

POLITECNICO DI TORINO

Corso di Laurea
in Ingegneria Aerospaziale

Tesi di Laurea Magistrale

Multi-fidelity methods for health monitoring of aerospace structures



Relatori

Dott. Ing. Laura Mainini
Prof. Marco Gherlone

Candidato

Matteo Fenoglio

Anno Accademico 2022-2023

Abstract

In engineering applications, common problems rely on optimization: improving system or structure design, health monitoring with damage assessment and control problems. Modern approaches solve these problems using computer simulations to find the best design among all the possibilities or to find the right damage parameters configuration. High-fidelity simulations, like the finite element in structural design or CFD codes for fluid dynamics, have high computational cost that can be computationally untractable.

To accelerate the optimization process two main approaches can be used to reduce the overall execution cost: use a small number of model evaluations and use cheaper, but with reduced accuracy, numerical models.

In this work of thesis we combine these two techniques: we employ a surrogate based optimization which is built along the optimization process by choosing the most informative sample reduce the number of function evaluations. Furthermore, the other main feature consists in using different simulation fidelity levels; the lower and cheaper ones purpose is to extensively explore the design space, the highest is used to improve the overall accuracy of the surrogate model in specific points. More in detail is used a Bayesian Multi-fidelity framework where the surrogate is modeled with a Gaussian Process, it merges and fuses the fidelities sampling in the design space using the acquisition functions: these indicate the next point to sample with the relative fidelity to maximise information gain. In this work of thesis we use and compare 3 different acquisition functions: Probability of Improvement, Expected Improvement and Max value Entropy Search; then we apply them on a damage identification problem in a composite specimen.

We consider a composite carbon fiber specimen with a modeled cut in the fiber direction that means discontinuity in the stress transmission; composites damages are critical because often difficult to spot but reduces greatly the structural strength hence the monitoring is crucial to ensure the health of the structure. The problem uses the strain reference field of a damage configuration and the algorithm actively builds a surrogate model that minimizes the Root Mean Square Error between the reference and the surrogate strain fields to find the cut geometrical properties.

We observe major reduction in cost, and number of evaluations, respect to the singlefidelity counterpart.

Contents

List of Tables	3
List of Figures	4
1 Introduction	6
2 Surrogate modeling	10
2.1 Design Of Experiment (DOE)	11
2.2 Building the surrogate	13
2.3 Multifidelity surrogate	15
2.4 Multifidelity surrogate based optimization	17
3 Bayesian Optimization	20
3.1 Gaussian Process	22
3.1.1 Priors distribution	22
3.1.2 Covariance function	23
3.2 Acquisition functions	25
3.2.1 Probability of Improvement (PI)	25
3.2.2 Expected Improvement (EI)	26
3.2.3 Predictive Entropy Search (PES)	27
3.2.4 Max-Value entropy search	28
4 Multifidelity Bayesian optimization	29
4.1 Multifidelity Probability of Improvement	30
4.2 Multifidelity Expected Improvement	31
4.3 Multifidelity Predictive Entropy Search	32
4.4 Multifidelity Max-value Entropy Search	32
5 Numerical implementation of Bayesian Framework	35
5.1 Single fidelity Bayesian Optimization	35
5.2 Multifidelity Bayesian Optimization	37

6	Composite specimen damage identification	41
6.1	Theoretical background	43
6.2	Patran modeling	44
6.3	Low fidelity model	44
6.4	Mid fidelity model	46
6.5	High fidelity model	47
6.6	Border cut damage	48
6.7	Small cut damage	49
6.8	Large cut damage	50
7	Problem set up	51
8	Results	56
8.1	Aggregated results	57
8.2	Cut damage on border	61
8.2.1	Strain field evolution	64
8.3	Small cut damage	68
8.3.1	Strain field evolution	71
8.4	Large cut damage	75
8.4.1	Strain field evolution	78
9	Conclusions	82

List of Tables

5.1	Single fidelity framework	36
5.2	MFEI pseudocode implementation	38
5.3	MFPI pseudocode implementation	39
5.4	MFMES pseudo code	40
6.1	Hexcel IM7/8552 AS4 Plain Weave mechanical properties	42
7.1	Model fidelities description	52
7.2	Cut damage properties	52
7.3	Damage test conditions, measures expressed in mm	52
8.1	Mean evaluations times and mean number of evaluations for each fidelity	58
8.2	Mean evaluations times and mean number of evaluations for each fidelity level	62
8.3	Mean evaluations times and mean number of evaluations for each fidelity	69
8.4	Mean evaluations times for each fidelity	76

List of Figures

2.1	Full factorial sampling plan and a projection on the x1-x2 plane . . .	12
2.2	Latin Hypercube sampling plan and a projection on the x1-x2 plane	13
3.1	Example of a 1D problem optimization [57]	21
3.2	Bayes optimization process	22
3.3	A 1D function approximated with a Gaussian Process, the actual max value is for the sample \mathbf{x}^+ and the shaded green area over the dashed horizontal line indicates the improvement of sampling at \mathbf{x}_3 : it can be clearly seen that sampling at \mathbf{x}_1 or \mathbf{x}_2 doesn't give any improvement finding the max value.[57]	26
6.1	Specimen dimensions and lamination stacking sequence	42
6.2	Example of strain distribution through the thickness of a laminate [77]	43
6.3	Specimen boundary conditions and mesh	45
6.4	Mesh bias on the specimen direction of increasing dimension on the specimen	46
6.5	Low fidelity 2D model mesh	46
6.6	Higher fidelity model with 3D elements	47
6.7	Strain field for a cut on the border of specimen with positions $x_c = 74.66mm$ $y_c = 258.58mm$ and length of $l_c = 25.74mm$	48
6.8	Strain field of a small cut on the specimen with position $x_c = 86.11mm$ $y_c = 189.30mm$ and length of $l_c = 0.96mm$	49
6.9	Strain field of a large cut on the specimen $x_c = 11.86mm$ $y_c = 203.97mm$ and length of $l_c = 59.34mm$	50
7.1	Damage test samples distribution. Orange circle small cut. Yellow circle large cut. Red dot cut on the border.	53
7.2	Diagram showing the optimization framework	54
8.1	RMSE minimizing process results for each acquisition function . . .	57
8.2	Error estimating the x_c position	59
8.3	Error estimating the y_c position	59
8.4	Error estimating the l_c position	60

8.5	RMSE minimizing process results for each acquisition function cut on the border $x_c = 74.66mm$, $y_c = 258.58mm$ $l_c = 25.74$	61
8.6	Error estimating the x_c damage parameter cut on the border	62
8.7	Error estimating the y_c damage parameter cut on the border	63
8.8	Error estimating the l_c damage parameter cut on the border	63
8.9	HF Border cut strain field evolution using EI acquisition function	64
8.10	HF Border cut strain field evolution using MFEI acquisition function	65
8.11	HF Border cut strain field evolution using MFPI acquisition function	66
8.12	HF Border cut strain field evolution using FMES acquisition function	67
8.13	RMSE minimizing process results for each acquisition functions small cut $x_c = 86.11mm$ $y_c = 189.30mm$ $l_c = 0.96mm$	68
8.14	Error estimating the x_c damage parameter small cut	69
8.15	Error estimating the y_c damage parameter small cut	70
8.16	Error estimating the l_c damage parameter small cut	70
8.17	HF Small cut strain field evolution using EI acquisition function	71
8.18	HF Small cut strain field evolution using MFEI acquisition function	72
8.19	HF Small cut strain field evolution using FMES acquisition function	73
8.20	HF Small cut strain field evolution using MFPI acquisition function	74
8.21	RMSE minimizing process results for each acquisition function large cut	75
8.22	Error estimating the x_c damage parameter large cut	76
8.23	Error estimating the y_c damage parameter large cut	77
8.24	Error estimating the l_c damage parameter large cut	77
8.25	HF Large cut strain field evolution using EI acquisition function	78
8.26	HF Large cut strain field evolution using MFEI acquisition function	79
8.27	HF Large cut strain field evolution using FMES acquisition function	80
8.28	HF Large cut strain field evolution using MFPI acquisition function	81

Chapter 1

Introduction

Across different engineering branches optimization processes cover a broad range of problems like minimizing the drag by improving the design of an airfoil shape [1], optimizing the design of a wind turbine plant to reduce the wave effect [2], select the right sensor considering accuracy and cost [3]. Typically the optimization process requires to consider lots of data outputs, with different combinations of inputs, that can be expensive in terms of acquisition time and monetary cost: because the data necessary can come from real experiments, previous designs, or deterministic computer models simulations that aims to reproduce the physical behavior of the system. All the data that flows in the optimization process permits to correctly choose the best solution between many different feasible designs reducing the overall cost of production, maintenance and usage of the system. The higher is the data quality, in terms of representing with high precision and fidelity the real system, the higher is the price to obtain them. Considering using numerical models to simulate the behavior of the system, in order to reduce the data acquisition cost, even with high computational power the time to run the simulations can be extremely high.

One example can be taken from fluid dynamic simulations: the Reynolds Averaged Navier-Stokes formulation (RANS), which consists in solving several partial differential equations, requires an high amount of computational effort to obtain satisfactory results. Another example relies on the large model that simulates the oil reservoir level used by [4] that requires between 1-3 days to complete a single evaluation: it's clear that running an optimization process just through costly but extremely accurate models is not feasible.

A method to reduce the cost and the evaluation time during the optimization process is to use a *surrogate model*: it is an approximation of the real system, built from numerical models evaluations with random input data points chosen from the design domain in order to catch the global system behavior. During the optimization process the surrogate is employed instead of using the costly, but more

accurate, models leading to saving time. Two main techniques of creating a surrogate can be used: passive and active learning [5]. The first uses an offline/online phase, during which the model is trained with sampled data in the offline phase and used toward optimization in the online phase. The latter starts with a model build from a preliminary offline/online approach and iteratively improves the surrogate with new collected outputs from model evaluations [6].

Using different model fidelities, if available, permits to speed up the surrogate construction by extensively *exploring* the design space with lower and cheaper models and drive the sampling of the higher and costly fidelities that are used to *exploit* the real response of the system. The highest fidelity is considered the representation of real physical behavior of the system and enriches the observations where necessary to develop an appropriate surrogate [7].

Several examples of multifidelity surrogate modeling employed to speed up an optimization process can be found in literature. Forrester and Sóbester [8] describe the usage of a Gaussian process to built the surrogate with several level of fidelities, they focus on a wing optimization aiming to reduce the drag changing respectively the taper ratio and the swept angle. In [9] multiple RANS simulations are used to optimize the DLR wing maximizing the lift over drag ratio. The same wing model is optimized with a co-Kriging surrogate built with a low fidelity Potential method code and a high RANS simulation [10]: this permits to extensively explore the design space due to the reduced computational cost of the low fidelity model. A different set of problems is addressed by Pehlivanoglu [11], which uses a multifidelity approach to optimize the active flow control over an airfoil, optimizing also the shape in a transonic flight condition. A further step in the optimization process lays on considering different physical fields strongly interconnected in a multifidelity framework. Considering the extension to the whole aircraft the work [12] where the propulsion, structural and aerodynamic field are taken into account to minimize the fuel burn of a regional transport aircraft modifying the wing structural and aerodynamic characteristic. Moving to the space environment an application can be found in [13] where the aero-thermodynamic problem of a terrestrial reentry vehicle is addressed. An interesting application in the control field is studied in [14] where the altitude of a spacecraft is assured using an active learning based technique, considering the external disturbances, parameter uncertainty and an eventual actuator fail.

In the aerospace context another problem, which is related to damage identification, can be addressed leveraging a multifidelity approach. An example can be found in [15] where a surrogate model of a cantilevered beam is build up using two different level of fidelity finite element code to evaluate the residual strength. A classic problem is addressed by [16] which investigates the usage of different FEM models of an UAV wing to determine the damage location, evaluating the change in modal response of the system. A different approach is employed by Mainini

and Willcox [17] where the flight envelope is progressively limited considering the damage state of an UAV vehicle, making the vehicle self-aware of its condition. The surrogate model is built in the offline phase, i.e. using physic based aerostructural computer code considering different damage configurations, then is used and updated (with signals from on board sensors) during flight to predict the structure damage level.

Evaluating structural integrity, without disassembling or destructing the structure itself, is important to schedule punctual maintenance reducing the inoperative time of the aircraft. As stated in [18] the main achievements in the Structural Health Monitoring (SHM) are due to the several data algorithms developed: the so-called pattern recognition, which leverages supervised machine learning techniques, after being instructed with several healthy and damaged samples are able to indicate the health status of the structure [15].

Our approach will focus directly on a damage configuration and the framework utilised will find the more reliable damage parameters. This thesis will focus on a damage identification problem: we will evaluate the change in strain field of a composite specimen with the presence of a fiber cut in a specific layer. With different fidelity levels of finite element codes, the strain field is evaluated and used to build a surrogate: this approximation will speed up the localization process of the cut position and extension. This problem is highly multimodal, i.e. due to almost infinite combinations of cut positions and lengths which can lead to similar strain distribution on the specimen. It's difficult to estimate the problem variables (cut position) with similar outputs (the strain field). Different models, with three levels of fidelity, will be used to verify the best suited and fastest approach to correctly identify the damage.

In order to efficiently solve this damage identification problem a Bayesian Framework will be employed. The use of a Bayesian Framework is particularly indicated in active learning surrogate modeling because it further improves performance in terms of reducing the surrogate creation cost and the overall number of iterations [19]. The method is founded on the Bayes theorem that predicts the *posterior* distribution of an event using the *prior* probability and observed data. Specifically, the data used to build the surrogate is chosen using acquisition functions that aim to select the appropriate next point to query and the level of fidelity in order to maximize information gain: using the resource budget correctly balancing the exploration and exploitation phases [20]. These acquisition functions are: the Multifidelity Expected Improvement (**MFEI**), Multifidelity Probability of Improvement **MFPI** and the Multifidelity Max-Value Entropy Search (**MFMES**) and their formulation with pros and cons will be discussed in the chapter 3.

The content of the following chapters will be indicated hereafter. Chapter 2 explores the main methods to build up a surrogate model and describes the design

space point picking techniques employed. Chapter 3 focuses on the Bayes theorem and describes its usage to build a surrogate in a single fidelity context, this is extended in a multifidelity context in chapter 4. In chapter 5 the framework numerical implementation is presented. In chapter 6 the damage identification problem is extensively described along with the different fidelity models. In chapter 7 the problem set up is described and finally in chapter 8 the results obtained using different methods are presented.

Chapter 2

Surrogate modeling

Many engineering applications need to run lots of simulations to explore extensively the design space, then select the best solution among all the possibilities to improve or to create from scratch an optimal design. Optimization processes are very common in engineering, but using just high fidelity models would be computationally infeasible. Typically during the process it would be likely to use the best model available, in terms of accuracy in describing the physic of the analyzed system. This choice would be made to maintain an high accuracy on the optimized design: that would be computationally not possible due to high cost of accurate models. Instead of using high accuracy-high cost models, an approximated version called a *surrogate model* can be employed. A first classification of surrogate building methods can be made between passive and active learning: the first builds the surrogate from an offline/online phase using sampled data or data from previous experiment in the offline phase, and then using the built surrogate in the online phase; the active learning uses the information about previously sampled data to improve the seeding selection of the next point.

In particular a surrogate can be constructed in different ways that can be more intrusive like using a reduced model, where a cheaper model is obtained by projecting data from more costly models (or high fidelities) to a reduced subspace locally enriching the surrogate [21]. The main advantage relies on preserving the underline physic structure of the model, hence it's particularly indicated to problems that evolve dynamically in time. The reduction models can be used in those situations where can be tolerated an offline expensive phase, to permit an online faster evaluation of the airplane state [22].

Another technique is the Hierarchical models reduction that uses simpler physic laws or reduced grid size. When there is a physic law that describes approximately the phenomenon and it's computationally inexpensive this technique could be applied.

Lastly, the technique used in this work of thesis is the data-fit or regression modeling: typical approaches are Gaussian Process [23, 24, 25], response surfaces, radial basis and Kriging models that uses a particular polynomial base with a series of tunable hyperparameters, that can be modified to improve the interpolation accuracy. The main difference between regression methods and the reduction ones is being non-intrusive on the data collected from the high fidelity models, that can be considered as black-box functions [26].

A classification of the problems that can be addressed using surrogate modeling is given by [27]:

- **Uncertainty propagation:** the input of the model is described as a random variable and the output distribution is collected and analyzed. Typically, Monte Carlo simulations are employed leading to high computational costs due to the great number of random input needed to obtain a good domain coverage.
- **Statistical Inference** it's an inverse problem where the objective is to estimate the problem input by observing noisy and random output.
- **Optimization** looks for a minimum of an objective function in order to obtain the best solution, for example, obtain the shape of satellites that minimize the drag [28], changing input variables among the entire design space. That is the problem considered in this work with the damage identification in a composite specimen described in 6.

2.1 Design Of Experiment (DOE)

Before choosing the right surrogate building strategy the data point selection, from which the model is evaluated, should be considered. First, the origin of the data should be addressed: if the sources of information are previous experiments or a lab test, so the configuration and the combination of variables chosen can't be changed, the data sampled should be taken as is; instead if it is used a deterministic computer simulation, that yields the same output for the same set of input variable, i.e. there is not errors on the simulation process itself, the input points can be chosen quite freely in the design space [29]. Other differences between data gathered with computer based simulations and real simulations rely on the absence of noise in the signal and the repeatability of the experiment, which is due to the lack of random errors in the computer simulations. Generally, the data gathering process should be chosen wisely in order to reduce the number of acquisitions to reduce the cost of this design phase[30].

Another important aspect to consider is the choice of input variables: should be chosen that have a great influence on the output meaning that the parameter is

essential to describe correctly the phenomenon behind the model. Otherwise one possible issue will be in the *Curse of Dimensionality* [6]: for each dimension k assume n sampling points are needed for each dimension in order to represent with accuracy the response of the system, a total of n^k points are needed. Increasing the number of dimensions, i.e. the input variable, the total data points number grows exponentially: to reduce the numerical burden the input variables of the problem, and the sampled points, should be limited and chosen such as have a great impact on the observed outputs of the system [31]. This selection can be made excluding, when possible based on the nature of the problem, the variables that have a predictable influence on the output (i.e. a linear relation persists between the variable and the output value).

Another consideration should be made regarding point spreading in the domain that should be uniform to improve accuracy and prevent the clustering of points. The first intuitive method of sampling is the *full factorial* [32] which is the selection of the sample points based on all the possible combinations of the input variables state, this permit to gasp the interconnection and influence between different variables. A reduced version is called fractional factorial sampling, which considers just a portion of all the combinations. The major drawback is the limit on the dimension of applicability due to the rapid increase of total points; another problem rely on the overlap of the points when projected along each dimension, meaning that the sampling could be improved by sufficiently cover the design space and reduce the duplicate sample [33].

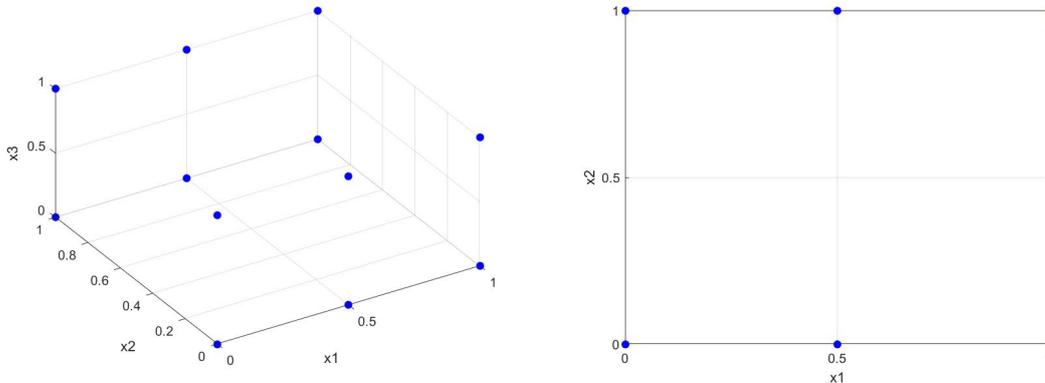


Figure 2.1: Full factorial sampling plan and a projection on the x_1 - x_2 plane

An alternative way of sampling data with quasi-random technique is the *Latin Hypercube Sampling* developed by McKay and Beckman [34] that allows an uniform distribution along the axis, i.e. the k dimensions, of the point projections. This method relies on splitting the design space in \mathbf{N} (which is the total number of

points that are sampled) non-overlapping intervals (bins) and select randomly one point from each bin.

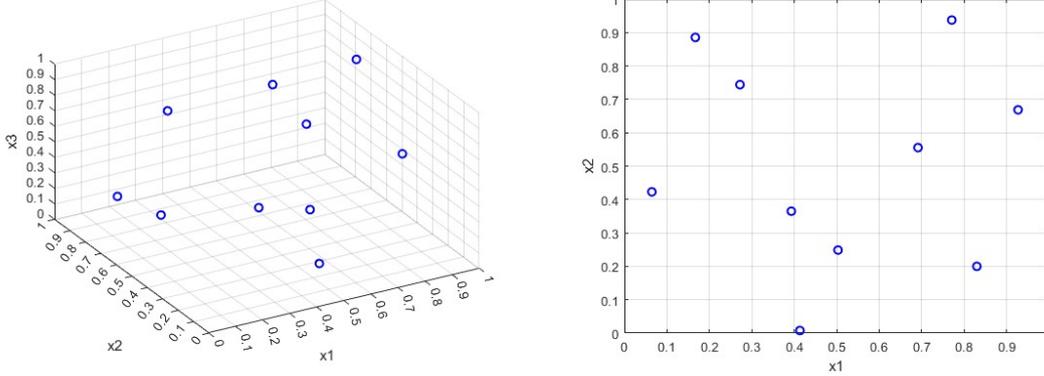


Figure 2.2: Latin Hypercube sampling plan and a projection on the x1-x2 plane

It's clear from figure 2.2 that the projection of points over a selected dimension (or on a plane) are more uniformly spreaded comparing that with the situation in figure 2.1, hence is assured good coverage of the design space.

2.2 Building the surrogate

The next step after choosing the right sampling plan, paying attention selecting the right input variables and a meaningful and representative output value for the state of the system, is to build up the surrogate with the suitable polynomial interpolation or regression technique (like kriging). Next, an important phase of the development is the validation, which is made by splitting the design space into two groups: the main one is used to build the surrogate while the second pair of data is used to validate the model. In particular the surrogate can be built with $n - 1$ sampled points and be validated on the remaining sample, this process can be repeated with all the data. These controls are called 'cross validation' because no additional test points are employed [35].

Assume that the sampling point is a vector containing all the input variables $\mathbf{x}^{(i)} = \{x_1^{(i)}, \dots, x_k^{(i)}\}$ in k dimensions and the corresponding observed objective function value is $y^{(i)} = y(\mathbf{x}^{(i)})$, with $i = 1, \dots, n$ and n is total number of sampled points, the regression model that represent the objective function linking to the input variables is [36]:

$$y(\mathbf{x}^{(i)}) = \sum_h^K \beta_h f_h(\mathbf{x}^{(i)}) + \epsilon^{(i)} \quad (2.1)$$

where $f_h(\mathbf{x})$ is a linear or non-linear function, in the latter case the function will be with a higher degree that is indicated by \mathbf{K} , β_h are the unknown coefficients that need to be estimated and the $\epsilon^{(i)}$ is the error that is considered normally distributed with zero mean and variance σ^2 . In order to determine the right degree of the polynomial, find the right terms that minimize the cross-correlation error is a good choice [37]. This is to avoid using many terms losing in generalization and with the risk of overfitting the function also catching the spurious noise oscillations. However a polynomial regression model cannot address non-linear, multimodal and multidimensional design space due to the high intrinsic complex relations between the input variables and the output, so this approach can be used in low-dimensional problems [7].

The main issue with regression models is the assumption that the errors are independent, while that's not true for a deterministic computer code[38]: if the function $y(\mathbf{x})$ is continuous (which is a typical valid assumption with some exception regarding aerodynamic, dynamic and crash analysis [7]) also the error ϵ is continuous because it's the difference between the function and the regression terms, hence, it can be said that the error is dependent from the variables input. Taking two points relatively close $\mathbf{x}^{(i)}$ and $\mathbf{x}^{(j)}$ the corresponding associated errors should be also close $\epsilon(\mathbf{x}^{(i)})$ and $\epsilon(\mathbf{x}^{(j)})$. Instead in a stochastic process modeled with random variables [39], in contrast to the regression approach, more importance is attributed to correctly represent the errors: in particular is assumed that the errors are correlated to the distance between the sampling points measured not by the Euclidean norm, because it weights all variables equally, but using a weighted distance which puts more attention on closer points with the assumption of them having more influence between each other [38].

$$d(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \sum_{h=1}^k \theta_h |x_h^{(i)} - x_h^{(j)}|^{p_h} \quad (\theta_h \geq 0, p_h \in [1,2]) \quad (2.2)$$

In equation 2.2 the parameter θ_h describes the importance of the variable x , meaning with a high value of θ_h a small change in the input produces a great change in the distance. The p_h term represents the smoothness of the function, a value close to 2 means a smooth function [7]. The correlation between errors is given by:

$$Corr [\epsilon(\mathbf{x}^{(i)}), \epsilon(\mathbf{x}^{(j)})] = exp [-d(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})] \quad (2.3)$$

With this expression for the correlation when $\mathbf{x}^{(i)}$ and $\mathbf{x}^{(j)}$ are close, hence the distance is low, the correlation is near to 1 and when two points are far away from each other, hence the distance tend to ∞ , the correlation is 0. Modeling the correlation in this way is useful to get rid of the regression terms replacing them with a constant value:

$$y(\mathbf{x}^{(i)}) = \mu + \epsilon(\mathbf{x}^{(i)}) \quad (i = 1, \dots, n) \quad (2.4)$$

Where μ is the mean value of the stochastic process and $\epsilon \mathbf{x}^{(i)}$ is the error which is normal distributed $(0, \sigma^2)$.

Kriging combines linear regression (with a linear combination of the y functions) using radial basis functions, which output value depends on the distance between the input variable and the center of the distribution [40], and a normal distribution of the standard error. Therefore the standard error goes to 0 in the sampled points and is high far from the sampled points, this feature provides a useful indication where to sample the unexplored parts of the design space to improve the interpolation approximation [39].

In general, the approximation with interpolating using radial basis, which is centered around a chosen point \mathbf{x}_j , can be expressed with:

$$\hat{y}(\mathbf{x}^*) = \sum_{j=1}^n b_j \varphi(\mathbf{x}^* - \mathbf{x}_j) \quad (2.5)$$

Where b_j are the weights that need to be determined and the φ are the radial basis which can have different expression such as linear $\varphi(x) = ||x||$, cubic $\varphi(x) = ||x||^3$ or represent thin plate $\varphi(x) = ||x||^2 \log(||x||)$. The main difference from these bases and the Kriging one (eq 2.2) is that the latter can be tuned with the parameters θ_h and p_h for each dimension, meaning a higher computational cost to determine these parameters with the advantage of an higher accuracy of the approximation [41], so typically kriging is used in lower-dimension problems with a high computational time.

