

POLITECNICO DI TORINO

Corso di Laurea in Ingegneria Matematica

Tesi di Laurea Magistrale

**Parameter identification on a
Carbon-Fiber Reinforced Polymer
using LS-OPT**

Optimization and Metamodeling Strategies



Relatore

prof. Roberto Fontana

Correlatore:

prof. Giorgio Chiandussi

Marco PASSARELLO

matricola: 240280

Supervisione aziendale

Nova Analysis

ing. Simone Ferrero

ing. Caterina Tribuzi

ANNO ACCADEMICO 2017 – 2018

Summary

Parameter identification fa riferimento ad un insieme di tecniche utilizzate per risolvere problemi inversi non lineari in cui la risposta di un sistema dipende da un certo numero di parametri incogniti. Fra le possibili applicazioni, la *parameter identification* può essere utilizzata per individuare le proprietà meccaniche dei materiali, basandosi sul loro comportamento sperimentale. L'idea alla base è di cercare i parametri del modello tali per cui il comportamento simulato del materiale corrisponda al comportamento reale. La ricerca dei parametri ottimali è svolta tramite l'utilizzo combinato di diverse tecniche, quali *metamodeling*, design degli esperimenti, metriche di *curve matching* e algoritmi di ottimizzazione, implementati nel software commerciale LS-OPT.

I dati sperimentali, necessari per l'ottimizzazione, sono stati ottenuti da test di trazione eseguiti su provini forniti dall'azienda CarboSix. I provini sono stati ricavati da lastre in resina epossidica rinforzata con fibre di carbonio e sono stati testati con diversi orientamenti delle fibre (0° , 45° , 90°) e diversi spessori.

In una prima fase del lavoro, le varie tecniche di *metamodeling*, *sampling*, *curve matching* e ottimizzazione sono state approfondite e uno studio su un problema semplificato è stato svolto con lo scopo di capire accuratezza e costo computazionale delle varie opzioni implementate in LS-OPT.

L'ottimizzazione vera e propria dei parametri del materiale è stata eseguita nella seconda parte del lavoro. Due modelli di materiale, identificati come MAT54 e MAT58, sono stati considerati. Tutte le simulazioni sono state svolte utilizzando il software agli elementi finiti LS-DYNA, mentre la procedura di ottimizzazione è stata svolta in LS-OPT. È stato osservato che il MAT54 riesce a rappresentare bene il comportamento del materiale testato solamente quando il carico è applicato nella stessa direzione delle fibre. Il MAT58 riesce invece a riprodurre adeguatamente il comportamento osservato sia per il caso di trazione a 0° che per quello a 45° . Per entrambi i materiali, la modellazione dei singoli layer di fibre di carbonio porta a risultati migliori rispetto alla modellazione con un solo layer. Infine, entrambi i modelli sembrano avere il potenziale per riprodurre il comportamento del materiale a rottura, anche se degli studi più approfonditi sono necessari per capire quali parametri relativi alla rottura utilizzare ed ottimizzare.

Acknowledgements

I would like to thank Simone and Caterina from Nova Analysis, whose help has been essential throughout the entire work, and CarboSix who has kindly provided the specimens for the experimental tests.

Special thanks go to my parents, my brothers and Kristjane for supporting and motivating me during all these years.

Contents

1	Introduction	1
2	Parameter Identification	3
2.1	Introduction	3
2.2	Parameter identification in LS-OPT	4
2.2.1	Metamodel-based Optimization	4
2.2.2	Direct Optimization	7
3	Metamodeling and Sampling Techniques	9
3.1	Metamodels	9
3.1.1	Polynomial response surfaces	10
3.1.2	Feedforward neural networks	11
3.1.3	Radial Basis Function networks	12
3.1.4	Kriging	14
3.1.5	Support Vector Regression	14
3.2	Design of Experiments	16
3.2.1	Full factorial designs	16
3.2.2	Koshal designs	17
3.2.3	Central composite designs	17
3.2.4	D-optimal designs	18
3.2.5	Space filling designs	19
3.3	Sensitivity analysis	20
3.4	Which method to choose?	21
4	Optimization	23
4.1	The objective function	23
4.1.1	Targeted composites	23
4.1.2	Curve matching composites	24
4.1.3	Multi-objective optimization	28
4.2	Constraints	28
4.2.1	Sampling constraints	28
4.3	Optimization algorithms	29
4.3.1	Leap-Frog Optimization	29
4.3.2	Genetic Algorithms	30
4.3.3	Adaptive Simulated Annealing	30

4.3.4	Differential Evolution	31
4.3.5	Hybrid algorithms	31
5	Experimental tests	33
5.1	Introduction	33
5.2	Materials	33
5.2.1	Composites	33
5.2.2	Tested material	34
5.3	Tensile tests	35
5.3.1	Specimens	35
5.3.2	Setup	36
5.3.3	Results	38
6	Numerical model	45
6.1	Introduction	45
6.2	General settings	45
6.2.1	Geometry and mesh	45
6.2.2	Boundary conditions	46
6.2.3	Numerical method	46
6.3	Material models	47
6.3.1	Material directions	47
6.3.2	Enhanced composite damage (MAT54)	48
6.3.3	Laminated composite fabric (MAT58)	50
6.3.4	Modeling of laminated composites	52
7	Parameter optimization	55
7.1	Preliminary study	55
7.1.1	Setup	55
7.1.2	Results	56
7.1.3	Conclusion	59
7.2	Optimization of MAT54	60
7.2.1	Single layer model	60
7.2.2	Multiple layer model	63
7.3	Optimization of MAT58	65
7.3.1	Two-steps optimization	67
7.3.2	One-step optimization	68
8	Conclusion	73
A	Failed specimens	75
B	LS-DYNA input files	79
	Bibliography	85

Chapter 1

Introduction

Parameter identification problems are particular types of non-linear inverse problems, in which certain parameters of a system have to be determined based on its response. The system is not completely unknown, in the sense that its equations are (assumed to be) known but they depend on unknown parameters, which are the objective of parameter identification. In this thesis, a specific approach for solving parameter identification problems is presented, which involves many different techniques related to metamodeling, design of experiments and optimization. This approach has then been used to characterize mechanical properties of a composite material.

In order to perform parameter identification, two softwares have been used: the actual optimization procedure was carried out using LS-OPT, while all simulations of the mechanical system were performed using the finite element solver LS-DYNA.

This thesis focuses on two main objectives:

1. to test the optimization, metamodeling and sampling techniques that are implemented in LS-OPT, so as to understand which technique is the most appropriate for the particular parameter identification problem that has been addressed;
2. to characterize the mechanical behaviour of the tested composite material using parameter identification techniques in LS-OPT.

The entire work has been carried on with the supervision of the company Nova Analysis and the composite material has been provided by the company CarboSix. Finally, experimental tests were performed at Politecnico di Torino mechanical laboratories.

The work is organized as follows. In Chapter 2 the basic idea behind parameter identification is presented and different approaches to solve the problem in LS-OPT are explained. Chapters 3 and 4 give the theoretical notions necessary to fully understand all steps in a parameter identification procedure. In particular, Chapter 3 concerns metamodeling and sampling techniques, whereas Chapter 4 concerns optimization algorithms and related topics, such as the choice of the objective function and constraints. In Chapter 5 the experimental tests performed on the available specimens are described in detail and experimental results are shown. The numerical model used for the simulations in LS-DYNA, including modeling of composite materials, is described in Chapter 6. Finally, in Chapter 7 all the optimization runs that have been performed are described and optimization results of the studied material models are presented. Chapter 8 summarizes all the obtained results.

For completeness, additional photos concerning the experimental tests are shown in Appendix A, while some relevant parts of the LS-DYNA input files used for simulations are given in Appendix B.

Chapter 2

Parameter Identification

2.1 Introduction

Complex materials are more often employed in many engineering fields, due to the advancement of material technologies. Such new materials are in many cases composite materials, i.e. they are created by combining two or more simple materials, but they exhibit none of the characteristics of the single constituents. Since in most cases the properties of such complex materials cannot be deduced from the properties of their constituents, mathematical modeling of composites is not an easy task. Even when there is reason to believe that a constitutive model could be more appropriate than another, there is the problem of appropriately choosing the many parameters which describe the model. A relatively recent technique, called parameter identification, has been developed in order to overcome this problem.

Parameter identification, often called parameter estimation or system identification, refers to a class of analytical and numerical techniques used to solve non-linear inverse problems in which the output of the system usually depends on a set of unknown parameters. The general idea can be applied to a variety of problems and fields, such as aerospace engineering [7], manufacturing process of metal parts [13] and crash analysis [20]. However, in most cases, parameter identification has been used to identify thermo-mechanical properties of materials. For example, some parameter estimation studies have been carried out on concrete, steel, foam and fabric materials [4][6][23]. Also, many studies involved the identification of plastics, carbon-fiber reinforced polymers and laminated composite materials [2][3][5][14][22].

Parameter estimation is based on a simple but effective idea: the unknown parameters of the material are optimized so that the output of simulations is as close as possible to experimental observations on the same material. The optimization process could be performed on the actual model which describes the material, but this would be computationally expensive. Instead, a metamodel (i.e. a model of the model) is built based on a reduced number of simulations and the optimization is then performed on such metamodel (see Figure 2.1). In this way, optimal parameters can be found with high accuracy and efficiency.

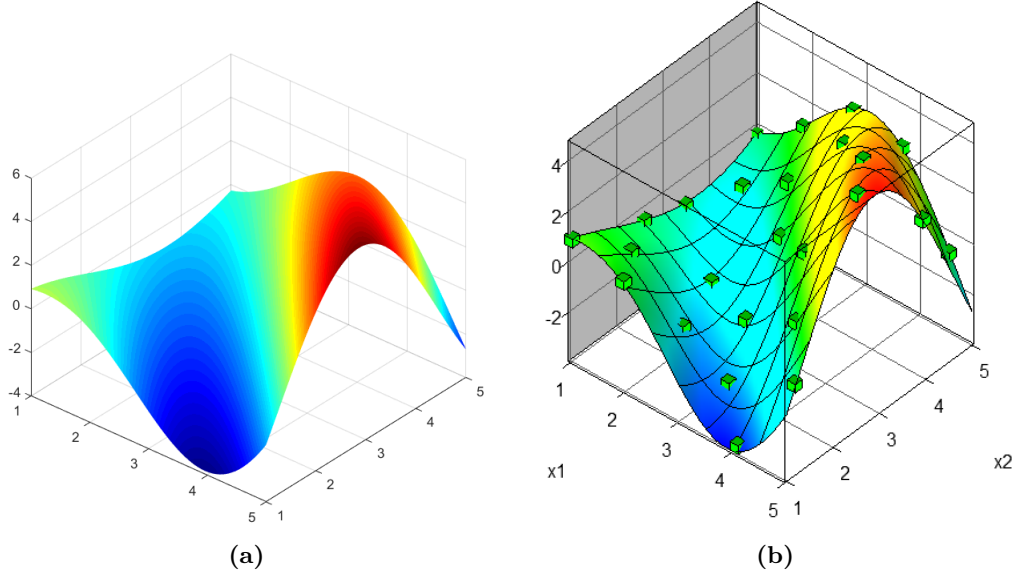


Figure 2.1: Comparison between a surface representing the model and the metamodel built from sampling the design space.

2.2 Parameter identification in LS-OPT

2.2.1 Metamodel-based Optimization

In this study parameter identification has been performed using LS-OPT [9], which is a software produced by Livermore Software Technology Corporation (LSTC), together with the finite element solver LS-DYNA and other minor softwares. It has been designed to solve general optimization problems and its capabilities range from basic optimization to probabilistic and robustness analyses and topology optimization problems. In particular, one important application of LS-OPT is parameter identification. In order to understand how the software solves the problem, it is useful to have a look at its graphical user interface.

Figure 2.2 shows the general approach for solving a parameter estimation problem in LS-OPT using metamodels. Apart from the Setup and Verification dialogs, the important steps are the following ones:

1. Sampling (or Design of Experiments, see Section 3.2): in this dialog points that have to be analysed are chosen from all possible combinations of parameters. These points will be used to build the metamodel. It is possible to add more Sampling dialogs, but each one of them must always have at least one Stage dialog;
2. Stage: the Stage dialog allows LS-OPT to communicate with an external solver in order to perform numerical simulations and gather output data from them. LS-OPT has a built-in interface for simulating with LS-DYNA, but it is possible to interface the software with any external solver. More stages can be present in the same project if the analysis requires to simulate different systems or multiple load cases;

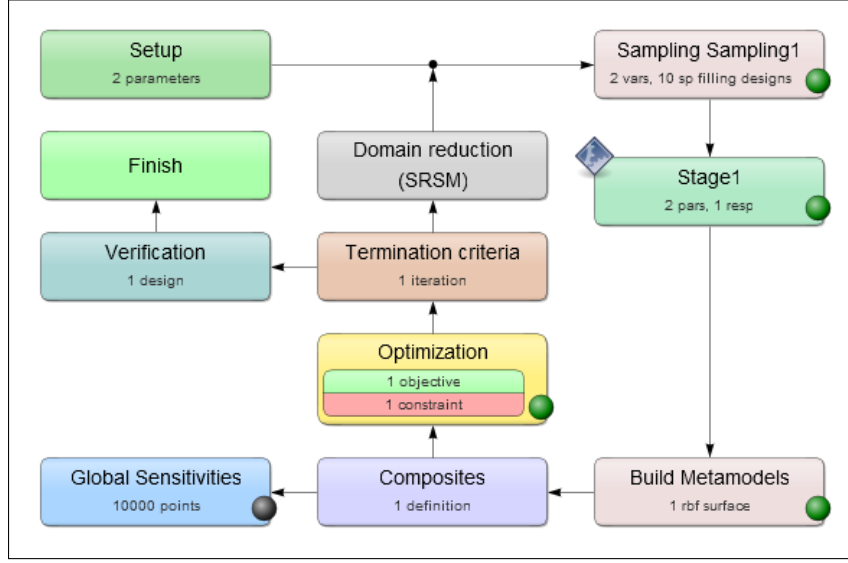


Figure 2.2: Flowchart of a metamodel-based parameter estimation problem from the LS-OPT graphical user interface.

3. Build Metamodels (see Section 3.1): after all simulations have been performed, LS-OPT uses their outputs (responses or histories) to build a metamodel on which the optimization will later be performed;
4. Composites: in most cases it is necessary to combine parameters or responses in order to define other dependant quantities, called composite expressions or simply composites. This is always the case in parameter identification problems, because of the particular form of the function to be minimized (see Section 4.1);
5. Global Sensitivities (see Section 3.3): this dialog is optional and it allows to compute sensitivities of the parameters based on the metamodel;
6. Optimization: after having defined the objective function and eventual constraints, the actual optimization can be performed in order to compute the optimal value of each parameter. Several optimization algorithms are implemented in the software (see Section 4.3).

Task and strategies

Steps 1 to 6, as described above, establish the basic run of a parameter estimation problem. However, some automatic procedures have been implemented to increase the accuracy of the final result. In order to explain this procedures it is convenient to introduce tasks.

As stated above, LS-OPT is capable of solving many different problems. The type of problems that has to be analysed is chosen by specifying a certain task. Tasks are divided into two main categories. The first one includes metamodel-based tasks, in which a metamodel is constructed based on some simulation runs. On such metamodel it is possible to perform simple design exploration, optimization, robustness analysis, reliability-based

optimization and Monte Carlo analysis. The second category includes direct simulation tasks, such as direct optimization and direct Monte Carlo analysis.

Ideally, parameter identification could be solved using direct optimization, but it is clear that this approach would be too much expensive since it requires to run hundreds of simulations. For this reason, the obvious choice is a Metamodel-based Optimization Task. For this particular task, three additional strategies can be selected:

1. Single Iteration: this strategy consists of performing steps 1 to 6 one single time, as described in the previous section. It is generally used to build a global metamodel in order to perform design exploration;
2. Sequential: by selecting this strategy, the basic procedure is repeated a certain amount of times (specified in the Termination Criteria dialog, see Figure 2.2) and at each iteration the metamodel is rebuilt or updated with new points. This strategy can be used for design exploration as well as for parameter identification if the user is interested in finding the optimal value while having some global information on the system at the same time. It has been observed that Single Iteration and Sequential strategies lead to the same results (in terms of accuracy and computational time) if the total number of simulation runs is the same;
3. Sequential with Domain Reduction: this strategy is similar to the previous one, but after each iteration the domain (design space) is shrunk and panned using specific parameters and algorithms in order to chase the current optimum point, as it is shown in Figure 2.3. Since in parameter identification problems the objective is to find the global optimum, this strategy is the most appropriate one of all three available.

