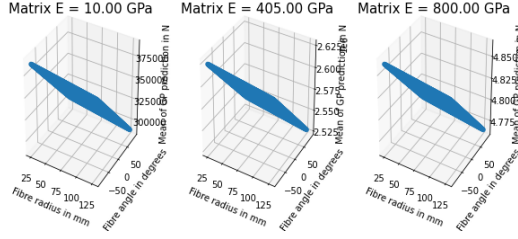


Figure 21: The mean plots of the predictions of the GPR with DotProduct + White Noise kernel for $E_m = 10GPa$, $E_m = 405GPa$ and $E_m = 800GPa$

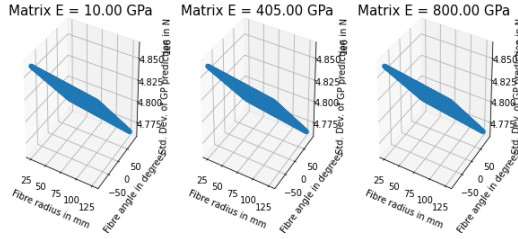


Figure 22: The standard deviation plots of the predictions of the GPR with DotProduct + White Noise kernel for $E_m = 10GPa$, $E_m = 405GPa$ and $E_m = 800GPa$

The DotProduct + White Noise kernel has better predictive capability as compared to the squared exponential kernel but does not seem to capture the inherent nonlinearities in the mapping between the design parameters like the Matern kernel is able to do. As a result the Matern kernel seems to be the kernel of choice for the nonlinear mapping between the design parameters and the buckling load expected in the final optimization routine. Other kernels can also be explored, like the γ -exponential kernel or the rational quadratic kernel.

The code and plots shown here can be found in the IPython notebook "GP_Regression.ipynb". The next subsection briefly discusses the nature of the role played by the GPR in the multifidelity constrained Bayesian Optimization algorithm.

4.5 GPR Deployment in Optimization Algorithm

The multifidelity constrained Bayesian Optimization algorithm as detailed in Ghoreishi-Allaire[6] is an information fusion based optimization algorithm. In brief, each low fidelity model, which for this project is the analytical model and the Neural Network, is used to train a GPR that surrogates that low fidelity model. These GPRs are then fused using the method described in Winkler[8]. Essentially the Gaussian Processes are combined based on the degree of overlap they have in knowledge of the mapping between the design parameters and the objective, which is captured in a correlation metric. This fusion of the Gaussian Processes

corresponds to fusion of the mapping information possessed by each low fidelity model. The use of Gaussian surrogates enables the fusion of the low fidelity models taking the similarity in their mapping information into consideration. The choice of covariance kernel will play a huge role in the fusion process as it contributes to the correlation metric between the low fidelity models. Thus the previous study was warranted.

Now that the functional and data-driven forward maps have been evaluated and the choice of the Matern kernel for Gaussian Process Regression is certain, the next section presents the details of the multi-fidelity constrained Bayesian Optimization approach and the results of its implementation on a test problem and the attempt at the composite plate problem.

5 Multifidelity Constrained Bayesian Optimization Algorithm

As mentioned before, the optimization algorithm of choice for this project is taken from Ghoreishi-Allaire[6]. However, some modifications have been made to the original algorithm for ease of implementation and improvement of accuracy. These modifications will be discussed in the algorithm description provided below.

5.1 Algorithm Working

Algorithm 1 shows the working of the multi-fidelity constrained Bayesian Optimization (MFCBO) algorithm employed for this project.

The algorithm begins with the generation of design datasets for the training of the GPRs for the ground truth and lower fidelity models for the objectives and constraints (shown as $X_{GT,f}$, X_f , $X_{GT,c}$ and X_c respectively in the Algorithm) using LHS. The GPR for the ground truth is only used for the computation of the fidelity variance (shown as $\sigma_{f,f}$ and $\sigma_{f,c}$ for the objective and constraint lower fidelity models respectively). The key idea here is that the total variance of a low-fidelity model at a design point is the sum of the GPR variance and the fidelity variance, i.e.

$$\sigma_f(\mathbf{x}) = \sigma_{f,GP}(\mathbf{x}) + \sigma_{f,f}(\mathbf{x})$$

The fidelity variance captures the discrepancy between the low-fidelity model and the ground truth as estimated by their GPRs. Either any of the training datasets or a completely new dataset can be used to compute the fidelity variance. The designs in the dataset are evaluated using the ground truth GPR and the low-fidelity GPRs and the square of the average difference between the predictions of the ground truth and a low fidelity model is used as the constant fidelity variance for that model. This fidelity variance is dynamically recomputed in every iteration to reflect the updating of the GPRs (low-fidelity and/or ground truth).

In each iteration, a different set of alternate, test and indicator design samples are generated using LHS (shown as

X_{alt} , X_{test} and X_{ind} respectively). The GPRs for the low-fidelity constraints are fused using the Winkler Fusion process, detailed in Winkler[8]. This fusion process will be described in detail later, This fused constraint GPR is used to filter the infeasible designs from the three design datasets. The condition for feasibility of a design is given by the following:

$$\mu_{c,fused}(\mathbf{x}) + 3\sigma_{c,fused}(\mathbf{x}) \leq 0$$

This condition is robust to the prediction uncertainty of the fused constraint GPR. The filtered designs are denoted as $X_{alt,feas}$, $X_{test,feas}$ and $X_{ind,feas}$ respectively.