2.3 Multifidelity surrogate

In order to speed up the surrogate creation, and to use efficiently the computational resources, instead of using just complex models can be employed different fidelities models with increasing cost and accuracy. The main goal is to use the lower fidelities to extensively *explore* the design space while using few high fidelity (**HF**) accuracy evaluations to *improve* the whole model. The complexity of the problem further increases because the criterion that guides the sampling process should also predict the level of fidelity to use. With different fidelities used the surrogate can be built according to [27] with several methods:

- **Adaption:** the lower fidelity models are enriched and "corrected" to match better the HF ones using for example Taylor series expansion based on gradients.
- **Fusion:** The output from the HF and LF are combined to create the surrogate. A typical example is co-Kriging that is an extension of Kriging, a regression method developed by the mining engineer Danie Krige to estimate

the quantity of gold in an extracted block [42], that uses different sources of information [43].

- **Filtering:** Using the LF extensively explore the design space and select the appropriate points, based on some criterion based on error evaluation, to be sampled with the HF to improve the surrogate quality [44].

Several techniques can therefore be used to combine different fidelities information gathered from different sources (experiments, expert suggestion, past database, and computer codes [4]). The surrogate built is essential in those applications where the usage of high accuracy computer simulations is not possible due to high execution times and cost. Using different fidelities permits to improve the quality of the surrogate where necessary for the problem involved.

A further classification between different surrogate building techniques relies on the passive and active approach.

The first kind is the offline/online approach used in the thesis of La Mantia [45] and Ermacora [46], which both constitute the background and the prior efforts of this work.

La Mantia in [45] addresses a delamination identification problem in a composite plate. He developed a computer finite element code that evaluates the modal response of the plate, then the curvatures are calculated to estimate the presence of the damage. The presence of the damage is assessed considering the difference between the calculated curvatures and the smoothed ones: these differences give an indication on the damage level. To decrease the numerical burden a reduced model, using several low fidelity code evaluations, is obtained by using the Proper Orthogonal Decomposition (POD): this reduces the overall dimension of the problem, identifying a set of orthonormal basis such as the model can be described as a linear combination of such bases. Then the reduced model is enriched and corrected using a low number of high fidelity code evaluations. All this phases are made to build the surrogate, which is trained with initial input samples and then is used in the online phase to obtain a fast and approximated position and entity of the delamination damage.

A different approach is used in the work of Ermacora [46]. The problem addressed is estimating the characteristic of a cut damage in a composite specimen subject to a traction strain. To evaluate the position and the length of the cut he uses the strain field. He builds a surrogate based model, with the techniques of kriging, co-kriging, regression trees and self organization maps to determine the reduced POD coefficient for each fidelity. The space dimension of the vector of each fidelity are incompatible, so he projects and rotates the high and mid fidelity POD basis on the low fidelity one in order to fuse the information from the different fidelities. He then uses the trained reduced model to evaluate the strain field in the specimen: combining both the Self Organizing Maps and the Regression Trees the

damage can be identified correctly reducing the number of false positive. The problem that can be addressed using multifidelity surrogates can be classified as Uncertainty propagation, Statistical inference and Optimization. In this section we will focus on the last kind of problem, because will be used to address the damage identification case study.

2.4 Multifidelity surrogate based optimization

The problem that we are going to solve is an optimization one: this relies on the minimization of an objective function that depends on the problem inputs. As shown in the previous section 2.3 to reduce the overall execution time and cost a viable solution is to use an approximated model called surrogate. A feasible technique is the trust region method. The search of the optimum is adapted iteratively by enlarging or restricting the zone. In the trust region, the quadratic Taylor expansion of high fidelity model, is used to evaluate the further improvement in optimizing the design [27]: if the effective next value obtained from the lower fidelity model improves better than predicted, the optimization step is increased otherwise is reduced, with the assumption that the model is sufficiently good near the actual minimum found until that iteration[47]. In [48] a consistency first order requirement is stated to ensure the convergence: at the center of the trust region the surrogate and the high fidelity model must have the same value of the gradient.

In case the high fidelity gradient is not available then a gradient-free approach should be used as in [49] where the low fidelity models are interpolated using radial basis function tailored for the specific problem [50]. This kind of search iteratively adapts the step length to accelerate the process towards finding the minimum. An example can be found in [51] where a profile shape optimization problem is addressed. In this paper two different surrogates, one for each level of fidelity considered as high and low, are built via a Kriging process using a trust region approach. Then the low fidelity surrogate is corrected with the high fidelity one which relies on costly CFD code simulations. Finally, the optimization process is made using the corrected low-fidelity surrogate. A similar approach is used by Alexander et al. [52] where a 2D profile and 3D complete wing problem are addressed. The low fidelity code is used to build a surrogate and follows the trend of the high fidelity code, this is granted by imposing the consistency condition by using a multiplicative coefficient approach. Their implementation has shown, by using the consistency condition, greater performance, and a fast convergence speed. If the gradient can't be calculated (for example the code works as a black box) then gradient-free method should be employed.

The method used by Wilcox and March [53] does not compute the gradient but

uses the low fidelity code to gain sensitivity information that speeds up the optimization process. This method showed to use around the same number of high fidelity evaluations as the gradient methods, hence they have the pretty the same performances. The main difference between the gradient-free and the finite difference gradient approximation approach rely on the convergence behavior: the first initially starts fast and then slows down near an optimum while the latter starts slowly and then near an optimum increases the convergence speed. The main issue with trust region approaches is that the quality of the approximation strongly depends on the set of the first high fidelity samples [27] with the downside of not finding the local minimum due to the probability of being in a flat zone or in a saddle point.

The co-Kriging method is an extension of Kriging models to a multifidelity approach [8, 54]. In [55] they describe how a surrogate is built to approximate the high fidelity and computationally high cost function, by fusing information from using information with different fidelities. The surrogate is updated iteratively during the optimization process and leads to finding the global optimum faster than using just the high fidelity code [55]. The main issues of this method are that the convergence to the global optimum cannot be guaranteed, the need to correctly balance the exploration and improve phases to avoid wasting iterations and obtaining a representative surrogate.

An improvement to the iterative adaptation of a surrogate model is the use of a Bayesian framework. In the single fidelity settings it was proposed as the Efficient Global Optimization (EGO) by Jones and Welch [38] where a convergence criterion is employed. In this approach the objective function is approximated as the mean value of a Gaussian process, while the uncertainty is modeled with its variance. The main improvement relies on using an acquisition function that drives the sampling of new points: this balances the exploitation and the exploration phase. The EGO approach overcomes also the problem of ill conditioning of the correlation matrix, that occurs when sampled points are close together: observing the improvement on finding the minimum with the acquisition function, can be translated in a stopping criterion when the improvement between two iterations is under a certain tolerance [38]. However the biggest issue with this approach relies on the influence on the first sampled data, from which the surrogate is initially built and assumed as correct [41]; in the same paper the problem is addressed with a one-stage approach: an initial guess of the minimum of the objective function is assumed, then the "credibility" of that point being the minimum is measured to improve the accuracy of the surrogate. The EGO approach was extended to deal with multifidelity problems by Kennedy and O'Hagan [56] which used a Gaussian process to model the low fidelity code in order to guide the high fidelity evaluations.

In this work we move away from the works of La Mantia and Ermacora, and

propose an active surrogate modeling technique based on a Bayesian optimization framework to guide the damage identification process. This represents an advancement step up in the surrogate modeling, due to the iterative process and the capability of improving the surrogate while already using it. This approach combined with the multifidelity nature of the problem is the key to speed up the surrogate creation process and solve the problem addressed in a much shorter time. In the chapter 3 the singlefidelity bayesian framework is described along with the singlefidelity acquisition functions and then in chapter 4 the framework is illustrated in its multifidelity fashion, which will be employed in this work.

Chapter 3

Bayesian Optimization

In chapter 2 the methods to build up a surrogate are described, and one main distinction is made between passive and active learning: in this chapter we will focus on the latter describing in particular the statistical inference model with Bayesian technique. This is particularly well suited to find the minimum of an objective function which expression is not known, for example black-box functions that are common in aerospace applications. The peculiarity of the Bayesian method consists in using particular functions called *acquisition functions* (or also *Utility function* as they measure the utility of sampling in specific points) that drive the search of the next point to sample: the objective is to reduce the overall number of sampling points while maximizing the accuracy of the surrogate model, this is iteratively updated with the new selected point. These acquisition functions try to balance the exploration of the domain space, spreading the sampled points to catch the global function shape and general behavior, and local exploitation to improve the accuracy of the model in specific areas. Typically, the acquisition functions assume a high value where there is high uncertainty in the model (exploitation) and where the actual model predicts a high value of the acquisition function (exploration) hence the next sampled point is selected accordingly.

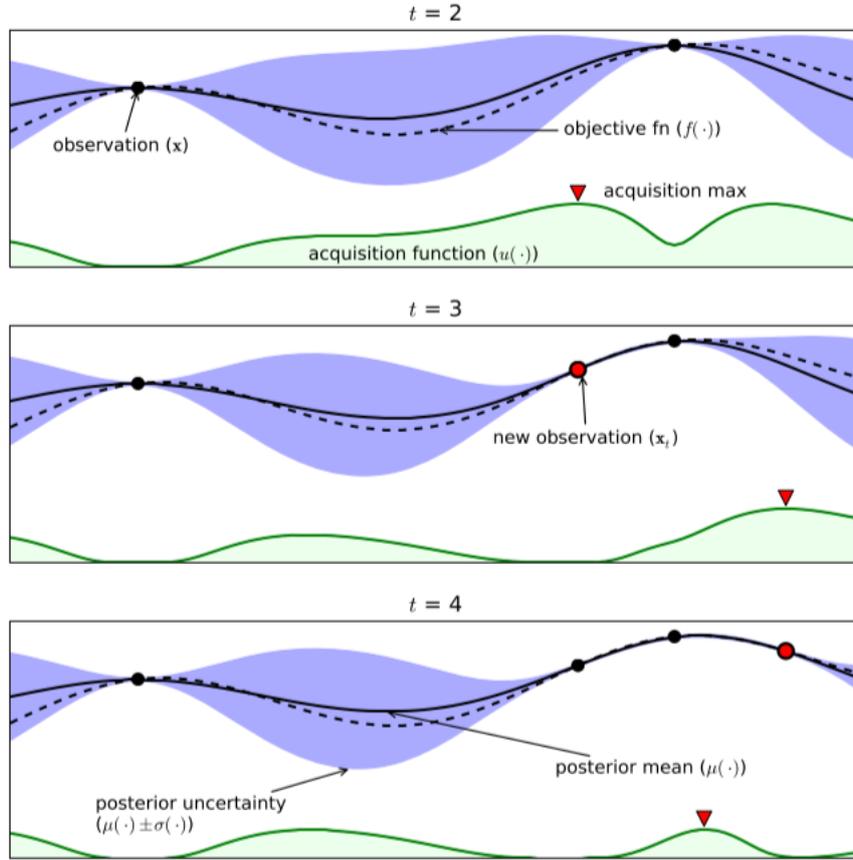


Figure 3.1: Example of a 1D problem optimization [57]

Figure 3.1 taken from [57], as the main considerations of this chapter, represents a 1D function approximated with a Gaussian process (GP) and in the lower part of the graph there is the acquisition function. The function is initially sampled in two points and between them the posterior uncertainty is high. The next point is sampled where the acquisition function assumes its maximum value and the model is updated. This process is repeated until a fixed number of iterations is reached, or all the computational budget is spent.

The Bayesian approach takes the name from the Bayes theorem that links the *posterior probability* of an event or theory called M given some data observation E and says that is proportional to the *likelihood* of the data E given M multiplied for the prior probability of M .

$$P(M|E) \propto P(E|M)P(M)$$

Where the prior indicates the information based on the belief of plausible objective functions: choosing a smooth function and noise-free across all possibilities permits

to consider less likely the data with high variance and oscillations [57]. Considering that, accumulating the data from various objective function observations $D_{1:t} = \{\mathbf{x}_{1:t}, f(\mathbf{x}_{1:t})\}$ the prior distribution is combined with the likelihood $P(D_{1:t}|f)$ hence the posterior probability is updated:

$$P(f|D_{1:t}) \propto P(D_{1:t}|f)P(f)$$

The Bayesian process can be summarized as reported in [57]:

Algorithm 1 Bayesian Optimization

- 1: **for** $t = 1, 2, \dots$ **do**
 - 2: Find \mathbf{x}_t by optimizing the acquisition function over the GP: $\mathbf{x}_t = \operatorname{argmax}_{\mathbf{x}} u(\mathbf{x}|\mathcal{D}_{1:t-1})$.
 - 3: Sample the objective function: $y_t = f(\mathbf{x}_t) + \varepsilon_t$.
 - 4: Augment the data $\mathcal{D}_{1:t} = \{\mathcal{D}_{1:t-1}, (\mathbf{x}_t, y_t)\}$ and update the GP.
 - 5: **end for**
-

Figure 3.2: Bayes optimization process

The problem state is a maximizing one using $\mathbf{x}^* = \operatorname{argmax}_{\mathbf{x}} f(\mathbf{x})$ but can be transformed in a minimization problem using $g(\mathbf{x}) = -f(\mathbf{x})$. A further assumption is that the objective function is *Lipschitz-continuous* meaning that there is some positive constant C typically unknown, such given $x_1, x_2 \in \mathcal{A}$ then $\|f(x_1) - f(x_2)\| \leq C\|x_1 - x_2\|$. Another assumption about the bounds of search space is made stating that are axis-aligned, the search space itself is an hyperrectangle of dimension d . With all these assumptions to guarantee the best observation $f(\mathbf{x}^+) \geq f(\mathbf{x}^*) - \epsilon$ requires $(C/2\epsilon)^d$ samples which can be extremely costly: this "worst case" assurance about the max value identified could be relaxed due to its unlikely probability to occur. The main goal is to maximize the posterior probability at each step reducing the distance between the real global maximum and the value given from the model. From figure 3.2 with the letter u is indicated the utility (acquisition) function, so the problem shifts from finding the maximum value on the objective function to finding the maximum value of the acquisition function, with the main advantage that the latter is typically easy in its formulation and cheap to compute.

3.1 Gaussian Process

3.1.1 Priors distribution

As reported in the previous section the Bayes theorem updates the prior probability distribution, but this distribution initially should be carefully chosen to

guarantee the problem convergence toward the optimum. Typically, a Gaussian Process Regression is employed as is well suited for this application [19]. In particular the author states these conditions to use the GP (i) the acquisition function is continuous and approximately minimizes the risk (defined as the expected deviation from the global minimum at a fixed point \mathbf{x}); (ii) conditional variance converges to zero, or to a minimum value in presence of noise, if the distance to the nearest observation is zero; (iii) the objective function is continuous; (iv) the prior probabilities are homogeneous meaning do not depend on the coordinate origin; (v) the optimization is independent of the m^{th} differences, meaning these can be considered as discrete m^{th} order derivatives, hence the sample functions are nearly non-derivable everywhere. A Gaussian Process is an extension to an infinite dimension stochastic process where every finite collection of variables has a multivariate normal distribution: it gives a mean m and a covariance k over the possible functions f at location \mathbf{x}

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$$

The mean value is initially assumed 0 and then updated in further evaluations, with this assumption the focus is entirely on the covariance function.

3.1.2 Covariance function

The main characteristic of a Gaussian Process is the choice of the right covariance function as that determines the properties of samples drawn from it, such as their overall smoothness. A popular type is the squared distance:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{1}{2} \|\mathbf{x}_i - \mathbf{x}_j\|^2\right)$$

which assumes a value near 1 when the two points \mathbf{x}_i and \mathbf{x}_j get closer and reaches 0 as the points distance increases. This is a necessary condition to assure the convergence as stated in [19]: the influence between points is greater if they are close to each other. Choosing to sample from the prior, the points $\{\mathbf{x}_{1:t}\}$ are picked with the relative objective function values $\mathbf{f}_{1:t} = f(\mathbf{x}_{1:t})$ creating the pair $\{\mathbf{x}_{1:t}, \mathbf{f}_{1:t}\}$. The functions value are chosen using a multivariate normal distribution $\mathcal{N}(0, \mathbf{K})$ where the \mathbf{K} is the covariance matrix:

$$\mathbf{K} = \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & \cdots & k(\mathbf{x}_1, \mathbf{x}_t) \\ \vdots & \ddots & \vdots \\ k(\mathbf{x}_t, \mathbf{x}_1) & \cdots & k(\mathbf{x}_t, \mathbf{x}_t) \end{bmatrix}$$

The values on the diagonal are equal to 1 just in a noise-free environment. The GP is fitted using data from an external model in our case, so the pair $\{\mathbf{x}_{1:t}, \mathbf{f}_{1:t}\}$

of previous samples are known. Then the objective is to choose the next point to query denoted with \mathbf{x}_{t+1} with the associate objective function value $f_{t+1} = f(\mathbf{x}_{t+1})$, by the property of the Gaussian processes $\mathbf{f}_{1:t}$ and f_{t+1} are jointly Gaussian:

$$\begin{bmatrix} \mathbf{f}_{1:t} \\ f_{t+1} \end{bmatrix} \sim \mathcal{N} \left(\mathbf{0}, \begin{bmatrix} \mathbf{K} & \mathbf{k} \\ \mathbf{k}^T & k(\mathbf{x}_{t+1}, \mathbf{x}_{t+1}) \end{bmatrix} \right)$$

where $\mathbf{k} = [k(\mathbf{x}_{t+1}, \mathbf{x}_1) \quad k(\mathbf{x}_{t+1}, \mathbf{x}_2) \quad \cdots \quad k(\mathbf{x}_{t+1}, \mathbf{x}_t)]$ using the Sherman-Morrison-Woodbury formula [58] the predictive posterior distribution can be obtained:

$$P(f_{t+1}|D_{1:t}, \mathbf{x}_{t+1}) = \mathcal{N} \left(\mu_t(\mathbf{x}_{t+1}), \sigma_t^2(\mathbf{x}_{t+1}) \right)$$

where

$$\begin{aligned} \mu_t(\mathbf{x}_{t+1}) &= \mathbf{k}^T \mathbf{K}^{-1} \mathbf{f}_{1:t} \\ \sigma_t^2(\mathbf{x}_{t+1}) &= k(\mathbf{x}_{t+1}, \mathbf{x}_{t+1}) - \mathbf{k}^T \mathbf{K}^{-1} \mathbf{k} \end{aligned}$$

In order to generalize and change the influence of each \mathbf{x} in the covariance (also called kernel) for an isotropic model just one *hyperparameter* θ is necessary:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp \left(-\frac{1}{2\theta^2} \|\mathbf{x}_i - \mathbf{x}_j\|^2 \right)$$

and for an anisotropic model:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp \left(-\frac{1}{2} (\mathbf{x}_i - \mathbf{x}_j)^T \text{diag}(\theta)^{-2} (\mathbf{x}_i - \mathbf{x}_j) \right)$$

where $\text{diag}(\theta)$ is a diagonal matrix with d entries θ on the diagonal. It can be seen that if a particular value of θ_1 is small, the covariance is also close to 0 becoming independent from l^{th} input value hence automatically discarding that. The hyperparameters are typically obtained by "seeding" some random samples and then maximizing the log-likelihood of the evidence given θ [59]. Another important kernel in Bayesian Optimization problems is the *Matérn kernel* [60, 61] which defines a smoothness parameter ζ that permits great flexibility in modeling:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{2^{\zeta-1} \Gamma(\zeta)} (2\sqrt{\zeta} \|\mathbf{x}_i - \mathbf{x}_j\|)^{\zeta} H_{\zeta}(2\sqrt{\zeta} \|\mathbf{x}_i - \mathbf{x}_j\|)$$

where $\Gamma(\cdot)$ is the Gamma function and the $H_{\zeta}(\cdot)$ is the Bessel function of order ζ . When $\zeta \rightarrow \infty$ the kernel reduces to the squared exponential and when $\zeta = 0.5$ it degenerates in the unsquared exponential kernel.

3.2 Acquisition functions

As stated in the previous chapter the search of new sampling points is guided by an appropriate acquisition function. In this section the main type, in their single fidelity fashion, will be analyzed and described.

3.2.1 Probability of Improvement(PI)

This acquisition function suggests *maximizing the probability of improvement* over the $f(\mathbf{x}^+)$ with $\mathbf{x}^+ = \operatorname{argmax}_{\mathbf{x}_i \in \mathbf{x}_{1:t}} f(\mathbf{x}_i)$:

$$\begin{aligned} \text{PI}(\mathbf{x}) &= P(f(\mathbf{x}) \geq f(\mathbf{x}^+)) \\ &= \Phi\left(\frac{\mu(\mathbf{x}) - f(\mathbf{x}^+)}{\sigma(\mathbf{x})}\right) \end{aligned}$$

where $\Phi(\cdot)$ is the normal cumulative probability distribution function. The main issue with this formulation is that it relies on pure exploitation: will be chosen points that have a $f(\mathbf{x})$ value slightly higher than the actual maximum but with high probability instead of point that will provide a higher objective function increase but with less certainty, i.e. further from the actual maximum point. This formulation can be improved by inserting a trade-off parameter $\xi \geq 0$ that is set with high value at the start of the process and then reduced towards 0 as the iterations increase, this modification permits to select initial points that assure a certain amount of improvement:

$$\begin{aligned} \text{PI}(\mathbf{x}) &= P\left(f(\mathbf{x}) \geq f(\mathbf{x}^+) + \xi\right) \\ &= \Phi\left(\frac{\mu(\mathbf{x}) - f(\mathbf{x}^+) - \xi}{\sigma(\mathbf{x})}\right) \end{aligned} \tag{3.1}$$

In [41] Jones clearly indicates that this approach is very dependent on the selection of the target: if the parameter ξ is small, i.e. the desired improvement is small, than the search process is highly local and tends to be more global after extensively searching around the current maximum; meanwhile if the parameter is settled with a high value, i.e. a desired high improvement, than the search is almost just global which means a slower overall algorithm in fine searching. Finding the right starting value of ξ is really important and on the other hand hard to accomplish: is necessary to balance the performance of the code with the accuracy of the surrogate model built.

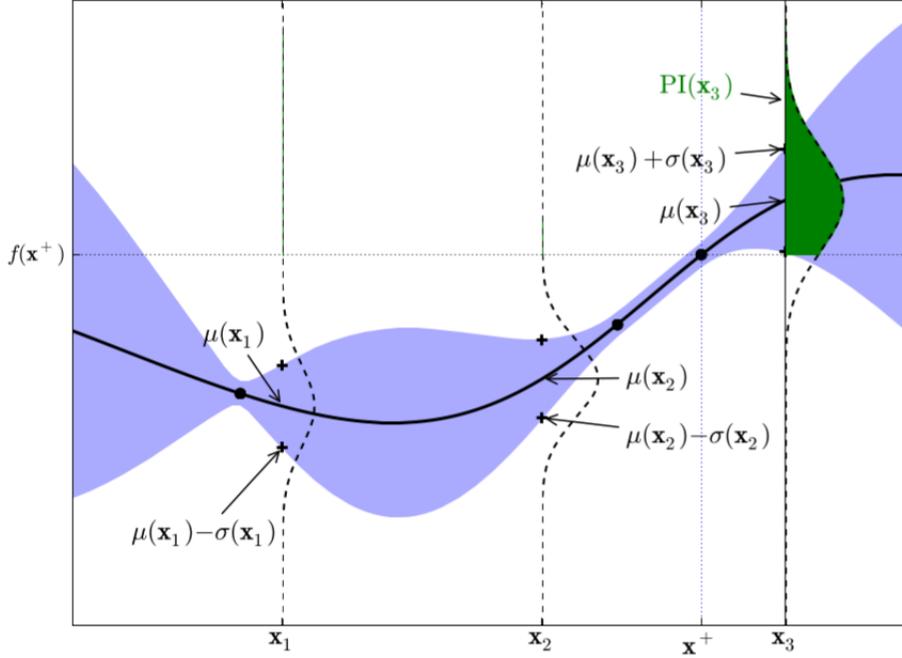


Figure 3.3: A 1D function approximated with a Gaussian Process, the actual max value is for the sample \mathbf{x}^+ and the shaded green area over the dashed horizontal line indicates the improvement of sampling at \mathbf{x}_3 : it can be clearly seen that sampling at \mathbf{x}_1 or \mathbf{x}_2 doesn't give any improvement finding the max value.[57]

3.2.2 Expected Improvement (EI)

A viable solution to improve the acquisition criterion is to consider the magnitude of improvement that a specific sampling point can lead to, i.e. minimizing the expected deviation from the actual maximum value:

$$\begin{aligned} \mathbf{x}_{t+1} &= \underset{\mathbf{x}}{\operatorname{argmin}} \mathbb{E} (|f_{t+1}(\mathbf{x}) - f(\mathbf{x}^*)| \mid \mathcal{D}_{1:t}) \\ &= \underset{\mathbf{x}}{\operatorname{argmin}} \int \|f_{t+1}(\mathbf{x}) - f(\mathbf{x}^*)\| P(f_{t+1} \mid \mathcal{D}_{1:t}) df_{t+1} \end{aligned} \quad (3.2)$$

This formulation considers just one step ahead, but it can be applied recursively to extend to further iterations. Instead [19] proposed a new formulation for the improvement criterion:

$$I(\mathbf{x}) = \max\{0, f_{t+1}(\mathbf{x}) - f(\mathbf{x}^+)\} \quad (3.3)$$

The improvement is positive if the expected value of the function is greater than the actual maximum, otherwise will be zero. The next point to sample is identified

by maximizing the expected value of the improvement:

$$\mathbf{x} = \underset{\mathbf{x}}{\operatorname{argmax}} \mathbb{E} \left(\max \left\{ 0, f_{t+1}(\mathbf{x}) - f(\mathbf{x}^+) \right\} \mid \mathcal{D}_t \right) \quad (3.4)$$

The likelihood of improvement $I(\mathbf{x})$ of a normal distribution with mean $\mu_{\mathbf{x}}$ and variance $\sigma(\mathbf{x})^2$ can be calculated with the normal density function:

$$\frac{1}{\sqrt{2\pi}\sigma(\mathbf{x})} \exp \left(-\frac{(\mu(\mathbf{x}) - f(\mathbf{x}^+) - I)^2}{2\sigma^2(\mathbf{x})} \right) \quad (3.5)$$

The Expected Improvement is obtained integrating by part this quantity:

$$\begin{aligned} \mathbb{E}(I) &= \int_{I=0}^{I=\infty} I \frac{1}{\sqrt{2\pi}\sigma(\mathbf{x})} \exp \left(-\frac{(\mu(\mathbf{x}) - f(\mathbf{x}^+) - I)^2}{2\sigma^2(\mathbf{x})} \right) dI \\ &= \sigma(\mathbf{x}) \left[\frac{\mu(\mathbf{x}) - f(\mathbf{x}^+)}{\sigma(\mathbf{x})} \Phi \left(\frac{\mu(\mathbf{x}) - f(\mathbf{x}^+)}{\sigma(\mathbf{x})} \right) + \phi \left(\frac{\mu(\mathbf{x}) - f(\mathbf{x}^+)}{\sigma(\mathbf{x})} \right) \right] \end{aligned} \quad (3.6)$$

That can be evaluated analytically as reported in [38]:

$$\begin{aligned} \text{EI}(\mathbf{x}) &= \begin{cases} (\mu(\mathbf{x}) - f(\mathbf{x}^+)) \Phi(Z) + \sigma(\mathbf{x})\phi(Z) & \text{if } \sigma(\mathbf{x}) > 0 \\ 0 & \text{if } \sigma(\mathbf{x}) = 0 \end{cases} \\ Z &= \frac{\mu(\mathbf{x}) - f(\mathbf{x}^+)}{\sigma(\mathbf{x})} \end{aligned} \quad (3.7)$$

where $\phi(\cdot)$ denotes the Probability Density Function (PDF) and the $\Phi(\cdot)$ indicates the Cumulative Density Function (CDF). To balance exploitation and exploration a parameter ξ can be introduced as in the previous section for Probability of Improvement.