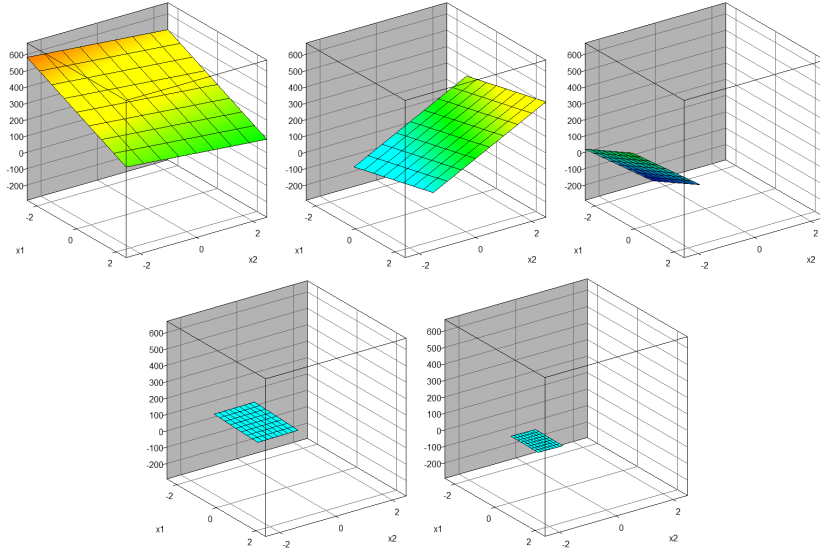


Figure 2.3: Example of a linear metamodel at the first five iterations of a Metamodel-based Optimization task with Sequential Domain Reduction strategy.

In general, both "Sequential" and "Sequential with domain reduction" strategies have proven to be very efficient and robust for parameter identification problems [21][22]. The domain reduction technique is by far the most efficient since it significantly reduces the total number of iterations, compared to the simple sequential strategy.

One flaw of sequential strategies is that they could lead to local optimal solutions in some cases. It is the user's job to verify that the experimental and simulated results are close enough at the end of the optimization. If this is not the case, then the optimization procedure has converged to a local optimum or the model used to describe the real phenomenon is not the appropriate one.

It is worth noticing that also Reliability-based Optimization (RBDO) tasks could be used for solving parameter estimation problems. In fact, this task should be preferred to Metamodel-based Optimization in those problems where there is uncertainty on the parameters and, as a consequence, on the response.

2.2.2 Direct Optimization

It has already been mentioned that solving parameter identification problems using direct optimization would lead to too many simulations, thus resulting in a very computationally expensive process. Nonetheless, it is useful to understand how direct optimization is performed in LS-OPT since this lead to a better understanding of the metamodel-based optimization.

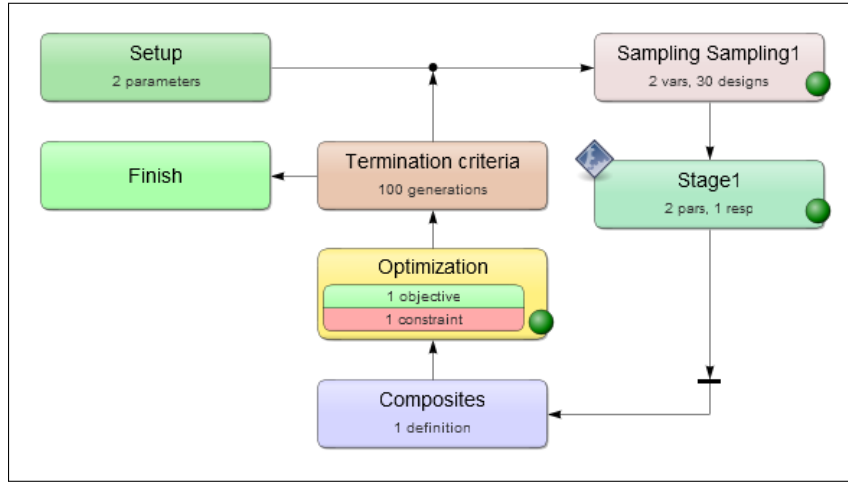


Figure 2.4: Flowchart of a direct optimization parameter estimation problem from the LS-OPT graphical user interface.

As it is shown in Figure 2.4, the flowchart of direct optimization is not very dissimilar to the previous one: sampling, simulations, computation of composites and optimization are still present. The main difference is the absence of the metamodel, which is not build after simulations have been performed. Other differences concern the choice of sampling and optimization algorithm. In fact, LS-OPT automatically performs direct optimization using Genetic Algorithms, therefore sampling simply consists in randomly generating points in the design space (according to specific criteria), whereas in the optimization dialog the

best simulated points are selected and transferred to the next iteration. Details on various steps of Genetic Algorithm (which applies both to direct optimization and to metamodel optimization) are given in Section 4.3.2.

Chapter 3

Metamodeling and Sampling Techniques

In this chapter, metamodeling and sampling techniques are reviewed from a theoretical point of view. These two topics are profoundly related to each other. In fact, the purpose of sampling is the construction of the metamodel and, viceversa, a metamodel could not be built without sampling the design space in the first place¹. For sake of simplicity, metamodeling techniques are firstly discussed and Design of Experiment theory is then presented. At the end of the chapter, a brief introduction to metamodel-based sensitivity analysis is presented and some considerations are made about the choice of sampling and metamodeling techniques for parameter identification. All theoretical notions presented in this Chapter are referenced in Chapters 21 and 22 of the LS-OPT User's Manual [9].

3.1 Metamodels

With the progress of computational power, more and more processes can be simulated. At the same time, however, more and more complex models have been developed to better describe real phenomena, so that a single simulation could take days or weeks to run. It is obvious that a parameter estimation problem cannot be solved by optimizing the actual model, because this would mean running lots of expensive simulations. By using metamodels, this cost can be drastically reduced without affecting too much the quality of results.

Given a physical phenomenon, a model is a set of mathematical laws which describe it. A metamodel is a model of the model, i.e. it is a model which describes the underlying model (see Figure 3.1). There are several advantages in using metamodels:

- they have in general a simpler mathematical formulation than the actual model, which is in fact the set of governing differential and algebraic equations describing the phenomenon;

¹This is the reason why the Sampling and Metamodel dialog of LS-OPT have actually the same content.

- they can be constructed using a relatively small number of simulations, thus lowering the total computational cost of the parameter optimization procedure;
- by construction, metamodels are smooth approximations of the underlying model; in this way local optima in the model caused by noise are avoided.

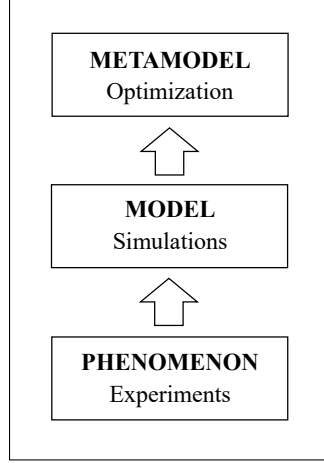


Figure 3.1

Since they are very useful for parameter identification problems, several classes of meta-models, such as polynomial response surfaces, neural networks, Kriging and support vector regression, have been studied. More details about the topics presented in the following sections are given in Chapters 21.1 and 22 of the LS-OPT User's Manual [9].

3.1.1 Polynomial response surfaces

Suppose that a generic system has to be studied and that it is simply represented by the function

$$y = \eta(\mathbf{x}), \quad (3.1)$$

where $\mathbf{x} = (x_1, \dots, x_N)$ is the vector of inputs (from now on also referred to as variables or parameters) and y is the output or response of the system. According to Response Surface Methodology (RSM), the system can be approximated by a function

$$\hat{y} = f(\mathbf{x}) \approx \eta(\mathbf{x}), \quad (3.2)$$

which can be expressed as a weighted sum of some basis functions

$$f(\mathbf{x}; \mathbf{a}) = \sum_{l=1}^L a_l \phi_l(\mathbf{x}) = \mathbf{a} \cdot \boldsymbol{\phi}(\mathbf{x}). \quad (3.3)$$

Polynomial response surfaces are defined by considering a polynomial basis for f . For example, a linear response surface is defined by

$$\boldsymbol{\phi} = (1, x_1, \dots, x_N), \quad (3.4)$$

while a fully quadratic one is defined by

$$\phi = (1, x_1, \dots, x_N, x_1^2, \dots, x_N^2, x_1 x_2, \dots). \quad (3.5)$$

Based on the degree of non-linearity of the underlying system, one could take into consideration polynomials of any order. However, it has been observed that polynomials of degree strictly greater than 2 are poorly efficient and could lead to over-fitting problems. In practice, common polynomials are linear, quadratic, elliptic (linear with diagonal terms) and interaction (linear with off-diagonal terms) ones.

The unknown coefficients \mathbf{a} in (3.3) are computed using the classic Least Square solution, namely

$$\mathbf{a} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \hat{\mathbf{y}}, \quad (3.6)$$

where \mathbf{X} is the dispersion matrix defined by

$$X_{pl} = \phi_l(\mathbf{x}_p), \quad 1 \leq p \leq P, \quad 1 \leq l \leq L, \quad (3.7)$$

and

$$\hat{y}_p = f(\mathbf{x}_p), \quad 1 \leq p \leq P, \quad (3.8)$$

being $\{\mathbf{x}_p\}_{1 \leq p \leq P}$ the set of points for which the output \hat{y}_p is known.

Polynomial metamodels have the great advantage of being easy and efficient to construct, because they do not require an optimization process (unlike, for example, neural networks, see Sections 3.1.2 and 3.1.3). However, due to their nature, they are only local approximations of the underlying model, therefore they have poor prediction capabilities outside the region of interest and their main application is in conjunction with sequential optimization strategies (with domain reduction).

3.1.2 Feedforward neural networks

Polynomial metamodels are very robust and efficient, especially if coupled with a sequential optimization strategy (with domain reduction). However, by definition, they are incapable of capturing high non-linearities in the behaviour of the response. To overcome this problem, other metamodeling techniques, such as neural networks, have been explored.

Just like polynomial response surfaces, neural networks try to model the relationship between inputs and outputs of a system. They do this by using single computational units, called neurons, which communicate according to specific topologies. In particular, feedforward neural networks (FNN) are characterized by having an initial layer which takes input variables, a series of hidden layers and a final layer which returns the output of the network, as shown in Figure 3.2. The information is passed through the layers only in the input-output direction.

Typically, input neurons are transparent (i.e. they simply transfer input values to the next layer), while output neurons perform a weighted sum of values coming from the previous layer. Differently, hidden neurons perform a weighted sum of input values which is then fed into an activation function (typically a sigmoid function, see Figure 3.3a); the output is finally passed to the next layer of neurons. In mathematical form a FNN with, for example, one hidden layer is defined as

$$\hat{y} = f(\mathbf{x}; \mathbf{w}) = w_0 + \sum_{h=1}^H w_h s \left(w_{h0} + \sum_{n=1}^N w_{hn} x_n \right), \quad (3.9)$$

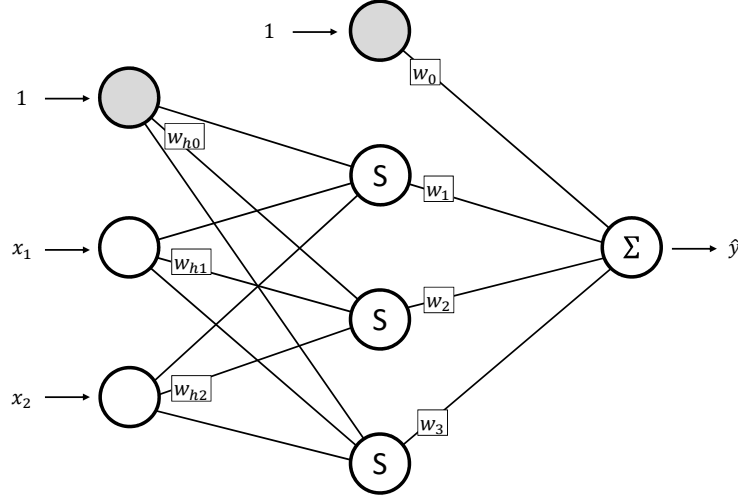


Figure 3.2: Example of a feedforward neural network with one 3-node inner layer. Grey neurons with constant input 1 are inserted in order to account for biases w_0 and w_{h0} .

where $s(\cdot)$ is the activation function, $\mathbf{w} = (w_0, \dots, w_H, w_{h0}, \dots, w_{HN})$ is the vector of weights and H is the number of neurons in the hidden layer. It is important to notice that FNNs are a non-linear regression technique, contrarily to polynomial response surfaces.

In order to find the unknown parameters \mathbf{w} , the network has to be trained using a set of inputs $\{\mathbf{x}_p\}$ with known outputs $\{\hat{y}_p\}$. These inputs are actually the design points chosen during sampling of the design space (see Section 3.2). The training algorithm is based on the well established backpropagation technique.

The accuracy of a FNN highly depends on the size of the training set. Scarce training datasets can lead to networks with large fitting error and poor prediction capabilities. The only solution to this problem is to use a larger dataset to train the network, although this increases the computational cost.

It is worth noticing that, with different initial weights, the training algorithm usually ends up in different local minima (in the space of weights), with the same quality. To counteract this phenomenon, LS-OPT considers a committee of networks with different initial weights and settings, trains them individually and finally averages them. The size of the committee significantly affects the time consumed in building the metamodel.

3.1.3 Radial Basis Function networks

Radial Basis Function (RBF) networks are a particular class of neural networks, characterized by having a single hidden layer, whose neurons have a radial basis function as activation function. Radial basis functions are functions whose value at some point only depends on the distance of the point from a fixed point called center of the function. Several RBF are present in literature (see Table 3.1), however only the Gaussian and Hardy's Multiquadratics functions are implemented in LS-OPT.

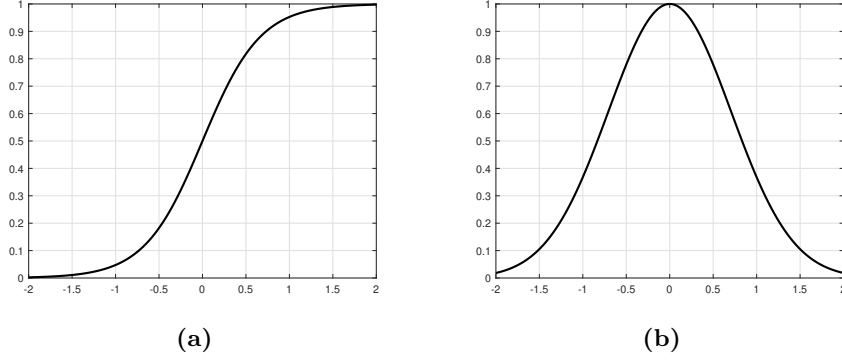


Figure 3.3: Sigmoid (a) and Gaussian (b) activation functions.

Name	$g(r; \varepsilon)$
Gaussian	$e^{-(\varepsilon r)^2}$
Hardy's Multiquadratic	$\sqrt{1 + (\varepsilon r)^2}$
Inverse Quadratic	$\frac{1}{1 + (\varepsilon r)^2}$
Inverse Multiquadratic	$\frac{1}{\sqrt{1 + (\varepsilon r)^2}}$

Table 3.1

The most general form of RBF networks is the following

$$\hat{y} = f(\mathbf{x}; \mathbf{w}) = w_0 + \sum_{h=1}^H w_h g(r_h; \varepsilon_h), \quad (3.10)$$

with

$$r_h = \left(\sum_{n=1}^N (x_n - w_{hn})^2 \right)^{\frac{1}{2}} \quad (3.11)$$

being the Euclidean distance of the point \mathbf{x} from the centre (w_{h1}, \dots, w_{hN}) of the h -th RB function.

It is clear from equations (3.10) and (3.11), that many variables are involved in the definition of the network. A simplified form of RBF networks assumes RBF centres coincident with design points $\{\mathbf{x}_p\}$, thus eliminating the need to compute weights w_{hn} . As a consequence, exactly P radial basis functions are used to interpolate the points.

The value of the shape parameter ε determines the "resolution" of the approximation: a too high ε could generate a function with poor prediction capability between points, while a

too small ε could cause poor prediction capability far away from design points. A very basic choice is to have all ε_h equal to some constant value, but clearly this is not an intelligent choice, when design points are not uniformly distributed over the design space. In order to account for this case, shape parameters are chosen such that each RBF overlaps with S neighbour functions, where S is a constant integer. The optimal value of S is computed by considering several trial networks with different values of S and then choosing the best one according to some criterion (prediction error, generalized cross validation, ect...).