Now, the idea is to choose the next design point to query from the $X_{alt,feas}$ dataset and the next information source to query by maximization of the respective acquisition functions for the objectives and constraints. The choice of acquisition function for the objectives in Ghoreishi-Allaire[6] is the two step look-ahead Utility function which is computed using a combination of Monte Carlo simulations and Knowledge Gradient. However, in this project, due to the lack of availability of a package capable of easy integration that can compute the knowledge gradient and the lack of time for creating a function for this from scratch, the acquisition function of choice is the one step Expected Improvement function. It is computed according to Equation below, referenced from Jones et. al.[34].

$$EI(\mathbf{x}) = \begin{cases} (\mu(\mathbf{x}) - y_{current}^* - \xi)\Phi(\mathbf{Z}) + \sigma(\mathbf{x})\phi(\mathbf{Z}) & \text{if } \sigma(\mathbf{x}) > 0 \\ 0 & \text{if } \sigma(\mathbf{x}) = 0 \end{cases}$$

where

$$\mathbf{Z} = \begin{cases} \frac{\mu(\mathbf{x}) - y_{current}^* - \xi}{\sigma(\mathbf{x})} & \text{if } \sigma(\mathbf{x}) > 0 \\ 0 & \text{if } \sigma(\mathbf{x}) = 0 \end{cases}$$

Each alternative point is added to the training dataset for each of the GPRs for the low-fidelity objective models one at a time. Then samples are drawn around that GPR's prediction at that design point. For each of the samples drawn, a fused temporary GPR (shown as $TempGP_f$) is constructed using the augmented dataset for that GPR and the other GPR. The predictions from this fused GPR on the designs in $X_{test,feas}$ are used to compute the expected improvement. Instead of computing the expected improvement at the means of each feasible alternate design point, the expected improvement of a design point is taken as the average of the expected improvements computed for the samples drawn around the GPR prediction at the design point. In this manner, the expected improvement values divided by the cost for the corresponding low-fidelity model for each feasible alternate design point and each low-fidelity model is computed and stored. Finally, the design point and model corresponding to the maximum utility metric (expected improvement divided by the cost) is chosen and stored.

Algorithm 1: MFCBO Algorithm

Input: budget,
 $cost_f, cost_c, \mathbf{x}_{range}, n_f, n_c, n_{alt}, n_{test}, n_{ind}$

- 1 $X_{GT,f}, X_f, X_{GT,c}, X_c = LHS(\mathbf{x}_{range}, n_f, n_c)$
- 2 $GP_f = GpTrain(X_f)$
- 3 $GP_{GT,f} = GpTrain(X_{GT,f})$
- 4 $GP_c = GpTrain(X_c)$
- 5 $GP_{GT,c} = GpTrain(X_{GT,c})$
- 6 $\sigma_{f,f} = FidelityVariance(GP_{GT,f}, GP_f)$
- 7 $\sigma_{f,c} = FidelityVariance(GP_{GT,c}, GP_c)$
- 8 **while** $cost \leq budget$ **do**
- 9 $X_{alt}, X_{test}, X_{ind} =$
 $LHS(\mathbf{x}_{range}, n_{alt}, n_{test}, n_{ind})$
- 10 $GP_{c,fused} = Fusion(GP_c, X_c)$
- 11 $X_{alt,feas}, X_{test,feas}, X_{ind,feas} =$
 $ConstFeas(X_{alt}, X_{test}, X_{ind}, GP_{c,fused})$
- 12 **for** $\mathbf{x} \in X_{alt,feas}$ **do**
- 13 // Objective Optimization
- 14 $X_f \leftarrow [X_f, \mathbf{x}]$
- 15 **for** $y \in GpSample(GP_f, \mathbf{x})$ **do**
- 16 $TempGP_f = GpTrain(X_f, y)$
- 17 $GP_{f,fused} =$
 $Fusion(GP_f, TempGP_f, X_{test,feas})$
- 18 $EI_y(y) =$
 $ExpImprov(TempGP_f, y, y_{current}^*, \mathbf{x})$
- 19 $EI_x(\mathbf{x}) = Average(EI_y)$
- 20 // Constraint Optimization
- 21 $GP_{c,fused,1} = Fusion(GP_c, X_{test,feas})$
- 22 $X_c \leftarrow [X_c, \mathbf{c}]$
- 23 **for** $c \in GpSample(TempGP_c, \mathbf{x})$ **do**
- 24 $TempGP_c = GpTrain(X_c, c)$
- 25 $GP_{c,fused,2} =$
 $Fusion(GP_c, TempGP_c, X_{test,feas})$
- 26 $IG_c(c) =$
 $InfoGain(GP_{c,fused,1}, GP_{c,fused,2}, c, \mathbf{x})$
- 27 $IG_x(\mathbf{x}) = Average(IG_c)$
- 28 // $X_{alt,feas}$ **for loop ends here**
- 29 $i_f, \mathbf{x}_{i,f} = Max(EI_x/cost_f)$
- 30 $i_c, \mathbf{x}_{i,c} = Max(IG_x/cost_c)$
- 31 // Find best design in current iteration
- 32 $GP_{f,fused} = Fusion(GP_f, X_{test,feas})$
- 33 $\mathbf{x}^*, y_{current}^*, \sigma_{current}^* =$
 $FindBest(GP_{f,fused}, X_{ind,feas})$
- 34 // Update Appropriate GPs and fidelity variance
- 35 $GP_{i,f} = GpTrain([X_{i,f}, \mathbf{x}_{i,f}])$
- 36 $GP_{i,c} = GpTrain([X_{i,c}, \mathbf{x}_{i,c}])$
- 37 $\sigma_{f,f} = FidelityVariance(GP_{GT,f}, GP_f)$
- 38 $\sigma_{f,c} = FidelityVariance(GP_{GT,c}, GP_c)$
- 39 $cost \leftarrow cost + cost_f + cost_c$
- 40 // Update Ground Truth GP
- 41 **if** $iter \% 10 == 0$ **then**
- 42 $GP_{GT,f} = GpTrain([X_{GT,f}, \mathbf{x}^*])$
- 43 $GP_{GT,c} = GpTrain([X_{GT,c}, \mathbf{x}^*])$
- 44 $cost \leftarrow cost + cost_f + cost_c$

Output: $\mathbf{x}^*, y^*, \sigma^*$

$$(\mathbf{x}_{N+1}, i_{N+1}) = \underset{i \in [1, \dots, S], \mathbf{x} \in X_{feas}}{\operatorname{argmax}} \frac{EI_{\mathbf{x},i}}{C_{\mathbf{x},i}}$$