3.2.3 Predictive Entropy Search (PES)

This method proposed by Lobato [62] uses the entropy to obtain global utility information. In particular the objective is to find the global maximizer $\mathbf{x}_* = \operatorname{argmax}_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x})$ where the function $f(\mathbf{x})$ is sampled from a Gaussian Process. The global parameter that encloses information about \mathbf{x}_* is the negative differential entropy of $p(\mathbf{x}_* | \mathcal{D}_t)$, hence to query the next point to sample the technique is to maximize the expected reduction of this quantity:

$$\alpha_n(\mathbf{x}) = \mathbb{H}[p(y | \mathcal{D}_t, \mathbf{x})] - \mathbb{E}_{p(\mathbf{x}_* | \mathcal{D}_n)} [\mathbb{H}[p(y | \mathcal{D}_t, \mathbf{x}, \mathbf{x}_*)]] \quad (3.8)$$

Calculating the $p(\mathbf{x}_* | \mathcal{D}_t)$ and approximating the various terms in the expression of the acquisition function can be difficult, computationally infeasible due to numerous iterations needed and the optimum could be not unique increasing the complexity of the problem[63].

3.2.4 Max-Value entropy search

To overcome the difficulties of the PES algorithm Wang & Jegelka [63] developed a new approach leveraging the information on the maximum objective function value $f_* := \max_x f(x)$: this reduces the computational cost because $f(\mathbf{x})$ is certainly in one dimension space. The information gain is measured between the actual maximum value of the function $y_* = f(\mathbf{x}_*)$ and the candidate point to sample, which can be approximated by evaluating the entropy of the predictive distribution:

$$\begin{aligned} \alpha_t(x) &= I(\{\mathbf{x}, y\}; y_* | D_t) \\ &= H(p(y | D_t, \mathbf{x})) - \mathbb{E}[H(p(y | D_t, \mathbf{x}, y_*))] \\ &\approx \frac{1}{K} \sum_{y_* \in Y_*} \left[\frac{\gamma_{y_*}(\mathbf{x}) \psi(\gamma_{y_*}(\mathbf{x}))}{2\Psi(\gamma_{y_*}(\mathbf{x}))} - \log(\Psi(\gamma_{y_*}(\mathbf{x}))) \right] \end{aligned} \quad (3.9)$$

where ψ is PDF and Ψ is the CDF and $\gamma_{y_*}(\mathcal{D}) = \frac{y_* - \mu_t(\mathbf{x})}{\sigma_t(\mathbf{x})}$. The expectation of the probability $p(y_* | D_n)$ is calculated with Monte Carlo estimation based on K values sampling. The probability term $p(y | D_t, \alpha)$ is a Gaussian distributed with mean value $\mu_t(\mathbf{x})$ and variance $k_t(x, x)$ while the term $p(y | D_n, \alpha, y_*)$ is distributed as a truncated Gaussian: given y_* the distribution of y needs to satisfy $y < y_*$. Sampling the y_* can be done in two ways according to [62] with an approximation Gumbel distribution and maximizing the function from the posterior Gaussian distribution. Sampling using the Gumbel distribution means approximating $\hat{p}(y_* | D_n)$ via its CFD $\widehat{\Pr}[y_* < z] = \prod_{\mathbf{x} \in \hat{\mathcal{X}}} \Psi(\gamma_z(\mathbf{x}))$ where $\hat{\mathcal{X}}$ is a finite discrete dense subset of representative points that replace the continuous set \mathfrak{X} . This means that is needed to sample r uniformly from $[0, 1]$ to find z such that $\Pr[y_* < z] = r$. A faster and cheaper way to sample is to approximate the CFD with a Gumbel distribution: $\widehat{\Pr}[y_* < z] \approx \mathcal{G}(a, b) = e^{-e^{-\frac{z-a}{b}}}$. This approximation is motivated by the Fisher-Tippett-Gnedenko theorem [64] which states that the maximum of a set of (Independent and Identically Distributed random variable) Gaussian variables is asymptotically described by a Gumbel distribution. To sample from the Gumbel distribution the Gumbel quantile function is used and the parameters a and b are estimated with percentile matching. An alternative sampling method consists of using the posterior GP functions and maximizing those functions.

Chapter 4

Multifidelity Bayesian optimization

In order to speed up the optimization process a feasible option is to gain information from multiple fidelities sources; the complexity of the problem formulation further increase, considering the counterpart of single fidelity implementation, because also the fidelity levels play a role in the sampling plan hence should be carefully chosen. More in detail the next point chosen the lower fidelity functions (which are used in the exploration phase) is selected maximizing the information gain with that fidelity. In order to accomplish that considering also the ratio between the gain and the cost of evaluation (to ensure that is preferred using low fidelity instead of an higher fidelity) and on the other hand preserve some budget for the exploitation phase with the higher fidelity models[65]. The first step consists in expanding the GP in a multifidelity approach, each observation embeds also a fidelity level $y_x^{(1)}, \dots, y_x^{(M)}$ where the highest fidelity level is (M) and the lowest is (1) . These function evaluations are taken considering the input from the design space $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^d$. The function evaluations are modeled with a GP $y_x^{(m)} = f_x^{(m)} + \epsilon$ where the term ϵ is random noise modeled with a normal distribution $\epsilon \sim \mathcal{N}(0, \sigma_{\text{noise}}^2)$. Each fidelity level is associated with a corresponding evaluation cost indicated with $\lambda^{(m)}$ that follows this inequality $\lambda^{(1)} \leq \lambda^{(2)} \dots \leq \lambda^{(M)}$ [66]: if $M = 1$ the problem goes back to single fidelity. The dataset is now composed by 3 variables: the inputs x_i , the fidelities m_i and the outputs $y_{x_i}^{(m_i)}$ constituting the n , that indicate the number of total samples, groups $\mathcal{D}_n = \{(x_i, y_{x_i}^{(m_i)}, m_i)\}_{i \in [n]}$. Considering an autoregressive method [56] with the fundamental assumption of the *Markov Property*:

$$\text{cov}\{f_m(\mathbf{x}), f_{m-1}(x') | f_{m-1}(\mathbf{x})\} = 0 \quad \forall x' \neq x$$

indicates that given the $f_{m-1}(\mathbf{x})$ nothing more can be learnt about $f_m(\mathbf{x})$ for every possible cheap evaluation $f_{m-1}(\mathbf{x}')$, this means that the highest fidelity is assumed correct, and the errors lie just in the lower fidelities models. The higher fidelity is modeled recursively using the lower fidelity and an associate error δ both modeled with a GP with zero mean and a kernel associated function k :

$$f^{(m)}(\mathbf{x}) = f^{(m-1)}(\mathbf{x}) + \delta^{(m)}(\mathbf{x}) \quad m = 2, \dots, M$$

In order to integrate all the fidelities in one GP model a Kernel Matrix $\mathbf{K} \in \mathbb{R}^{n \times n}$ is defined with each element i,j as $k((\mathbf{x}_i, m_i), (\mathbf{x}_j, m_j))$ the mean and the variance of the GP unified model are respectively:

$$\begin{aligned} \mu_{\mathbf{x}}^{(m)} &= \mathbf{k}_n^{(m)}(\mathbf{x})^\top \mathbf{C}^{-1} \mathbf{y} \\ \sigma_{\mathbf{x}}^{2(m)} &= k((\mathbf{x}, m), (\mathbf{x}, m)) - \mathbf{k}_n^{(m)}(\mathbf{x})^\top \mathbf{C}^{-1} \mathbf{k}_n^{(m)}(\mathbf{x}) \end{aligned}$$

where each term is:

$$\begin{aligned} \mathbf{I} &:= \text{Identity matrix} \\ \mathbf{C} &:= \mathbf{K} + \sigma_{\text{noise}}^2 \mathbf{I} \\ \mathbf{y} &:= (y_{\mathbf{x}_1}^{(m_1)}, \dots, y_{\mathbf{x}_n}^{(m_n)})^{-1} \\ \mathbf{k}_n^{(m)}(\mathbf{x}) &:= (k((\mathbf{x}, m), (\mathbf{x}_1, m_1)), \dots, k((\mathbf{x}, m), (\mathbf{x}_n, m_n)))^\top \end{aligned}$$

The next step consists in changing the acquisition functions in a multifidelity context, in order to consider also the cost of the single fidelities and accordingly choose the next point to query with an associated level of fidelity.

4.1 Multifidelity Probability of Improvement

The expression of the single fidelity Probability of Improvement is expanded according to [67] introducing new terms:

$$MFPI(x, m) = PI(\mathbf{x}) \cdot Corr(\mathbf{x}, m) \cdot CR(m) \cdot \eta(\mathbf{x}, m) \quad m = 1, \dots, M$$

where the single terms are defined as:

$$\begin{aligned} PI(x) &= \Phi\left(\frac{f(\mathbf{x}^+)^m - f(\mathbf{x})^m}{\sigma(\mathbf{x})^m}\right) \\ Corr(\mathbf{x}, m) &= Corr[f(\mathbf{x})^M, f(\mathbf{x})^m] \\ CR(m) &= \frac{\lambda_M}{\lambda_m} \\ \eta(\mathbf{x}, m) &= \prod_{i=1}^n [1 - R(\mathbf{x}, x_i^{(m)})] \end{aligned} \tag{4.1}$$

The contribution in the term of the PI for the same input value is maximized when using the higher fidelity available. This behaviour is also reflected on the correlation term that links two different fidelities and measure how much is reduced the PI when selecting the lower fidelities. Hence if the highest fidelity is used the correlation term assumes value 1 so the PI isn't penalized. If exists already a sample in the \mathbf{x} location, then the correlation is 0 (if there isn't random error conducting the experiment) this discard sampling in the same point twice.

The term $CR(m)$ refers to the *Cost Ratio* and indicates the necessary number of lower fidelity evaluations that have the same cost as one highest fidelity model run. This term weights the cost of the available fidelities and if the same value of PI can be obtained then the cheaper model is evaluated: the $CR(m)$ is 1 if the highest fidelity is employed, while assumes a value greater than 1, that tends to increase the overall MFPI value, for each lower fidelities that have a lower computational cost.

The term $\eta(\mathbf{x}, m)$ is the sample data density function and is used to prevent clustering of sample around a point. To model the distance between sampled data and the next query point the GP covariance function is employed, hence due to that formulation this term drops rapidly when the selected point moves toward an already sampled point.

4.2 Multifidelity Expected Improvement

In a similar manner the Expected Improvement for single fidelity introduced by Jones [38] is extended to multifidelity problems [68]:

$$MFEI(\mathbf{x}, m) \equiv E \left[\max \left(f^{(m)}(\mathbf{x}) - f^{(m)}(\mathbf{x}^+), 0 \right) \right] \cdot \alpha_m(\mathbf{x}, m) \cdot \alpha_2(\mathbf{x}, m) \cdot \alpha_3(l) \quad (4.2)$$

where the single α terms indicate:

$$\begin{aligned} \alpha_1(\mathbf{x}, m) &= \text{corr} \left[f^{(m)}(\mathbf{x}), f^{(M)}(\mathbf{x}) \right] \\ \alpha_2(\mathbf{x}, m) &= \left(1 - \frac{\sigma_\varepsilon}{\sqrt{\sigma^{2(m)}(\mathbf{x}) + \sigma_\varepsilon^2}} \right) \text{ where } \sigma(\mathbf{x})^{2(m)} = \text{cov} [f^m(\mathbf{x}), f^m(\mathbf{x})] \quad (4.3) \\ \alpha_3(m) &= \frac{\lambda_M}{\lambda_m} \end{aligned}$$

The term $\alpha_1(\mathbf{x}, l)$ indicated as the correlation between the highest fidelity M and the lower fidelity m sampled at the same \mathbf{x} , it reduces the Expected Improvement when sampling using the lower fidelity models (the correlation is exactly 1 if the model used is the highest fidelity available). Hence it avoids sampling just from the lowest fidelity because it's cheaper. This term also prevents sampling in points that

have been already sampled avoiding wasting computational resources. The term $\alpha_2(\mathbf{x}, l)$ modifies the EI when the simulations from the lower fidelity m contains random errors, as the prediction become more accurate it reduces the posterior standard deviation when a duplicate data point is added; hence it reduces the overall number of duplicates when the models are affected by random errors. This factor assumes value equal 1 when the variance of the random error is zero. The term $\alpha_3(l)$ adjust the EI to the evaluation cost of the model choosing, with the same information gain, the cheaper model available. This ratio is greater than 1 so it augments the EI if a cheaper model is used, avoiding using just the highest fidelity one.

4.3 Multifidelity Predictive Entropy Search

This acquisition function allows to balance between exploration and exploitation naturally without using parameters and discretization information which are difficult to estimate and also influences greatly the model. This goal can be achieved using an acquisition function that maximize the information gain only on the global target maximizer \mathbf{x}_{+t} observing the next sampling point which include the input and a corresponding fidelity $\langle \mathbf{x}, m \rangle$:

$$\alpha(y_{\mathbf{X}}, \langle x, m \rangle) = H(y_m(x)|y_{\mathbf{X}}) - \mathbb{E}_{p(x_{+t}, |y_{\mathbf{X}})}[H(y_m(x)|y_{\mathbf{X}}, x_{+})]$$

Due to the Gaussian Process posterior approximation the first term on the right hand side can be calculated as:

$$H(y_m(x)|y_{\mathbf{X}}) \triangleq 0.5 \log(2\pi e(\sigma_{\langle x, m \rangle | X}^2 + \sigma_{n_m}^2)) \quad \text{with} \quad \sigma_{\langle x, i \rangle | X}^2 \triangleq \Sigma_{\{\langle x, m \rangle\} \{\langle x, m \rangle\} | X}$$

The second term can be calculated considering the cross correlation between the target and auxiliary function exploiting just the target maximizer from $p(\mathbf{x}_{*t} | y_{\mathbf{X}})$. As reported in [69] the expectation can be approximated with:

$$\mathbb{E}_{p(x_{+t}, |y_{\mathbf{X}})}[H(y_m(x)|y_{\mathbf{X}}, x_{+})] \approx -\frac{1}{2S} \sum_{s=1}^S \log(v_{f_m}^{(s)} + \sigma_{n_m}^2)$$

4.4 Multifidelity Max-value Entropy Search

The multifidelity extension of the Max-value Entropy Search is founded on its singlefidelity approach: the goal is to maximize the information gain of the maximum value in the highest fidelity function using lower fidelities $f_+ := \max_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x})^{(M)}$. This approach, in opposition to the MES where the objective is to maximize the information gain about $\mathbf{x}_+ := \operatorname{argmax}_{\mathbf{x}} f(\mathbf{x})$ that is a global utility indicator

and automatically balances between exploration and exploitation, has the main advantage that f_+ is in one dimension-space which reduces the problem to simpler calculations. The extension is taken from [66] where the sampling due to the different fidelity cost can be extended in a paralleled way, but our implementation uses sequential querying. The acquisition function integrates also the evaluation cost of the m^{th} fidelity:

$$MFMES(\mathbf{x}, m) := I(f_*; f_{\mathbf{x}}^{(m)} | \mathcal{D}_t) / \lambda^{(m)} \quad (4.4)$$

where \mathcal{D}_t is the already sampled data and the Information term is defined between the actual maximum value f^* and the next considered candidate with its fidelity $f_{\mathbf{x}}^{(m)}$, hence the Information term can be formalized considering the entropy reduction:

$$I = (f_*; f_{\mathbf{x}}^{(m)} | \mathcal{D}_t) = H(f_{\mathbf{x}}^{(m)} | \mathcal{D}_t) - \mathbb{E}_{f_+ | \mathcal{D}_t} [H(f_{\mathbf{x}}^{(m)} | f_+, \mathcal{D}_t)] \quad (4.5)$$

The first term in on the right hand side can be calculated analytically:

$$H(f_{\mathbf{x}}^{(m)} | \mathcal{D}_t) = \log(\sigma_{\mathbf{x}}^{(m)} \sqrt{2\pi e}) \quad (4.6)$$

The Expectation can't be analytically calculated hence it is approximated using Monte Carlo estimation sampling the f_* from the GP:

$$\mathbb{E}_{f_+ | \mathcal{D}_t} [H(f_{\mathbf{x}}^{(m)} | f_+, \mathcal{D}_t)] \approx \sum_{f_+ \in \mathcal{F}_+} \frac{H(f_{\mathbf{x}}^{(m)} | f_+, \mathcal{D}_t)}{|\mathcal{F}_+|} \quad (4.7)$$

where \mathcal{F} is a collection of previously sampled f_* and because it's one dimensional the number of samples needed is small. In [66] is clearly reported how to manipulate the Entropy formulation to obtain a computationally tractable version. In particular the entropy of $p(f_{\mathbf{x}}^{(m)} | f_+, \mathcal{D}_t)$ is needed in ?? but is substituted with $p(f_{\mathbf{x}}^{(m)} | f_{\mathbf{x}}^{(M)} \leq f_+, \mathcal{D}_t)$, in this second expression the conditioning is given only on the \mathbf{x} and when the $m = M$ then the density function is Truncated Normal Distribution and the entropy according to Michalowicz [70]:

$$H(f_{\mathbf{x}}^{(M)} | f_{\mathbf{x}}^{(M)} \leq f_+, \mathcal{D}_t) = \log(\sqrt{2\pi e} \sigma_{\mathbf{x}}^{(M)} \Phi(\gamma_{f_+}^{(M)}(\mathbf{x}))) - \frac{\gamma_{f_+}^{(M)}(\mathbf{x}) \phi(\gamma_{f_+}^{(M)}(\mathbf{x}))}{2\Phi(\gamma_{f_+}^{(M)}(\mathbf{x}))} \quad (4.8)$$

where the term $\gamma_{\zeta}^{(m)}(\mathbf{x}) := (\zeta - \mu_{\mathbf{x}}^{(m)}) / \sigma_{\mathbf{x}}^{(m)}$ indicate a function normalization with $\zeta \in \mathbb{R}$, ϕ and Φ represent the probability density function and the cumulative distribution function of the normal distribution.

Considering a case where $m \neq M$ the density distribution of $p(f_{\mathbf{x}}^{(m)} | f_{\mathbf{x}}^{(M)} \leq f_+, \mathcal{D}_t)$

is not truncated normal hence the probability distribution can be obtained using the predictive distribution from the GP:

$$f_{\mathbf{x}}^{(M)} \mid f_{\mathbf{x}}^{(m)}, \mathcal{D}_t \sim \mathcal{N}\left(u(\mathbf{x}), s^2(\mathbf{x})\right) \quad (4.9)$$

where

$$\begin{aligned} u(\mathbf{x}) &= \sigma_{\mathbf{x}}^{2(mM)} \left(f_{\mathbf{x}}^{(m)} - \mu_{\mathbf{x}}^{(m)} \right) / \sigma_{\mathbf{x}}^{2(m)} + \mu_{\mathbf{x}}^{(M)} \\ s^2(\mathbf{x}) &= \sigma_{\mathbf{x}}^{2(M)} - \left(\sigma_{\mathbf{x}}^{2(mM)} \right)^2 / \sigma_{\mathbf{x}}^{2(m)} \end{aligned} \quad (4.10)$$

Finally, the entropy can be calculated solving a one dimension integral, simplifying the problem:

$$H\left(f_{\mathbf{x}}^{(m)} \mid f_{\mathbf{x}}^{(M)} \leq f_+, \mathcal{D}_t\right) = - \int Z \Psi\left(f_{\mathbf{x}}^{(m)}\right) \log\left(Z \Psi\left(f_{\mathbf{x}}^{(m)}\right)\right) \mathrm{d}f_{\mathbf{x}}^{(m)} \quad (4.11)$$

where the single terms are:

$$\begin{aligned} Z &:= 1/\sigma_{\mathbf{x}}^{(m)} \Phi\left(\gamma_{f_+}^{(M)}(\mathbf{x})\right) \\ \Psi\left(f_{\mathbf{x}}^{(m)}\right) &:= \Phi\left((f_+ - u(\mathbf{x})) / s(\mathbf{x})\right) \phi\left(\gamma_{f_{\mathbf{x}}^{(m)}}^{(m)}(\mathbf{x})\right) \end{aligned} \quad (4.12)$$

This simple 1-Dimension integral can be computed using standard numerical methods, improving the overall performance of the algorithm.

Chapter 5

Numerical implementation of Bayesian Framework

This chapter describes the general implementation of the framework in Matlab will be discussed. Both the single and multifidelity approach are reported: the main difference relies on the acquisition functions as reported in chapters 3 and 4.

5.1 Single fidelity Bayesian Optimization

The code starts with setting the framework parameters in the *opt* data structure. In particular the dimension of the problem, the range in which the problem variables can be selected, the maximum number of iterations, the budget, the number of tests to run to build a statistic and the number of points to use for the initial surrogate are defined. After this initial phase the effective optimization process can start.

An initial Latin Hypercube Sampling with the parameters in the *opt* structure is run. From this LHS an initial group of samples is selected randomly to build the initial surrogate.

After selecting the input point x_i the objective function of the problem is calculated $f(x_i)$. The calculated output values are collected and stored in a y vector. Before starting the optimization loop the initial budget spent to evaluate the various $f(x_i)$ is calculated. The minimum value are calculated and collected in a vector *minvalues* with their corresponding input values *minsamples*. The surrogate with these initial sampled points is built.

Then the optimization for loop can start. The surrogate is built by fitting a Gaussian process regression that uses a square kernel. The posterior is obtained in terms of μ and σ^2 from the Gaussian Process. Then the acquisition function is calculated and maximized to select the next point x_{i+1} from the LHS samples set.

The objective function is evaluated on the selected point which is then removed from the design space in order to avoid its reuse.

After each iteration the minimum value of the objective function is calculated and the vectors *minvalues* and *minsamples* are updated along with the budget. Then the cycle is repeated until the maximum iterations are reached, or the budget is terminated (typically the second condition is stricter than the first one) and the surrogate is updated iteratively with the new evaluations.

Singlefidelity EGO algorithm	
1.	Sample points with a LHS and create a Design of Experiment
2.	Evaluate the $f(x_i)$
3.	Calculate the minimum value $f(x^+)$ and store in <i>minvalues</i> and the x^+ in <i>minsamples</i>
4.	For i=1:max iterations or end of budget
5.	Update the GP surrogate
6.	Obtain μ and σ^2 from GP
7.	Calculate $EI = \max\{f(x) - f(x^+)\}$ Eq. 3.3
8.	Select next query point $x_{i+1} = \text{argmax}(EI)$
9.	Remove the point x_{i+1} from the design of experiment
10.	Evaluate the objective function on the x_{i+1} point
11.	Calculate the minimum of the objective function and update the <i>minvalues</i> and <i>minsamples</i> and calculate used budget
12.	end for

Table 5.1: Single fidelity framework

5.2 Multifidelity Bayesian Optimization

The multifidelity code implementation is an extension of the singlefidelity as described in chapter 4. The first phase of the code consists in setting the framework parameters, which have more terms considering the single fidelity case. In particular the parameters are: the dimension of the problem, the range in which the problem variables can be selected, the maximum number of iterations, the budget, the number of tests to run to build a statistic, the number of points from the design space to use for the initial surrogate and the cost of each level of code λ^m (the highest fidelity equals 1 and the lower fidelities cost is calculated accordingly).

An initial set of input points are selected from the LHS and the objective function is evaluated with the code of different fidelities indicated as m . The values are stored in a vector $y^m = f(x_1)^m \dots f(x_N)^m$ for each level of fidelity. The minimum value of the objective function is calculated between the evaluation on the highest fidelity code and stored in the vector *minvalues* and the relative input data in *minsamples*.

The evaluated objective function is then used to fit a Multifidelity Gaussian Process regression model and then the optimization process starts.

The mean value μ and the variance σ^2 are obtained from the posterior and are used in the calculation of the acquisition functions. Maximizing those, that are an extension of the acquisition functions for the single fidelity case, the next input point x_{i+1}^m along with an appropriate fidelity level m are selected.

Then the objective function is evaluated on the selected point x_{i+1}^m using the code of selected $m - th$ fidelity level.

The x_{i+1}^m point is removed from the design space to prevent replicates. The *minvalues* and *minsamples* vectors are updated with the minimum value of the objective function, along with the calculation of the total evaluations cost.

Then the code is repeated until the budget or the maximum number of iterations is reached.

In the next tables the pseudocode implemented for the 3 multifidelity acquisition functions are reported (table ??).