Finally, the simplified formulation of RBF networks is

$$\hat{y} = f(\mathbf{x}; \mathbf{w}) = w_0 + \sum_{p=1}^P w_p g(\|\mathbf{x} - \mathbf{x}_p\|; S). \quad (3.12)$$

It is important to notice that this simplified version is actually a linear regression method with respect to the weights (w_0, \dots, w_P) . Therefore, less computational effort is required to build RBF networks compared to FNN, especially if FNN with large committees are considered. Moreover, thanks to internal cross validation, RBF networks have in general good prediction capabilities.

3.1.4 Kriging

The basic idea behind Kriging is to split the unknown response in two parts as follows:

$$f(\mathbf{x}) = F(\mathbf{x}) + Z(\mathbf{x}), \quad (3.13)$$

where F is the deterministic component (or trend) of f , while Z is its stochastic component. The trend is usually taken as a constant, linear or quadratic polynomial with coefficients to be determined. Z is characterized by having mean zero and covariance

$$\text{Cov}(Z(\mathbf{x}_p), Z(\mathbf{x}_q)) = \sigma^2 R(\mathbf{x}_p, \mathbf{x}_q), \quad 1 \leq p, q \leq P \quad (3.14)$$

where $\sigma^2 = \text{Var}(Z(\mathbf{x}_p))$ and R is a correlation function. Two common choices for the correlation function are Exponential

$$R(\mathbf{x}_p, \mathbf{x}_q) = \prod_{n=1}^N \exp\left[-\Theta_n |(x_p)_n - (x_q)_n|\right] \quad (3.15)$$

and Gaussian

$$R(\mathbf{x}_p, \mathbf{x}_q) = \prod_{n=1}^N \exp\left[-\Theta_n ((x_p)_n - (x_q)_n)^2\right] \quad (3.16)$$

In both expressions constants Θ_n have to be determined. In LS-OPT, the maximum likelihood estimates for Θ_n are computed by solving an optimization problem, using Genetic Algorithms (see Section 4.3.2).

3.1.5 Support Vector Regression

Support Vector Regression (SVR) is a metamodeling technique derived as a generalization of Support Vector Machine algorithms. The particular method implemented in LS-OPT

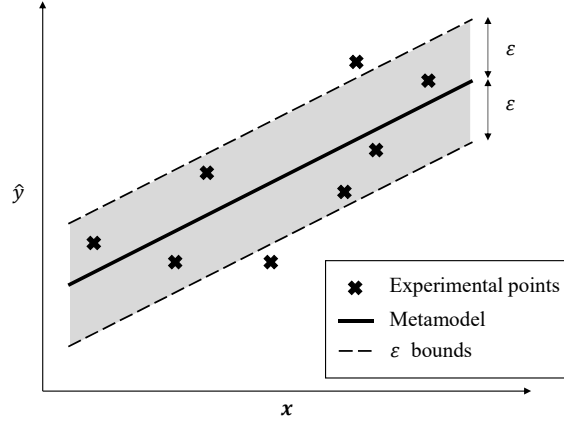


Figure 3.4: Representation of a one-dimensional linear SVR metamodel.

is ε -SVR [16], which consists in finding the "flattest" function that interpolates the data points, such that the deviation of each point from the function is at most ε (see Figure 3.4).

In the simplest case, the approximating function is assumed to be linear:

$$\hat{y} = f(\mathbf{x}; \mathbf{w}, b) = \mathbf{w} \cdot \mathbf{x} + b. \quad (3.17)$$

In order to find the weights \mathbf{w} and the bias b , the following constrained optimization problem has to be solved

$$\text{minimize} \quad \frac{1}{2} \|\mathbf{w}\|^2 \quad (3.18)$$

$$\text{subject to} \quad \begin{cases} y_p - \mathbf{w} \cdot \mathbf{x}_p - b \leq \varepsilon \\ \mathbf{w} \cdot \mathbf{x}_p + b - y_p \leq \varepsilon \end{cases}, \quad 1 \leq p \leq P. \quad (3.19)$$

The "flatness" of the approximating function f is achieved by minimizing $\|\mathbf{w}\|^2$.

The problem, as stated above, could not always have a solution. In order to allow for some errors, slack variables ξ_p, ξ_p^* and a cost parameter C are introduced:

$$\text{minimize} \quad \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{p=1}^P (\xi_p + \xi_p^*) \quad (3.20)$$

$$\text{subject to} \quad \begin{cases} y_p - \mathbf{w} \cdot \mathbf{x}_p - b \leq \varepsilon \\ \mathbf{w} \cdot \mathbf{x}_p + b - y_p \leq \varepsilon \\ \xi_p, \xi_p^* \geq 0 \end{cases}, \quad 1 \leq p \leq P. \quad (3.21)$$

As described in detail in [16], the optimization problem (3.21) is actually solved introducing Lagrangian multipliers α_p and α_p^* . The important theoretical result is that the unknown weights are actually a linear combination of input data points, i.e.

$$\mathbf{w} = \sum_{p=1}^P (\alpha_p + \alpha_p^*) \mathbf{x}_p, \quad (3.22)$$

therefore

$$f(\mathbf{x}; \mathbf{w}, b) = \sum_{p=1}^P (\alpha_p + \alpha_p^*) \mathbf{x}_p \cdot \mathbf{x} + b. \quad (3.23)$$

Equation (3.23) suggests a natural generalization to non-linear approximations by substituting the dot products with a non-linear kernel function k , so that the approximating function becomes

$$f(\mathbf{x}; \mathbf{w}, b) = \sum_{p=1}^P (\alpha_p + \alpha_p^*) k(\mathbf{x}_p, \mathbf{x}) + b. \quad (3.24)$$

Two common kernels, both implemented in LS-OPT, are polynomial kernels

$$k_m(\mathbf{x}_p, \mathbf{x}) = (1 + \mathbf{x}_p \cdot \mathbf{x})^m \quad (3.25)$$

and Gaussian kernels

$$k_\sigma(\mathbf{x}_p, \mathbf{x}) = \exp\left(-\frac{\|\mathbf{x}_p - \mathbf{x}\|^2}{2\sigma^2}\right). \quad (3.26)$$

Both optimization parameters ε , C and kernel parameters m or σ are internally optimized by LS-OPT using k -fold cross validation. This procedure is quite time consuming but has the advantage of generating a really good approximating function.

It is worth noticing that the approximating surface f only depends on the number of data points, not on their dimension. This fact suggests that SVR could be an efficient metamodeling technique for high dimensional optimization problems.

3.2 Design of Experiments

Design of Experiments (DoE) is the selection procedure for finding points in the design space that have to be analysed. Originally used for actual experiments, DoE has recently been applied also to numerical experiments, i.e. computer simulations. DoE theory offers useful techniques for accurately choosing which and how many experiments to perform in order to extract as much information as possible from a limited amount of experiments. Although numerous classes of designs are present in literature, only those implemented in LS-OPT will be discussed (see Chapter 21.2 of the LS-OPT User's Manual [9]).

We recall that $\mathbf{x} = (x_1, \dots, x_N)$ is the vector of N parameters/variables and we will assume that each x_n varies within an interval I_n . The Cartesian product of all I_n is the design space D , which, for sake of simplicity, will always be normalized to an N -dimensional centred hypercube in the following examples. The whole design, i.e. the set of points $\{\mathbf{x}_p\}_{1 \leq p \leq P}$, can be summarized in a P -by- N matrix, called the sampling matrix \mathbf{S} , whose columns are the chosen design points.

3.2.1 Full factorial designs

In full factorial designs, l uniformly spaced values are considered for each variable x_n within its range I_n . The total number of designs is l^N , therefore the design is called an l^N or l -level full factorial design. Full factorial designs can be expensive for large values of N , since the number of points increases exponentially. A simple way to overcome this problem is to use

the so-called fractional factorial designs, in which only a fraction of the full factorial points is considered. However, fractional factorial designs are not implemented in LS-OPT.

Usually, 2^N and 3^N full factorial designs are used for linear and quadratic surfaces respectively, since they provide a good oversampling percentage over the minimum required number of designs.

An example of a 3^2 full factorial design is shown in Figure 3.5 and its sampling matrix is

$$\mathbf{S} = \begin{bmatrix} -1 & 0 & 1 & 0 & 0 & 1 & -1 & 0 & 1 \\ 1 & 1 & 1 & -1 & 0 & 0 & -1 & -1 & -1 \end{bmatrix}^T. \quad (3.27)$$

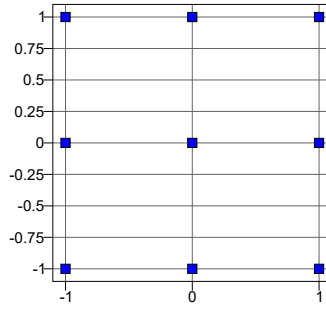


Figure 3.5: Example of a 3^2 full factorial design.

3.2.2 Koshal designs

Koshal designs are saturated designs that can only be used in conjunction with linear or quadratic polynomial response surfaces (see Section 3.1.1). Saturated means that the number of designs is the minimum required to find the unknown coefficients in the polynomial expansion, i.e. $N + 1$ for linear polynomials and $(N^2 + 3N + 2)/2$ for quadratic ones.

The sampling matrices of two-dimensional linear and quadratic Koshal designs are respectively

$$\mathbf{S}_1 = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}^T \quad \text{and} \quad \mathbf{S}_2 = \begin{bmatrix} 0 & 1 & 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & 1 & 0 & -1 \end{bmatrix}^T. \quad (3.28)$$

3.2.3 Central composite designs

Central composite designs (CCD) are specially designed for quadratic polynomial response surfaces. They consist of a 2-level full factorial design, enhanced by axial points and a central point (see Figure 3.7). The number of design is then $2^N + 2N + 1$. Two different types of CCD exist, which differ on the position of the axial points. In the first type, called Inscribed CCD, axial points are placed on the "faces" of the design space, whereas in the second type, called Circumscribed CCD, they are chosen so that they have the same distance from the design space centre as the factorial points. Since Circumscribed CCD has better properties (such as rotatability) than Inscribed CCD, it is the only central composite design implemented in LS-OPT.

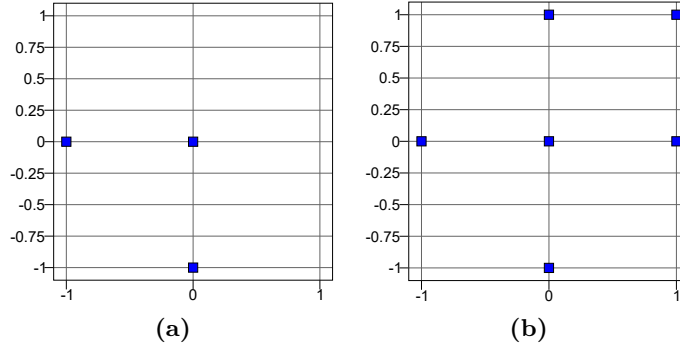


Figure 3.6: Example of two-dimensional Koshal designs for linear (a) and quadratic (b) polynomials.

The sampling matrix for CCD is

$$\mathbf{S} = \begin{bmatrix} 0 & 1 & 0 & -1 & 0 & \alpha & \alpha & -\alpha & -\alpha \\ 0 & 0 & 1 & 0 & -1 & \alpha & -\alpha & \alpha & -\alpha \end{bmatrix}^T, \quad (3.29)$$

where $\alpha = \sqrt{2}/2$ for Circumscribed CCD (see Figure 3.7) or $\alpha = 1$ for Inscribed CCD.

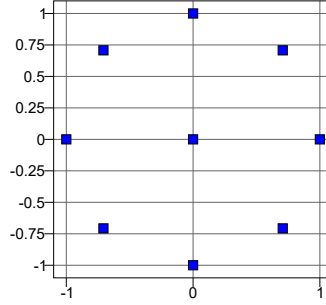


Figure 3.7: Example of a two-dimensional Circumscribed Central Composite Design.

3.2.4 D-optimal designs

D-optimal designs are generated by solving the following optimization problem²

$$\text{maximize } \det(\mathbf{X}^T \mathbf{X}) \text{ over designs } \{\mathbf{x}_p\} \subset \{\mathbf{y}_q\}, \quad (3.30)$$

where $\{\mathbf{y}_q\}$ is a basis set of Q designs, $\{\mathbf{x}_p\}$ is the D-optimal set of P designs and \mathbf{X} is the dispersion matrix, defined as in (3.7). D-optimal designs are constructed in a way that

²LS-OPT uses Genetic Algorithm to solve this problem.

maximizes the information present in the set of points, in fact $\mathbf{X}^T \mathbf{X}$ is the information matrix. Since their construction depends on \mathbf{X} , they can be used only with response surfaces of the form (3.3).

In LS-OPT, the user can specify the basis $\{\mathbf{y}_q\}$ from which to generate the D-optimal design. Common options are full factorial designs, LHS or space filling designs.

Because of the way they are defined, D-optimal designs are very useful when some constraint is defined on parameters (see Section 4.2.1). In this cases, D-optimal designs are the best option among all techniques implemented in LS-OPT, when used to build a polynomial response surface.

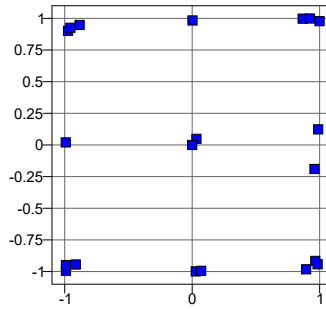


Figure 3.8: Example of 20 D-optimal designs, generated from a basis of 1000 space filling designs, for a quadratic polynomial surface.

3.2.5 Space filling designs

Space filling designs are a large category of designs, whose purpose is to generate a certain number of points uniformly distributed across the entire design space. From a programming point of view this is not an easy task, however a variety of algorithm exists in order to solve this problem.

The basic approach is to generate purely random points in the design space. This is referred to as Monte Carlo designs.

Another basic but more effective approach is Latin Hypercube Sampling (LHS). According to this algorithm, each variable interval is divided into P levels (or subintervals) and points are generated such that there is a design point for each level of each variable. In particular, this is called Central LHS. A slight variation, called Generalized LHS, introduces a small offset to each point, but preserves the original LHS structure, thus generating a partially random and partially structured design. The sampling matrices for both methods are defined as

$$S_{ij} = \frac{\eta_{ij} - \zeta_{ij}}{P}, \quad (3.31)$$

where $(\eta_{1j}, \dots, \eta_{Pj})$ are uniform random permutations of integers $(1, \dots, P)$; $\zeta_{ij} = 0.5$ for Central LHS, whereas it is a uniform random numbers between 0 and 1 for Generalized LHS. LHS designs have the advantage of representing all levels of all variables still using a few number of designs.

More advanced algorithms try to uniformly fill the design space by maximizing the minimum distance between any pair of points (maximin criterion). Three variants of the

same algorithm are implemented in LS-OPT:

1. the first one, called "maximin distance", starts from an arbitrary design and moves points randomly so as to optimize the maximin criterion;
2. the second one, called "maximin LHD permute", starts from an LHS design, then columns of the sampling matrix are permuted so as to optimize the maximin criterion;
3. the third, called "maximin LHD subinterval", also starts from an LHS design, then points are moved within their subinterval so as to optimize the maximin criterion; this algorithm preserves the original structure of the LHS design.

In all three cases, the optimization is internally performed in LS-OPT using Adaptive Simulated Annealing (see Section 4.3.3).

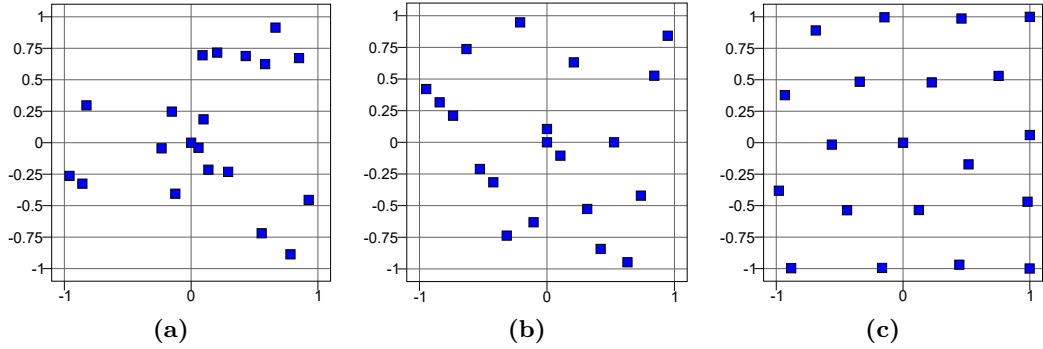


Figure 3.9: Examples of 20 designs points generated with Monte Carlo sampling (a), Central LHS (b) and Space filling with the "LHD subinterval" algorithm (c). It is evident how the structure of the design points increases from left to right.