Constraint Optimization is carried out in a similar manner. The acquisition function of choice is the Information Gain computed by KL-Divergence. It is given by Equation 16. The KL-Divergence metric is a quantitative metric measuring the difference between two distributions (shown in the Equation as the current fused GPR $\hat{g}_j^{fused}(\mathbf{x}'_s)$ and the temporary fused GPR $\hat{g}_{\mathbf{x},i,j}^{fused,r}(\mathbf{x}'_s)$).

$$IG_{\mathbf{x},i}^j = \frac{1}{N_r} \sum_{r=1}^{N_r} \sum_{s=1}^{N_s} D_{KL} \left(\hat{g}_{\mathbf{x},i,j}^{fused,r}(\mathbf{x}'_s) \parallel \hat{g}_j^{fused}(\mathbf{x}'_s) \right) \quad (16)$$

Ghoreishi-Thomison et. al.[35] provides the formulation for computing this metric analytically. For a lack of space, this equation is omitted here but can be found in the IPython notebook for this implementation ("Bayesian_Optimization.ipynb").

In each iteration of the constraint optimization, the current low-fidelity constraint GPRs are fused to create the initial fused constraint GPR (showed as $GP_{c,fused,1}$). Then one of the low-fidelity constraint GPRs is used to sample points at the current alternate feasible design point. Similar to the objective optimization, the each sample is used as the response for the alternate feasible design point and is appended to the training dataset for the sampling GPR. A new temporary constraint GPR (shown as $TempGP_c$) is constructed and fused with the other constraint GPR to create the temporary fused constraint GPR (shown as $GP_{c,fused,2}$). The information gain between $GP_{c,fused,2}$ and $GP_{c,fused,1}$ is computed for each sample from the sampling GPR and the information gain for all samples is averaged and divided by the cost for that constraint information source to obtain the information gain associated with that alternate feasible design point and that information source. The alternate design point and constraint information source corresponding to the maximum information gain to cost ratio is chosen as the next design point and next information source to query.

$$(\mathbf{x}_{N+1}^j, i_{N+1}^j) = \underset{i \in [1, \dots, G_j], \mathbf{x} \in X_g}{\operatorname{argmax}} \frac{IG_{\mathbf{x},i}^j}{C_{\mathbf{x},i}^j}$$

The appropriate information sources for the objectives and the constraints are updated by appending $(\mathbf{x}_{N+1}, i_{N+1})$ and $(\mathbf{x}_{N+1}^j, i_{N+1}^j)$ to the training datasets of the corresponding low-fidelity objective and constraint GPRs respectively and retraining those GPRs using the low-fidelity model evaluations.

The feasible indicator design dataset is used to determine the best design in that current iteration. The low-fidelity objective GPRs are fused and the fused objective GPR is used to query at each design in the feasible indicator dataset. For

robustness, the best design is chosen to be the one with the highest sum of the mean and three sigma prediction values. The corresponding mean is used in the Expected Improvement acquisition function. A memory based approach is also implemented here, which is different from the original algorithm, wherein the maximum response at each iteration is compared with the highest response value found so far. This enables a reduction in the reliance on the exploratory nature of the algorithm to find the optimum. However, if the goal is also to improve the accuracy of the lower fidelity objective GPRs, a different acquisition function must be used. This leads to the usage of the two step Knowledge Gradient acquisition function in the original algorithm.

An additional modification of the original algorithm implemented here is the process of updating the ground truth GPRs every 10 iterations. The idea is that updating the ground truth GPRs with the current optimum enables a more accurate prediction of the objective and the constraint in the region around the optimum since the fidelity variances for the low-fidelity models will be updated to reflect the same.

Once the budget is exhausted, the current best design, best estimated objective and the uncertainty associated with the best objective are outputted from the algorithm. This design is guaranteed to satisfy the constraint since it is deemed robustly feasible by the fused constraint GPR. The next subsection discusses the Winkler method for fusion of two information sources in more detail, which sheds light on the ability of this optimization algorithm to determine the optimum efficiently.

5.1.1 Winkler Fusion

The method to fuse two information sources in general is discussed in detail in Winkler[8]. The idea is that the degree of influence of the predictions of each information source on the fused prediction depends on the uncertainty of those predictions. In other words, information sources which are more accurate at a certain region in the design space will exert greater influence on the fused prediction. This point is elaborated upon below.

The fused Winkler predicted mean and variance at a given design point is given by Equations 17 and 18 respectively.

$$\mu_{Wink}(\mathbf{x}) = \frac{\mathbf{e}^T \tilde{\Sigma}^{-1}(\mathbf{x}) \boldsymbol{\mu}(\mathbf{x})}{\mathbf{e}^T \tilde{\Sigma}^{-1}(\mathbf{x}) \mathbf{e}} \quad (17)$$

$$\sigma_{Wink}^2(\mathbf{x}) = \frac{1}{\mathbf{e}^T \tilde{\Sigma}^{-1}(\mathbf{x}) \mathbf{e}} \quad (18)$$

where $\boldsymbol{\mu}(\mathbf{x}) = [\mu_1(\mathbf{x}) \ \mu_2(\mathbf{x}) \ \dots \ \mu_S(\mathbf{x})]^T$ is the vector of the predicted means of the S information sources to fuse, $\mathbf{e} = \mathbf{1}^S$ and

$$\tilde{\Sigma}_{ij}^{-1}(\mathbf{x}) = \rho_{ij}(\mathbf{x}) \sigma_i^2(\mathbf{x}) \sigma_j^2(\mathbf{x})$$

is the correlated covariance matrix for the S information sources. The σ values represent the total variance for each

information source. $\rho_{ij}(\mathbf{x})$ is the correlation coefficient which captures the amount of deviation between the predictions of each information source at the design point. This correlation coefficient is computed using the theory from Thomason et. al.[36]. Details about the computation of the correlation coefficient can be found in that paper. The correlation coefficient at any design point is given by Equation 19.