MFEI algorithm	
1	Sample point with a LHS
1	Evaluate the $f(x_i)$ from random points selected from LHS
3	Calculate the minimum value $f(x^+)$ and store in <i>minvalues</i> and the x^+ in <i>minsamples</i>
4	for i=1:max iterations or end of budget
5	Update the GP surrogate
6	Extract $\mu^{(m)}$ and $\sigma^{2(m)}$ from GP
7	Calculate the minimum value $f(x^+)$ and store in <i>minvalues</i> and the x^+ in <i>minsamples</i>
8	for m=1: M^{th} fidelity
9	Calculate $\alpha_1(\mathbf{x}, m) = \text{corr} \left[f^{(m)}(\mathbf{x}), f^{(M)}(\mathbf{x}) \right]$
10	Calculate $\alpha_2(\mathbf{x}, m) = \left(1 - \frac{\sigma_\varepsilon}{\sqrt{\sigma^{2(m)}(\mathbf{x}) + \sigma_\varepsilon^2}} \right)$
11	Calculate $\alpha_3(m) = \frac{\lambda_M}{\lambda_m}$
12	end for
13	Calculate EI= $\max\{f(x)^{(M)} - f(x^+)^{(M)}\}$ (calculated on the highest fidelity code)
14	Calculate MFEI(\mathbf{x}, m)= $\text{EI} \cdot \alpha_1(\mathbf{x}, m) \cdot \alpha_2(\mathbf{x}, m) \cdot \alpha_3(m)$ 4.2
15	Select next query point $[x_{i+1}, m^{th}] = \text{argmax}(\text{MFEI})$ (the m^{th} is selected between all the available fidelities)
16	Remove the point x_{i+1} from the design of experiment
17	Evaluate the objective function on the x_{i+1} point
18	Update the <i>minvalues</i> and <i>minsamples</i>
19	end for

Table 5.2: MFEI pseudocode implementation

MFEI algorithm	
1	Sample point with a LHS
1	Evaluate the $f(x_i)$ from random points selected from LHS
3	Calculate the minimum value $f(x^+)$ and store in <i>minvalues</i> and the x^+ in <i>minsamples</i>
4	For i=1:max iterations or end of budget
5	Update the GP surrogate
6	Extract $\mu^{(m)}$ and $\sigma^{2(m)}$ from GP
7	Calculate the minimum value $f(x^+)$ and store in <i>minvalues</i> and the x^+ in <i>minsamples</i>
8	for m=1: M^{th} fidelity
9	Calculate $Corr(\mathbf{x}, m) = Corr [f(\mathbf{x})^M, f(\mathbf{x})^m]$
10	Calculate $CR(m) = \frac{\lambda_M}{\lambda_m}$
12	end for
13	Calculate $\eta(\mathbf{x}, m)$
14	Calculate $PI(x) = \Phi \left(\frac{f(x^+) - f(x)}{\sigma(x)} \right)$??
15	Calculate $MFPI(\mathbf{x}, m) = PI(\mathbf{x}) \cdot Corr(\mathbf{x}, m) \cdot CR(m) \cdot \eta(\mathbf{x}, m)$
16	Select next query point $[x_{i+1}, m^{th}] = \text{argmax}(MFPI)$ (the m^{th} is selected between all the available fidelities)
17	Remove the point x_{i+1} from the design of experiment
18	Evaluate the objective function on the x_{i+1} point
19	Update the <i>minvalues</i> and <i>minsamples</i> and budget
20	end for

Table 5.3: MFPI pseudocode implementation

MFEI algorithm	
1	Sample point with a LHS
2	Evaluate the $f(x_i)$ from random points selected from LHS
3	For i=1:max iterations or end of budget
4	Update the GP surrogate
5	Extract $\mu^{(m)}$ and $\sigma^{2(m)}$ from GP
6	Calculate the minimum value $f(x^+)$ and store in <i>minvalues</i> and the x^+ in <i>minsamples</i>
7	Calculate $H0=H(f_x^{(m)} D_t) = \log(\sigma_x^{(m)}\sqrt{2\pi e})$ eq. 4.6 and $\lambda^{(m)}$
8	for m=1: M^{th} fidelity
9	for k=1:Max num Montecarlo iterations
10	Calculate $\gamma_{f_+}^{(M)}(\mathbf{x}), \phi(\gamma_{f_+}^{(M)}(\mathbf{x})), \Phi(\gamma_{f_+}^{(M)}(\mathbf{x}))$
11	if m=M
12	Calculate $H1=\sum_{f_+ \in \mathcal{F}_+} \frac{H(f_x^{(M)} f_+^{(M)}, \mathcal{D}_t)}{ \mathcal{F}_+ }$ eq 4.7 and eq 4.8
13	else
14	$Z, u(\mathbf{x}), s^2(\mathbf{x}), \gamma_{f_x}^{(m)}, \Psi(f_x^{(m)})$ eq. ??
15	$H1=\sum_{f_+ \in \mathcal{F}_+} \frac{H(f_x^{(m)} f_+^{(M)}, \mathcal{D}_t)}{ \mathcal{F}_+ }$ eq4.7 and eq 4.11
16	end if
17	end for
18	Calculate MFMES= $(H0 - H1)/\lambda^{(m)}$ eq. ??
19	Select next query point $[x_{i+1}, m^{th}] = \text{argmax}(\text{MFMES})$ (the m^{th} is selected between all the available fidelities)
20	Remove the point x_{i+1} from the design of experiment
21	Evaluate the objective function on the x_{i+1} point
22	Update the <i>minvalues</i> and <i>minsamples</i> and budget
23	end for

Table 5.4: MFMES pseudo code

Chapter 6

Composite specimen damage identification

The goal of this work of thesis consists in using previously discussed technique to correctly identify a cut in a composite specimen. Health monitoring of engineering structures is a complex, but important branch that allows using structures and systems for many years, ensuring a high level of safety which is normally requested by the normative. In particular the concept of safety in the aerospace engineering field is crucial and strictly regulated because it should be minimized the occurrence of catastrophic events, that will cause the damage or loss of airplanes, involved structures and people. In the last decades composite materials have become widely used due to their excellent mechanical properties and their reduced weights. The major drawback of composites, which is also a strength point, lies in their complex and in some cases unpredictable mechanical behavior. Hence also their failure modes differ greatly from an isotropic material. We will focus on the fiber cut damage, which is critical because is difficult to detect (especially if that regards the inner plies of the composite) and deteriorates the mechanical performance. This is due to the discontinuity of the fiber itself which causes the discontinuity of stress transfer. Hence in the region of the cut just the matrix, which has lower mechanical properties than the whole laminate, will transfer the load across the structure. Moreover a fibercut can lead to matrix cracking and delamination, that further reduces the mechanical properties of the laminate [71]. In order to monitor the damage level of the structure and identify the position of the cut, the strain field will be calculated and analyzed because it's a common parameter employed in health monitoring and damage assessment. The strain max value can be used to evaluate structural failure and it's a quite easy measure to take in laboratory, with the goal of using different sources of information building the surrogate model [72, 73]. In those cited works the strain field is read in discrete points using a

finite number of Fiber Bragg gratings, while in recent advance SHM techniques the complete strain field is evaluated even on complex structure using optical image correlation [74, 75]. These newer approaches are particularly suited dealing with the complex strain field that is present in a damaged composite structure. The dimensions and material of the specimen is taken from the work of professor Mainini¹. In particular we are using 4 layers of plain-weave prepreg carbon fiber composite (IM7/8552) with a stacking sequence [45°/0°/0°/45°] with dimension 456X102 mm and the thickness of each ply 0.76 mm.

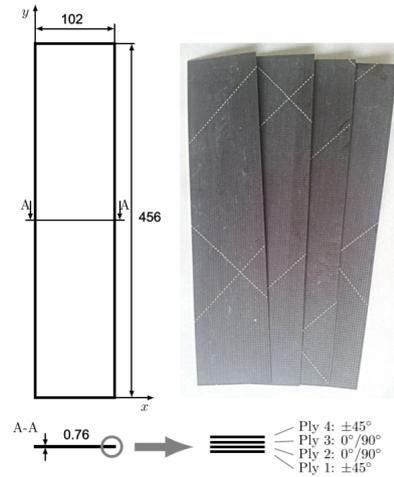


Figure 6.1: Specimen dimensions and lamination stacking sequence

The material properties of the specimen are taken from [76]:

E_{11}	64120 MPa	E_m	4600 MPa
E_{22}	64120 MPa	ρ_m	1300 kg/m ³
G_{12}	5777 MPa		
ν_{12}	0.031		
ρ	1700 kg/m ³		

Table 6.1: Hexcel IM7/8552 AS4 Plain Weave mechanical properties

¹Visiting professor program project: "A surrogate modeling approach to support real-time structural assessment and decision-making: Integrating data from experiments and numerical simulations"

6.1 Theoretical background

The finite element code used to solve the structural problem is NASTRAN. In particular the element chosen is a quadrilateral isoparametric shell element with linear shape functions. There are 6 degree of freedom (DOF) for each node (3 translations and 3 rotations) for a total of 20 DOF for each element. The code is employed to solve a linear static analysis, i.e. solve the typical structural problem stated as $[K]\{\mathbf{u}\} = \{\mathbf{f}\}$ where $[K]$ is the stiffness matrix of the entire structure, $\{u\}$ is the vector containing all the DOF of the structure. The stiffness matrix is calculated for each element in its coordinate system, then is rotated in the global reference system and finally all the matrix from each element is assembled together considering the connection between single elements. In Nastran the First Order Shear Deformation plate theory is implemented, which is an extension of the Reissner Mindlin to multilayered structure. This theory consists in an extension of the Kirchoff plate theory considering also the transverse shear. Solving the system leads to obtaining the displacement, for each DOF, in the nodes. The strains, which is the quantity monitored, are calculated with a post process routine knowing that are the derivative of the displacements. The displacement field is linear across the thickness of the laminate hence the strain is also linear through the thickness of all the laminate. This can be a limitation for our application because the sudden discontinuity in the strain field can be hard to catch. This limitation is in the low and mid fidelity models, due to the shell element employed.

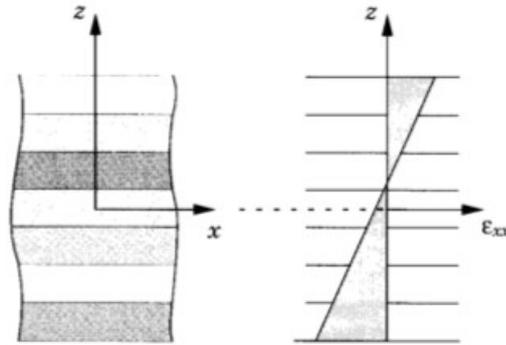


Figure 6.2: Example of strain distribution through the thickness of a laminate [77]

6.2 Patran modeling

Specifically different fidelities will be employed to speed up the optimization process. The models should be created considering variables that greatly influence the strain field in the specimen such as geometric dimensions, cut characteristics and different load levels; to permit this flexibility the models are developed leveraging the *Patran Command Language (PCL)* which is a programming language developed to code the interactions with the software and automate the modelization, maintaining the ability to tune the model parameters of interest. In order to avoid the *Curse of Dimensionality* the models were developed with a fixed geometry and lamination sequence in order to focus only on the damage problems. The variable chosen to model the damage are the cut location x_c and y_c , the cut extension in the x direction l_c . The strain recorded in a report file is in the y direction, the ϵ_{yy} .

6.3 Low fidelity model

The low fidelity model is modeled with a quad4 shell element that has 4 nodes with 6 DOF each. The boundary conditions are chosen to simulate a tensile test in accordance with the ASTM D3039 [78] imposing a clamp in the lower section and a displacement in the upper position; the boundary conditions are applied on an area that is extended by the 10% of the specimen total length. The remaining 80% of the central part of the specimen is discretized and used to calculate the strain field.

The cut is modeled as a rectangular zone. The extension of the cut in the y direction is fixed at 2 mm while in the x direction goes from 0 (there isn't any cut) and 60 mm (the cut is extended for more than half the specimen width). The cut zone moves across the entire specimen (excluding the clamp and load area). The damaged zone is modeled with a laminate property that assigns the matrix material in the damaged ply: in this zone the load is not carried by the fibre. In our case we chose to consider the cut on the 3rd layer, because internally positioned in the laminate is more difficult to observe and catch. The stiffness of elements in the cut region is reduced, the displacement increases, and the strain expected in this region should also increase. Laminated property is assigned to the other regions around the cut and has the composite material applied on all 4 layers.

The discretization of the specimen is made with a one-way bias mesh seed, starting from the cut region through the borders of the specimen: this method permits to increase mesh density in the cut region, which is a critical zone due to the sudden change in the strain field that should be caught. In addition to that, the mesh dimensions (the minor dimension of the elements is placed near the cut while the coarser is far from the cut) could be changed with different combinations to obtain

a sort of continuous variable fidelity model. The mesh dimensions could also be considered as model variable.

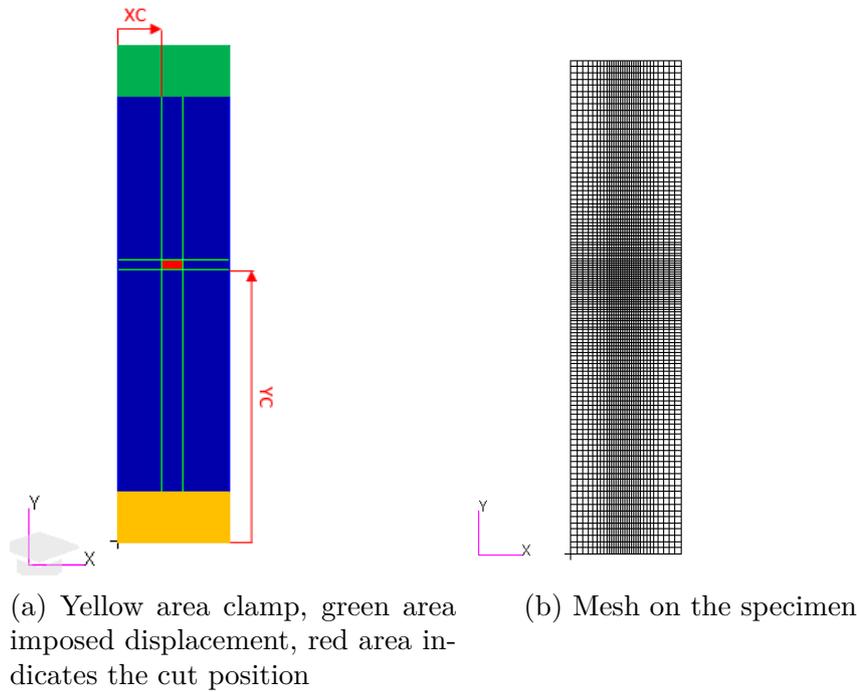


Figure 6.3: Specimen boundary conditions and mesh

From figure 6.3b can be clearly seen that in the cut zone the elements shrink and are coarser far from the damaged zone. While the mesh bias is shown on the next figure 6.4. The x_c and y_c positions are given between the origin of the specimen (the lower left corner) and the lower left corner of the cut region. Moving away from the cut zone the dimension of the elements increase in both the x and y directions.

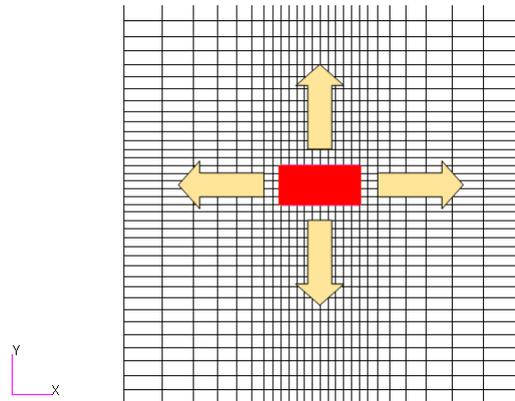


Figure 6.4: Mesh bias on the specimen direction of increasing dimension on the specimen

The mesh discretization for the low fidelity model is with a min mesh dimension of $1mm$ and a max of $2mm$.

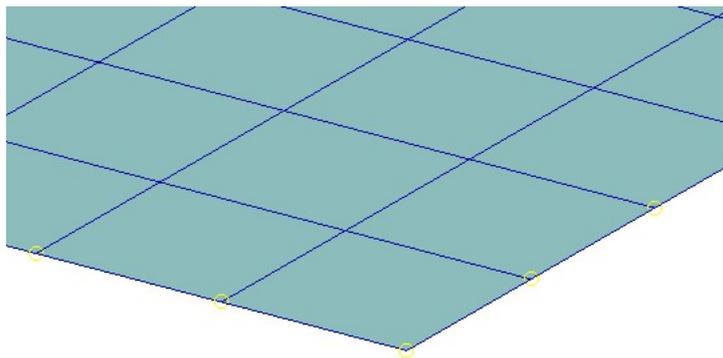


Figure 6.5: Low fidelity 2D model mesh

6.4 Mid fidelity model

The mid fidelity model uses the same quad4 elements and the same material definition and lamination of the low fidelity model. Also, the boundaries condition applied are the same from both the lower section, where a clamp is applied and the upper where an imposed displacement of $5mm$ is applied. The main difference, with the objective to focus the attention on the cut zone, is the mesh discretization with a min dimension of $0.5mm$ and a max dimension of the grid of $1mm$.

6.5 High fidelity model

In order to further increase the fidelity of our model we choose to use solid elements, the HEX8 that for our application where the load condition is mainly tension is appropriate. The HEX8 solid element is a brick with 8 nodes, one for each corner, and there are 3 DOF for each node: the 3 translations in the different directions for a total of 24 DOF for each element. The model is created using the 2D geometry of the specimen, used for the low and mid fidelity models, and extruding that in the transverse direction. Each layer is discretized using 3 elements in the thickness direction, in order to improve the accuracy in the calculation of the strain field and catch the discontinuity. In the preliminary phase was used more elements in the thickness direction and then was reduced due to the lack of strain field significative change. The elements material is modeled as a 3D orthotropic material, where the transverse property is the same of the matrix $E_{33} = 4600MPa$. The elements in the cut region are associated with homogeneous material, associated with the elastic modulus of the matrix. The boundary conditions are the same applied in the 2D models: the 10% lower part is clamped (the 3 DOF are imposed equal 0) while the upper 10% section imposes a vertical displacement of $5mm$. The higher computational cost, in opposition to the low and mid fidelity models, is due the overall higher number of nodes and hence total DOF: in the 2D model each element represent the mechanical characteristic of the whole laminate, while with the solid model each layer has 3 different elements linked together. The assumption of these layer, which is typical for the FSDT theory, leads in the perfect bond between different layers.

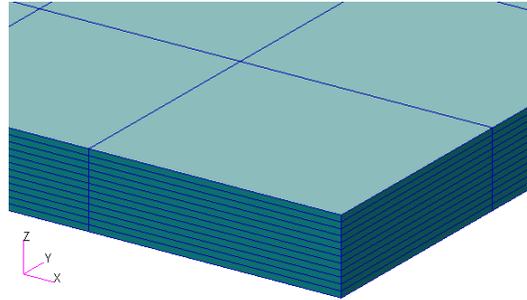


Figure 6.6: Higher fidelity model with 3D elements

The highest fidelity model is obtained using a grid on the 3D model with dimension $1 \times 3mm$ (intended as minimum and maximum mesh dimension), this is considered the "ground truth", approximating with high accuracy the specimen behavior. In the next sections the strain field is evaluated and discussed on 3 peculiar cut conditions: a cut on the border of the specimen, a small cut and a large cut.

6.6 Border cut damage

The cut positioned on the border is Case 17 reported in table 7.3 with the position $x_c = 74.66mm$ $y_c = 258.58mm$ and the length of $l_c = 25.74mm$.

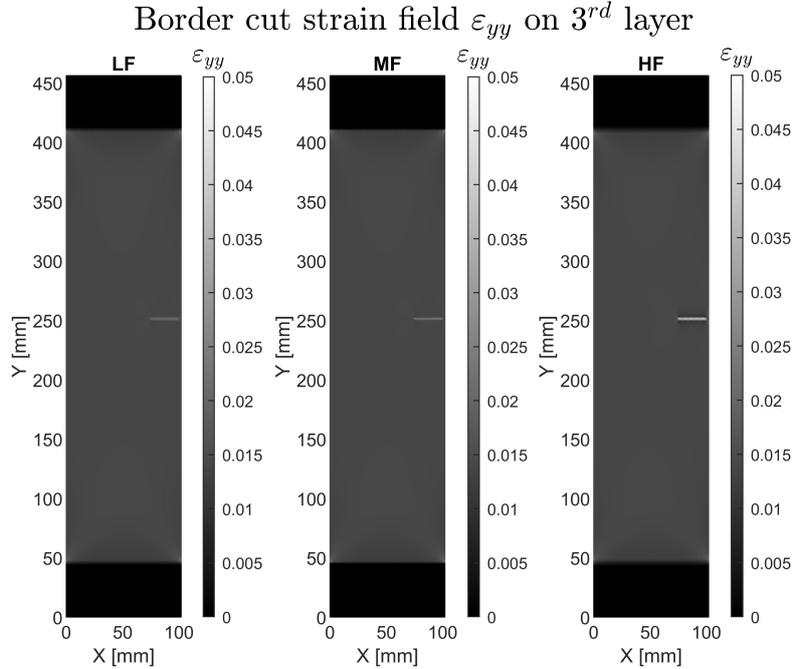


Figure 6.7: Strain field for a cut on the border of specimen with positions $x_c = 74.66mm$ $y_c = 258.58mm$ and length of $l_c = 25.74mm$

From the image 6.7 that reports the strain field in the 3rd layer, where the damage is located, can be seen that for each level of fidelity the cut is visible: for the low and mid fidelity is barely visible while is clear and easy to detect from the high fidelity model. In particular in the cut region the strain for each fidelity, which are also the maximum strain recorded in the specimen, are 2.19% for the Low fidelity code, 2.21% for the Mid fidelity code, 4.99% for the High fidelity code. The strain field in the other regions of the specimen are 1.32% for the low fidelity, 1.37% for the mid fidelity and 1.36%. The difference between the strain caught from the low and mid fidelity models is negligible but are slightly higher for the mid fidelity model: this behavior can be addressed to the overall reduction of stiffness for the mid fidelity model due to the usage of a higher number of nodes, while it's clear the greater ability of the solid model to catch the sudden jump in material and strength change. The presence of the border seems to have little effect on the overall strain field.

6.7 Small cut damage

The small cut damage considered is Case 4 reported in table 7.3 with the position $x_c = 86.11mm$ $y_c = 189.30mm$ and the length of $l_c = 0.96mm$

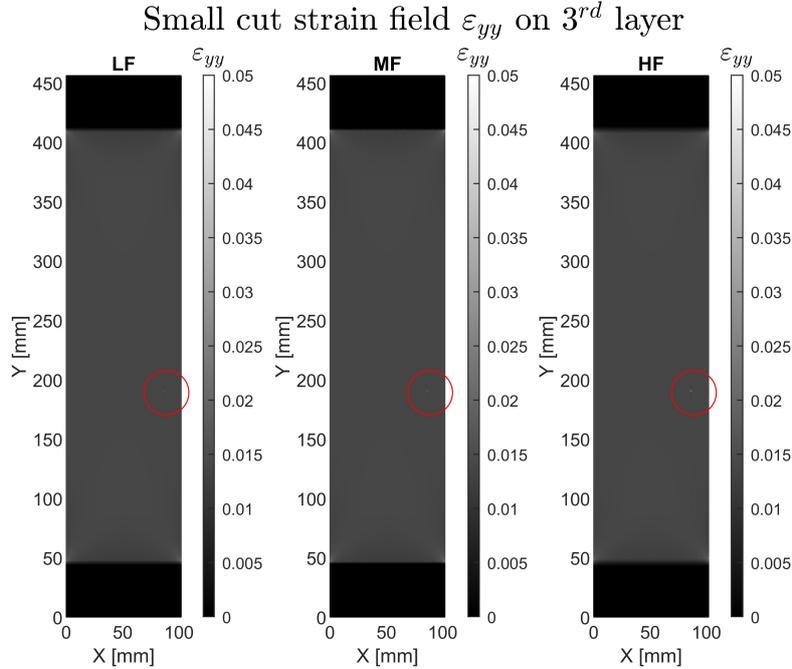


Figure 6.8: Strain field of a small cut on the specimen with position $x_c = 86.11mm$ $y_c = 189.30mm$ and length of $l_c = 0.96mm$

The image, as for the previous case reported, shows the strain field of the 3rd layer of the specimen for each fidelity. In particular in the cut region the strain for each fidelity, which are also the maximum strain in the specimen, are 1.59% for the Low fidelity code, 1.71% for the Mid fidelity code, 2.04% for the High fidelity code. The strain field in the other parts of the specimen are 1.39% for the low fidelity, 1.38% for the mid fidelity and 1.38%. As for the case discussed previously, the highest fidelity catches the greater strain discontinuity, while the mid and more the low fidelity struggle a bit more on. The difference for each level of code between the cut region and the remaining part of the specimen is small, this is because the cut length is short hence the reduction of the overall stiffness influences a small region on the specimen. This lack of reduction in the stiffness doesn't allow the cut region to deform much, hence the strain is similar to a "healthy" specimen. This complicates and stresses out the capability of the code to individuate small cuts.

6.8 Large cut damage

The largest cut extension is Case 25 reported in table 7.3 with the position $x_c = 11.86mm$ $y_c = 203.97mm$ and the length of $l_c = 59.34mm$

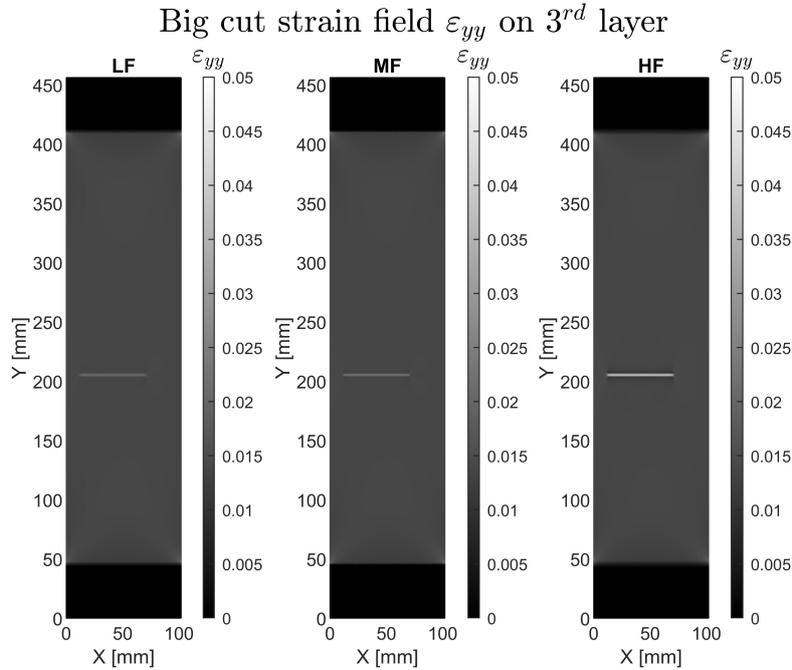


Figure 6.9: Strain field of a large cut on the specimen $x_c = 11.86mm$ $y_c = 203.97mm$ and length of $l_c = 59.34mm$

The image, as for the previous case reported, shows the strain field of the 3rd layer of the specimen for each fidelity. In particular in the cut region the strain for each fidelity, which are also the maximum strain in the specimen, are 2.13% for the Low fidelity code, 2.14% for the Mid fidelity code, 4.06% for the High fidelity code. The strain field in the other parts of the specimen are 1.37% for the low fidelity, 1.37% for the mid fidelity and 1.37%. With a larger cut the difference between the strain field in the cut region and on the remaining part of the specimen is higher leading to easily catch the strain discontinuity hence individuate the damaged zone. From the 3 test cases analyzed we can conclude that bigger is the cut damage, a greater part of the specimen will be affected by a reduced strength allowing the specimen to deform more, hence a higher discontinuity in the strain field is present. Then the mid fidelity model has a strain field that is slightly higher in value compared to the low fidelity model. Finally, the highest fidelity model is able to catch a greater discontinuity in the material, hence permits to identify the damage easily.