3.3 Sensitivity analysis

Sensitivity analysis refers to a class of techniques used for determining the influence that each parameter has on the response. In parameter identification problems, sensitivity analysis is often used in an early stage for variable screening, i.e. for identifying which variable actually contribute to the response and which does not.

In LS-OPT, two methods are implemented for computing sensitivities, namely ANOVA and GSA. ANOVA, i.e. Analysis of Variance, is a regression-based sensitivity measure. Sensitivities are computed as

$$b_n = \frac{\partial f_{\text{lin}}}{\partial x_n} \cdot |I_n| \quad 1 \leq n \leq N, \quad (3.32)$$

where f_{lin} is a linear approximation of experimental points and $|I_n|$ is the range of variable x_n . Since ANOVA sensitivities are computed using linear approximations, they are significant only for quasi-linear models.

In order to accurately compute sensitivities for non-linear responses, Global Sensitivity Analysis (GSA) is used. GSA is based on Sobol's indices theory [19] and sensitivities are found using Monte Carlo integration on the metamodel. This means that GSA sensitivities are highly influenced by the accuracy and quality of the metamodel. Another important advantage of GSA over ANOVA is that the influence of each parameters on more than one response can be visualized at the same time. This is useful for determining the overall influence of each parameter when dealing with multiple load cases.

3.4 Which method to choose?

Considering the variety of sampling and metamodeling techniques that are implemented in LS-OPT, some considerations are needed in order to properly choose the right method.

Since the "Sequential with Domain Reduction" strategy will be used, using a polynomial metamodel is the first choice. In particular, linear metamodels should be preferred rather than quadratic ones, since they require less sampling points for their construction and the only side effect is a slightly larger number of iterations. The better accuracy gained from quadratic metamodels is not worth the time spent to simulate extra design points, especially for a large number of parameters.

When using polynomial response surfaces, D-optimal designs have two main advantages over other techniques:

1. they are expressly defined for maximizing the amount of information in the design based on the specific form of the polynomial surface, and
2. they can be used even when sampling constraints are defined, contrarily to non-optimized designs such as the full factorial, Koshal and central composite ones.

The choice of the basis from which to extract the D-optimal design does not seem to be too relevant for the quality of the design, if a large number of basis designs is provided. The LHS basis represents a good compromise between quality and efficiency for D-optimal designs.

Other classes of metamodels should not be discarded *a priori*. For example, neural networks are good candidates for sequential strategies in general, as it is possible and computationally inexpensive to use designs from previous iterations to update the metamodel. This is particularly useful when the objective of the study is to find the optimal value but global informations are needed for other reasons since neural networks can be locally refined while maintaining their global structure unvaried. Among neural networks, RBF should be preferred to FNN for two reasons. Firstly, RBF networks can be constructed much faster than FNN, since they are based on linear regression and they do not require training. The internal cross-validation may be expensive but, at least, it has the advantage of generating a metamodel with good prediction capabilities. Secondly, RBF networks are more robust than FNN when a small number of designs is used to build the metamodel.

Regarding Kriging metamodels, they have the advantage of exactly³ interpolating the

³Actually, the interpolation is almost exact since LS-OPT introduces a small error in order to avoid ill-conditioning of the correlation matrix.

input points. However this leads to a surface highly sensitive to noise and local refinements could cause global variations of the metamodel [8]. Moreover, the construction of Kriging metamodels could be time consuming because of the calculation of Θ_n for each n .

Finally, SVR metamodels are a valid choice for high dimensional problems, since their construction only depends on the number of input designs. However, it is not evident which kernel function should be used.

For non-polynomial metamodels, only full factorial, Monte Carlo, LHS and space filling designs are available. Space filling designs should be preferred to the other ones, regardless of the particular algorithm selected. However, when using a small number of points and/or parameters, full factorial designs are a convenient choice in terms of computational cost. Monte Carlo and LHS are rarely used for optimization; instead, they are useful for computing metamodel-based statistics or for Monte Carlo analyses.

Chapter 4

Optimization

In this chapter, the optimization algorithms that are implemented in LS-OPT are reviewed, together with important considerations about the choice of the objective function and constraints specifically for parameter identification problems. Detailed information about the optimization algorithms can be found in Chapter 23 of the LS-OPT User's Manual [9].

4.1 The objective function

An essential ingredient of an optimization process is the definition of the objective function, i.e. the function to be minimized or maximized. The objective function can be simply defined as an output quantity from simulations. However, in many cases, it is necessary to manipulate the simulation results in order to properly define the objective of the optimization. This is always the case for parameter identification problems, where the objective function is defined as the "distance" between experimental and simulated results. This concept of distance is vague as it assumes different expressions based on the actual problem that has to be solved. In the following sections, two common scenarios will be presented together with suitable approaches for defining the objective function.

4.1.1 Targeted composites

Let us suppose that a certain quantity (e.g. a force) is measured during an experiment at some fixed times. Then the experimental data are in the form

$$t_j, F_j \quad 1 \leq j \leq J, \quad (4.1)$$

where J is the total number of measurements. The correspondent time history f resulting from simulations will be evaluated at the same instants¹, leading to simulated data

$$t_j, f_j = f(t_j) \quad 1 \leq j \leq J. \quad (4.2)$$

¹If simulation data at times t_j are not directly computed, it is possible to extrapolate them using, for example, a linear approximation.

In this situation, the distance between experimental and simulated data can be defined using the Mean Square Error (MSE) expression

$$\text{MSE} = \sum_{j=1}^J w_j \left(\frac{f_j - F_j}{s_j} \right)^2, \quad (4.3)$$

where w_j and s_j are some weighting and scaling factors, specified by the user. Similarly, the same metric can be defined in its square root form as

$$\text{SqrtMSE} = \sqrt{\text{MSE}} \quad (4.4)$$

These two types of composites are referred to as targeted composites because they measure the distance of simulated data f_j from target values F_j .

In a more general case, also target values for parameters can be added in the definition of the MSE as a way of regularizing the objective function. The complete expression of MSE is then

$$\text{MSE} = \sum_{j=1}^J w_j \left(\frac{f_j - F_j}{s_j} \right)^2 + \sum_{n=1}^N \omega_n \left(\frac{x_n - X_n}{\sigma_n} \right)^2, \quad (4.5)$$

where x_n is the n -th component of the parameter vector \mathbf{x} and X_n is its target value.

4.1.2 Curve matching composites

In order to illustrate curve matching metrics it will be assumed that experimental data are summarized in a single two-dimensional curve, e.g. a force vs. displacement curve resulting from a tensile test. The data extracted from simulations will be consistent with the available experimental data. In this case, the distance between experimental and simulated curves can be computed in several different ways.

Mean Square Error

The first option is to compute the distance using the MSE formulation, namely:

$$\text{MSE}(\mathbf{x}) = \frac{1}{J} \sum_{j=1}^J w_j \left(\frac{f_j(\mathbf{x}) - F_j}{s_j} \right)^2 \quad (4.6)$$

where J is the total number of points on the experimental curve², F_j is the j -th experimental target value and $f_j(\mathbf{x})$ is the metamodel constructed using the j -th points of simulated curves (see Figure 4.1). By default, $w_j = 1$ and $s_j = \max\{|F_k| : 1 \leq k \leq J\}$ for each j .

Although this formulation is very similar to the MSE targeted composite, it differs in some important aspects: firstly, the MSE targeted composite is intended to be used also for data coming from different curves or simulations, whereas the MSE curve matching

²In LS-OPT, it is actually possible to use equally spaced points for computing the MSE, instead of experimental ones, which is recommended if the experimental curve has too many points (approximately greater than 100).

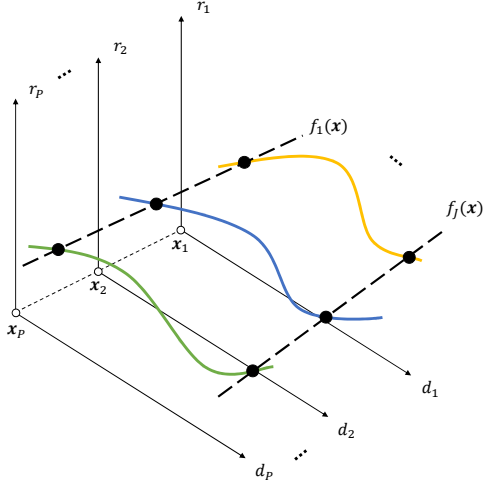


Figure 4.1: Graphical representation of how metamodels are constructed from simulated curves: solid coloured lines are the histories obtained from simulations, while dashed black lines are (linear) metamodels built from points with the same indices on the simulated curves.

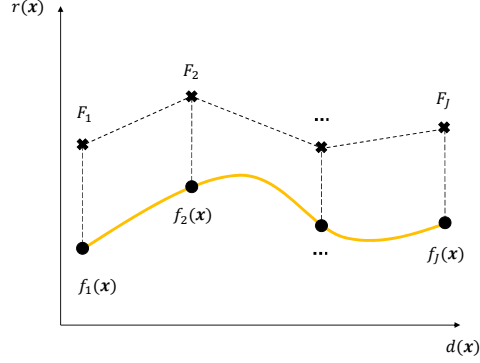


Figure 4.2: Simulated histories can be extended over all the design space by constructing metamodels $f_j(\mathbf{x})$ like in Figure 4.1. Then, the MSE distance between experimental and simulated curves can be computed for each point \mathbf{x} in the design space.

composite is specially defined as a metric for measuring curve distances; secondly, when the experimental curve consists of many points, the targeted MSE is far less convenient as the user should insert points one by one.

The graphical meaning of definition (4.6) is shown in Figure 4.2. It is evident that this formulation cannot be used for curves with non-monotonically increasing abscissa, such as hysteretic curves. In this cases, other options are required.

It is important to notice that the MSE has not the same form as the chosen metamodel type for the optimization. For example, if a linear metamodel has been selected, then all $f_j(\mathbf{x})$ will be linear functions while the MSE will be quadratic. Furthermore, in LS-OPT only linear, quadratic and RBF metamodels can be selected to approximate histories, while all other metamodels can be used for scalar responses.

Partial Curve Mapping

The second way of computing distance between curves is known as the Partial Curve Mapping algorithm [24], which consists of the following simple steps:

1. In the first phase, the two (polygonal) curve parameterizations are normalized and their total polygonal lengths are computed;
2. Then a finite number of offsets is defined in order to slide the shortest curve onto the other. The minimum offset is zero, while the maximum coincides with the difference between the two curve lengths;

3. For each offset, points of the shortest curve are mapped onto the the other ones: the first point is mapped using the offset value, while the successive ones are mapped in such a way that the mapped curve maintains its original length;
4. The discrepancy between the curves, as a function of the offset, is found by computing the area between them, using the map computed in the previous step;
5. Finally, the distance is given as the minimum over all offsets of all discrepancies computed previously.

A graphical representation of how the algorithm works is shown in Figure 4.3. It is important that both curves are free of noise in order to not overestimate their lengths, therefore filtering is recommended for noisy curves.

This algorithm has been developed on purpose for hysteretic curves, for which the MSE metric is undefined. However, it presents other advantages over the MSE: firstly it takes into account for steep parts of the curves to be matched and secondly it can be used even when the two curves have partially overlapped or even disjoint ranges. The latter case is particularly frequent in parameter estimation problems since the experimental curve could be restricted to a certain range due to the measurement procedure.

If $\text{map}(\cdot, \cdot)$ is the function that generates the map between the curves and computes the area, then the composite expression depends on the parameter vector in the following way

$$\text{PCM}(\mathbf{x}) = \text{map}(\{f_j(\mathbf{x})\}, \{F_j\}), \quad (4.7)$$

where $f_j(\mathbf{x})$ are defined as for the MSE curve matching composite. $\text{PCM}(\mathbf{x})$ is the actual surface that will be minimized through the optimization algorithm. Similarly to the previous case, PCM will not have the same form as the chosen metamodel, but it will be a non-linear function of it.

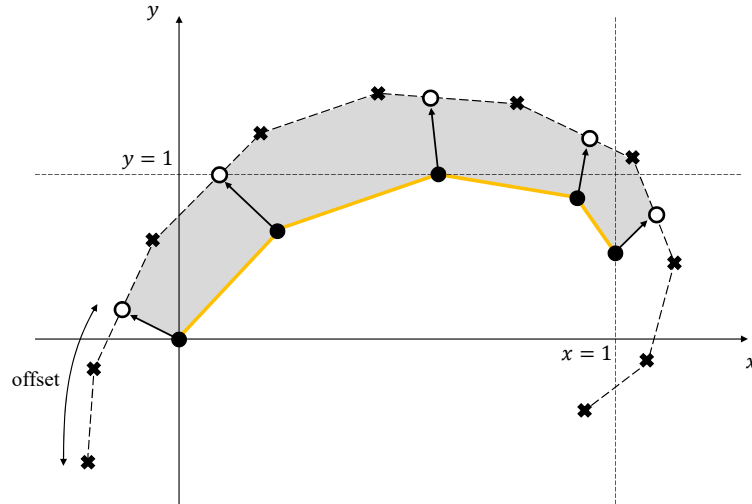


Figure 4.3: Representation of the Partial Curve Mapping algorithm.

Dynamic Time Warping

A third way of computing distance between curves is the so-called Dynamic Time Warping (DTW) algorithm. This algorithm has been widely used in many fields such as speech recognition, signature recognition or general shape comparison. Recently, DTW has been introduced in parameter identification problem as a powerful way to match simulated and experimental curves³ [3].

Similarly to Partial Curve Mapping, the idea behind DTW is to consider the discrepancy of the two curves in the abscissa direction, apart from the ordinate one which is the only one used in the MSE composite. In particular, in the first stage, the algorithm builds a map between points from the curves, following a simple specific set of rules. This map does not need to be bijective, therefore curves could have a different number of points as it is often the case. After the map has been defined, one way to compute the overall distance is to sum the distances between each pair of points in the map (in this way the DTW distance may be considered as an extension of the MSE formulation). An example of DTW mapping is shown in Figure 4.4 where each pair defined by the map is connected by dashed lines.

DTW is surely better than the MSE curve matching composite since the shape of the curves is considered in the mapping, rather than the ordinate distance only. For this reason, the curves do not need to be aligned in the abscissa direction in order to be correctly compared. Moreover, DTW does not require filtering or smoothing of the curves, contrarily to the Partial Curve Mapping algorithm, because the curve lengths are not involved in the algorithm.

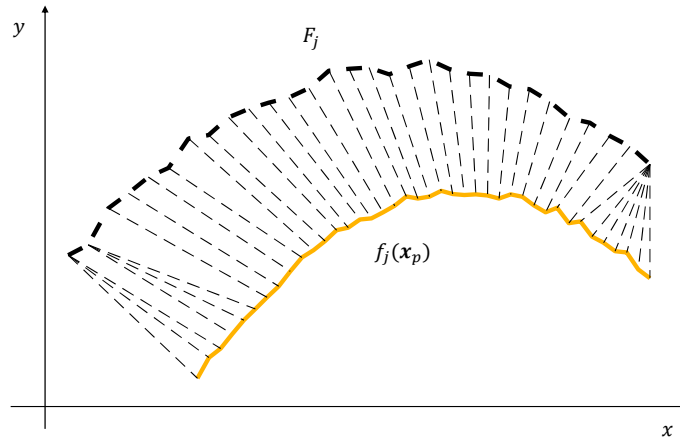


Figure 4.4: Mapping between curves generated with the Dynamic Time Warping algorithm. Notice how the map takes into account for the shape of the curves.

³DTW is not available in the LS-OPT version used in this thesis, but it will be implemented in future versions. It is still possible to manually implement the algorithm and insert it as a post-processor stage in LS-OPT.

4.1.3 Multi-objective optimization

In some optimization problems it is necessary to define more than one objective function. For example, this is the case when multidisciplinary analyses are performed or multiple load cases of the same system are considered. Multi-objective optimization (MOO) problems can be solved in LS-OPT by defining several objectives in the Optimization dialog. However MOO is a wide topic and will not be used in this work. Since in parameter identification we are not interested in trade-off analyses or Pareto optimal solutions, it is convenient to combine different objectives into one single quantity, thus removing the need of multi-objective optimization. The different objectives are usually weighted and added together.