$$\rho_{ij}(\mathbf{x}) = \frac{\sigma_j^2(\mathbf{x})}{\sigma_i^2(\mathbf{x}) + \sigma_j^2(\mathbf{x})} \tilde{\rho}_{ij}(\mathbf{x}) + \frac{\sigma_i^2(\mathbf{x})}{\sigma_i^2(\mathbf{x}) + \sigma_j^2(\mathbf{x})} \tilde{\rho}_{ji}(\mathbf{x}) \quad (19)$$

where $\sigma_i^2(\mathbf{x})$ is the total variance of the i th information source at design \mathbf{x} and $\tilde{\rho}_{ij}$ can be computed by reifying the models i and j using the following formulation

$$\tilde{\rho}_{ij}(\mathbf{x}) = \frac{\sigma_i(\mathbf{x})}{\sqrt{(\mu_i(\mathbf{x}) - \mu_j(\mathbf{x}))^2 + \sigma_i^2(\mathbf{x})}}$$

$\tilde{\rho}_{ji}$ is computed by switching the indices. It can be seen that the correlation metric in Equation 19 is the variance weighted average of the $\tilde{\rho}_{ij}$ and $\tilde{\rho}_{ji}$ coefficients. $\tilde{\rho}_{ij}$ represents a comparative uncertainty of model i with respect to model j . This metric influences an element in the covariance matrix based on the uncertainty of the other model(s) in their predictions at design \mathbf{x} . Essentially the comparative uncertainty $\tilde{\rho}_{ij}$ will have a higher influence on the covariance matrix element if $\sigma_j(\mathbf{x}) > \sigma_i(\mathbf{x})$, so the model with less predictive uncertainty at any design point will govern the fused predictive uncertainty at that point.

To enable the use of this combined knowledge from the information sources, the fused Winkler means at the feasible test points are used to construct a fused GPR for the objectives and constraints. A demonstration of the Winkler Fusion process is provided in "Winkler_fusion.ipynb" using the two low-fidelity objective models from the test problem.

Now that the optimization algorithm has been discussed in detail, the next subsections deal with the training of the data-driven surrogate and the implementation of this algorithm on both a test problem and the composite plate problem. We start with the discussion on the process of generating the training data for the neural network.

5.2 Generation of Training Data

The neural network is trained using high-fidelity evaluations from Abaqus python script (named "abaqus_fea_macro.py"). 2000 design datapoints are generated using LHS within the following design space

$$25 \leq r_f \leq t/2$$

$$0 \leq \theta_f \leq 90$$

$$10 \leq E_m \leq 800$$

The fibre radius r_f is measured in mm, fibre angle θ_f is measured in degrees and matrix Young's Modulus E_m is

measured in GPa. It must be noted that the design space for θ_f was changed from $[-90, 90]$ to $[0, 90]$ since some negative angle cases would result in the overlap of fibres which would cause runtime issues in the Abaqus python script. Each generated design is evaluated through the command window as a subprocess, with the design parameters being provided as command line arguments. The Abaqus python script outputs the buckling eigenvalues as a string. This string is passed through the standard output pipe of the subprocess and read by the python script for training dataset generation. The string is parsed for the eigenvalues, which are stored alongside the corresponding design parameters in a csv file. The code carrying out this implementation can be found in "training_dataset_generation.py". The csv file containing the neural network training dataset is named "NeuralNet_training_dataset.csv". Due to the re-meshing cycle limit, 250 out of the 2000 designs could not be evaluated and were excluded from the training dataset. The exclusion takes place in the code for training the neural network and the unevaluated datasets are part of the "NeuralNet_training_dataset.csv". For these designs, in the eigenvalue fields "Error: cae exited with an error" is printed, which is a direct output from the Abaqus python script. The generation of the training dataset took 78182.43 seconds in run time. Now that the training dataset is generated, the next subsection discusses the process of training the neural network.

5.3 Neural Network Training

The remaining 1750 designs are used to train the neural network. The neural network is trained to regress 3 eigenvalues. A k-fold cross validation procedure is employed to determine the optimal neural network. The 1750 datapoints are first pre-processed to centre them at the mean and scale them to unit variance. The mean and variance of the pre-processed data is saved to a csv file (named "normalization_constants.csv") in order to scale the outputs of the neural network to the true eigenvalues. Following this, the datapoints are split into training and testing datasets in 80-20 proportion. The number of folds for k-fold cross validation is chosen as 10 and training and validation datasets are generated at random in 80-20 proportion for each of the 10 folds. For each fold, a neural network is trained using the training dataset and evaluated using both the validation and testing datasets. The batch size is set to 64 and the number of training epochs to 150. The chosen architecture for the neural network is shown in Figure 29. The actual image (named "Neural_Net_norm.png") is also provided separately in case the image in the report is not very legible.

A large architecture size was observed to be favourable in reducing the final training mean square error (MSE) for the folds. The results of the k-fold cross validation based determination of the best neural network is shown in Figure 23. The oscillation of the validation MSE around the training MSE across the different folds is very strange, however the

strangest observation is the near constant and consistently low value of the test MSE for all folds. The condition to determine the optimal neural network was the lowest test MSE. However, the consistent value of the test MSE across all fold makes it difficult to justify choosing one over the other. Nevertheless, the neural net corresponding to fold 3 was determined to be optimal. Figure 24 shows the MSE as a function of epochs for that particular fold.

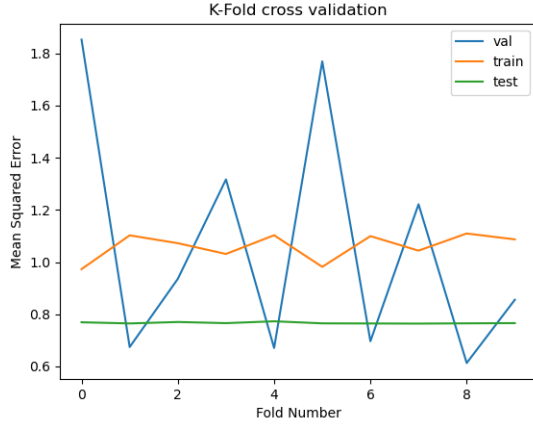


Figure 23: Results of the k-fold training and evaluation of the neural networks. It can be observed that for different folds the validation MSE oscillates around the training MSE. But the constant and consistently low test MSE is extremely strange.