Chapter 7

Problem set up

With the set of previous models, we want to tune the mesh dimensions to obtain and investigate different cases focusing the attention on two key factors: using 3 fidelities to see if that speeds up the optimization process and using a fixed load, imposed on the upper part of the specimen equals to $5mm$. These assumptions are made to limit the curse of dimensionality, that is due to the multimodality of the problem which is typical of inverse problems. In particular we choose this case: the high fidelity uses the 3D model with min dimension of the mesh equals $1 mm$ and max dimension equals $3 mm$, the mid fidelity uses a 2D model with a minimum dimension of the mesh equals $0.5 mm$ and the max dimension equals $1 mm$ and finally the lowest fidelity employs a 2D model with a minimal mesh dimension equals $1 mm$ and the maximum equals $2 mm$. The cost of each model is calculated using the ratio between the number of DOF of each fidelity and the number of DOF in the model with the highest fidelity (for an arbitrary cut position taken). Also, the time to completely calculate the strain field and generate the report is used, because are taken into account the IO operations and gives a real estimation of the overall costs. While the model with their level of fidelities can be summarized as: To summarize the whole strain field and gain information about the position of the cut the RMSE between the strains on the highest fidelity model which is settled in a reference condition (that represents the "real" damage condition, and the optimization process should indicate the cut position and dimensions) and the model of an iteration is calculated; in particular due to the different mesh dimensions and cut positions, to calculate the error the nearest elements between the two models are considered. To emphasize the different strains between different configurations a normalization with the reference strain is made.

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N \frac{(S^{(i)} - S_{ref}^{(i)})^2}{S_{ref}^i}} \quad (7.1)$$

	High fidelity	Mid fidelity	Low fidelity
Max mesh Dimension	3 mm	0.5 mm	1 mm
Min mesh Dimension	2 mm	1 mm	2 mm
Cost	1	0.54	0.12
Element Type	Solid HEX8	Shell QUAD4	Shell QUAD4
Peculiarity in models	3 element for layer in the thickness direction	Lamination with orthotropic material and isotropic in the cut region	

Table 7.1: Model fidelities description

The cut damage properties boundaries and test cases are presented in the table:

Table 7.2: Cut damage properties

Test Case	X cut (x_c)	Y cut (y_c)	Cut length (l_c)	Test Case	X cut (x_c)	Ycut (y_c)	Cut length (l_c)
Limits	0-102	60-400	0-60	Case 13	61.5	137.14	15.32
Case 1	81.42	123.33	0.09	Case 14	16.1	392.53	18.17
Case 2	44.15	282.41	0.25	Case 15	24.07	66.09	20.61
Case 3	96.33	73.97	0.56	Case 16	11.84	105.99	21.73
Case 4	86.11	189.3	0.96	Case 17	74.66	258.58	25.74
Case 5	83.92	377.98	1.65	Case 18	50.96	240.2	29.1
Case 6	56.56	91.35	3.3	Case 19	37.49	362.51	32.04
Case 7	28.9	334.21	3.68	Case 20	61.09	172.46	37.33
Case 8	16.7	266.24	5.17	Case 21	58.34	295.1	39.41
Case 9	46.91	164.31	6.59	Case 22	32.65	150.25	45.28
Case 10	25.99	218.81	9.51	Case 23	1.81	305.67	49.22
Case 11	72.06	347.87	10.16	Case 24	38.05	235.1	52.28
Case 12	6.57	318.68	12.8	Case 25	11.86	203.97	59.34

Table 7.3: Damage test conditions, measures expressed in mm

The cut vertical position y_c is chosen between the constrained zones, while the cut dimension could be between 0 (the specimen is not damaged) and 60 (the cut relies on more than half the specimen width). In order to ensure the geometric consistency of the model if the sum of x_c and l_c exceeds the specimen width an arbitrary high value of the RMSE is assumed, leading the optimization process

to search the minimum value of the RMSE with other cut parameters. A visual representation of the damage test cases sample considered is reported.

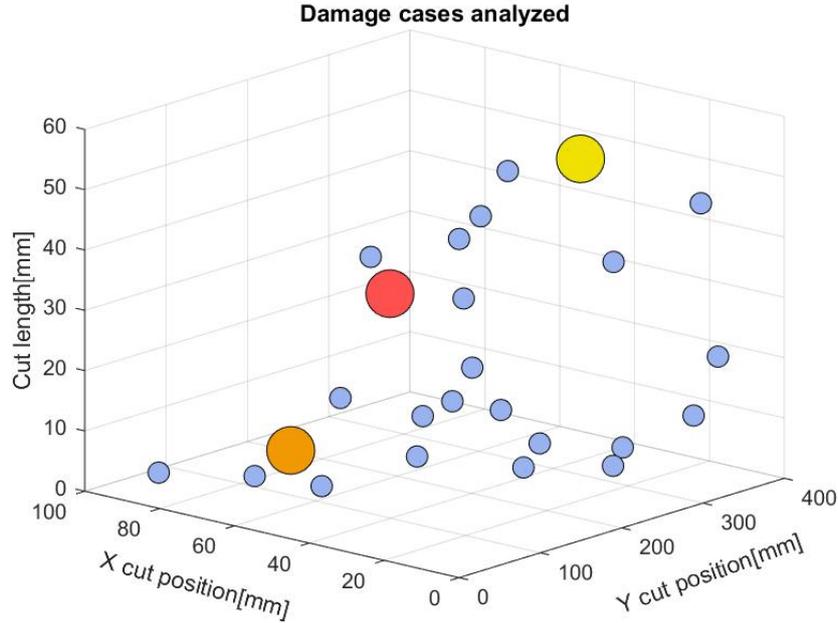


Figure 7.1: Damage test samples distribution. Orange circle small cut. Yellow circle large cut. Red dot cut on the border.

From figure 7.1 and more in detail from table 7.3 can be seen that the cut lengths are shifted to the lower bound, i.e. smaller cuts are considered because ideally the framework should be able to catch the damage in its incubating phase: this is in order to prevent its propagation and perform a repair on the structure in a short period of time. This shift of the cut length toward zero is possible due to the modified LHS presented in [79], the points sampled are closer to the origin of the damage length values distribution. This is possible due to raising the typical latin hypercube terms to a fixed power: this shifts the center of the sampling distribution towards the origin of the damage length value. The bigger dots in the figure indicate 3 damage test conditions which will be monitored to evaluate the performance of the code due to their representativeness of the different cases of possible damage position and dimensions. In particular the orange point is a small cut damage with length equal 0.96mm (the case 4 in table 7.3). The yellow point is a larger cut damage, with a cut length of 59.34mm (the case 25 in table 7.3) The red dot is a cut near the border of the specimen, with a distance of 2mm from the right border (the case 17 in table 7.3).

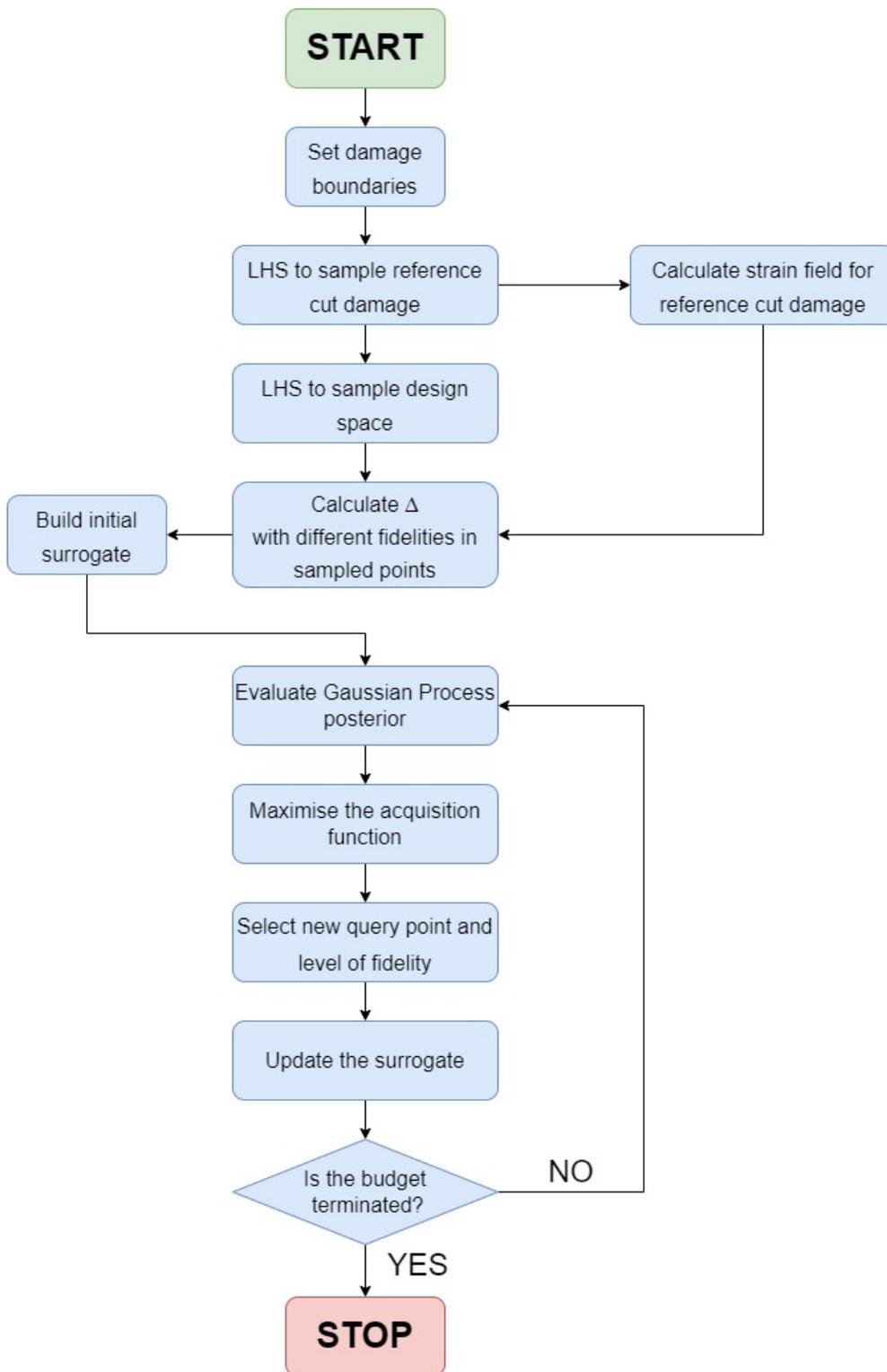


Figure 7.2: Diagram showing the optimization framework

The diagram in figure 7.2 illustrates the generic code implementation on the damage identification problem. The code starts by setting the damage parameters boundaries which will be imposed during the model creation. Then an initial LHS is used to randomly select the 25 different damage conditions with a focus on small cuts. The reference strain field is calculated for a damage condition from the 25 different cases. Then a LHS is used to sample the design space, which will be used during the optimization process. Initially the strain fields are evaluated using 20 Low fidelity code runs, 10 Mid fidelity runs, and 2 High fidelity runs selecting the damage parameters from the LHS. For each evaluation the RMSE is calculated using the reference strain as reported in 7.1. With this evaluation the initial surrogate, which uses a Gaussian Process regression is built. The minimum value of the RMSE along with the corresponding damage parameters are stored in the *minvalues* and *minsamples* vectors (the values are stored calculating the minimum of the RMSE on the high fidelity evaluations). Then the optimization process begins. The posterior is evaluated in terms of the mean μ and variance σ^2 of the Multifidelity Gaussian Process. These quantities are used to calculate the acquisition functions: by maximizing them the fidelity level and the next damage parameters are selected in order to improve the surrogate accuracy. Then the model with the queried fidelity and damage parameters is run. This new evaluation is used to update the surrogate. The minimum of the RMSE is evaluated again and the vector *minvalues* and *minsamples* are updated. The total cost is calculated and if is lower than the budget or the number of iterations is lower than the maximum the code is repeated. For each damage condition this optimization process is repeated 10 times to build a suitable statistic, this is due to the stochastic nature of the optimization process. Then the optimization process is repeated considering each of the 25 damage conditions using the different acquisition functions: Multifidelity Expected Improvement, Multifidelity Probability of Improvement, Multifidelity Max value Entropy Search.

Chapter 8

Results

In this section the results of the damage identification problem are presented. In order to obtain meaningful results, due to the intrinsic stochastic nature of the process, 25 different damage conditions are picked randomly with a Latin Hypercube sampling which is shifted towards short cut length: with this modification the damage considered is relatively small and allow to stress to code and verify if it is capable of identifying such small damages.

Then to build a suitable statistic for the problem 10 different tests for each damage condition, which means building 10 different surrogates and running 10 different optimization processes, are considered. This approach is repeated for the 3 different acquisition functions, considering the same 25 damages and the same sampling points to initially build the surrogate: this is made to have a common baseline for each method. Finally, the median value, of the minimum value of the RMSE taken from the High Fidelity evaluation, is calculated from the total 250 optimization runs and is represented as a solid line. The shaded area around each median represents the collected values in the confidence interval between the 15th and the 75th percentile.

The graphs representing the normalized errors on the estimation of the parameters x_c , y_c and l_c between the reference cut damage and the actual estimation of the code are reported: these are the real quantities of interest that needs to be found. These quantities representation is the same as the RMSE: the solid line indicates the median calculated from the 250 optimization runs and the shaded area shows the confidence interval between the 15th and the 75th percentile.

After reporting the overall results, where the median and the confidence boundary are calculated on all the 25 damage samples, the results on 3 peculiar cut damages are selected to show the performance of the code with different conditions. In particular from the 25 samples, we will select a cut condition where the damage is near the border of the specimen, a cut with short and another cut with large length.

8.1 Aggregated results

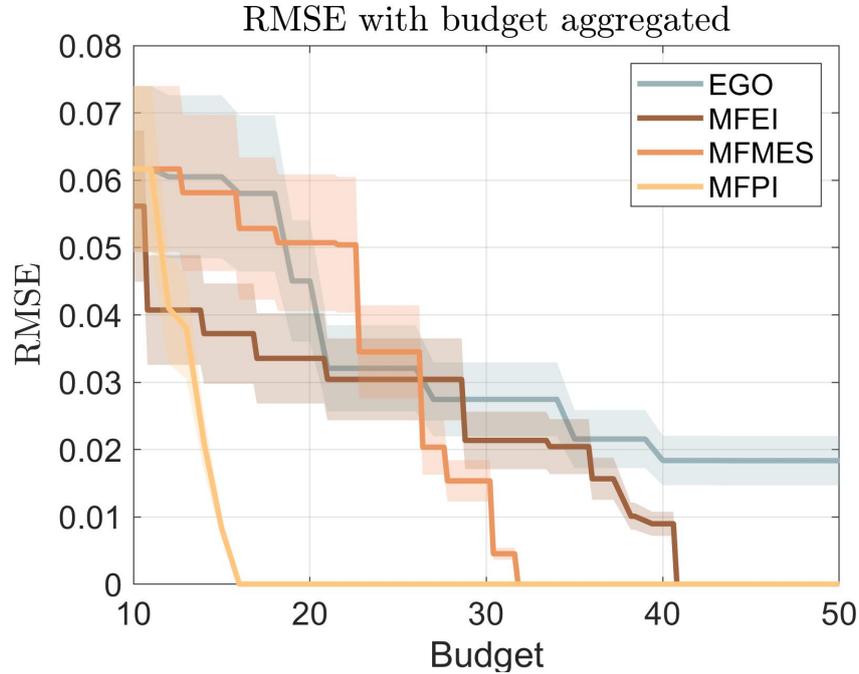


Figure 8.1: RMSE minimizing process results for each acquisition function

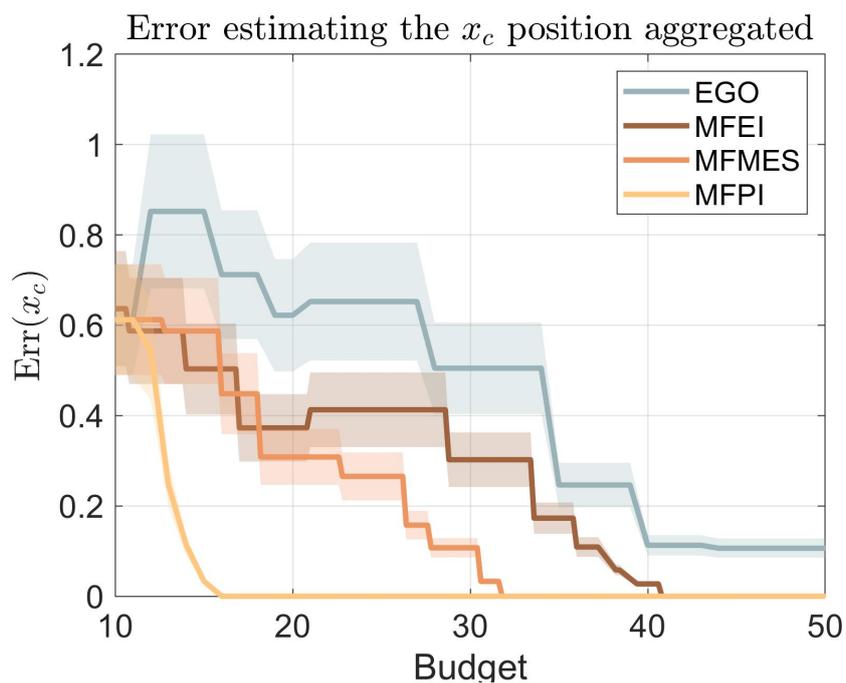
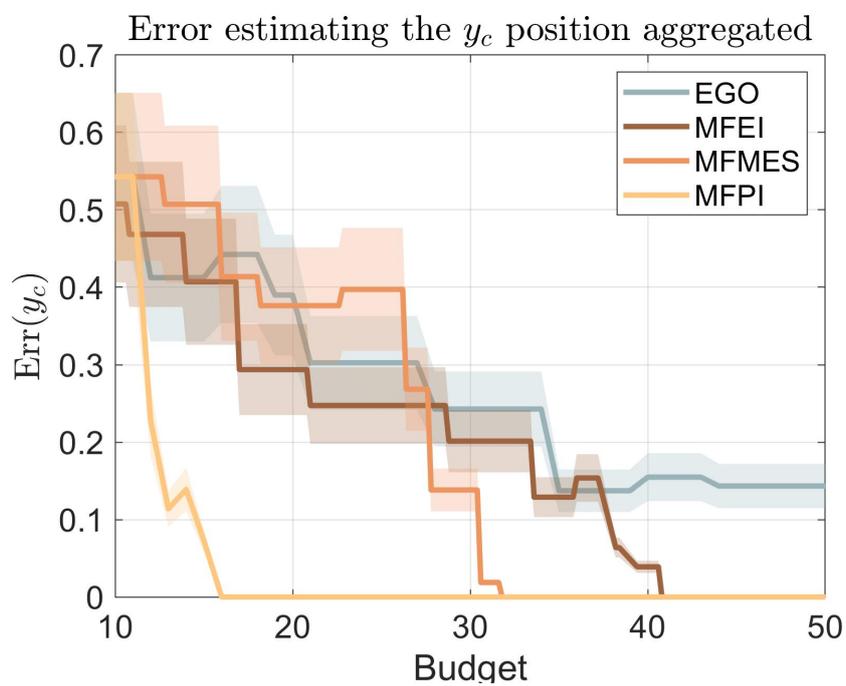
In the graph are represented the 3 multifidelity acquisition functions and the singlefidelity Efficient Global Optimization (EGO). It's clear the advantage on using the multifidelity approach since the EGO is the only case where the RMSE doesn't drop to zero. It can also be noted that the EGO curve doesn't fall as fast as the other ones. This can be explained with the limited exploration of the EGO due to the usage of just high fidelity code, hence with the same budget of other methods (fixed at 50) the number of sampled points is exactly 50. The fastest code to bring the RMSE to zero is the MFPI that has a higher number of High Fidelity evaluations meaning that more focus is brought on the exploitation that, in a low dimension problem, leads to rapidly find the minimum. On the opposite end, the slowest code is the MFEI. The MFEI is the most balanced algorithm with the right balance between exploration and exploitation. This characteristic doesn't lead to great performance due to the nature of the inverse problem: a more exploitative approach for a low dimensions problem greatly speeds up the optimization process. The MFMES performance relies between the other two. The exploration phase can be seen between budget 10 and 23 where the graph is mostly flat (which means that are made mainly low and mid fidelity evaluations) and then suddenly the slope of the curve is similar to the MFPI. This more explorative behavior can be addressed

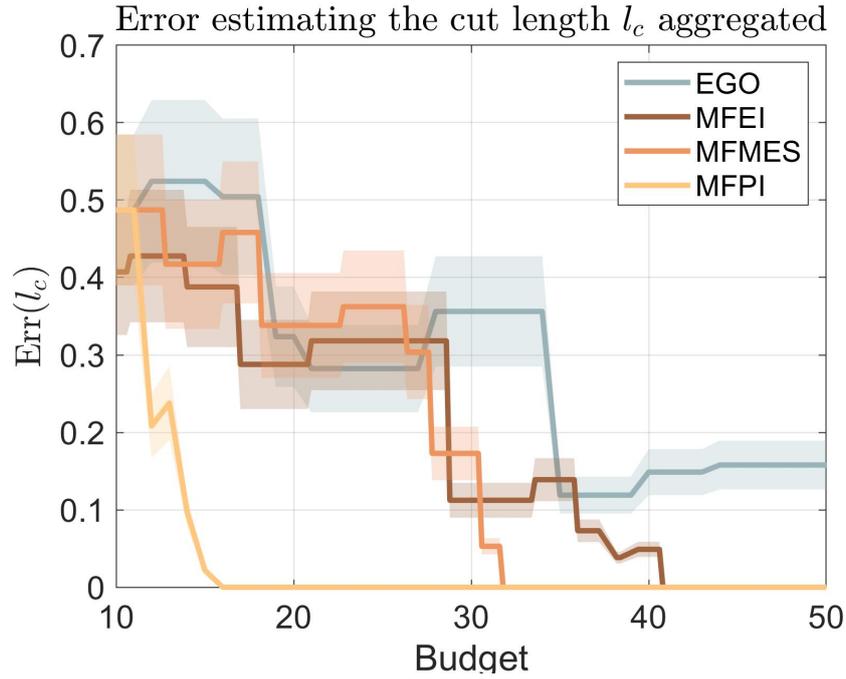
to the formulation: it calculates the entropy to measure the accuracy between the surrogate and the high fidelity code. A closed form to calculate the entropy is not available, hence Monte Carlo simulations are used. With this method, which is an approximation of the real entropy, the accuracy term is underestimated in favor of focusing on reducing the evaluations cost hence using much low fidelity evaluations. Considering that the mean duration of a test is 4.2 hours the mean time for each acquisition functions to bring the RMSE to 0 can be calculated.

	Time to reach RMSE=0 [min]	Evaluation High	Evaluation Mid	Evaluation Low
EGO	-	50	-	-
MFEI	206.4	10.5	47.8	97.5
MFMES	160	11.7	22.6	189.4
MFPI	80.4	18.1	36.4	88.1

Table 8.1: Mean evaluations times and mean number of evaluations for each fidelity

It's clear from the table 8.1 the superiority, in this particular application, of the MFPI that is twice as faster as the MFEI and more than that considering the FMES. In order to further appreciate the difference between the methods the mean number of evaluations for each fidelity can be observed. It's clear, as stated previously, that the FMES acquisition function selects the lower fidelity, and cheaper, hence it extensively explores the design space. The MFPI has a more aggressive approach on the exploitation phase, which in a low dimension problem helps to fast drive the optimization towards the optimum. While the MFEI is a balanced mix of exploration and exploitation and leads to poorer performance. Finally, another consideration can be made observing the relative errors that are measured considering the cut parameters between the damaged specimen and the high fidelity current evaluation.

Figure 8.2: Error estimating the x_c positionFigure 8.3: Error estimating the y_c position

Figure 8.4: Error estimating the l_c position

For the multifidelity frameworks the errors drop to zero when the RMSE also drops to zero meaning that the right cut configuration parameters are found. Observing the normalized errors it can be seen, in contrast to the RMSE value, that don't decrease monotonically but in some regions increase their value. This means that the damage identification case, which is an inverse problem, is ill-conditioned due to its probably natural multimodality nature. In the next sections we will discuss the 3 particular cases, a small, large and near the border cut, which strain field are analyzed in the previous chapter. Reporting these results permit to understand the robustness and the overall performance of the code in finding the correct damage parameters. In the next sections the results are presented for the peculiar cases of a cut on the border, a small and large cut selected from the sampled damage parameters and highlighted in table 7.3.

8.2 Cut damage on border

In this section the results of the cut damage positioned near the border are analyzed. The reported graphs are similar to those in the section on the aggregated results but in these cases the data utilized is referred to a single damage case (in this section we are referring to the cut near the border case 17 in table 7.3 with $x_c = 74.66mm$, $y_c = 258.58mm$ and $l_c = 25.74$). We are considering the 10 test evaluated, used to obtain a suitable statistic. The solid line indicates the median value of the RMSE calculated with the 10 test cases data while the shaded areas indicate the distribution of data between the 15th and 75th percentile. The initial points sampling to build the surrogate are different between each of the 10 tests but are shared between different methods in order to start the comparison from the same surrogate baseline.