4.2 Constraints

The presence of constraints either on the objective function or on the parameters is an important aspect of an optimization process. Constraints usually have the following form

$$L_i \leq g_i(x_1, \dots, x_N, \hat{y}) \leq U_i, \quad 1 \leq i \leq I, \quad (4.8)$$

where I is the total number of constraints, L_i and U_i are respectively the lower and upper bounds of the i -th constraint and g_i is a generic composite expression, i.e. a mathematical expression involving the response \hat{y} and/or the parameters x_1, \dots, x_N . It is important to notice that LS-OPT gives the possibility to specify the strictness of the constraints as well as constraints with only one bound.

Bilateral constraints such as

$$h_k(x_1, \dots, x_N, \hat{y}) = H_k, \quad 1 \leq k \leq K, \quad (4.9)$$

can also be defined by using the equivalent formulation

$$\begin{cases} h_k(x_1, \dots, x_N, \hat{y}) \leq H_k \\ h_k(x_1, \dots, x_N, \hat{y}) \geq H_k \end{cases}, \quad 1 \leq k \leq K. \quad (4.10)$$

However, this class of constraints will never be exactly satisfied from a numerical point of view, so that all constraints actually are as in (4.8).

All constraints are internally normalized by the software so as to avoid numerical problems due to large differences in the order of magnitude of defined quantities.

4.2.1 Sampling constraints

In some problems, it could be necessary to define constraints on parameters x_n only. Since it would be inefficient to sample the whole design space and then discard infeasible designs (i.e. designs that violate the constraints), it is possible in LS-OPT to specify sampling constraints in the Sampling dialog. The actual definition of the constraint is the same, as it makes use of composite expressions. However, in this way only feasible designs will be considered in the sampling process. It must be said that not all sampling techniques are compatible with this option. For example, it is obvious that non-optimized designs, such as full factorial, Koshal and central composite ones, do not have the ability of generating constrained designs. This is not the case for the other available techniques (D-optimal and space filling), which are capable of sampling from design space with irregular shapes (see Figure 4.5).

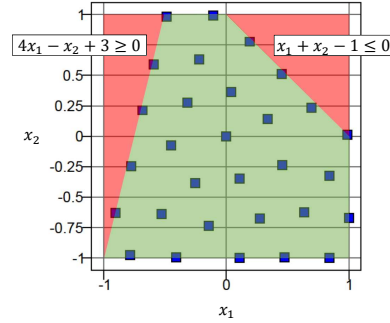


Figure 4.5: Example of 30 space filling designs in the domain $[-1,1] \times [-1,1]$, subject to sampling constraints. The space filling algorithms takes into account constraints and guarantees uniformly distributed points inside the feasible region.

4.3 Optimization algorithms

In a generic parameter identification problem, once the metamodel has been built and the objective function and constraints have been defined, the actual optimization process can begin. From a mathematical viewpoint, the only theoretical result for constrained optimization problems is represented by the Karush-Kuhn-Tucker (KKT) conditions, which are an extension of the Lagrangian multiplier method to the case of unilateral constraints. However KKT conditions are not useful for conceiving an optimization algorithm capable of finding global optima. For this reason, algorithms presented here are based on heuristic ideas, rather than proved mathematical results.

4.3.1 Leap-Frog Optimization

The Leap-Frog Optimization algorithm (LFOP), conceived by J. A. Snyman [17], is a gradient-based method that generates a dynamic trajectory towards a local optimum. The basic algorithm seeks the minimum of the objective function f by considering the associated dynamic problem of a particle of unit mass moving in a conservative force field, in which the potential energy of the particle is given by f itself. Although LFOP is based on a physical idea, the algorithm has been developed introducing modifications on a heuristic basis, in order to improve its performances.

The constrained version of the algorithm, called LFOPC [18], applies the original unconstrained algorithm to a penalty function, which involves the objective function f and a penalty parameter. The algorithm consists of three phases, which will be briefly explained:

- In Phase 0, the objective function is minimized with a moderate value of the penalty parameter, starting from a random point;
- In Phase 1, in order to satisfy still active constraints, the objective function is minimized with a high value of the penalty parameter, starting from the result of Phase 0;
- Finally, in Phase 2, if the number of active constraints exceeds the number of variables, another optimization run is performed in order to find a compromised solution.

Since it is a gradient-based method, the LFOP algorithm can only find local optima. However, for a sufficiently smooth surface, an effective strategy for seeking global optima with LFOP, is a multi-start approach, in which several optimization runs are performed with different starting points and the best optimum is chosen.

4.3.2 Genetic Algorithms

Genetic Algorithms (GA) seek the global optimum of the objective function (also called cost function) trying to imitate the natural process of evolution. Basically, an initial population of individuals (i.e. design points) evolves through various generations. In order to explore the entire design space, mutation and crossover operators are applied to some individuals. At the end of each generation, individuals with the highest cost function are discarded and the process is repeated.

In more details, at each generation, the main steps of GA are:

1. Selection: individuals with the lowest cost function are selected and will survive to the next generation. Selection can be implemented in several ways: common ones are Tournament selection, Roulette selection and Stochastic Universal Sampling;
2. Crossover: genes from two individuals (parents) are mixed to create one or two new individuals (children). The number of crossovers and the amount of similarity between parents and children can be adjusted selecting the crossover operator (SBX or BLS) as well as crossover distribution and probability;
3. Mutation: genes of some individuals randomly change, based on mutation distribution and probability.
4. Elitism: some of the best individuals of each generation are automatically passed to the next generation without undergoing the selection process. This feature can significantly increase the performance of the algorithm, by ensuring a monotonic decrease of the cost function throughout the generations.

It is important to mention that LS-OPT automatically uses Genetic Algorithms for discrete and mixed-discrete optimization problems, as well as for Direct Optimization.

Genetic Algorithms are mainly suited for Multi-Objective Optimization because they are very effective in finding Pareto optimal solutions and then computing the Pareto optimal frontier. Since they are very expensive compared to other algorithms, they are not recommended for single-objective optimization problems.

4.3.3 Adaptive Simulated Annealing

Simulated Annealing (SA) is a heuristic global optimization algorithm whose idea is based on the annealing process used to strengthen metals. In this process the metal is heated up and then slowly cooled down so as to allow for crystals inside the material to assume a low energy configuration. In SA, the objective function is interpreted as energy and the intent is to find the point (also called "state") with smaller energy by slowly decreasing a temperature parameter. At each temperature step the algorithm explores neighbours of the current state and moves randomly to one of them according to an acceptance function.

In this way the algorithm can potentially explore the whole design space, thus having more probability to end up in the global optimum basin.

The algorithm is essentially defined by three main functions:

1. the candidate distribution function describes what are the neighbours of a given state in the design space, thus defining the topology of the state space graph;
2. the cooling schedule determines the rate of cooling as well as the initial and final temperatures;
3. the acceptance function represents the probability of moving to a low energy state at each step. It depends on the current temperature value and tends to 1 as the temperature tends towards its final value.

Adaptive Simulated Annealing (ASA), which is the actual algorithm implemented in LS-OPT, is an improvement of SA in which some temperature parameters are automatically adjusted during the optimization, thus increasing the overall efficiency of the algorithm.

4.3.4 Differential Evolution

Differential Evolution (DE) is an optimization algorithm similar to Genetic Algorithms in that it uses a population that evolves over time. However, mutation and crossover operators are directly specified in the algorithm, so that the user has not to specify any option. Unfortunately, DE cannot be used for optimization with constraints in LS-OPT, although a constrained version of the algorithm (which uses a penalty function formulation) exists in literature.

4.3.5 Hybrid algorithms

Hybrid algorithms have been introduced in order to increase efficiency of GA and ASA. It has been observed that GA and ASA have the ability of rapidly converging into the basin of the global optimum, but then it takes several iterations for them to make small improvements towards the actual optimum. On the other hand, LFOP is very fast at converging once a point near the global optimum has been chosen as the starting point. These considerations suggest to use GA or ASA for the first few iterations and then switch to LFOP in order to quickly and accurately converge to the optimum. The combination of a local and a global algorithm is referred to as a hybrid optimization algorithm. Both hybrid algorithms GA/LFOP and ASA/LFOP are implemented in LS-OPT.

Chapter 5

Experimental tests

5.1 Introduction

Experimental data are a fundamental ingredient for solving a parameter estimation problem, as they describe the real behaviour of the material which has to be reproduced by numerical simulations.

In this chapter the experimental tests that were carried out are described in detail. Among all possible mechanical tests, tensile tests were chosen to be performed because of their simplicity and effectiveness, although a better characterization of the material would be possible by performing a more complete set of experiments (such as compression tests or flexural tests). It is worth noticing that, when dealing with parameter identification, experimental tests are not performed with the purpose of directly compute physical constants of the material based on the test data. Instead, experimental data are used to "guide" the material model towards the real behaviour through the optimization algorithm. Therefore, all experimental data are valid and usable as long as the tests can be consistently reproduced by numerical simulations.

5.2 Materials

5.2.1 Composites

Composites are a particular class of materials made of two or more constituents combined in some way with the purpose of obtaining properties that none of the single constituent has. Basic composite materials are generally made of two constituents, namely the matrix and the reinforcement. The matrix has the function of keeping in place the reinforcement and it cannot support high loads, while the reinforcement has the function of supporting very high loads in specific directions (depending on the actual type of reinforcement). Composite materials can be classified based on the type of matrix, such as metal, ceramic or polymer matrices, or on the type of reinforcement. The latter can be made of metal particulates or fibers (carbon or glass fibers) with different lengths.

Another class of composites is that of laminated and sandwiched composites. The former ones are made of single anisotropic layers stratified with different directions, while

the latter ones are generally made of two thin external layers and one thick internal layer, which fulfill different mechanical functions.

5.2.2 Tested material

The material tested for this study is a carbon-fiber reinforced epoxy resin manufactured by the company CarboSix. The material is produced in large planar sheets made of several layers of carbon-fiber fabric kept in place by the resin. Because of this, the material can be classified both as a matrix/reinforcement and as a laminated composite.

The number of layers of the sheets can vary depending on their thickness. From the manufacturer, it is known that the 2mm sheets are made of 4 layers while the 3mm ones are made of 6 layers. Therefore it can be deduced that each layer has a thickness of 0.5mm. The layers are made of the same material (carbon-fiber fabric), but with different fiber densities: the two external layers have a surface density of 200g/m^2 while all the internal ones (2 or 4 depending on the total thickness) have a surface density of 430g/m^2 . It is very important to specify that fibers in each layer run in both longitudinal and transverse direction ($0^\circ/90^\circ$) and all layers are stratified with the same direction (see Figure 5.1). This is important since the orientation of fibers highly influences the resulting mechanical properties of the material in different directions, i.e. its degree of anisotropy.

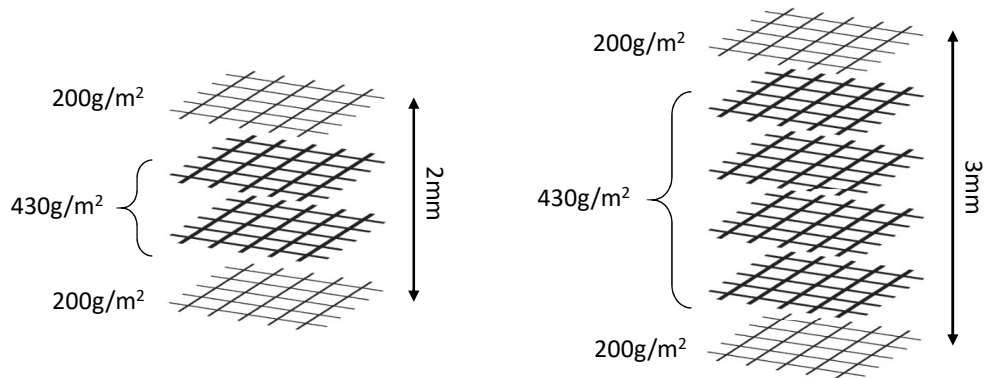


Figure 5.1: Spatial arrangement of single carbon-fiber fabric layers inside sheets with 2mm thickness (left) and 3mm thickness (right), with respective fiber densities.

Apart from the material previously described, another one was tested too. This material is again a carbon-fiber reinforced epoxy resin, but with long continuous unidirectional fibers. CarboSix produces pultruded profiles (such as that in Figure 5.2) with a specific manufacturing process that allows to embed uninterrupted carbon fibers in the resin. The resulting product is a profile with fibers running inside the material from one end to the other, which give the material really high strength for bending and longitudinal traction.

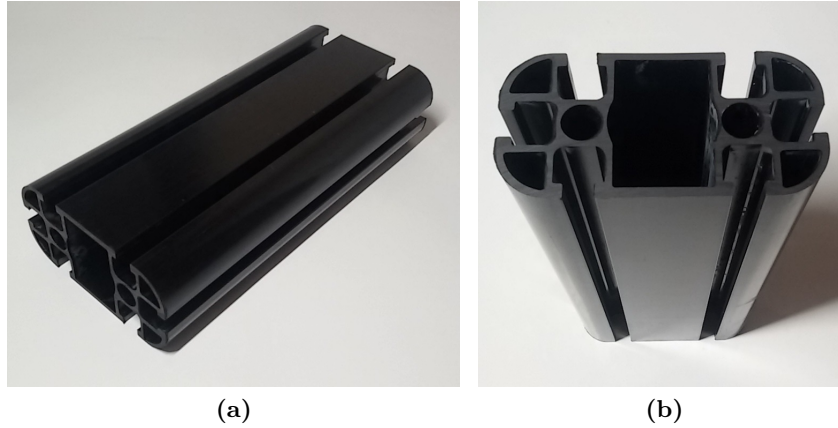


Figure 5.2: One of the unidirectional-reinforced pultruded profiles produced by CarboSix; principal dimensions of the cross section are 45mm and 90mm.

5.3 Tensile tests

5.3.1 Specimens

Specimens for tensile tests were categorized in 6 different groups (A, B, C, D, E, E') based on their specifications, as it is shown in Table 5.1. Groups from A to D were obtained by cutting carbon-fiber reinforced sheets of different thicknesses. For the 2 millimeters thick one, specimens were cut in 3 different directions with respect to the main orientation of the fibers. Since each layer of fabric in the sheets consists of bi-directional braided fibers and all layers are superimposed with the same orientation, specimens cut at 0° and 90° should exhibit the same behaviour.

Another group of specimens was cut from the flat region of a 45x90 pultruded profile with unidirectional fibers¹ (see Figure 5.2). Since there were discrepancies between the top and bottom thicknesses of the profile walls, the resulting specimens turned out to have different thicknesses. Therefore it was decided to subdivide this group into two subgroups referred to as E and E'.

All specimens were cut using a Type I profile, except for specimens in group C which were cut using a Type II profile because of their greater thickness. Type I and Type II profiles, shown in Figure 5.4, were determined in accordance to the standard ASTM D638 [1]. The exact dimensions of each profile are reported in Table 5.2. Photos of one specimen of each group (before testing) are shown in Figure 5.3.

¹It should be pointed out that fibers might not have constant direction through the material, due to the manufacturing process. However, only slight (thus negligible) variations from the longitudinal direction should be expected.

Group	Type	Thickness	Angle	Quantity
A	I	2mm	0°	3
B	I	2mm	45°	3
C	I	2mm	90°	3
D	II	3mm	0°	3
E	I	2.8mm	0°	3
E'	I	2.4mm	0°	3

Table 5.1: List of all specimens used in tensile tests. See text for details on the manufacturing of the specimens.