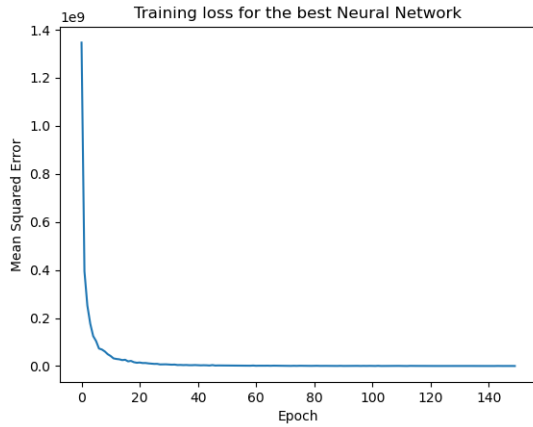


Figure 24: MSE for the best neural network as a function of the epochs. It can be seen that the MSE quickly converges to a minimum after around 20 epochs

The near constant test MSE combined with the fast convergence of the MSE for the best neural network, starting from a suspiciously high MSE value for normalized data which is not very high in number warranted a comparison of the predictions of the best neural network with the true

outputs for the test dataset. The sum of the three eigenvalues was used for comparison purposes. Figure 25 shows this very comparison. It is clear that the neural network predicts the same value for different designs. This could be attributed to a local minima, size of the training dataset and/or the inability of the current network architecture to capture the complexity of the mapping between the designs and the objectives. Multiple network architectures were used and the learning rate for the Adam (reference Kingma et. al.[37]) was also adjusted to different values. The AMS-Grad variant of Adam, introduced in Reddi et. al.[38] was also employed but to no avail. In any case, further investigation into this behaviour is necessary and will be carried out in the future.

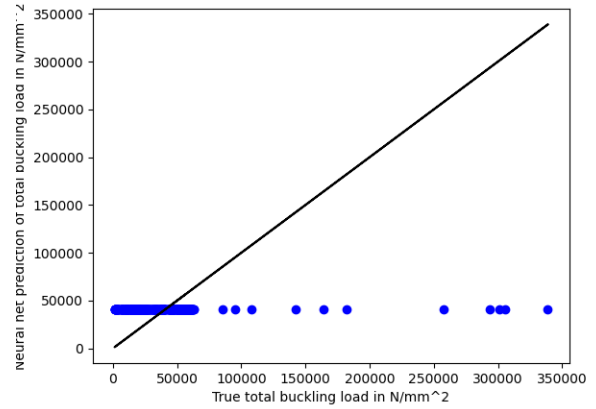


Figure 25: Comparison of the predicted outputs from the best neural network with the true outputs. It can be seen that the neural network is predicting the same value for different designs. This could be attributed to a local minima, size of the training dataset and/or the inability of the current network architecture to capture the complexity of the mapping between the designs and the objectives.

This behaviour of the neural network creates issues during the actual implementation of the optimization algorithm on the composite plate problem and shall be discussed in that subsection. First, the next subsection presents the results of the implementation of the optimization algorithm on a test problem. The code for training the neural network can be found in "Neural_net_training_kfold.py".

5.4 Implementation on Test Problem

The MFCBO algorithm is implemented on a one-dimensional test problem taken from Ghoreishi-Allaire[6]. The goal of the problem is formally stated as the following

$$x^* = \underset{x \in [0,1.2]}{\operatorname{argmax}} -(1.4 - 3x)\sin(18x)$$

subject to the following constraint

$$x^2 - 1.2 \leq 0$$

Figure 26 illustrates the objective and constraint curves. The feasible optimum for this problem is given as $(x^*, f(x^*)) = (1.0954, 1.4388)$.

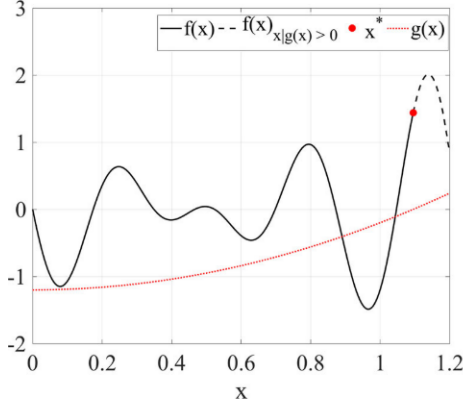


Figure 26: Plot showing the objective $f(x)$ and constraint $g(x)$ curves. It can be observed that the feasible optimum is around the $x=1.1$ region. The image is taken from Ghoreishi-Allaire[6]

The above functions are taken as the high-fidelity objective and constraint functions respectively. It is assumed that two low-fidelity models are available for computation of $f(x)$, which are

$$f_1(x) = -(1.6 - 3x)\sin(18x)$$

$$f_2(x) = -(1.8 - 3x)\sin(18x + 0.1)$$

Similarly, two low-fidelity constraint models are also available, they are shown below

$$g_1(x) = (x - 0.001)^2 - 1.2$$

$$g_2(x) = (x + 0.02)^2 - 1.2$$

The original problem in Ghoreishi-Allaire[6] only considers the cost of the low-fidelity models and assumes constant fidelity variances for the low-fidelity objectives and constraints. In this research, the cost of the ground truth queries is also considered and the fidelity variances are dynamically computed in each iteration.

The number of alternate and test samples to generate in each iteration is set to 30 and 50 respectively. The number of indicator points to sample is set to $1000d$ where d is the dimension of the design vector. The initial number of samples to train the ground truth GPRs for the objectives and constraints is set to 10 each, whereas the GPRs low-fidelity models for the objectives and constraints were trained using 40 samples each. The cost of querying the objective ground truth $f(x)$ is set at 100, whereas the cost of querying low-fidelity models $f_1(x)$ and $f_2(x)$ is set at 25 and 20 respectively. The cost of querying the constraint ground truth $g(x)$ is set at 40, whereas the cost of querying

low-fidelity models $g_1(x)$ and $g_2(x)$ is set at 10 and 5 respectively. The maximum budget is set to 5000. Figures 27 and 28 illustrate the highest function values and their corresponding constraint values for each iteration with their 3σ uncertainty bounds. The constraint values are computed using the fused constraint GPR for that iteration.