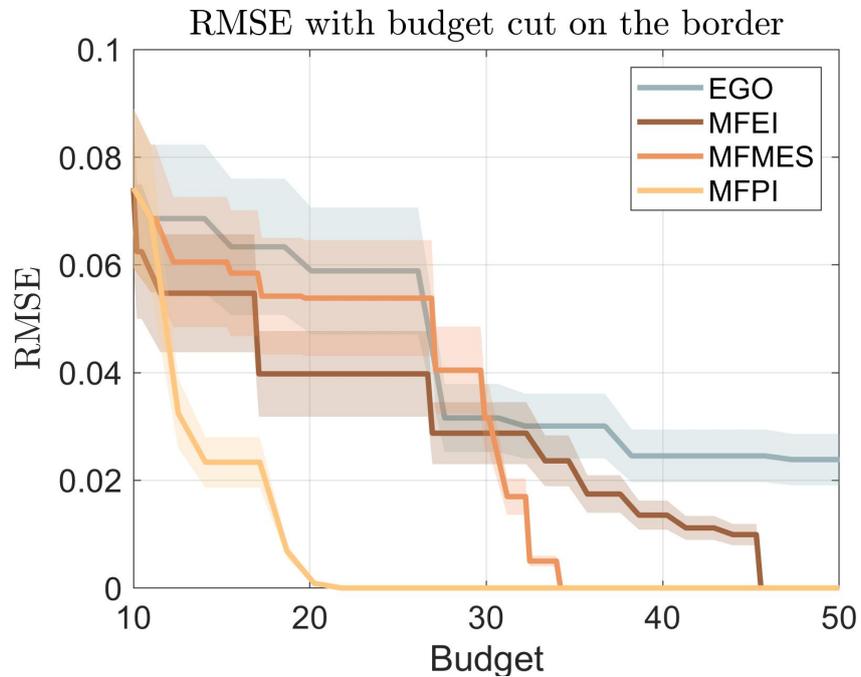


Figure 8.5: RMSE minimizing process results for each acquisition function cut on the border $x_c = 74.66mm$, $y_c = 258.58mm$ $l_c = 25.74$

As shown in the previous overall RMSE reduction graph on fig 8.1 the difference in performance between each method is similar. The single fidelity EGO model doesn't reach the RMSE zero, meaning that the singlefidelity approach will need a higher budget in contrast to the multifidelity cases. The fastest acquisition function that leads to bring toward zero the RMSE is the MFPI, which in this case has a little more exploration phase between budget 14 and 17 which leads to a

higher budget to reach the solution. The other 2 multifidelity acquisition functions, MFEI and FMES, explore also intensively between budget 17 and 30 indicating the difficulty of the code in individuating the right damage characteristics. The FMES is the second acquisition function that reach the convergence while the MFEI that balances exploitation and exploration is the last acquisition function that reaches convergence, and that can be due to the nature of the problem that leads to a more exploitative approach due to the low dimension problem. The times to reach convergence for each acquisition function can be reported:

	EGO	MFEI	FMES	MFPI
Time to reach RMSE=0 [min]	-	229.62	172.39	109.79

Table 8.2: Mean evaluations times and mean number of evaluations for each fidelity level

The next figures report the errors in estimating the damage parameters.

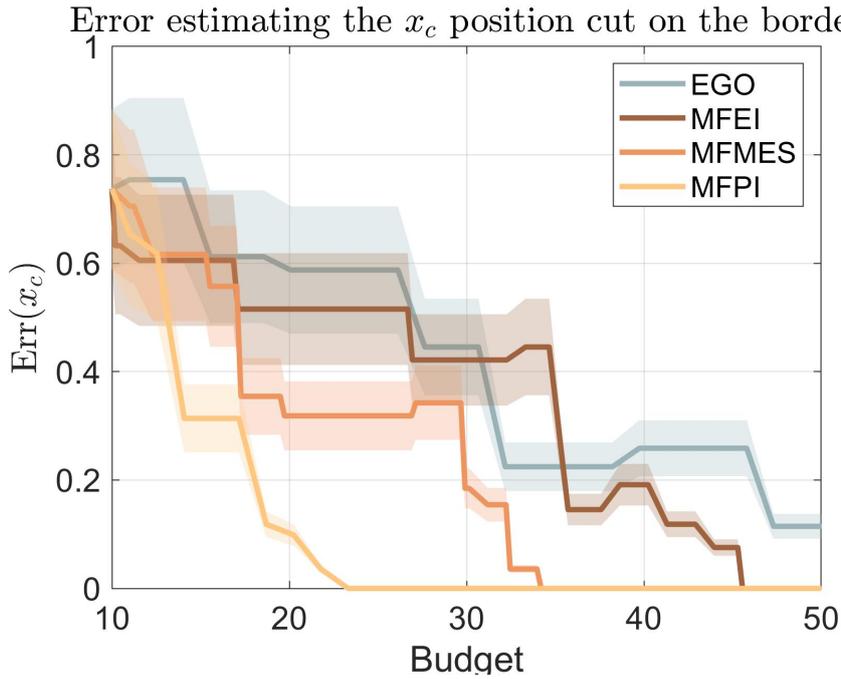
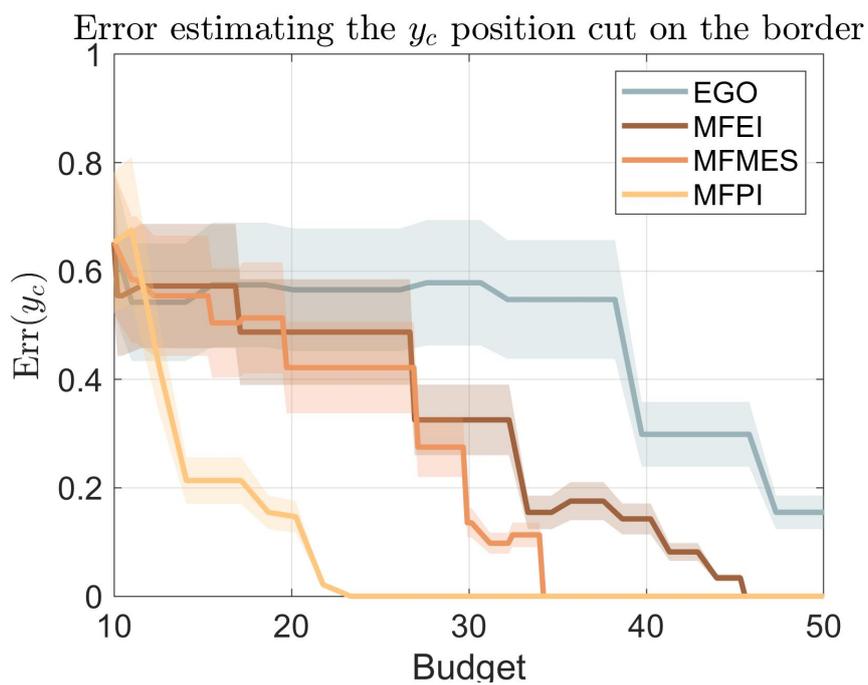
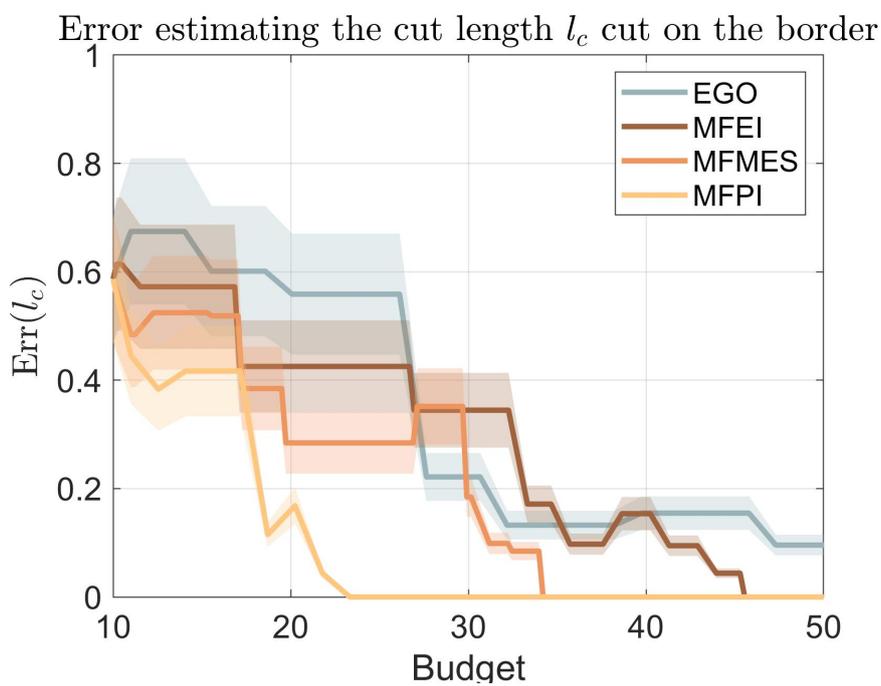


Figure 8.6: Error estimating the x_c damage parameter cut on the border

Figure 8.7: Error estimating the y_c damage parameter cut on the borderFigure 8.8: Error estimating the l_c damage parameter cut on the border

The error on the cut position is immediately reduced for the MFPI while the error on estimating the cut length in figure 8.8 is initially flat and then suddenly reduces. Then in the zone of budget between 20 and 30 as the models selected for the other acquisition functions are the lower and middle fidelity, the errors remain flat to then suddenly reduce, non monotonically during the next iterations.

8.2.1 Strain field evolution

The evolution of strain field is presented for each acquisition function in order to visually show the considerations made in the previous chapter.

EI Strain field evolution ε_{yy} on the 3rd layer border cut

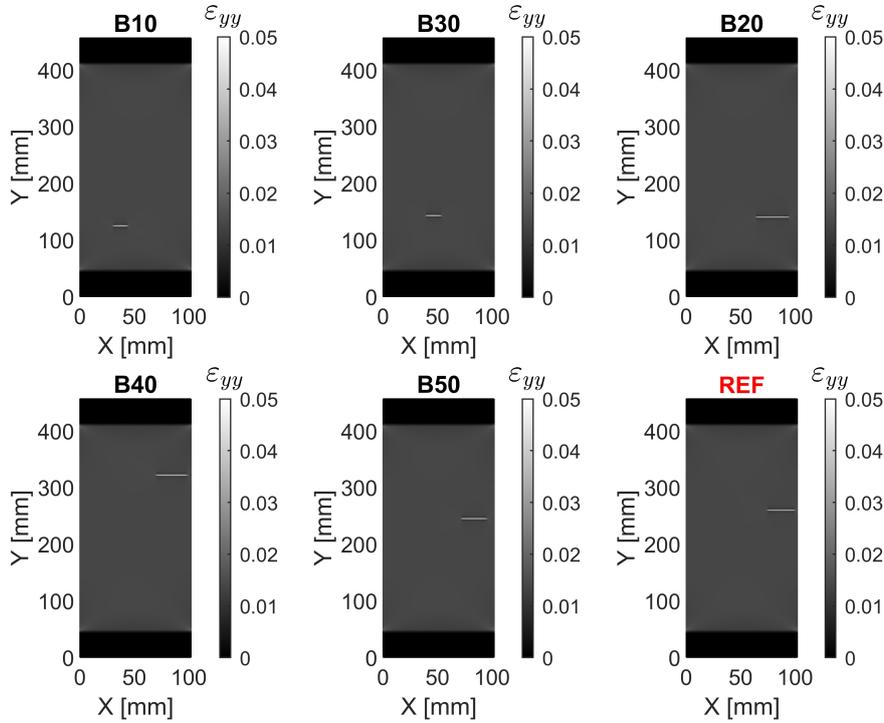


Figure 8.9: HF Border cut strain field evolution using EI acquisition function

The singlefidelity EI shows that for the damage on the border the estimation of the cut position gets pretty accurate. At the beginning of the optimization framework the cut length estimated is shorter than the real one, also the position is close to the center of the specimen. After some iterations the length of the cut, hence the strain in that zone, increases moving toward the right edge. This can be due to the little interaction between the cut and border that slightly modify the strain field

distribution. Finally, the last parameter to correctly catch is the y_c position and finally the right position and length are nearly obtained. This shows how the large cut properties can be identified with just the singlefidelity code with the downside of employing all the available budget.

MFEI Strain field evolution ε_{yy} on the 3rd layer border cut

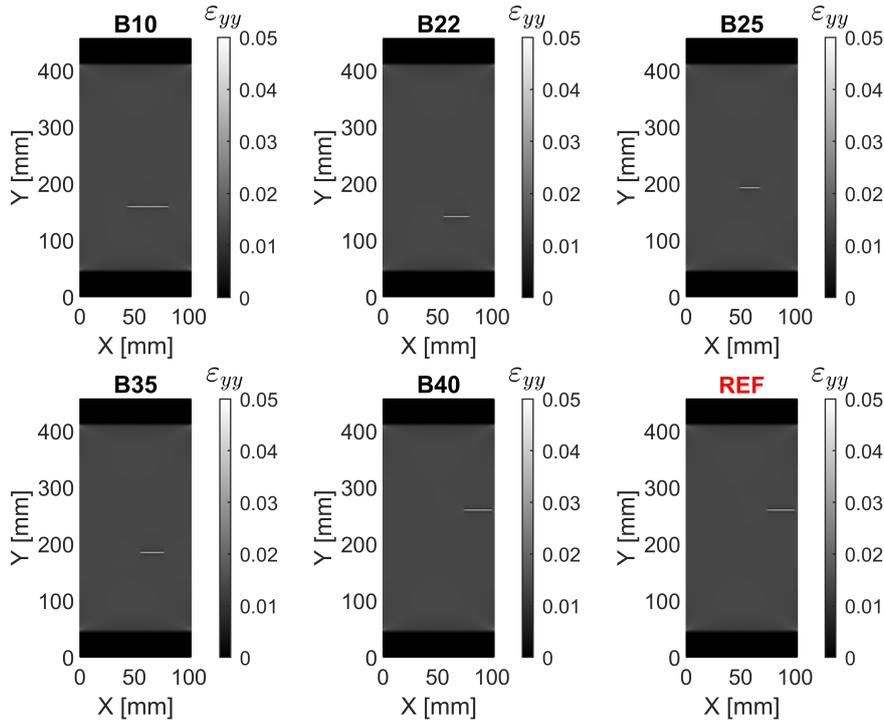


Figure 8.10: HF Border cut strain field evolution using MFEI acquisition function

Using the acquisition function MFEI the progression of the strain field starts with the position of the cut near the border and then further moves toward the border. In the initial phase to approximately budget 35 the code extensively explores the design space hence the predicted position of the cut doesn't change a lot. Then between budget 35 and 40 using high fidelity evaluation the right damage parameters are identified. This shows the advantage of using the multifidelity version of the Expected Improvement: a reduction in the necessary budget to correctly identify the damage parameters.

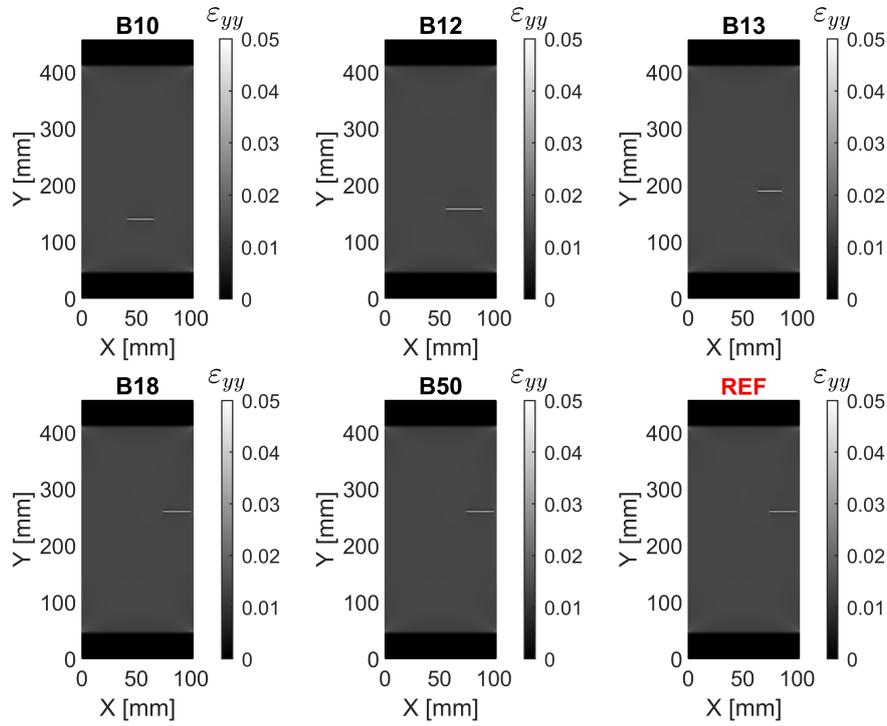
MFPI Strain field evolution ε_{yy} on the 3rd layer border cut

Figure 8.11: HF Border cut strain field evolution using MFPI acquisition function

The MFPI acquisition function shows its superiority in this application: within a small number of high fidelity evaluations the cut parameters are correctly identified for a budget around 18. The cut position moves towards the border while the length of the damage is estimated correctly after few iterations.

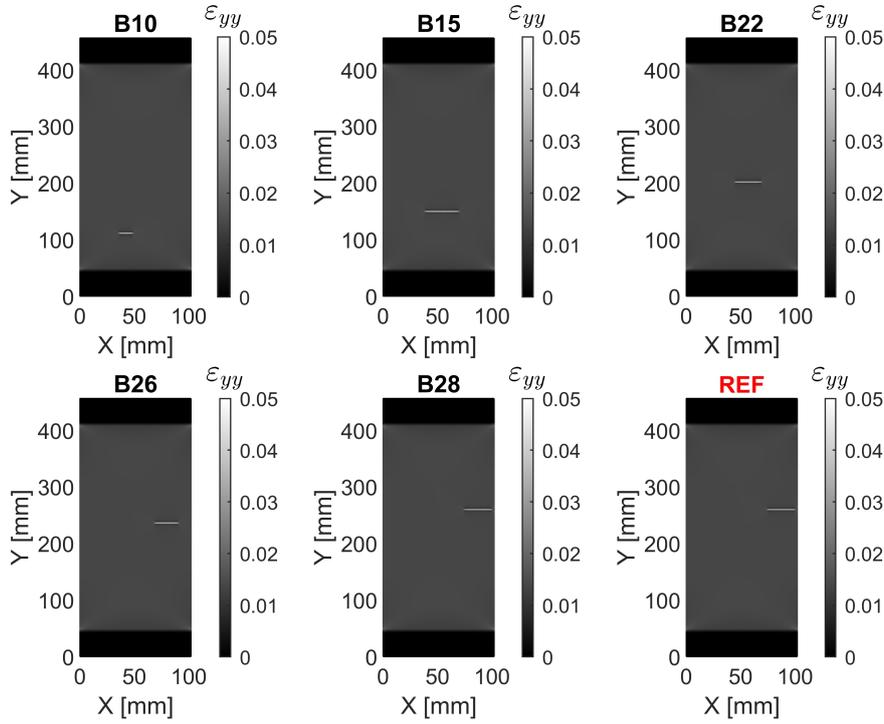
MFMES Strain field evolution ε_{yy} on the 3rd layer border cut

Figure 8.12: HF Border cut strain field evolution using MFMES acquisition function

The MFMES acquisition function uses a higher budget, considering the MFPI, to correctly identify the damage parameters. In the first iterations through budget 22 the cut length is overestimated and slightly moves toward the border. With further iterations the cut finally reaches the border with an approximate budget of 28. The behaviour of the different acquisition functions is similar with slight differences and budget used.

8.3 Small cut damage

In this section the result of the cut damage of small dimension is analyzed. In particular the damage condition is reported in table 7.3 as Case 4 (with $x_c = 86.11mm$, $y_c = 189.30mm$ and $l_c = 0.96mm$).

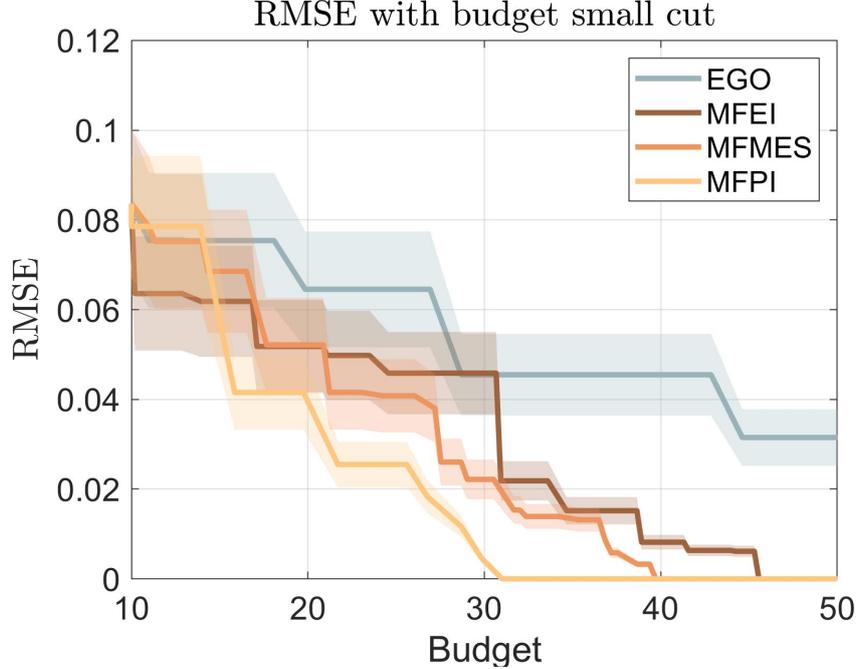


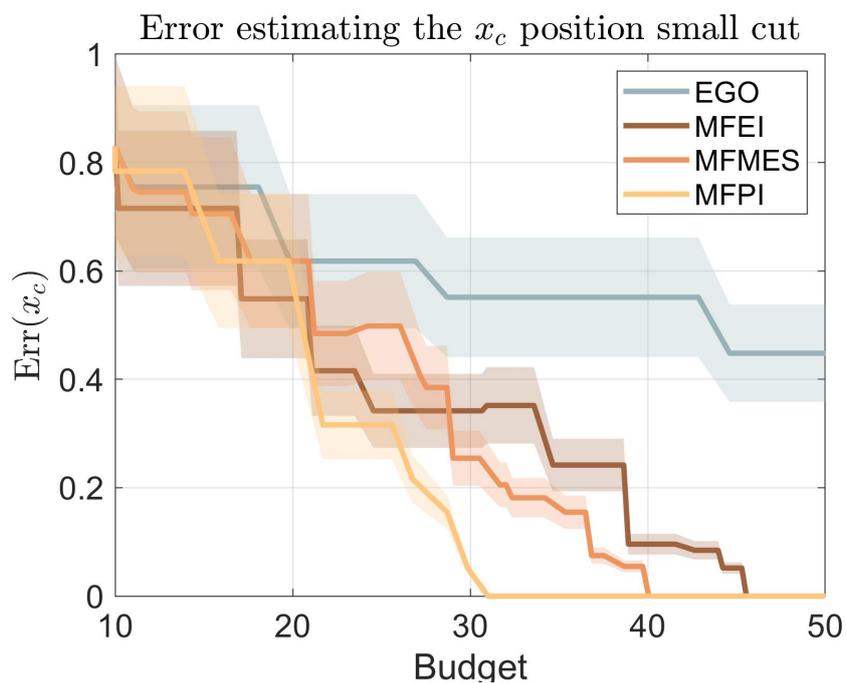
Figure 8.13: RMSE minimizing process results for each acquisition functions small cut $x_c = 86.11mm$ $y_c = 189.30mm$ $l_c = 0.96mm$

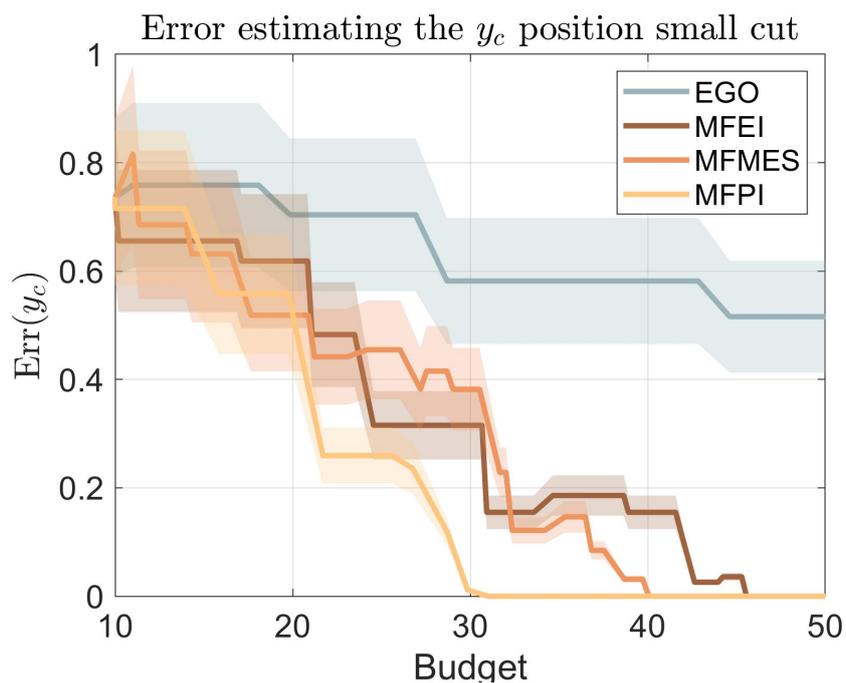
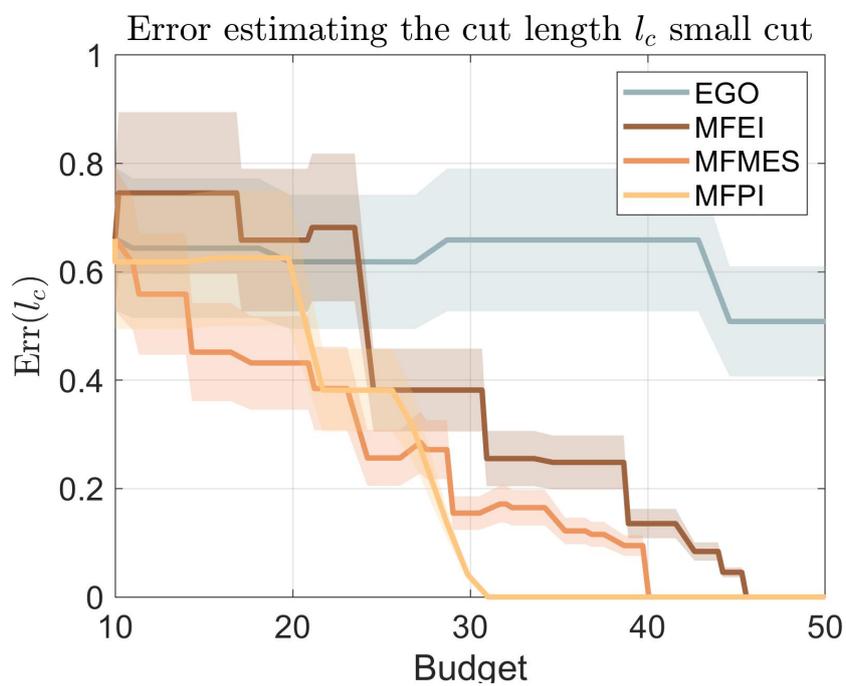
From the figure 8.13 is immediately noticeable that the Budget used from each fidelity to reach RMSE zero is way higher than in previous case. This can be addressed to the peculiar strain field and the difficulty to catch strain discontinuity, as reported in paragraph 6.7. The singlefidelity EGO code doesn't reach convergence and the value of the RMSE when the budget is finished is higher than the previous case: the surrogate needs an even higher budget to overcome the difficulty on finding the right damage parameters. The difference in performance between the MFEI and the MFPI is closer than previously: the higher number of low fidelity evaluations can help to find the damage characteristics quickly but a more exploitative approach in this scenario, such as the MFPI, still wins. The MFPI which puts more emphasis on the exploitation, in this particular case needs to explore also more extensively before reaching RMSE zero: this indicates that this peculiar cut condition stresses the code and requires a higher budget to find the right cut parameters.