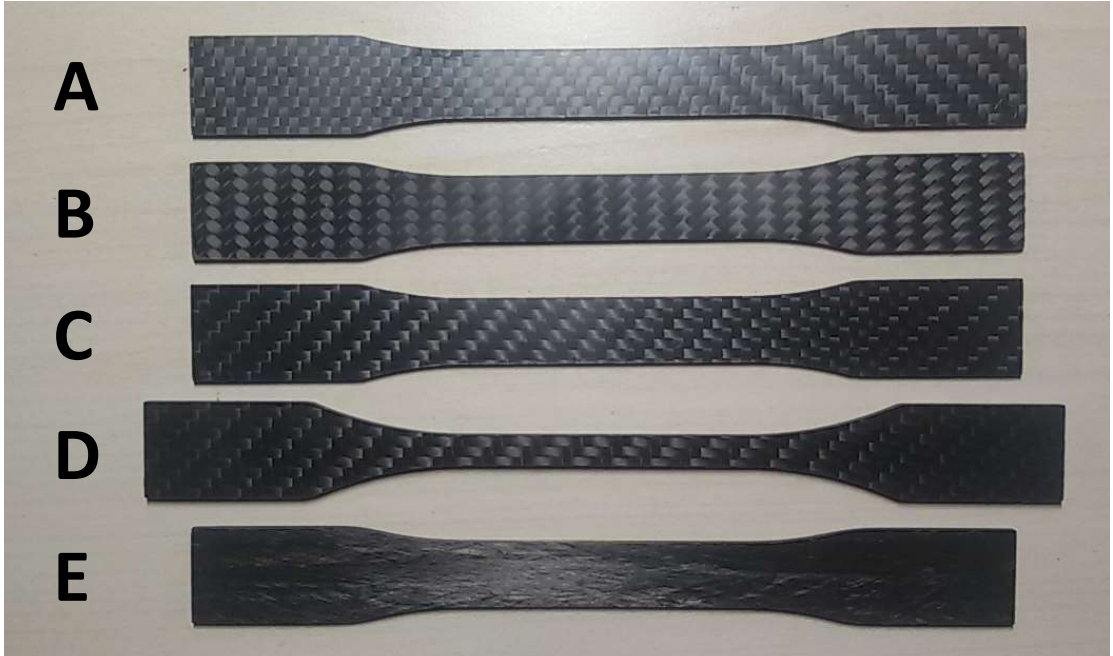


Figure 5.3: Photos of untested specimens for each test group. Notice that specimens A, C and D have the same fiber directions and differences in the photos are due to lighting effects.

5.3.2 Setup

All experiments were performed on a servohydraulic machine (Instron 8801) with force capacity up to 100kN, which is shown in Figure 5.5. All tests were conducted at room temperature (about 20°C) and no conditioning procedure was applied to the specimens. Tests involving thermal variations were not considered in this study although it is known that many composite materials may have different properties depending on their temperature, due to the viscoplastic behaviour of the resin.

Before running the tests, all specimens were uniquely marked in accordance with their

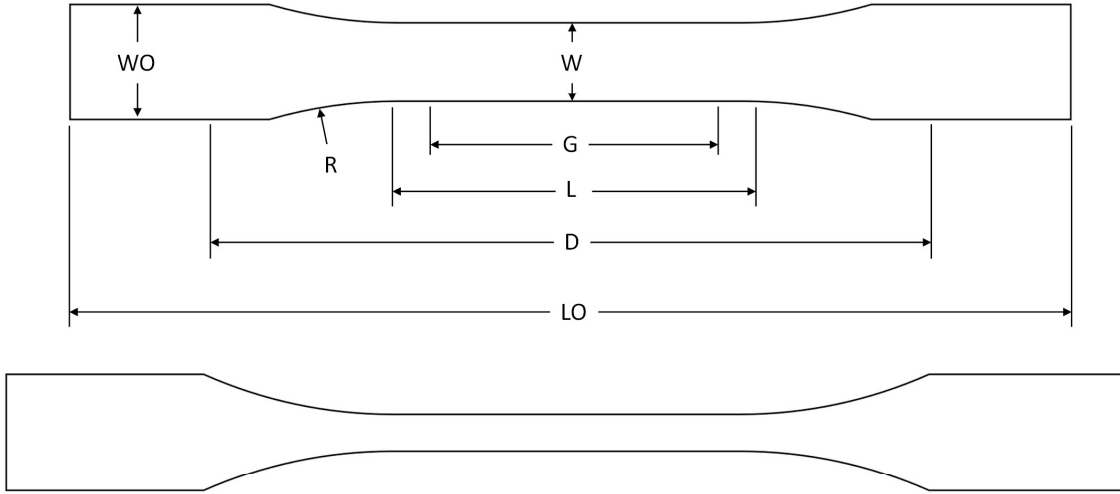


Figure 5.4: Type I (top) and Type II (bottom) profiles for tensile tests and dimensions. Dimension labels for the second profile are analogous to the first one.

Dimensions	Type I	Type II
W - Width of narrow section	13	6
L - Length of narrow section	57	57
WO - Width overall	19	19
LO - Length overall	165	183
G - Gage length	50	50
D - Distance between grips	115	135
R - Radius of fillet	76	76

Table 5.2: Dimensions of specimens for tensile tests according to the standard ASTM D638. All dimensions are expressed in millimeters.

group and equally spaced reference marks were traced in the gage section in order to estimate the final deformation in case of failure of the acquisition system.

At the beginning of each test, each specimen was placed in the machine and an extensometer was applied in order to measure deformation of the specimen over time. Tests were performed by imposing a constant speed of the moving part of the machine and simultaneously measuring the applied load through a load cell. A 2mm/min speed of testing was used for all tests except for specimens in group C which were tested at 5mm/min².

²It is known to the author that composite materials often present strain-rate effects, therefore all tests should have been performed at the same rate. However, by observing the first experimental curve for group B (Figure 5.8), it can be noticed that there is only a subtle difference between 2mm/min and

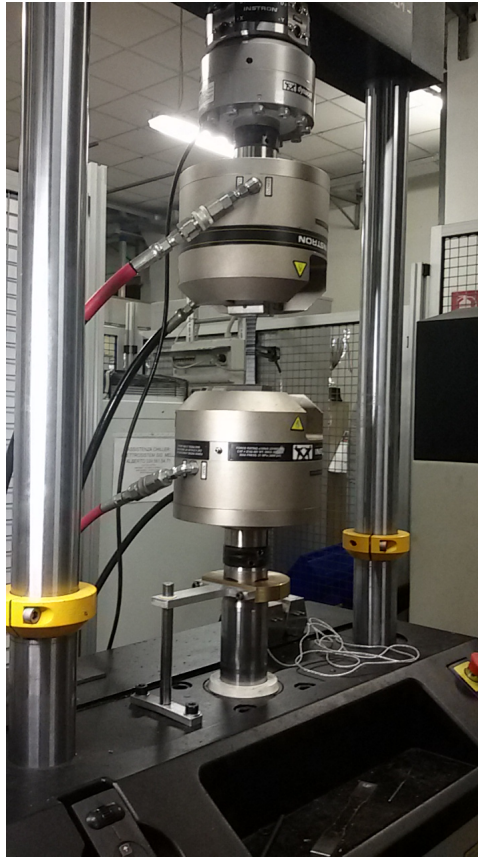


Figure 5.5: Servohydraulic machine Instron 8801 used for all the experimental tests.

Although seven specimens per group were available, only three per group were tested since there was not great discrepancy between the acquired data.

5.3.3 Results

The force vs. displacement curves resulting from all tensile tests are shown in Figures 5.6 to 5.16.

Group A

For group A (Figure 5.6) the behaviour of the specimen is almost linear until failure occurs between 16kN and 18kN, corresponding to a displacement of about 2.2mm. There is no evident plastic deformation and the failure is brittle. The little sharp decreases in the curves immediately before failure correspond to the breaking of single layers in the composite material and they starts after a 14kN load. The Young modulus can be directly

5mm/min testing speed.

computed from the stress vs. strain curves and it is about 63.73GPa. One of the failed specimens is shown in Figure 5.7 (see Appendix A for photos of all failed specimens). Almost all specimens of type A failed at the end of the gage section and were cracked on both sides, with the cracks being oriented in the 90° direction.

Notice that the vertical offset of curves (both in this and other tests) is due to the fact that the specimen is inevitably slightly loaded (often in compression) when it is fixed in position through the grips. The load cell should not be recalibrated after this step since this could invalidate the experiment. To match the curves it is simply necessary to shift them horizontally so that the point of zero load corresponds to the point with zero displacement.

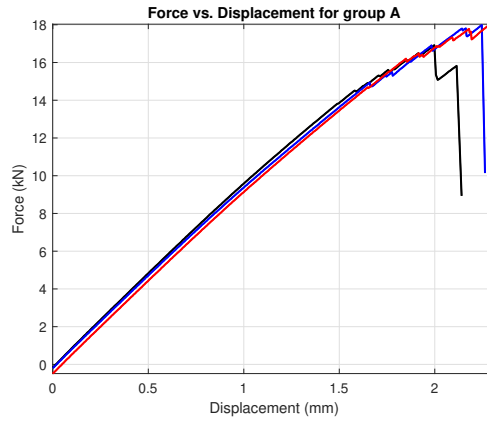


Figure 5.6: Results of tensile tests for 2mm thick specimens of Type I with 0°/90° fiber angles (group A).



Figure 5.7: Failed specimen of group A.

Group B

In group B (Figure 5.8) there is an evident plastic behaviour after a displacement of about 2mm. The total displacement until failure is much greater than other cases (oscillating between 10mm and 12mm), but the maximum supported load is much smaller (around 4kN). The small step in the plastic range of the black line is due to a change in the speed of testing from 2 to 5mm/min. The other two curves were obtained with 5mm/min speed of testing. Contrarily to the previous case, there is no failure of single fibers but all of them fail at once along the 45° direction, as it is shown in Figure 5.9. The elastic modulus computed from the first linear part of the curve is about 17.68GPa.

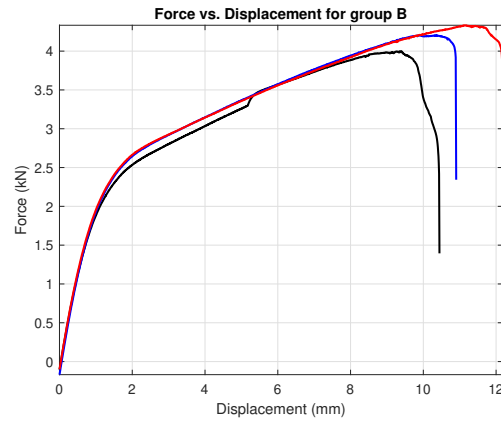


Figure 5.8: Results of tensile tests for 2mm thick specimens of Type I with $-45^{\circ}/45^{\circ}$ fiber angles (group B).



Figure 5.9: Failed specimen of group B.

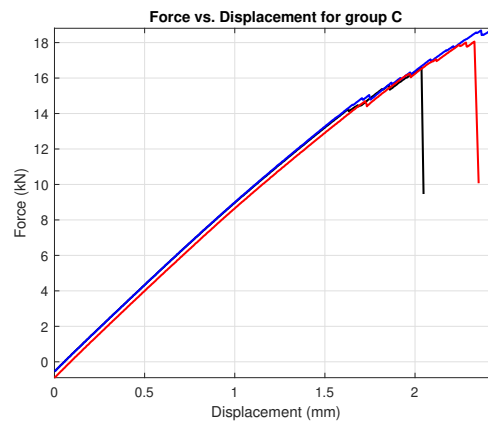


Figure 5.10: Results of tensile tests for 2mm thick specimens of Type I with $0^{\circ}/90^{\circ}$ fiber angles (group C).



Figure 5.11: Failed specimen of group C.

Group C

Results for group C, shown in Figure 5.10, are identical to those for group A. This fact confirms that the material has the same behaviour along the 0° and 90° directions as a result of the manufacturing of the specimens. The Young modulus is equal to 62.65GPa, which is very similar to that from group A. Also characteristics of failed specimens are the same as in group A (see Figure 5.11).

Group D

Specimens from group D exhibit the same behaviour of groups A and C, with no plastic deformation and brittle failure (see Figure 5.12). Due to the different cross section, failure occurs at around 11kN. The Young modulus, computed from the stress vs. strain curves, is approximately equal to 86.02GPa, which is greater than that obtained for groups A and C. This can be explained considering the different fiber densities of the specimens due to the different manufacturing. One of the failed specimens is shown in Figure 5.13. The type of failure is very similar to A and C specimens as they all have the same fiber orientations, however failure occurred in the gage section because of the narrower section of the specimens. Delamination effects are clearly visible by looking at the side of the specimen as shown in Figure 5.14.

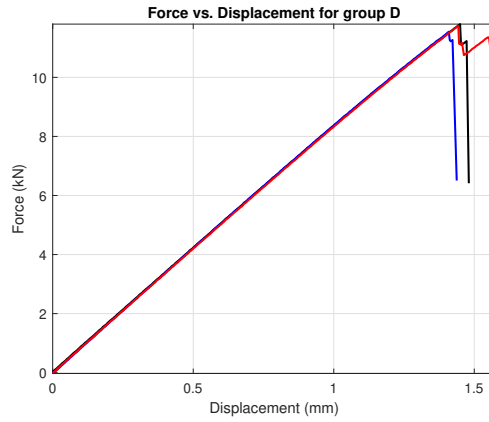


Figure 5.12: Results of tensile tests for 3mm thick specimens of Type II with $0^\circ/90^\circ$ fiber angles (group D).



Figure 5.13: Failed specimen of group D.



Figure 5.14: Side view of a failed specimens of group D with evident delamination of the layers.

Group E

Results for group E are shown in Figure 5.15. Since these specimens have unidirectional fiber reinforcement, they have a much higher modulus of elasticity and strength. Therefore it was not possible to break the specimens. In the first tests the specimens slipped as the pressure of the grips was not high enough. By increasing the grip pressure slip problems were partially solved but the specimens failed in correspondence of the grips (see photos of tested specimens of group E in Appendix A). For this tests only the first elastic part of the force vs. displacement curve is valid. The Young modulus can be computed and it is about 160.3GPa, which is very close to the value claimed by the manufacturers of the material.

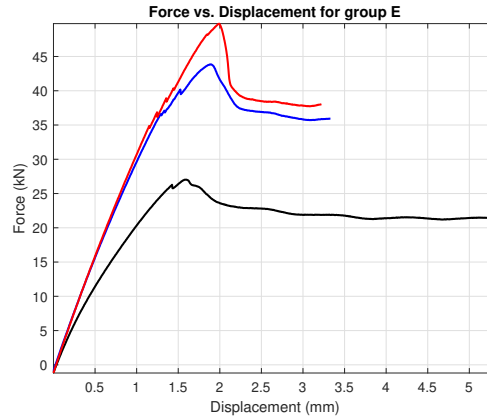


Figure 5.15: Results of tensile tests for 2.8mm thick specimens of Type I with 0° fiber angle (group E).

Group E'

Results of group E' (Figure 5.16) are very similar to the previous ones, since they have very similar thicknesses. Also in this case there were slip problems of the specimens and only one specimen, shown in Figure 5.17, failed at around 44kN (black line in the graph). The Young modulus computed from these tests is 159,61GPa, which is coherent with results of group E, since the tested material is the same.

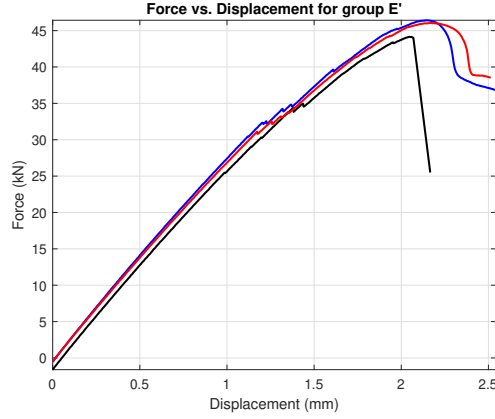


Figure 5.16: Results of tensile tests for 2.4mm thick specimens of Type I with 0° fiber angle (group E').



Figure 5.17: Failed specimen of group E'.

Comparison of all groups

In order to compare different behaviours of the specimens, all the experimental curves obtained are plotted together in Figure 5.18. It can be noticed that groups A, C and D show the same behaviour apart from the maximum supported load which depends on the cross section of the specimens. This is reasonable since these specimens have almost the same characteristics in terms of manufacturing process and fiber direction. The same material, but with 45° fiber direction, exhibits a very different behaviour with evident plastic deformation and smaller maximum load. Specimens from groups E and E', which were cut from unidirectional fiber reinforced pultruded profiles, have a much greater modulus of elasticity. Unfortunately, no information can be gathered from the preformed tests about the plastic and failure behaviours. However, looking at the one specimen from group E' which failed, it seems that the behaviour is similar to groups A, C and D, with very small plastic deformations and brittle failure. The uni-directionality of the reinforcement gives a more rigid and strong material than bi-directional reinforcement.

Finally, a summary of the moduli of elasticity computed for each test is given in Table 5.3.

Group	A	B	C	D	E	E'
E (GPa)	63.73	17.68	62.6	86.025	160.30	159.62

Table 5.3: Summary of Young's moduli obtained from tensile tests.