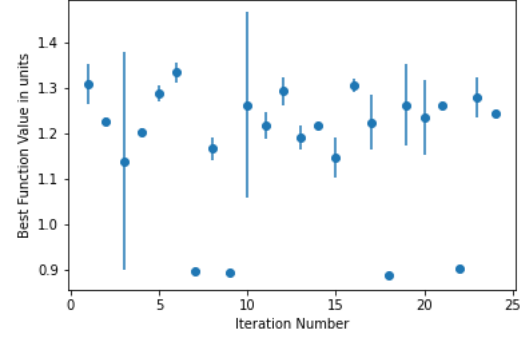


Figure 27: Plot showing the optimal function values obtained in each iteration along with their 3σ uncertainty bounds. It can be seen that the optimal function values are close to the true optimum and the uncertainty bounds reduce consistently as the number of iterations increases.

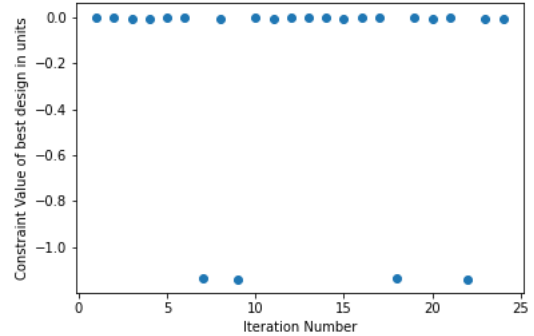


Figure 28: Plot showing the constraint values for the optimal designs obtained in each iteration along with their 3σ uncertainty bounds. It can be seen that the constraint values corresponding the optimal function values near the true optimum have a near zero constraint value, which is expected according to Figure 26

The algorithm took 24 iterations to exhaust its budget and was able to obtain the optimal design of 1.09597 but estimates the optimal function value as 1.33565 with a standard deviation of 0.00767. It is reasonable to say that a higher optimum could be obtained on an averaged basis over multiple runs with the same budget. The four optimal values that are noticeably lower than the rest can be attributed to the feasible indicator dataset not being large enough. Since the optimum is so close to the constraint surface, it is possi-

ble that the fused constraint GPR did not deem a potentially optimal point as feasible. However, it can be seen that the fused constraint GP estimates the constraint values for the optimal designs very well.

The next subsection details the implementation of the MFCBO algorithm on the composite plate optimization problem.

5.5 Implementation on Composite Plate Problem

For the implementation of the MFCBO algorithm on the composite plate problem, the length l and width w are set to 10ft.(3048 mm) and the thickness t is set to 254mm. The fibre Young's Modulus E_f is set at 121 GPa (the Young's Modulus of copper). These values are the same as in the Abaqus python script. The range of the design variables was chosen to be the following

$$25 \leq r_f \leq (t/2) - 20$$

$$0 \leq \theta_f \leq 90$$

$$10 \leq E_m \leq 800$$

As usual, r_f is measured in mm, θ_f is measured in degrees and E_m is measured in GPa. The objective to optimize is chosen to be a measure of the effective buckling load.

$$f(\mathbf{x}) = P_{cr,e1}(\mathbf{x}) + 0.5P_{cr,e2}(\mathbf{x}) + 0.25P_{cr,e3}(\mathbf{x})$$

where $P_{cr,e1}$, $P_{cr,e2}$ and $P_{cr,e3}$ are the first, second and third buckling eigenvalues respectively.

To reiterate, the constraint is set on the volume fraction and is given as

$$g(\mathbf{x}) = v_f(\mathbf{x}) - 0.75$$

It is assumed that two low-fidelity versions of $g(\mathbf{x})$ are available. They are

$$g_1(\mathbf{x}) = \frac{r_f^2}{(1000\cos(\theta_f))} - 0.75$$

$$g_2(\mathbf{x}) = \frac{r_f^2}{(1000\sin(\theta_f) + 0.1)} - 0.75$$

The first term in $g_1(\mathbf{x})$ is the same as the volume fraction except for a multiplicative constant and $g_2(\mathbf{x})$ is a variation on $g_1(\mathbf{x})$. It must be noted here that the constraint models use θ_f in radians.

The number of alternate and test samples to generate in each iteration is set to 100 and 150 respectively. The number of indicator points to sample is set to $1000d$ where d is the dimension of the design vector. The initial number of samples to train the ground truth GPRs for the objectives and constraints is set to 10 each, whereas the GPRs low-fidelity models for the objectives and constraints would be trained using 50 samples each. The cost of querying the objective ground truth is set at 2000, whereas the cost of querying

low-fidelity analytical and neural network models is set at 50 and 30 respectively. The cost of querying the constraint ground truth $g(\mathbf{x})$ is set at 100, whereas the cost of querying low-fidelity models $g_1(\mathbf{x})$ and $g_2(\mathbf{x})$ is set at 20 and 10 respectively. The maximum budget is set to 40000.

Due to the presence of the 3 re-mesh cycle limit, it is highly possible that some of the designs would not be evaluated by Abaqus. In this case, the design would be evaluated using the analytical model. This would affect the computation of the fidelity variance for the analytical model but would ensure that the algorithm continues to operate.

As discussed in the neural network training subsection, the best neural network obtained returns the same output for all inputs. As a result, the GPR trained using the responses from this neural network will have zero variance. This implies a zero value for expected improvement for any of the samples. As a result, there is no clear candidate design for query and the algorithm throws an error here. As a result the deployment of the MFCBO algorithm for the composite plate optimization algorithm was unsuccessful. It is a firmly held belief that the neural network must be rectified for the algorithm to show some results. Further investigation will be done in this respect.