	EGO	MFEI	MFMES	MFPI
Time to reach RMSE=0 [min]	-	229.62	200.17	156.26

Table 8.3: Mean evaluations times and mean number of evaluations for each fidelity

The table 8.3 further represents the difference between each acquisition function and is clear the increase in time to find the cut parameters considering the overall mean times from table 8.1. To understand which parameter is more difficult to find the errors associated with each of them are presented.

Figure 8.14: Error estimating the x_c damage parameter small cut

Figure 8.15: Error estimating the y_c damage parameter small cutFigure 8.16: Error estimating the l_c damage parameter small cut

As expected the main time consuming element to search is the cut length as reported in figure 8.16. For the MFPI the graph is mainly flat, meaning that in those zones even if the RMSE is decreasing, the model doesn't improve its estimate on the damage parameter. The other multifidelity acquisition functions have a similar decrease but the MFEI and more the MFMES can leverage more exploration and a stepper reduction in the errors (more relaxed). From all the figures can be noted that the EGO residual level of error is higher considering the cut positioned on the border: this means, and is clear from the large flat zones, that the performance of the EGO is worse due to the higher budget needed to reach convergence.

8.3.1 Strain field evolution

EI Strain field evolution ε_{yy} on the 3rd layer small cut

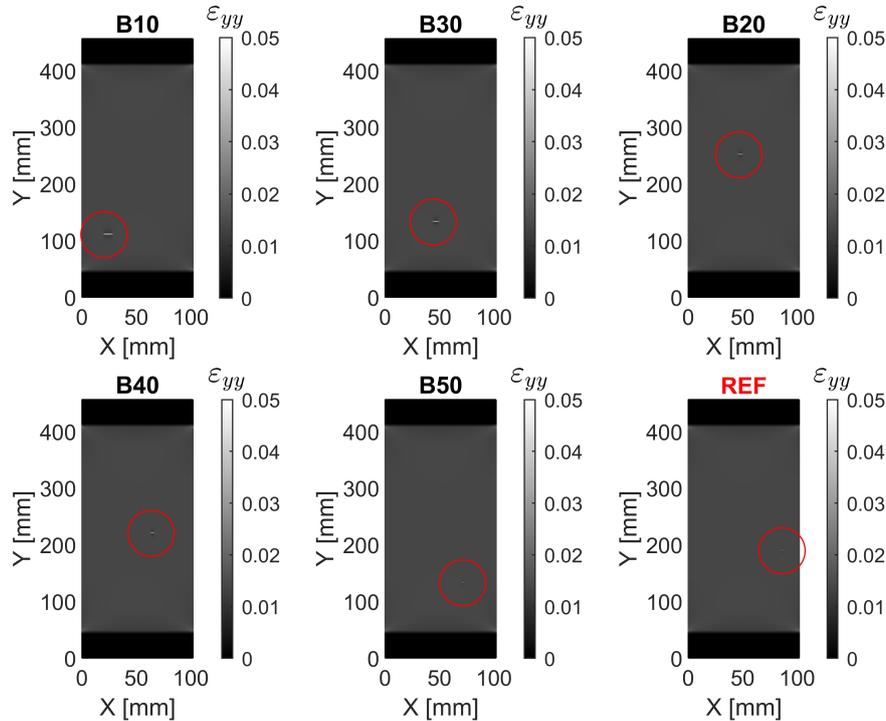


Figure 8.17: HF Small cut strain field evolution using EI acquisition function

The cut damage is indicated in the red circle due to its low dimension hence the difficulty to spot. From figure 8.17 can be seen that initially the cut length is overestimated: this due to the difficulty to catch the strain field with a smaller

damage. Between budget 20 and 30 the dimension of the cut estimated decreases along with the position that moves toward the border. Then thought budget 40 and 50 the length further decreases and moves near the border, but there is a bigger difference between the final parameter estimated at budget 50 with the reference, that are the right parameters. This case highlights the low performance of the singlefidelity code with a difficult strain field to be spotted. With a higher budget the singlefidelity code will reach convergence due to the necessity of an extensive search of the right damage parameters.

MFEI Strain field evolution ε_{yy} on the 3rd layer small cut

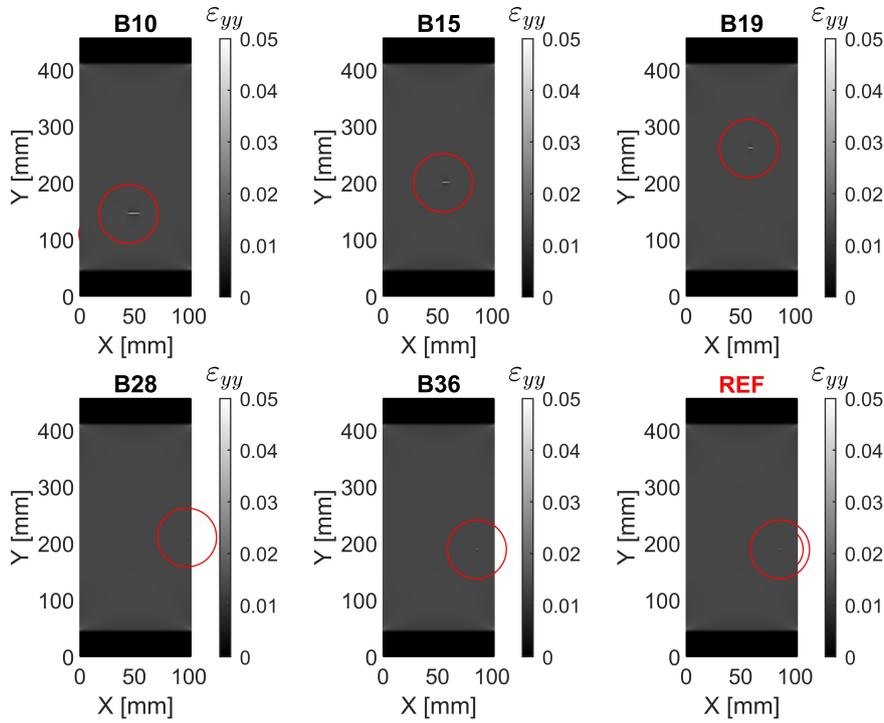


Figure 8.18: HF Small cut strain field evolution using MFEI acquisition function

The MFEI acquisition function starts with an estimation of a larger cut dimension and progressively reduces that through budget 19. Between budget 19 and 38 the cut length is further reduced and finally also the position is correctly identified. This acquisition function shows an initial greater uncertainty on the cut dimension and employs the highest budget to reach convergence (as indicated by the mean times reported in table 8.3).

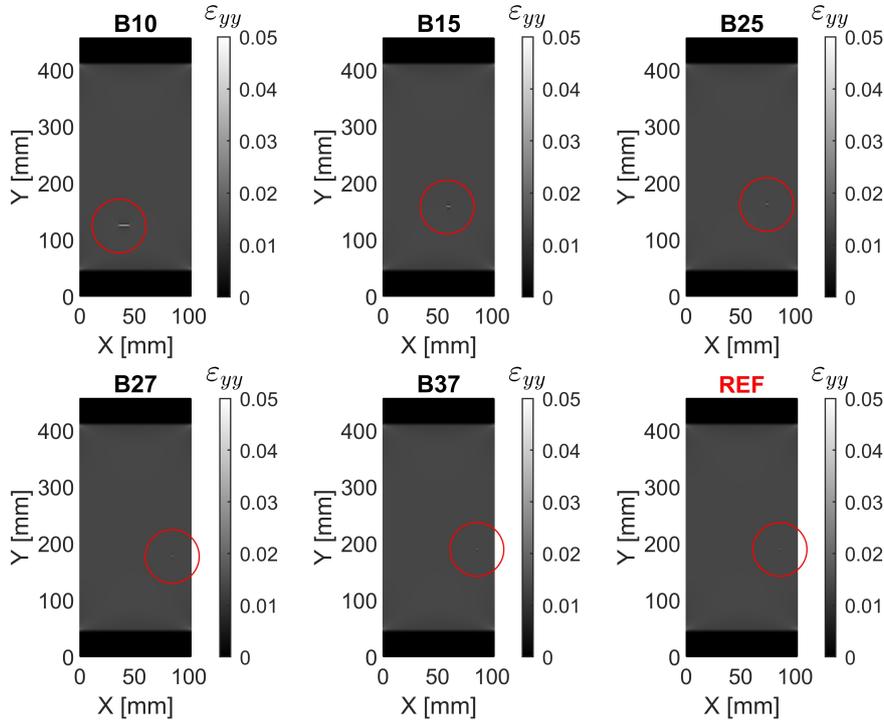
MFMES Strain field evolution ε_{yy} on the 3rd layer small cut

Figure 8.19: HF Small cut strain field evolution using MFMES acquisition function

With the MFMES acquisition function due to the extensive exploration phase the cut dimension is shorter than in the MFEI case in figure 8.18 and the reduction is noticeable from the first iterations. Also, the estimated cut position is close to the actual position of damage. The prediction of the cut position is correctly estimated at around budget 37 after an exploration phase between budget 25 and 37. At around budget 27 the position of the cut is really close to the effective position. This acquisition function shows that for a demanding case, such as with a small damage, using a more explorative approach as the MFMES can reduce the used budget.

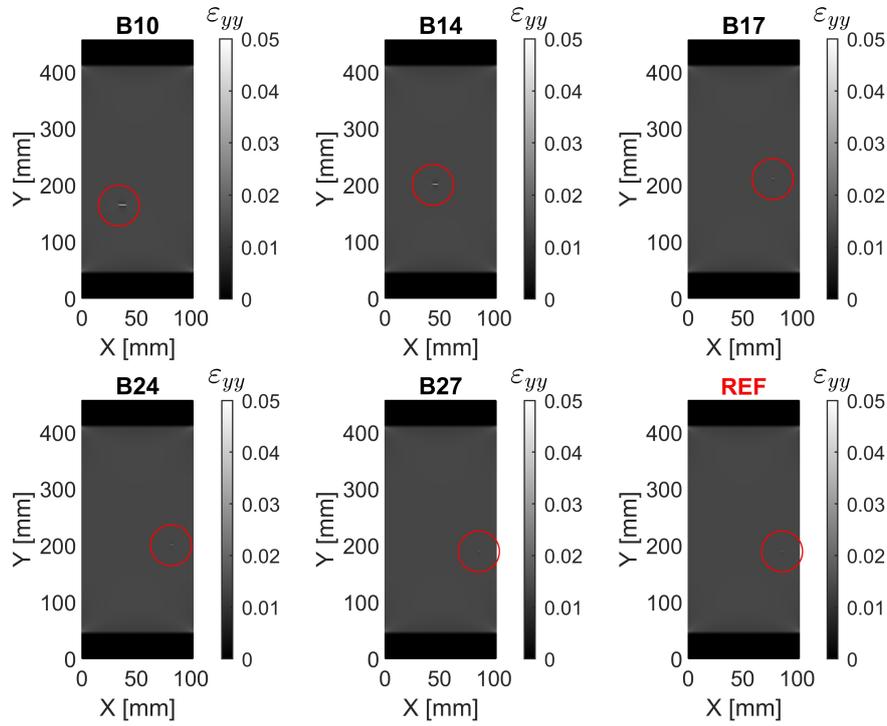
MFPI Strain field evolution ε_{yy} on the 3rd layer small cut

Figure 8.20: HF Small cut strain field evolution using MFPI acquisition function

Using the MFPI acquisition function within budget 17 the cut position and dimension are close to the effective damage characteristics. Leveraging the exploitation phase the code is able to approach the correct damage condition rapidly. Then between budget 17 and 27 the code continues to improve the accuracy on the estimation of the parameters that are correctly identified.

8.4 Large cut damage

In this section the results of the cut damage with large length are analyzed. The damage case considered is Case 25 in table 7.3 ($x_c = 11.86mm$, $y_c = 203.97mm$ and $l_c = 59.34mm$) considering the 10 tests evaluated.

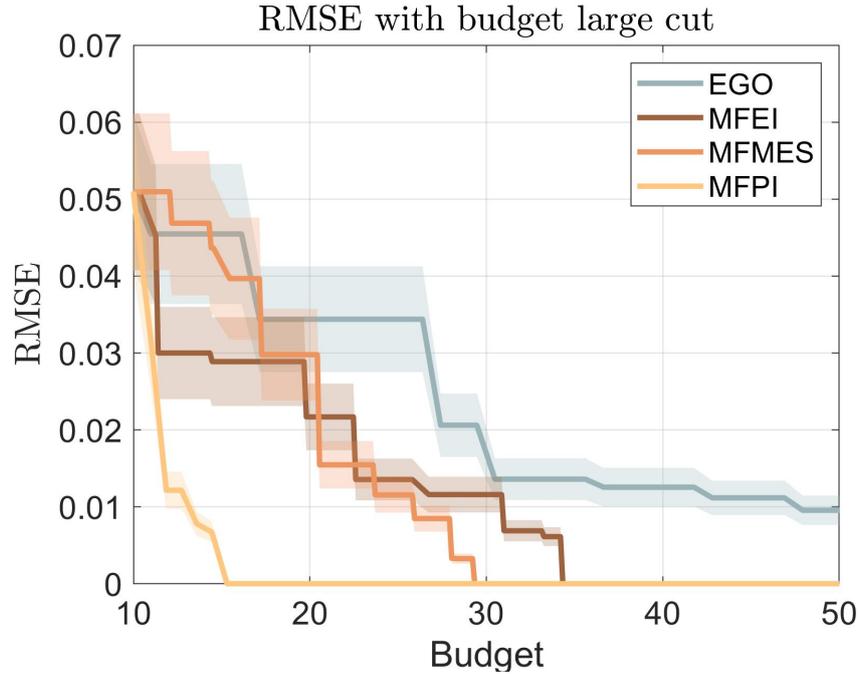
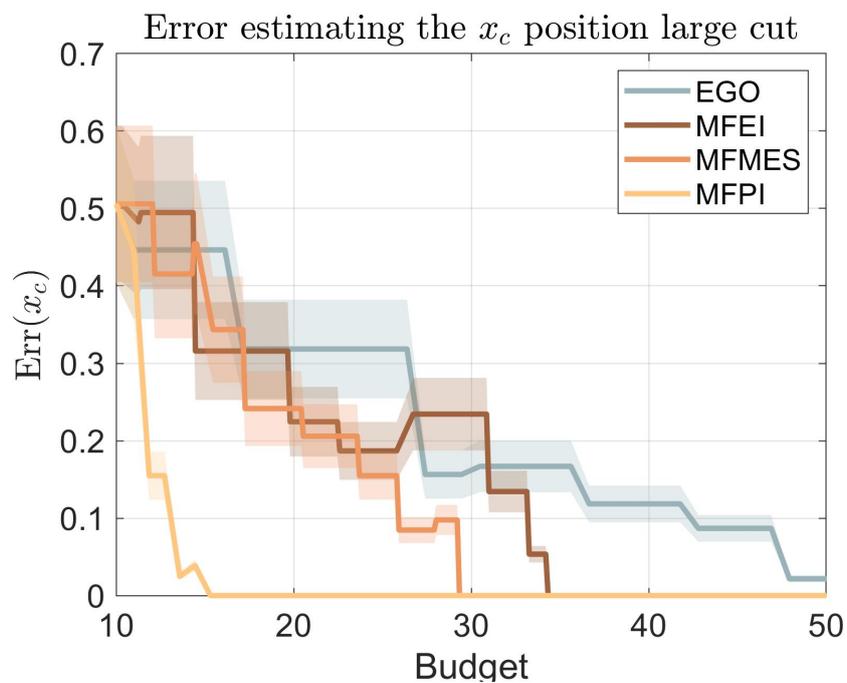


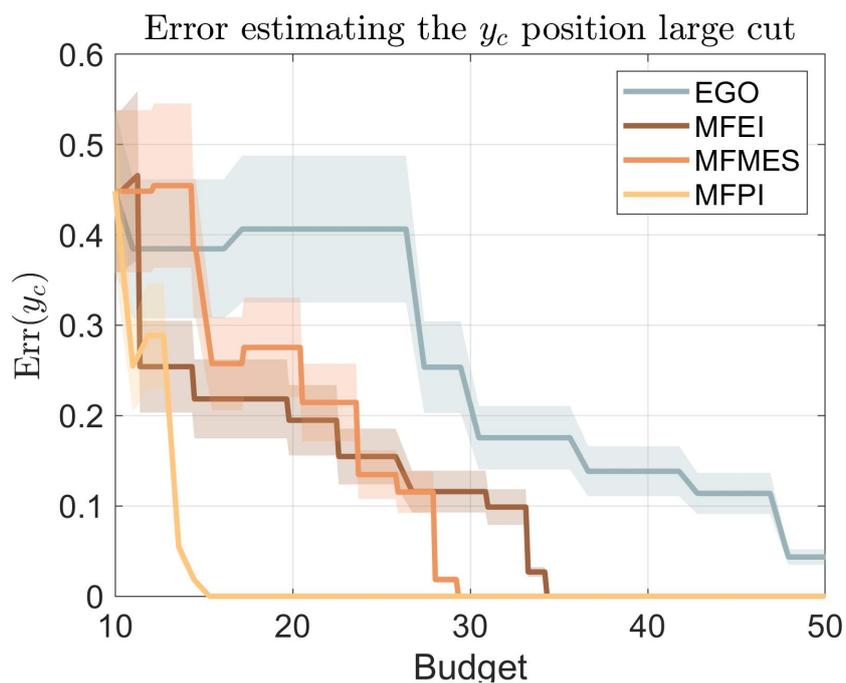
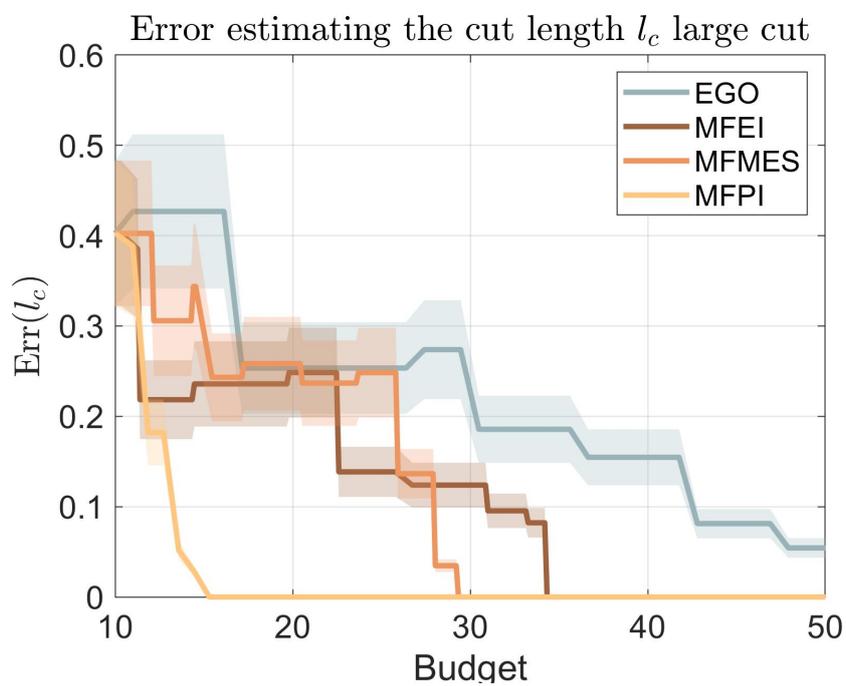
Figure 8.21: RMSE minimizing process results for each acquisition function large cut

The time to reach RMSE zero for a large cut is drastically reduced. The comparison between different acquisition functions is the same as the other cases analyzed. The singlefidelity EGO doesn't reach RMSE zero but its value is inferior to previous small and border cut conditions: this means that this condition is easier to spot and leads to a lower RMSE residual. Moreover, there is a great reduction in the RMSE in the initial phase meaning that is not necessary much exploration to reduce the overall solution time. This trend can be seen also in the MFEI which reduction comes mainly in the first phase while the MFMES tends to have more explorative zones with slight reduction in the RMSE. In particular the fastest method to reach convergence is the MFPI with a budget near 15, with a big RMSE reduction in the first phase. The times to reach convergence for each acquisition function can be reported:

	EGO	MFEI	MFMES	MFPI
Time to reach RMSE=0 [min]	-	173.04	147.87	77.14

Table 8.4: Mean evaluations times for each fidelity

Figure 8.22: Error estimating the x_c damage parameter large cut

Figure 8.23: Error estimating the y_c damage parameter large cutFigure 8.24: Error estimating the l_c damage parameter large cut

The trend for the errors are similar to the RMSE trends: the biggest reduction in the errors, especially for the cut length l_c , are in the first phase. This means that to catch this large cut, which strain field distribution deviate greatly from a "healthy" specimen, more details in chapter 6.8, is easier than the other peculiar short and border cut. The decrease of the error on estimating the various parameters is fast for the MFPI, showing that an exploitative approach doesn't waste much budget in exploration for a situation where is not needed. It can also be noted that the residual error for the EGO code is way lower than for the small cut case, meaning that the code is closer to reach convergence on finding the right damage parameters. Also, the confidence bound around the EGO (and also the other acquisition functions) errors is tight, meaning that there is less dispersion of values that indicate the capability of the framework to, in this condition, catch rapidly the correct damage parameters.

8.4.1 Strain field evolution

EI Strain field evolution ε_{yy} on the 3rd layer large cut

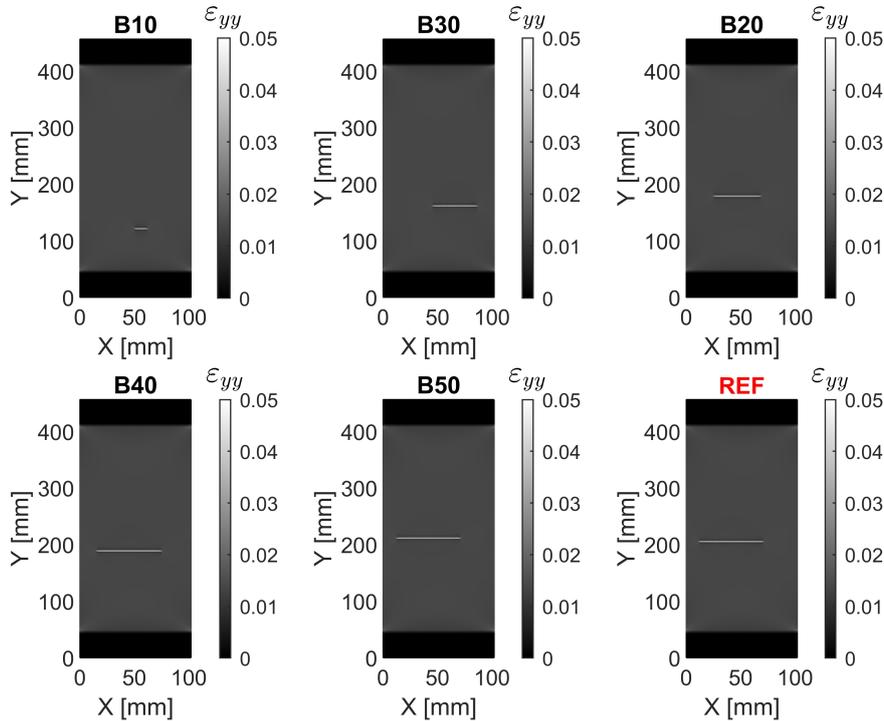


Figure 8.25: HF Large cut strain field evolution using EI acquisition function

The initial iterations from budget 10 to 30 immediately estimate a large cut on the specimen due to the easily catchable strain field. From budget 40 to 50 the extension and the position of the damage are fine adjusted. Comparing the final guess of the damage parameters between budget 50 and the reference can be seen a great similarity between the strain fields: this means that in this case with a large cut even a singlefidelity code is able to identify damage parameters that are close to the real damage condition.

MFEI Strain field evolution ε_{yy} on the 3^{rd} layer large cut

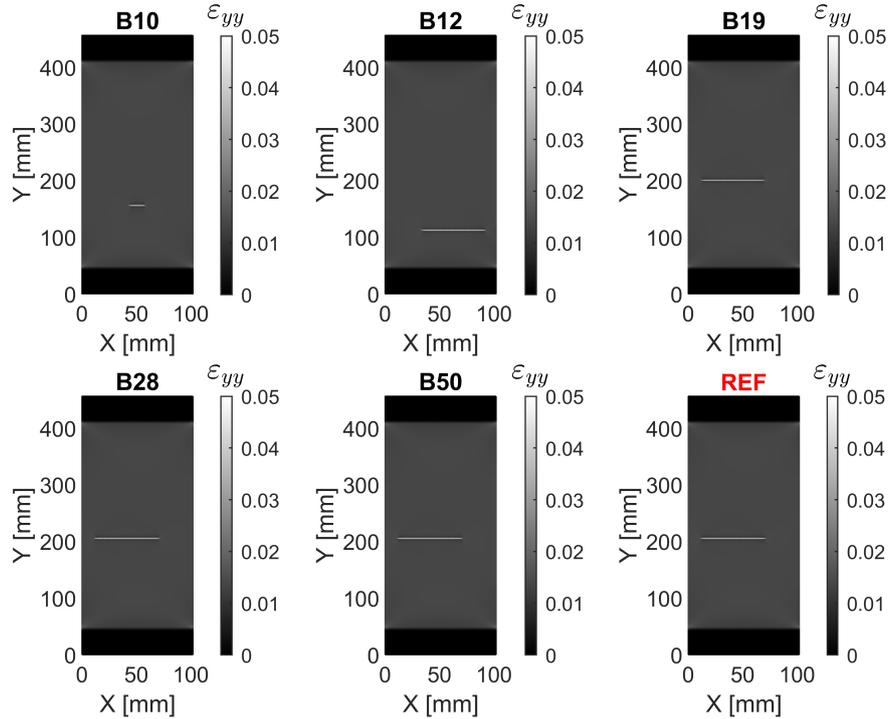


Figure 8.26: HF Large cut strain field evolution using MFEI acquisition function

The strain field evolution using the MFEI acquisition function reported in figure 8.26 immediately catches the right length magnitude of the real damage, due to the easier strain field pattern. Then the estimated position of the cut is adjusted with more iterations to reach the right position around budget 28.