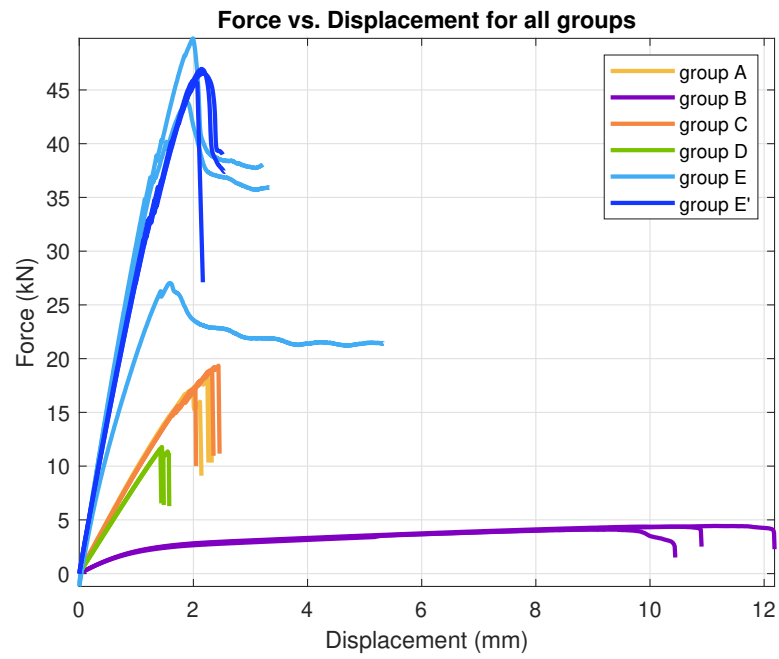


Figure 5.18: Results of all tests for comparison.

Chapter 6

Numerical model

6.1 Introduction

In this chapter the numerical model used to reproduce experimental tests is described and information about geometry, mesh, boundary conditions and other features of the model are given. Detailed informations are given about the material models which will be used in the next chapter for the optimization.

The simulations were performed using the software LS-DYNA, which is a finite element solver. It is mainly used for explicit analyses, but it can perform implicit analyses too. In order to perform a simulation the solver requires an input file in which all features and characteristics of the model have to be specified. Each feature is inserted using particular predefined keywords which are listed and explained in the LS-DYNA Keyword User's Manuals [10][11]. Details on theoretical notions about the solver can be found in [12].

Since LS-DYNA does not require any specification for the units of measure of the model, it is important to express all quantities in the same system. The units of measure used in all the simulations are given in Table 6.1.

Quantity	Time	Length	Force	(Stress)	(Density)
Unit	s	mm	N	(MPa)	(ton/mm ³)

Table 6.1: Units of measure used in all simulations. Quantities between brackets are derived from previous ones.

6.2 General settings

6.2.1 Geometry and mesh

The geometry and mesh used for simulations are shown in Figures 6.1 and 6.2. The geometry of the two models was generated in accordance with the tested specimens, by following the specifications in the ASTM D638 standard. The mesh of the wide (Type I) specimen has 927 nodes and 816 elements, while the narrow (Type II) one has 1143 nodes

and 1008 elements¹. All elements are quadrilateral shell elements, therefore the specimens are modeled as plane shells and the information about the actual thickness is given in another part of the input file (*SECTION_SHELL or *PART_COMPOSITE).

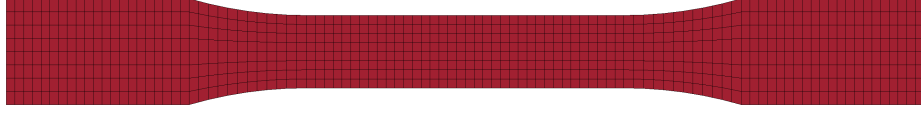


Figure 6.1: Type I specimen geometry and mesh.

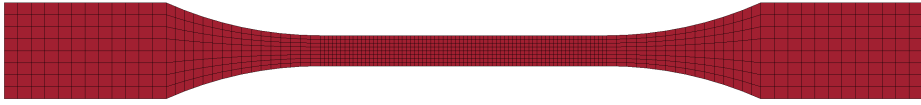


Figure 6.2: Type II specimen geometry and mesh.

6.2.2 Boundary conditions

The boundary conditions were assigned in accordance with the experimental tests. In particular nodes on the right side of the model were fixed in all directions, while nodes on the left side were assigned a displacement linearly increasing over time. The final value of the displacement was chosen based on the maximum displacement observed during the tensile tests for each group of specimens. In particular, maximum displacements of 3mm and 12mm were chosen for the 0° and 45° tests, respectively.

6.2.3 Numerical method

All simulations were performed using an implicit solver with adaptive time step. Since rate effects were not considered and the needed output of the simulations is a crossplot between displacement and force histories, all simulations were performed using a non-dimensional time. In particular, the end time was set to 1.

Among all element formulation implemented in LS-DYNA for shells, it was chosen to use a fully integrated (4 points per integration point) element formulation (ELFORM=16) because of its accuracy and cost. In a preliminary study of the optimization process, an under integrated (1 point per integration point) formulation (ELFORM=2) was used in order to decrease the cost and time of the simulation. Five through-thickness integration points were used in the single layer model, while one integration point per layer was used in the multiple layer model.

¹Finer meshes were available for both geometries, however they were not used since the coarse ones already gave good results and led to less computationally expensive simulations.

6.3 Material models

Apart from the boundary conditions, the physics of the problem is mainly described by the constitutive law of the material. The choice of the material model obviously depends on the actual material tested, however in many case it is not clear which model should be used, especially for composite materials which exhibits very different behaviours based on their composition.

In LS-DYNA hundreds of different material models are implemented for a variety of applications. However, only few of them are suitable for modeling composite materials. Common models used for composites are MAT22 (Composite damage), MAT54/55 (Enhanced composite damage) and MAT58 (Laminated composite fabric), although other models are implemented that might be more suitable for specific applications such as modeling of sandwiched composites, laminated glass or composite layups. During a preliminary study, material models were searched based on two essential requirements:

1. the material had to be anisotropic (or at least orthotropic), and
2. it should exhibit non-linear behaviour induced by some plasticity model or by some failure model.

Many materials were found to meet these two requirements, however the majority of them were discarded because of incompatibilities with the type of analysis performed. Eventually, MAT54 and MAT58 were chosen for parameter identification. Detailed information on these models are given in sections below.

6.3.1 Material directions

Since composites materials often exhibit anisotropic mechanical properties, it is important to correctly model the material by specifying its directions of anisotropy. This is done in LS-DYNA by using the AOPT (Axes Options) parameter, which allows to define material directions for each element of the mesh. The AOPT value is an integer varying between 0 and 4 and each option requires some input vector. For shells only options 0, 2 and 3 can be used. In this study, AOPT=2 has been chosen since the directions of orthotropy can be specified globally (i.e. they are the same for every element) and do not change during the simulation.

For option AOPT=2 the only input needed is vector \mathbf{A} (see material cards in Table 6.2), which defines the first axis of orthotropy (labeled with a), after being projected onto the plane of the element. The third direction, c , is obtained from the element normal vector, while the second direction, b , is computed from directions a and c so as to obtain an orthogonal system of reference: $b = c \times a$ (see Figure 6.3).

An additional control for specifying the material directions of the elements is given by the parameter BETA (actually called MANGLE in MAT54). If a non-zero value of BETA is specified, then the (a, b, c) reference system is rotated around the c direction of exactly BETA degrees.

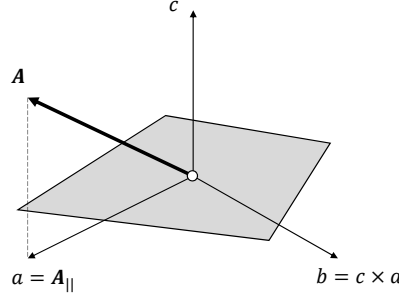


Figure 6.3: Construction of material directions for a single shell element using AOPT=2 with input vector \mathbf{A} .

6.3.2 Enhanced composite damage (MAT54)

Material 54, called Enhanced composite damage, is an extension of Material 22 (Composite damage)². It models an orthotropic linear elastic material with additional parameters for failure and damage.

The elastic constants of the material are Young's moduli in the 3 principal directions (EA, EB, EC), Poisson's ratios (PRBA, PRCA, PRCB) and shear moduli (GAB, GBC, GCA). The model is implemented both for solid and shell elements. Since all simulations done in this study concern shell elements, elastic constants EC, PRCA and PRCB will not be used.

In this model failure criteria are implemented both for fiber tension and compression and for matrix tension and compression. For material model MAT54 the failure is governed by the Chang-Chang criteria, in which strengths XC, XT, YC, YT and SC appear. The exact formulation of Chang-Chang is expressed in terms of normalized coefficients as follows:

- For fiber tension ($\sigma_{aa} > 0$):

$$e_{ft} = \left(\frac{\sigma_{aa}}{X_t}\right)^2 + \beta \left(\frac{\sigma_{ab}}{S_c}\right)^2 \quad (6.1)$$

- For fiber compression ($\sigma_{aa} < 0$):

$$e_{fc} = \left(\frac{\sigma_{bb}}{X_c}\right)^2 \quad (6.2)$$

- For matrix tension ($\sigma_{bb} > 0$):

$$e_{mt} = \left(\frac{\sigma_{bb}}{Y_t}\right)^2 + \left(\frac{\sigma_{ab}}{S_c}\right)^2 \quad (6.3)$$

- For matrix compression ($\sigma_{bb} < 0$):

$$e_{mc} = \left(\frac{\sigma_{bb}}{2S_c}\right)^2 + \left[\left(\frac{Y_c}{2S_c}\right)^2 - 1\right] \frac{\sigma_{bb}}{Y_c} + \left(\frac{\sigma_{ab}}{S_c}\right)^2 \quad (6.4)$$

²Material 22 is an orthotropic elastic material with failure criteria for matrix tension/compression and fiber tension and no damage parameters.

In all equations above, σ_{**} are the components of the stress tensor with respect to the coordinate system (a, b, c) defined by AOPT. If any of the quantities e_{**} exceeds 1, then the element fails, otherwise it behaves elastically.

Element deletion can be controlled through DFAILx parameters which represent the maximum strain the element can undergo in different directions. After an element has been deleted, it is possible to simulate damage of adjacent elements using other parameters, namely SOFT, FBRT and YCFAC.

In this study only XT, DFAILT and DFAILC have been considered in order to simulate the failure of the specimen. In particular DFAILC was included only for numerical reasons, because the solver needs a non-zero value of DFAILC when DFAILT is not zero. Since compressive failure was not investigated, DFAILC was chosen to be equal and opposite to DFAILT.

The general behaviour of the model MAT54 (as used in this study) is shown in Figure 6.4. These particular results were obtained from a 0° load case on the Type I specimen. The material is linear elastic until the maximum stress XT is reached. At this point the behaviour varies based on the value of DFAILT. If DFAILT is approximately equal to XT/EA, the specimen is not capable of supporting additional stress and fails. For higher values of DFAILT the stress remains constant until failure occurs. It is worth mentioning that the actual failure of the specimen depends on DFAILT, as it is shown in Figure 6.5. While the specimen at the top has failed, the specimen at the bottom has not due to the larger value of DFAILT. However, a large concentrated deformation is evident at the point where the specimen would have failed if DFAILT was set to a smaller value. Therefore, even for high values of DFAILT it is possible to identify where and when the specimen fails by looking at the strain plot.

MID	R0	EA	EB	EC	PRBA	PRCA	PRCB
1	1.6E-9	&E	&E		&PR		
GAB	GBC	GCA	KF	AOPT	2WAY		
&G	&G	&G		2			
XP	YP	ZP	A1	A2	A3	MANGLE	
			1.0	0.0	0.0	0.0	
V1	V2	V3	D1	D2	D3	DFAILM	DFAILS
TFAIL	ALPH	SOFT	FBRT	YCFAC	DFAILT	DFAILC	EFS
					&DFAILT	-&DFAILT	
XC	XT	YC	YT	SC	CRIT	BETA	
	&XT				54		
PFL	EPSF	EPSR	TSMD	SOFT2			
100.0							
SLIMT1	SLIMC1	SILMT2	SLIMC2	SLIMS	NCYRED	SOFTG	

Table 6.2: Cards of material 54. Values preceded by & corresponds to parameterized variables, while empty options are not used in this work.

Variable name	Meaning
MID	Material identification number
R0	Material density
EA, EB	Young's moduli in a - and b -direction
PRBA	Minor Poisson ratio in plane ab
GAB, GBC, GCA	Shear moduli
AOPT	Material axes option
XP, YP, ZP	Point P for axes option
A1, A2, A3	Vector A for axes option
MANGLE	Material angle for axes option
V1, V2, V3	Vector V for axes option
D1, D2, D3	Vector D for axes option
DFAILM	Maximum strain for matrix tension and compression
DFAILS	Maximum shear strain
TFAIL	Minimum time-step for element deletion
ALPH	Weighting factor for non-linear shear stress component
SOFT	Damage parameter
FBRT	Damage parameter
YCFAC	Damage parameter
DFAILT	Maximum strain for fiber tension
DFAILC	Maximum strain for fiber compression
XC, XT	Longitudinal compressive and tensile strengths
YC, YT	Transverse compressive and tensile strengths
SC	Shear strength
CRIT	Failure criterion
BETA	Weighting factor for fiber tensile failure (see Equation 6.1)

Table 6.3: Variables of material card MAT54.

6.3.3 Laminated composite fabric (MAT58)

Material 58 is called Laminated composite fabric. It is an orthotropic linear elastic material with non-linear effects introduced by the modeling of failure and damage. As for MAT54, the elastic constants are the Young's moduli, the Poisson's ratios and the shear moduli; however, EC, PRCA and PRCB are unused by the solver since this model is only implemented for shell elements.

Three failure model are implemented which can be chosen using the flag parameter FS (Failure Surface):

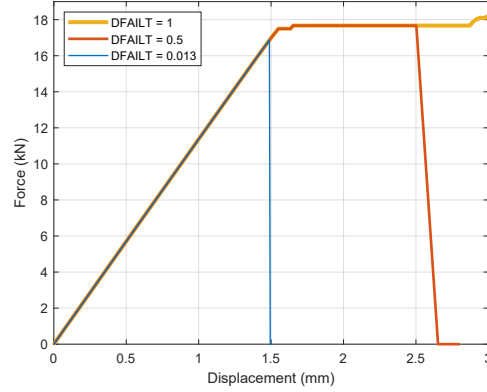


Figure 6.4: Force vs. displacement curves from simulations on MAT54 by varying the value of DFAILT. The strength XT is set to the constant value of 700MPa, while the longitudinal elastic modulus EA is equal to 62GPa.

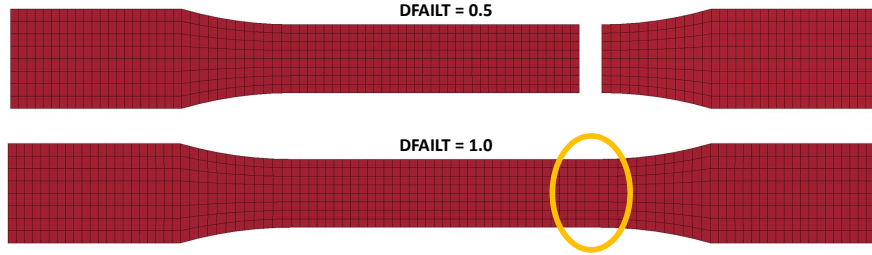


Figure 6.5: Effects of DFAILT on the element deletion. For low values of DFAILT elements are deleted while for high values of DFAILT elements are not deleted but they locally deforms in correspondence of the failure point.

- $FS=1$ corresponds to a smooth failure surface with a quadratic criterion for longitudinal and transverse directions and it is suited for laminated and fabrics;
- $FS=0$ corresponds to a smooth failure surface in the transverse direction only and it is mainly suited for composites with unidirectional fiber reinforcement;
- $FS=-1$ corresponds to a faceted failure surface with optional controls for shear behaviour and it is suited for laminates and fabrics as the first one.

Since in this study it has been chosen to model each layer of the composite material with equal properties in both longitudinal and transverse directions, failure surfaces 1 and -1 have been investigated. In particular, it was observed how the failure surface modifies the output of the simulation for the 45° load case. The difference, shown in Figure 6.7, is evident since the selection of $FS=-1$ introduces two extra parameters ($TAU1$, $GAMMA1$) in order to modify the shear stress vs. shear strain curve. By varying other parameters related to shear behaviour, it was observed that MAT58 with $FS=-1$ was able to qualitatively reproduce the experimental behaviour observed in the 45° load case, therefore this failure option was used for all optimization runs in the next chapter.