6 Conclusions

The problem of finding the optimal fibre radius, fibre angle and matrix Young's Modulus for a composite plate with maximum buckling load was defined and its bounds and constraints formulated. The problem was analyzed through the technological, materials and Olson's Framework perspectives. A short literature survey was conducted exploring the relationships between the chosen design parameters and the chosen objective and the knowledge of these relationships from literature was used to inform the discussion on the balance of design space exploration and exploitation as carried out by the acquisition functions for the Bayesian Optimization algorithm. The forward maps from the design space to the objective space, as captured by the analytical model, were observed and the performance of different regressors was investigated. A comparison of different kernels for the GPR resulted in the Matern kernel being chosen for the implementation of the optimization algorithm. The Multi-Fidelity Constrained Bayesian Optimization (MFCBO) algorithm was introduced and discussed in detail. The generation of the training data and the training process and results for the neural network were presented. Finally, the results of the implementation of the MFCBO algorithm on a test problem and the composite plate problem were presented and discussed.

Some clear conclusions have emerged from this study. Firstly, the MFCBO algorithm is well suited for a problem of this type as it takes into account the correlation between different low-fidelity models and only uses the GPRs in the Bayesian Optimization setting for the objective and con-

straint. Secondly, the training of the neural network can be especially tricky when using stochastic gradient descent. If a network architecture falls into a local minima during training, it can be difficult to determine the optimal architecture for the problem. Thirdly, the Abaqus FEA module implemented using python script must be improved in order to avoid the implementation of the re-meshing cycle. Finally, the current version of the algorithm, although well performing on the test problem, can be improved by using different acquisition functions.

7 Future Work

Some directions for future work include the following:

- In light of the last conclusion, the algorithm will be tested using the knowledge gradient acquisition function to compare its gains as compared to the expected improvement function. Other acquisition functions aimed at traversing the design space towards the optimum will also be tested.
- A clear area for further investigation is the training of the neural network. To avoid the local minima, a population or particle swarm based optimization method will be considered to either determine the optimal weights for a particular architecture or determine the optimal architecture and its weights for this particular problem.
- The Abaqus python script must be modified to get around the generation of zero volume mesh elements. Either an adaptive re-meshing strategy must be used or provisions must be made to remove the zero volume mesh elements as they are generated.
- The MFCBO algorithm can be generalized to incorporate the effects of manufacturing defects by modelling the design parameters as random variables and considering the effect of their uncertainties. An effort will be made to explore this direction of research.

8 Acknowledgements

The author would like to graciously thank Danial Khatam-saz of the Department of Mechanical Engineering at Texas A&M University for his help and guidance in understanding and modifying the Bayesian Optimization algorithm and Dr. Raymundo Arroyave of the Department of Materials Science and Engineering at Texas A&M University for his guidance, suggestions and support for the duration of this project.

References

- [1] J. Snoek, Hugo Larochell, and R. P. Adams, "Practical Bayesian Optimization of Machine Learning Algorithms,"
- [2] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas, "Taking the human out of the loop: A review of Bayesian optimization," *Proceedings of the IEEE*, vol. 104, no. 1, pp. 148–175, 2016.
- [3] P. I. Frazier, "A Tutorial on Bayesian Optimization," no. Section 5, pp. 1–22, 2018.
- [4] S. F. Ghoreishi, A. Molkeri, A. Srivastava, R. Arroyave, and D. Allaire, "Multi-Information Source Fusion and Optimization to Realize ICME: Application to Dual-Phase Materials," *Journal of Mechanical Design, Transactions of the ASME*, vol. 140, no. 11, pp. 1–14, 2018.
- [5] S. F. Ghoreishi, A. Molkeri, R. Arroyave, D. Allaire, and A. Srivastava, "Efficient use of multiple information sources in material design," *Acta Materialia*, vol. 180, pp. 260–271, 2019.
- [6] S. F. Ghoreishi and D. Allaire, "Multi-information source constrained Bayesian optimization," *Structural and Multidisciplinary Optimization*, vol. 59, no. 3, pp. 977–991, 2019.
- [7] M. F. Ashby, "Multi-objective optimization in material design and selection," *Acta Materialia*, vol. 48, no. 1, pp. 359–369, 2000.
- [8] R. L. Winkler, "Combining Probability Distributions from Dependent Information Sources,"
- [9] K. K. Chawla, "The Applicability of the "Rule-of-Mixtures" to the Strength Properties of Metal-Matrix Composites," *Revista Brasileira de Física*, vol. 4, no. 3, pp. 411–418, 1974.
- [10] W. Shan, T. Lu, and C. Majidi, "Soft-matter composites with electrically tunable elastic rigidity," *Smart Materials and Structures*, vol. 22, no. 8, 2013.
- [11] N. G. Cheng, A. Gopinath, L. Wang, K. Iagnemma, and A. E. Hosoi, "Thermally tunable, self-healing composites for soft robotic applications," *Macromolecular Materials and Engineering*, vol. 299, no. 11, pp. 1279–1284, 2014.
- [12] I. S. Bayer, M. K. Tiwari, and C. M. Megaridis, "Bio-compatible poly(vinylidene fluoride)/cyanoacrylate composite coatings with tunable hydrophobicity and bonding strength," *Applied Physics Letters*, vol. 93, no. 17, pp. 1–4, 2008.
- [13] A. F. Visentin and M. J. Panzer, "Poly(ethylene glycol) diacrylate-supported ionogels with consistent capacitive behavior and tunable elastic response," *ACS Applied Materials and Interfaces*, vol. 4, no. 6, pp. 2836–2839, 2012.