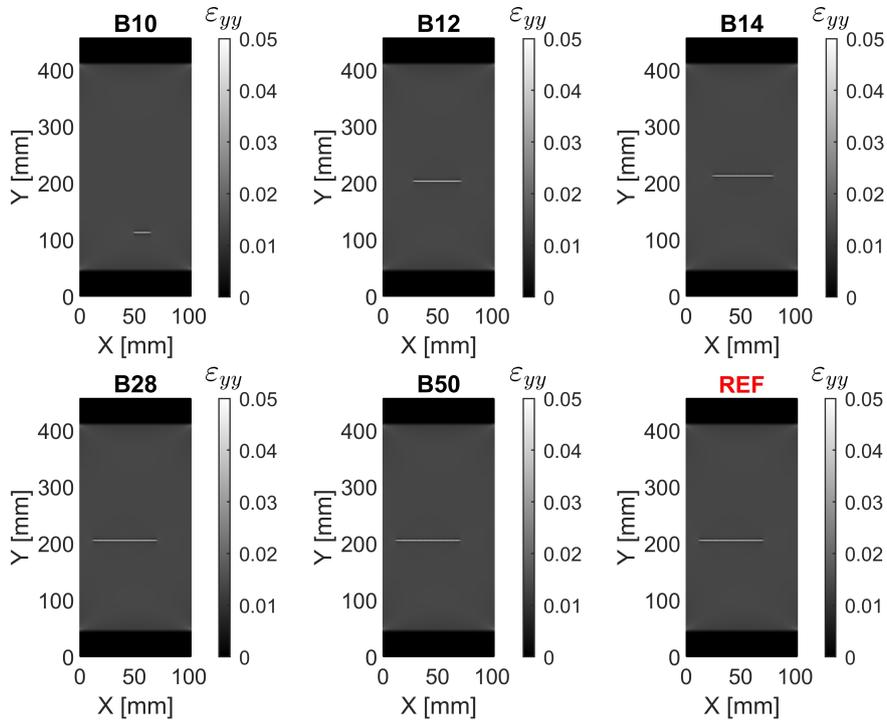
MFMES Strain field evolution ε_{yy} on the 3rd layer large cut

Figure 8.27: HF Large cut strain field evolution using MFMES acquisition function

The strain field evolution using the MFMES acquisition function reported in figure 8.27 within the first iterations correctly catches approximately the right strain field both with position and length. Then between budget 14 and 28 the further exploration brings to correctly identify the damage parameters.

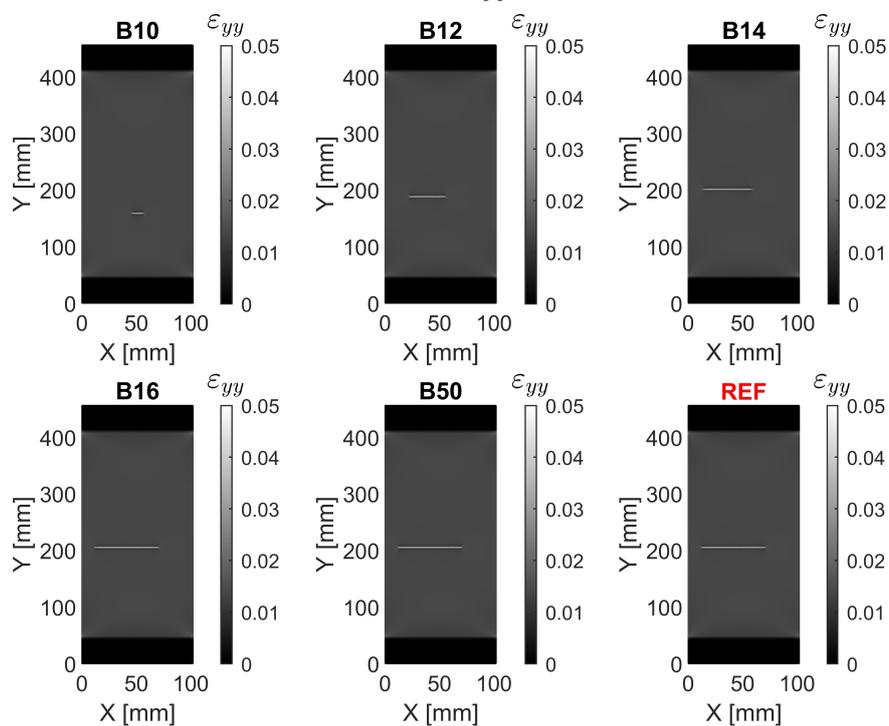
MFPI Strain field evolution ε_{yy} on the 3rd layer large cut

Figure 8.28: HF Large cut strain field evolution using MFPI acquisition function

The strain field evolution using the MFPI acquisition function is reported in figure 8.28. The cut length estimated is progressively increased through budget 16 where the correct position and extension are reached. The progression is faster than the other acquisition functions indicating the superior performance of the MFPI, especially in this case with a large cut where the strain field differs greatly from a "healthy" condition.

Chapter 9

Conclusions

The objective of this work of thesis is to investigate and implement a Bayesian surrogate multifidelity framework to speed up an optimization process. In particular the surrogate is built iteratively using different fidelities; the points to build the surrogate are selected by using acquisition functions that have the purpose of maximising the accuracy of the surrogate itself. The acquisition functions taken into account are the Multifidelity Expected Improvement MFEI, Multifidelity Probability of Improvement MFPI and Multifidelity Max value Entropy Search MFMES. These methods are applied to a damage identification problem on a composite specimen. Health monitoring problems are a fundamental aspect on assuring the integrity of structures in different engineering field such as aeronautical, civil, automotive and mechanical. The ability to develop a fast code that processes data from different sources to identify the damages is crucial to be able to repair the structure promptly.

In particular the problem addressed considers the modelization of a fibre cut damage in a carbon fibre composite specimen with fixed dimensions $102\text{ mm} \times 456\text{ mm}$ and stacking sequence $[45^\circ/0^\circ/0^\circ/45^\circ]$. The specimen is modeled with 3 different levels of fidelity: the low and mid use quad4 shell element with different mesh discretization and the high fidelity model employs 3D HEX8 elements. The models creation is automatized by using the Patran Command Language code. The parameters to be identified are the position and the extension of the cut considering a reference damage condition with an imposed fixed load. The quantity evaluated with Nastran models is the strain field in the direction of the load on the specimen, due to its noticeable interaction with the damaged cut region.

The Objective function to be minimized is the normalized root mean square error, calculated between the reference strain and the current code iteration. To find the right cut configuration the objective function needs to reach zero. This is verified for all the multifidelity acquisition functions, while the single fidelity Expected Improvement doesn't reach zero, hence the effective damage condition is not correctly

identified. This shows the power of using a multifidelity framework that greatly speeds up the optimization process finding the minimum of the objective function using a lower budget.

The codes consider 25 different damage conditions and run several different tests to build a suitable statistic. An initial surrogate is built using 20 low fidelity iterations, 10 mid fidelity iterations and 2 high fidelity iterations, then the optimization process starts and is guided by the surrogate information.

Considering the 3 different acquisition functions the performance differences are due to the different formulations. MFPI is the fastest code to reach convergence due to its main focus on exploitation, i.e. extensive use of high fidelity code to greatly improve the surrogate accuracy, that brings good results in a low dimension problem such as ours. MFEI is the slowest code to reach convergence between the three and that is due to its balance between exploitation and exploration that is not optimal for our application. FMES has performance between the other two acquisition functions due to its extensive exploration phase which in this case helps to find faster the optimum.

Future development of this work can be seen by enriching the surrogate using data from lab experiments by using a real damaged specimen, impose a constant displacement using a tensile testing machine and record the strain field using the Digital Image Correlation techniques.

Another interesting aspect can rely on increase the dimensionality of the problem to evaluate the change in performance of the code: for example, can be considered new variables the dimensions of the specimen, the number of layers, the stacking sequence, the layer where the cut is positioned and the type of damage.

Further future challenges can be seen on extending this approach to complex structures such a wing or a portion the airplane with the aim to apply the code on real application.

Acknowledgements

Desidero ringraziare la Dottoressa Laura Mainini per il continuo e costante supporto tecnico e morale, nonché la grande comprensione e incoraggiamento durante i momenti più difficili e di sconforto. Un grande ringraziamento va al Professor Marco Gherlone per la sua competenza e disponibilità dimostrate durante la stesura del lavoro. Inoltre desidero ringraziare sentitamente Francesco di Fiore per le sue preziose delucidazioni e per il tempo dedicatomi.

Infine ringrazio la mia famiglia che mi ha supportato durante questi anni, soprattutto durante i momenti più impegnativi e sfidanti, la vostra vicinanza mi ha permesso di raggiungere un traguardo che mi ero prefissato fin da bambino. Un ringraziamento speciale va a tutti gli amici e ai colleghi che mi hanno accompagnato durante il percorso con i quali ho affrontato le difficoltà e condiviso le gioie.

Bibliography

- [1] Andres S. Padron, Juan J. Alonso, and Michael S. Eldred. “Multi-fidelity Methods in Aerodynamic Robust Optimization”. In: *18th AIAA Non-Deterministic Approaches Conference*. 2016. DOI: 10.2514/6.2016-0680.
- [2] Michele Samorani. “The Wind Farm Layout Optimization Problem”. In: *Handbook of Wind Power Systems, Energy Systems* (Jan. 2010). DOI: 10.1007/978-3-642-41080-2_2.
- [3] Rami Debouk, Stéphane Lafortune, and Demosthenis Teneketzis. “On an Optimization Problem in Sensor Selection*”. In: *Discrete Event Dynamic Systems* 12.4 (Oct. 2002), pp. 417–445. ISSN: 1573-7594. DOI: 10.1023/A:1019770124060.
- [4] Peter S. Craig et al. “Pressure Matching for Hydrocarbon Reservoirs: A Case Study in the Use of Bayes Linear Strategies for Large Computer Experiments”. In: *Case Studies in Bayesian Statistics*. Ed. by Constantine Gatsonis et al. New York, NY: Springer New York, 1997, pp. 37–93.
- [5] Sarah E. Davis, Selen Cremaschi, and Mario R. Eden. “Efficient Surrogate Model Development: Optimum Model Form Based on Input Function Characteristics”. In: *27th European Symposium on Computer Aided Process Engineering*. Ed. by Antonio Espuña, Moisès Graells, and Luis Puigjaner. Vol. 40. Computer Aided Chemical Engineering. Elsevier, 2017, pp. 457–462. DOI: <https://doi.org/10.1016/B978-0-444-63965-3.50078-7>.
- [6] Forrester Alexander I J., Sóbester András, and Andy J. Keane. *Engineering Design via surrogate modelling: A practical guide*. Wiley, 2008.
- [7] Alexander I.J. Forrester and Andy J. Keane. “Recent advances in surrogate-based optimization”. In: *Progress in Aerospace Sciences* 45.1 (2009), pp. 50–79. DOI: <https://doi.org/10.1016/j.paerosci.2008.11.001>.
- [8] Alexander I.J Forrester, András Sóbester, and Andy J Keane. “Multi-fidelity optimization via surrogate modelling”. In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 463.2088 (2007), pp. 3251–3269. DOI: 10.1098/rspa.2007.1900.

- [9] Xinshuai Zhang et al. “Multi-fidelity deep neural network surrogate model for aerodynamic shape optimization”. In: *Computer Methods in Applied Mechanics and Engineering* 373 (2021), pp. 113–485. DOI: <https://doi.org/10.1016/j.cma.2020.113485>.
- [10] Likeng Huang, Zhenghong Gao, and Dehu Zhang. “Research on multi-fidelity aerodynamic optimization methods”. In: *Chinese Journal of Aeronautics* 26.2 (2013), pp. 279–286. ISSN: 1000-9361. DOI: <https://doi.org/10.1016/j.cja.2013.02.004>.
- [11] Y. Volkan Pehlivanoglu and Bedri Yagiz. “Aerodynamic design prediction using surrogate-based modeling in genetic algorithm architecture”. In: *Aerospace Science and Technology* 23.1 (2012). 35th ERF: Progress in Rotorcraft Research, pp. 479–491. ISSN: 1270-9638. DOI: <https://doi.org/10.1016/j.ast.2011.10.006>.
- [12] Anil Variyar, Thomas D. Economon, and Juan J. Alonso. “Multifidelity Conceptual Design and Optimization of Strut-Braced Wing Aircraft using Physics Based Methods”. In: *54th AIAA Aerospace Sciences Meeting*. DOI: 10.2514/6.2016-2000. eprint: <https://arc.aiaa.org/doi/pdf/10.2514/6.2016-2000>. URL: <https://arc.aiaa.org/doi/abs/10.2514/6.2016-2000>.
- [13] Francesco Di Fiore, Paolo Maggiore, and Laura Mainini. “Multifidelity domain-aware learning for the design of re-entry vehicles”. In: *Structural and Multidisciplinary Optimization* 64 (Oct. 2021), pp. 1–19. DOI: 10.1007/s00158-021-03037-4.
- [14] Zhifeng Gao et al. “Active fault tolerant control design approach for the flexible spacecraft with sensor faults”. In: *Journal of the Franklin Institute* 354.18 (2017), pp. 8038–8056. DOI: <https://doi.org/10.1016/j.jfranklin.2017.09.023>.
- [15] Keith Worden, Wieslaw Staszewski, and James Hensman. “Natural computing for mechanical systems research: A tutorial overview”. In: *Mechanical Systems and Signal Processing - MECH SYST SIGNAL PROCESS* 25 (Jan. 2011), pp. 4–111. DOI: 10.1016/j.ymsp.2010.07.013.
- [16] Gaurav Makkar et al. “A Machine Learning Framework for Physics-Based Multi-Fidelity Modeling and Health Monitoring for a Composite Wing”. In: *ASME International Mechanical Engineering Congress and Exposition*. Vol. 86625. American Society of Mechanical Engineers. 2022, V001T01A008.
- [17] D. Allaire et al. “Multifidelity DDDAS Methods with Application to a Self-aware Aerospace Vehicle”. In: *Procedia Computer Science* 29 (2014). 2014 International Conference on Computational Science, pp. 1182–1192. ISSN: 1877-0509. DOI: <https://doi.org/10.1016/j.procs.2014.05.106>.

- [18] Alfredo Güemes et al. “Structural Health Monitoring for Advanced Composite Structures: A Review”. In: *Journal of Composites Science* 4.1 (2020). ISSN: 2504-477X. DOI: 10.3390/jcs4010013. URL: <https://www.mdpi.com/2504-477X/4/1/13>.
- [19] Jonas Mockus. “Application of Bayesian approach to numerical methods of global and stochastic optimization”. In: *Journal of Global Optimization* 4 (1994), pp. 347–365.
- [20] Jialin Song, Chen Yuxin, and Yisong Yue. “A General Framework for Multifidelity Bayesian Optimization with Gaussian Processes”. In: (Nov. 2018).
- [21] Maziar Raissi and Padmanabhan Seshaiyer. *A multi-fidelity stochastic collocation method using locally improved reduced-order models*. 2013. DOI: 10.48550/ARXIV.1306.0132.
- [22] Laura Mainini and Karen Willcox. “Surrogate Modeling Approach to Support Real-Time Structural Assessment and Decision Making”. In: vol. 53. Jan. 2014. DOI: 10.2514/6.2014-1488.
- [23] D. Huang et al. “Sequential kriging optimization using multiple-fidelity evaluations”. In: *Structural and Multidisciplinary Optimization* 32 (Nov. 2006), pp. 369–382. DOI: 10.1007/s00158-005-0587-0.
- [24] Loic Le Gratiet and Claire Cannamela. *Kriging-based sequential design strategies using fast cross-validation techniques with extensions to multi-fidelity computer codes*. 2012. DOI: 10.48550/ARXIV.1210.6187.
- [25] Loic Le Gratiet. *Bayesian analysis of hierarchical multi-fidelity codes*. 2011. DOI: 10.48550/ARXIV.1112.5389.
- [26] Peter Benner, Serkan Gugercin, and Karen Willcox. “A Survey of Projection-Based Model Reduction Methods for Parametric Dynamical Systems”. In: *SIAM Review* 57.4 (2015), pp. 483–531. DOI: 10.1137/130932715.
- [27] Benjamin Peherstorfer, Karen Willcox, and Max Gunzburger. “Survey of Multifidelity Methods in Uncertainty Propagation, Inference, and Optimization”. In: *SIAM Review* 60 (Jan. 2018), pp. 550–591. DOI: 10.1137/16M1082469.
- [28] Jonathan Walsh, Lucy Berthoud, and Christian Allen. “Drag reduction through shape optimisation for satellites in Very Low Earth Orbit”. In: *Acta Astronautica* 179 (2021), pp. 105–121. DOI: <https://doi.org/10.1016/j.actaastro.2020.09.018>.
- [29] Box, J.S. Hunter, and W.G. Hunter. *Statistics for Experimenters: Design, Innovation, and Discovery*. Wiley Series in Probability and Statistics. Wiley, 2005.

-
- [30] Jacek Pietraszek, Norbert Radek, and Andrii V. Goroshko. “Challenges for the DOE methodology related to the introduction of Industry 4.0”. In: *Production Engineering Archives* 26.4 (2020), pp. 190–194. DOI: doi:10.30657/pea.2020.26.33.
- [31] D.C. Montgomery. *Design and Analysis of Experiments, 8th Edition*. John Wiley Sons, Incorporated, 2012.
- [32] Angelos Markopoulos et al. “Modelling and Optimization of Machining with the Use of Statistical Methods and Soft Computing”. In: Jan. 2016, pp. 39–88. DOI: 10.1007/978-3-319-23838-8_2.
- [33] NIST gov. *NIST/SEMATECH e-Handbook of Statistical Methods*. 2012. URL: <https://www.itl.nist.gov/div898/handbook/> (visited on 10/22/2022).
- [34] M. D. McKay, R. J. Beckman, and W. J. Conover. “A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code”. In: *Technometrics* 21.2 (1979), pp. 239–245. (Visited on 09/27/2022).
- [35] Jennifer Bone et al. “Hierarchical Machine Learning for High-Fidelity 3D Printed Biopolymers”. In: *ACS Biomaterials Science Engineering* 6 (Nov. 2020). DOI: 10.1021/acsbiomaterials.0c00755.
- [36] Nestor Queipo et al. “Surrogate-based analysis and optimization”. In: *Progress in Aerospace Sciences* 41 (Jan. 2005), pp. 1–28. DOI: 10.1016/j.paerosci.2005.02.001.
- [37] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Series in Statistics. Springer New York, 2013.
- [38] Donald Jones, Matthias Schonlau, and William Welch. “Efficient Global Optimization of Expensive Black-Box Functions”. In: *Journal of Global Optimization* 13 (Dec. 1998), pp. 455–492. DOI: 10.1023/A:1008306431147.
- [39] M. Lifshits. *Lectures on Gaussian Processes*. Jan. 2012. DOI: 10.1007/978-3-642-24939-6.
- [40] Tom Lefebvre, Frederik De Belie, and Guillaume Crevecoeur. *A Radial Basis Function based Optimization Algorithm with Regular Simplex set geometry in Ellipsoidal Trust-Regions*. 2018. DOI: 10.48550/ARXIV.1805.11830.
- [41] Donald Jones. “A Taxonomy of Global Optimization Methods Based on Response Surfaces”. In: *J. of Global Optimization* 21 (Dec. 2001), pp. 345–383. DOI: 10.1023/A:1012771025575.
- [42] D.G. Krige. “A statistical approach to some basic mine valuation problems on the Witwatersrand”. In: *Journal of the Southern African Institute of Mining and Metallurgy* 52.6 (1951), pp. 119–139. DOI: 10.10520/AJA0038223X/4792.

-
- [43] Donald E. Myers. “Matrix formulation of co-kriging”. In: *Journal of the International Association for Mathematical Geology* 14.3 (June 1982), pp. 249–257. DOI: 10.1007/BF01032887.
- [44] Akil Narayan, Claude Gittelsohn, and Dongbin Xiu. “A Stochastic Collocation Algorithm with Multifidelity Models”. In: 36 (2014), A495–A521. DOI: 10.1137/130929461.
- [45] Mainini L. La Mantia P. Gherlone M. “Modellazione surrogata multi-fedeltà per piastre danneggiate in materiale composito”. MA thesis. Politecnico di Torino, 2017-2018.
- [46] Mainini L. Ermacora M. Gherlone M. “Multifidelity modelling of damaged aerospace structures: A transfer learning based approach”. MA thesis. Politecnico di Torino, 2021-2022.
- [47] Natalia Alexandrov et al. “A Trust Region Framework for managing the use of approximation models in optimization”. In: *Structural Optimization* 15 (Feb. 1998), pp. 16–23. DOI: 10.1007/BF01197433.
- [48] Natalia Alexandrov et al. “A Trust Region Framework for managing the use of approximation models in optimization”. In: *Structural Optimization* 15 (Feb. 1998), pp. 16–23. DOI: 10.1007/BF01197433.
- [49] Andrew March and Karen Willcox. “Constrained multifidelity optimization using model calibration”. In: *Structural and Multidisciplinary Optimization* 46 (July 2012). DOI: 10.1007/s00158-011-0749-1.
- [50] Stefan Wild and Christine Shoemaker. “Global Convergence of Radial Basis Function Trust Region Derivative-Free Algorithms”. In: *SIAM Journal on Optimization* 21 (July 2011), pp. 761–781. DOI: 10.1137/09074927X.
- [51] Christopher Fischer and Ramana Grandhi. “Multi-Fidelity Design Optimization via Low-Fidelity Correction Technique”. In: June 2016. DOI: 10.2514/6.2016-4293.
- [52] Naks Alex et al. “Optimization with Variable-Fidelity Models Applied to Wing Design”. In: 2000 (Mar. 2000). DOI: 10.2514/6.2000-841.
- [53] Andrew March and Karen Willcox. “Constrained multifidelity optimization using model calibration”. In: *Structural and Multidisciplinary Optimization* 46 (July 2012). DOI: 10.1007/s00158-011-0749-1.
- [54] Jay Ver Hoef and Noel Cressie. “Multivariable Spatial Prediction”. In: *Mathematical Geology* 25 (Feb. 1993), pp. 219–240. DOI: 10.1007/BF00893273.
- [55] Stephen J. Leary, Atul Bhaskar, and Andrew J. Keane. “Global Approximation and Optimization Using Adjoint Computational Fluid Dynamics Codes”. In: *AIAA Journal* 42.3 (2004), pp. 631–641. DOI: 10.2514/1.9114.

-
- [56] Marc Kennedy and A O’Hagan. “Predicting the Output from a Complex Computer Code When Fast Approximations Are Available”. In: *Biometrika* 87 (Oct. 1998). DOI: 10.1093/biomet/87.1.1.
- [57] Eric Brochu, Vlad M. Cora, and Nando de Freitas. “A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning”. In: *CoRR* abs/1012.2599 (2010).
- [58] Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*. Vol. 2. 3. MIT press Cambridge, MA, 2006.
- [59] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, Nov. 2005. ISBN: 9780262256834. DOI: 10.7551/mitpress/3206.001.0001.
- [60] Bertil Matérn. *Spatial Variation*. Springer New York, 1986. DOI: 10.1007/978-1-4615-7892-5.
- [61] Michael L. Stein. *Interpolation of Spatial Data*. Springer New York, 1999. DOI: 10.1007/978-1-4612-1494-6.
- [62] José Hernández-Lobato, Matthew Hoffman, and Zoubin Ghahramani. “Predictive Entropy Search for Efficient Global Optimization of Black-box Functions”. In: *Advances in Neural Information Processing Systems* 1 (June 2014).
- [63] Zi Wang and Stefanie Jegelka. “Max-value Entropy Search for Efficient Bayesian Optimization”. In: *Proceedings of the 34th International Conference on Machine Learning*. Ed. by Doina Precup and Yee Whye Teh. Vol. 70. Proceedings of Machine Learning Research. PMLR, June 2017, pp. 3627–3635.
- [64] SEWALL WRIGHT. “THE GENETICAL THEORY OF NATURAL SELECTION: A Review”. In: *Journal of Heredity* 21.8 (Aug. 1930), pp. 349–356. DOI: 10.1093/oxfordjournals.jhered.a103361.
- [65] Jialin Song, Yuxin Chen, and Yisong Yue. *A General Framework for Multi-fidelity Bayesian Optimization with Gaussian Processes*. 2018.
- [66] Shion Takeno et al. “Multi-fidelity Bayesian Optimization with Max-value Entropy Search”. In: *ICML*. 2020.
- [67] Xiongfeng Ruan et al. “Variable-fidelity probability of improvement method for efficient global optimization of expensive black-box problems”. In: *Structural and Multidisciplinary Optimization* 62 (2020), pp. 1–32.
- [68] Deng Huang et al. “Sequential kriging optimization using multiple-fidelity evaluations”. In: *Structural and Multidisciplinary Optimization* 32 (2006), pp. 369–382.

- [69] Yehong Zhang et al. “Information-Based Multi-Fidelity Bayesian Optimization”. In: 2017.
- [70] Joseph V. Michalowicz, Jonathan M. Nichols, and Frank Bucholtz. “Handbook of Differential Entropy”. In: 2013.
- [71] I.M. Daniel and A. Charewicz. “Fatigue damage mechanisms and residual properties of graphite/epoxy laminates”. In: *Engineering Fracture Mechanics* 25.5 (1986), pp. 793–808. ISSN: 0013-7944. DOI: [https://doi.org/10.1016/0013-7944\(86\)90042-1](https://doi.org/10.1016/0013-7944(86)90042-1).
- [72] Julián Sierra-Pérez, Alfredo Güemes, and Luis E Mujica. “Damage detection by using FBGs and strain field pattern recognition techniques”. In: *Smart Materials and Structures* 22.2 (Dec. 2012), p. 025011. DOI: [10.1088/0964-1726/22/2/025011](https://doi.org/10.1088/0964-1726/22/2/025011).
- [73] Wei Zhang et al. “The Strain Field Method for Structural Damage Identification Using Brillouin Optical Fiber Sensing”. In: *Smart Materials and Structures* 16 (May 2007), p. 843. DOI: [10.1088/0964-1726/16/3/034](https://doi.org/10.1088/0964-1726/16/3/034).
- [74] Miguel Caminero et al. “Damage Assessment of Composite Structures Using Digital Image Correlation”. In: *Applied Composite Materials* 21 (Jan. 2014). DOI: [10.1007/s10443-013-9352-5](https://doi.org/10.1007/s10443-013-9352-5).
- [75] H. Man and G. Prusty. “Neural network modelling for damage behaviour of composites using full-field strain measurements”. In: *Composite Structures* 93.2 (2011), pp. 383–391. ISSN: 0263-8223. DOI: <https://doi.org/10.1016/j.compstruct.2010.09.003>.
- [76] Kristin Marlett. *Hexcel 8552S AS4 Plain Weave Fabric Prepreg 193 gsm 38% RC Qualification Material Property Data Report*. Nazional institute for aviation research Wichita State University, 2011.
- [77] J.N. Reddy. *Mechanics of Laminated Composite Plates and Shells: Theory and Analysis, Second Edition*. CRC Press, 2003. ISBN: 9780203502808.
- [78] *Standard Test Method for Tensile Properties of Polymer Matrix Composite Materials*. D3039/D3039M-17. ASTM standard. Dec. 2017.
- [79] Pier Berri, Matteo Dalla Vedova, and Laura Mainini. “Real-time Fault Detection and Prognostics for Aircraft Actuation Systems”. In: Jan. 2019. DOI: [10.2514/6.2019-2210](https://doi.org/10.2514/6.2019-2210).