-
- [14] Z. Zhang, C. Yao, Y. Yu, Z. Hong, M. Zhi, and X. Wang, "Mesoporous Piezoelectric Polymer Composite Films with Tunable Mechanical Modulus for Harvesting Energy from Liquid Pressure Fluctuation," *Advanced Functional Materials*, vol. 26, no. 37, pp. 6760–6765, 2016.
- [15] M. Arian Nik, K. Fayazbakhsh, D. Pasini, and L. Lessard, "Optimization of variable stiffness composites with embedded defects induced by Automated Fiber Placement," *Composite Structures*, vol. 107, no. 1, pp. 160–166, 2014.
- [16] M. Y. Matveev, P. J. Schubel, A. C. Long, and I. A. Jones, "Understanding the buckling behaviour of steered tows in Automated Dry Fibre Placement (ADFP)," *Composites Part A: Applied Science and Manufacturing*, vol. 90, pp. 451–456, 2016.
- [17] T. J. Dodwell, R. Butler, and A. T. Rhead, "Optimum fiber steering of composite plates for buckling and manufacturability," *AIAA Journal*, vol. 54, no. 3, pp. 1139–1142, 2016.
- [18] T. L. Wu, K. K. Shukla, and J. H. Huang, "Post-buckling analysis of functionally graded rectangular plates," *Composite Structures*, vol. 81, no. 1, pp. 1–10, 2007.
- [19] Y. Zhu, Y. Qin, S. Qi, H. Xu, D. Liu, and C. Yan, "Variable Angle Tow reinforcement design for locally reinforcing an open-hole composite plate," *Composite Structures*, vol. 202, no. January, pp. 162–169, 2018.
- [20] N. Hamani, D. Ouinas, N. Taghezout, M. Sahnoun, and J. Viña, "Effect of fiber orientation on the critical buckling load of symmetric composite laminated plates," *Advanced Materials Research*, vol. 629, pp. 95–99, 2013.
- [21] G. Raju, Z. Wu, and P. M. Weaver, "Buckling analysis of Variable Angle Tow composite plates using Differential Quadrature Method," *ECCM 2012 - Composites at Venice, Proceedings of the 15th European Conference on Composite Materials*, no. June, pp. 24–28, 2012.
- [22] B. Almroth, "Design of composite material structures for buckling (state of art)," tech. rep., Wright-Patterson Air Force Base, Ohio, 1981.
- [23] B. Singh, N. Iyengar, and D. Yadav, "Effects of Random Material Properties on Buckling of Composite Plates," *Journal of Engineering Mechanics*, vol. 127, no. September, pp. 873–879, 2001.
- [24] Z. Zhang and K. Friedrich, "Artificial neural networks applied to polymer composites: A review," *Composites Science and Technology*, vol. 63, no. 14, pp. 2029–2044, 2003.
- [25] M. Hidayat and P. Yusoff, "Optimizing Neural Network Prediction of Composite Fatigue Life under Variable Amplitude Loading using Bayesian Regularization," in *Composite Materials Technology - Neural Network Applications*, ch. 9, pp. 222–248.
- [26] W. Hu, W. Zhao, Y. Wang, Z. Liu, J. Cheng, and J. Tan, "Design optimization of composite wind turbine blades considering tortuous lightning strike and non-proportional multi-axial fatigue damage," 2019.
- [27] A. L. Araújo, C. M. Mota Soares, and M. J. Moreira De Freitas, "Characterization of material parameters of composite plate specimens using optimization and experimental vibrational data," 1996.
- [28] S. Huang and P. Qiao, "A novel semi-analytical method for buckling analysis of stiffened laminated composite plates," *Thin-Walled Structures*, vol. 148, no. August 2019, p. 106575, 2020.
- [29] L. Grosset, R. Le Riche, and R. T. Haftka, "A double-distribution statistical algorithm for composite laminate optimization," *Collection of Technical Papers - AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, vol. 6, pp. 4191–4202, 2004.
- [30] V. J. Papazoglou, N. G. Tsouvalis, and A. G. Lazaridis, "A non-destructive evaluation of the material properties of a composite laminated plate," *Applied Composite Materials*, vol. 3, no. 5, pp. 321–334, 1996.
- [31] A. S. Milani, H. Wang, D. D. Frey, and R. C. Abeyaratne, "Evaluating three DOE methodologies: Optimization of a composite laminate under fabrication error," *Quality Engineering*, vol. 21, no. 1, pp. 96–110, 2009.
- [32] D. Brush, B. Almroth, and J. Hutchnison, *Buckling of Bars, Plates and Shells*. 1975.
- [33] C. E. Rasmussen and C. K. Williams, *Gaussian Processes for Machine Learning*. MIT Press, 2006.
- [34] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient Global Optimization of Expensive Black-Box Functions," , vol. 13, no. 4, pp. 455–492, 1998., *Journal of Global Optimization*, vol. 13, pp. 455–492, 1998.
- [35] S. F. Ghoreishi, W. D. Thomison, and D. L. Allaire, "Sequential information-theoretic and reification-based approach for querying multi-information sources," *Journal of Aerospace Information Systems*, vol. 16, no. 12, pp. 575–587, 2019.
- [36] W. D. Thomison and D. Allaire, "A model reification approach to fusing information from multifidelity information sources," *19th AIAA Non-Deterministic Approaches Conference, 2017*, no. January, 2017.

- [37] D. P. Kingma and J. L. Ba, “Adam: A method for stochastic optimization,” *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, pp. 1–15, 2015.
- [38] S. J. Reddi, S. Kale, and S. Kumar, “On the convergence of Adam and beyond,” *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*, pp. 1–23, 2018.

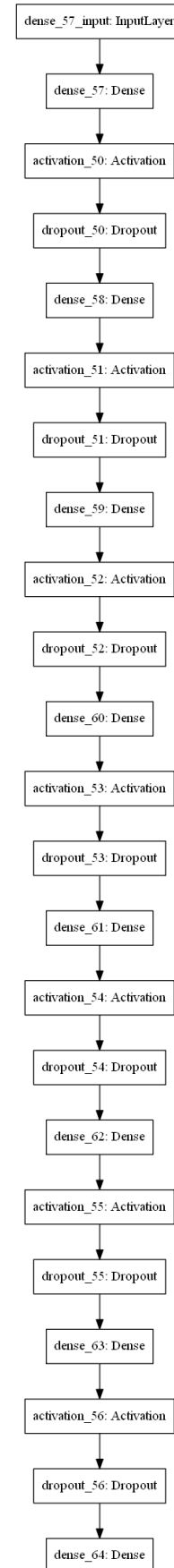


Figure 29: Architecture of the Neural Network trained using the k-fold cross validation approach