MID	RO	EA	EB	EC	PRBA	TAU1	GAMMA1
1	1.6E-9	&E	&E		&PR	&TAU	&GAMMA
GAB	GBC	GCA	SLIMT1	SLIMC1	SLIMT2	SLIMC2	SLIMS
&G	&G	&G	0.01	1.0	0.01	1.0	1.0
AOPT	TSIZE	ERODS	SOFT	FS	EPFS	EPSR	TSMD
2		0.0		-1			
XP	YP	ZP	A1	A2	A3	PRCA	PRCB
			1.0	0.0	0.0		
V1	V2	V3	D1	D2	D3	BETA	
						0.0	
E11C	E11T	E22C	E22T	GMS			
	&EPS		&EPS	&GMS			
XC	XT	YC	YT	SC			
	&XT		&XT	&SC			

Table 6.4: Cards of material 58. Values preceded by & corresponds to parameterized variables, while empty options are not used in this work.

Similarly to MAT54, strength values (XC, XT, YC, YT, SC) can be defined for MAT58 for both fiber and matrix tensile and compressive modes, as well as for shear stress. In addition, the strains which corresponds to those strengths can be defined (E11C, E11T, E22C, E22T, GMS). After the stress reaches the correspondent strength value, its evolution is governed by the stress limit parameters³, namely SLIMxy. Stress limits are used to model the fact that layers can support a percentage of stress at failure after they have failed. The effects of these parameters are shown in Figure 6.6 for the 0° load case and in Figure 6.7 for the 45° load case. Considering the experimental behaviour of the material, it was chosen to use a low value⁴ for SLIMT1 and SLIMT2, while the other stress limits where set to 1.

Finally, in order to avoid actual failure and deletion of elements, ERODS was set to 0. This was done for speeding up the simulation and avoid numerical problems after failure of the specimen, similarly to MAT54.

6.3.4 Modeling of laminated composites

In order to use a certain material model in LS-DYNA, it is basically required to define a part (with the keyword *PART) to which the material model and the element formulation will be assigned. In this way, the material is modeled as a single layer with thickness specified in the keyword *SECTION_SHELL. However, composite materials are often made of multiple layers with possibly distinct properties. In this case, a single layer model could not be able to correctly reproduce the real behaviour of the material, especially in conditions when different numbers of layers are considered. The definition of a part with multiple layers can be done in LS-DYNA in several ways, but the suggested one is to use

³Stress limits were also implemented in MAT54, but they were not used there for numerical stability problems.

⁴Ideally 0, but this was not recommended in the LS-DYNA User's Manual, therefore both stress limits for tensile mode were set to 0.01.

Variable name	Meaning
MID	Material identification number
RO	Material density
EA, EB	Young's moduli in <i>a</i> - and <i>b</i> -direction
PRBA	Minor Poisson ratio in plane <i>ab</i>
TAU1	Stress limit of the first slightly non-linear part of the shear stress vs. shear strain curve
GAMMA1	Strain limit of the first slightly non-linear part of the shear stress vs. shear strain curve
GAB, GBC, GCA	Shear moduli
SLIMT1, SLIMC1	Stress limit factor for fiber tension and compression
SLIMT2, SLIMC2	Stress limit factor for matrix tension and compression
SLIMS	Stress limit factor for shear stress
AOPT	Material axes option
TSIZE	Time step for automatic element deletion
ERODS	Maximum effective strain for element layer failure
SOFT	Softening factor for strength in crashfront
FS	Flag for failure surface type
EPSF	Damage initiation transverse shear strain
EPSR	Final rupture transverse shear strain
XP, YP, ZP	Point P for axes option
A1, A2, A3	Vector A for axes option
V1, V2, V3	Vector V for axes option
D1, D2, D3	Vector D for axes option
BETA	Material angle for axes option
E11C, E11T	Strain at longitudinal compressive and tensile strengths
E22C, E22T	Strain at transverse compressive and tensile strengths
GMS	Shear strain at shear strength
XC, XT	Longitudinal compressive and tensile strengths
YC, YT	Transverse compressive and tensile strengths
SC	Shear strength

Table 6.5: Variables of material card MAT58.

the keyword `*PART_COMPOSITE`, in substitution of both `*PART` and `*SECTION_SHELL`.

In `*PART_COMPOSITE` it is possible to define an arbitrary number of layers with different material models, thicknesses and material angles. For example, in order to model the Type I specimen, 4 layers can be defined with a thickness of 0.5mm (so as to sum up to 2mm), as it is shown in Table 6.6. Using the available informations on the manufacturing

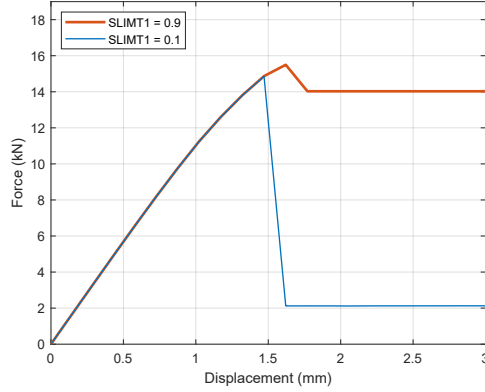


Figure 6.6: Force vs. displacement curves from simulations on MAT58 (0° load case) with FS=1, by varying SLIMIT1.

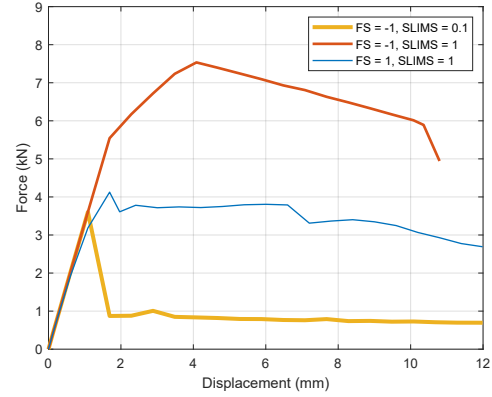


Figure 6.7: Comparison between FS=1 and FS=-1 for the 45° load case, with variation of SLIMS.

of the specimen, the two external layer and the two internal ones are assigned with two different material types and all layers have the same principal direction. The advantage of using the keyword `*PART_COMPOSITE` will be evident in the next chapter during the optimization of material parameters.

One important difference between the single layer and multiple layer definitions of a material relies in the failure behaviour. In fact, when using multiple layers with different properties it could happen that one layer fails before another does. In this cases, the material has partially failed but it is still capable of undergoing additional stress, until all layers fail. For some material, such as MAT54, it is possible to specify the percentage of layers that have to fail in order to completely delete the shell element, through the parameter PFL (see Table 6.2).

PID	ELFORM	SHRF	NLOC	MAREA	HGID	ADOPT	ITHELFRM
40	16						
MID1	THICK1	B1	TMID1	MID2	THICK2	B2	TMID2
1	0.5	0.0		2	0.5	0.0	
MID3	THICK3	B3	TMID3	MID4	THICK4	B4	TMID4
2	0.5	0.0		1	0.5	0.0	

Table 6.6: Cards of keyword `*PART_COMPOSITE`.

Chapter 7

Parameter optimization

In this chapter all results obtained from LS-OPT optimization runs are presented. A preliminary study was performed on a simplified problem in order to understand how different settings in LS-OPT affect the optimal result. Then, some parameters of material models MAT54 and MAT58 were optimized using different approaches with the objective of reproducing experimental results previously obtained in Chapter 5. Referring to notation used in Chapter 5, parameter identification has been performed on specimens from groups A, B, C and D, while results of group E and E' were discarded because of their inconsistency due to problems already described.

7.1 Preliminary study

7.1.1 Setup

The material model used for all simulations in this preliminary study was MAT58, modeled with 4 layers using the keyword `*PART_COMPOSITE` (see Section 6.3.4). Since in this stage the objective is not the optimization of model's parameter, it was chosen to use only 6 significant parameters, whose baseline values and ranges are given in Table 7.1: `E` is the Young's modulus, `XT` is the longitudinal tensile strength and `EPS` (actually named `E11T` in the material card, see Table 6.4) is the strain correspondent to the strength value; the first and second sets of parameters refer to the external and internal layers, respectively. With these settings the duration of each simulation was about 30 seconds. The complete material cards used for these simulations are given in Appendix B, where the exact values of other parameters can be found.

In total, 6 different configurations were tested, labeled with letters from A to F. Since the objective of parameter identification is to find a unique optimal value of the parameters, only Metamodel-based Optimization tasks were considered and the Sequential with Domain Reduction strategy was chosen, with a maximum of 10 iterations. The detailed settings used for each configuration are given below.

Par.	Baseline	Min	Max
E	8E4	2E4	8E4
EPS	0.01	0.01	0.03
XT	600	400	800
E2	8E4	4E4	1E5
EPS2	0.01	0.01	0.03
XT2	1000	800	1500

Table 7.1: Parameters used for optimization runs in the preliminary study.

Configurations A, B and C

In configuration A a linear polynomial metamodel was considered and design points were generated using the D-optimal criterion. 11 points per iteration were used to build the metamodel as this is the default number given by the software. Partial Curve Mapping, with 50 points in total, was used to compare experimental and simulated curves and the actual metamodel optimization was performed using the hybrid ASA/LFOP algorithm.

Configurations B and C are the same as A, except for the choice of the metamodel and the sampling technique. In B a quadratic polynomial metamodel was built using 43 D-optimal points per iteration. In C, Radial Basis Functions with 11 space filling designs were used. Since in this specific problem only histories were defined as output from simulations, there were not other choices for the type of metamodel used to approximate histories.

Configuration D

Configuration E is the same as A, but the MSE curve matching composite was used instead of the Partial Curve Mapping algorithm. This configuration was used to see how the choice of the curve matching composite affects the optimization run both in terms of cost and accuracy of the result.

Configuration E

Configuration F is identical to A, but the optimization algorithm was changed to the hybrid GA/LFOP.

Configuration F

Configuration G is the same as A, but all points from the experimental curve (692) were used to compute the curve matching composites, instead of a fixed number of 50.

7.1.2 Results

Results of the 6 configurations have been compared both in terms of computational time and accuracy of the optimized curve.

Computational times are listed in Table 7.2, as output from LS-OPT. They are divided into subcategories based on the individual steps of the optimization procedure and they are discussed in the following:

- Sampling: sampling times varies a lot depending on the number of total designs generated during all iterations and on the type of sampling technique. It can be seen that D-optimal designs (in A, D, E and F) take less time than space filling ones (in C); in particular it is possible to compute that a single D-optimal design costs about 0.3s against 0.7s for one space filling design. Sampling time for configuration B is obviously much greater than other cases since more points had to be generated.
- Simulation: all simulation times are, as expected, approximately equal to the time of one simulation multiplied by the total number of simulations (indicated as Function Evaluations in Table 7.2).
- Metamodel: the time spent for building the metamodel is approximately equal for all configurations and it is a small percentage of the total time. Since there were not any scalar response, but only histories, defined as output of the simulations, the metamodel are only used to approximate histories and to compute the curve matching composite. Therefore, the actual time spent for building the metamodel is included in the composite computation time (which is automatically included in the Optimization time, see below);
- Optimization: configurations A, B and C show similar times since the same algorithm (ASA/LFOP) was used for all of them. Optimization for configuration D took much less time with the same algorithm. This suggests that computation of composites is included in the Optimization time, since in configuration D the MSE composite, which is much less expensive than PCM, was used. This fact is confirmed by configuration E, in which almost 700 points were used to compute the curve matching composite. Finally, a comparison between configurations A and F shows how Genetic Algorithms take more time than ASA, as expected.

In terms of accuracy, all configurations lead to similar results, except for configurations C and F, as it can be seen by looking at the objective value in Table 7.2. It is important to notice that in almost all cases, the optimal curve was not obtained after the last iteration, because of the non-monotonicity of the optimization history. For example, for configuration A (see Figure 7.1) the optimum was reached at iteration 8 (out of 10). This was an unexpected aspect of the Sequential with Domain Reduction optimization strategy and it has to be kept in mind for future optimizations.

The optimized curves for all configurations are shown in Figure 7.2 together with the experimental curve used as the objective. It is evident that all configurations (except maybe for C and F) lead to very similar results, although the optimized parameters are very dissimilar from each other. Optimized parameters, normalized to their initial range, are shown in Figure 7.3.

Since a large variability was observed in the optimized parameters in different configurations, it was decided to perform a Direct Optimization in order to find the actual set of optimal parameters. The results of this optimization are also displayed in Figure 7.3 and they are labeled as configuration Z. From these results it is clear that there is not

	A	B	C	D	E	F
Initialization	4.88	2.95	3.04	2.71	2.90	3.21
Sampling	36.60	129.17	86.53	34.69	36.11	38.09
Simulation	3405.8	14089	3454.6	3561.7	3468.4	3505.1
Result extr.	583.32	2081.7	572.72	542.56	565.3	566.47
Metamodel	2.15	3.09	2.20	2.12	2.19	2.11
Optimization	1034.5	974.5	912.18	333.45	2109	8312.8
Total	1h 25m	4h 48m	1h 28m	1h 15m	1h 43m	3h 27m
Objective	2.129E-4	2.315E-4	1.131E-3	2.781E-3	2.124E-4	3.555E-3
Fun. Eval.	111	431	111	111	111	111

Table 7.2: Computational time spent for the optimization in each configuration. The optimal curve matching value (Objective) and the total number of simulations (Function Evaluations) are also reported.

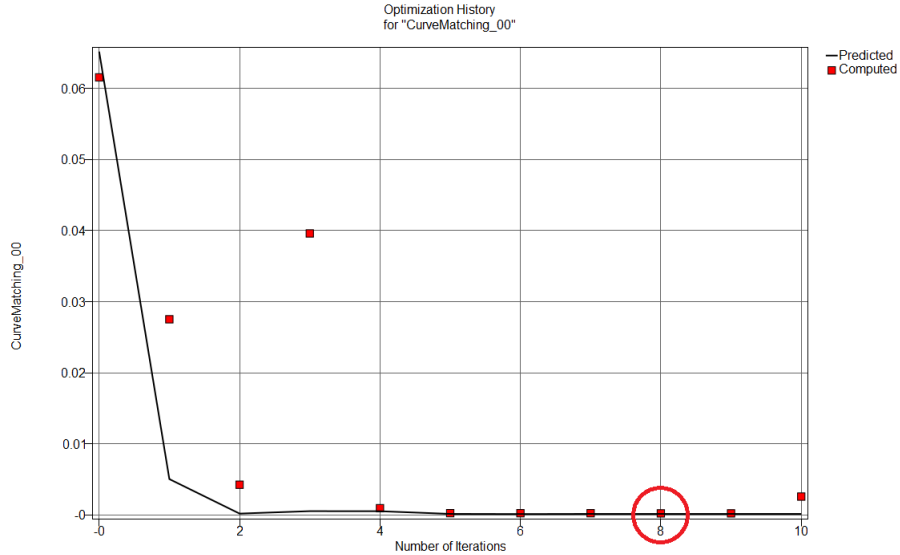


Figure 7.1: Optimization history of the curve matching composite for configuration A. The optimal value of the objective is not reached after the last iteration, but at iteration 8 instead.

one single set of optimal parameters, but many of them exist and different one are found depending on the algorithm and options used for the optimization process (as well as on the baseline point). However, all optimal points are equally good as long as experimental and simulated curves match sufficiently well. The presence of many optimal points could be related to the multiple layer model: since the model is described using more parameters, possible interactions between them could generate more optimal points.

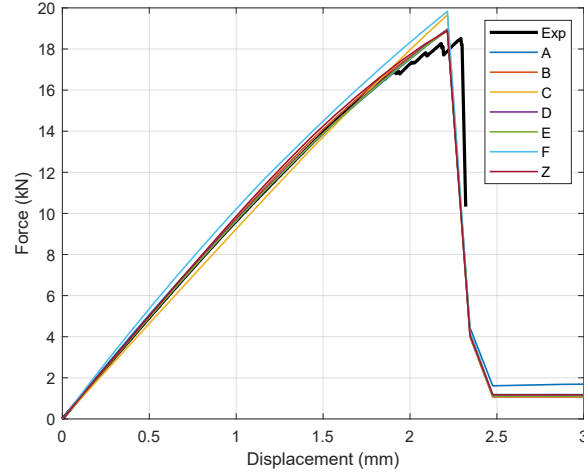


Figure 7.2: Optimized curves from all configurations A to G, together with the experimental curve and the curve resulting from direct optimization (configuration Z).

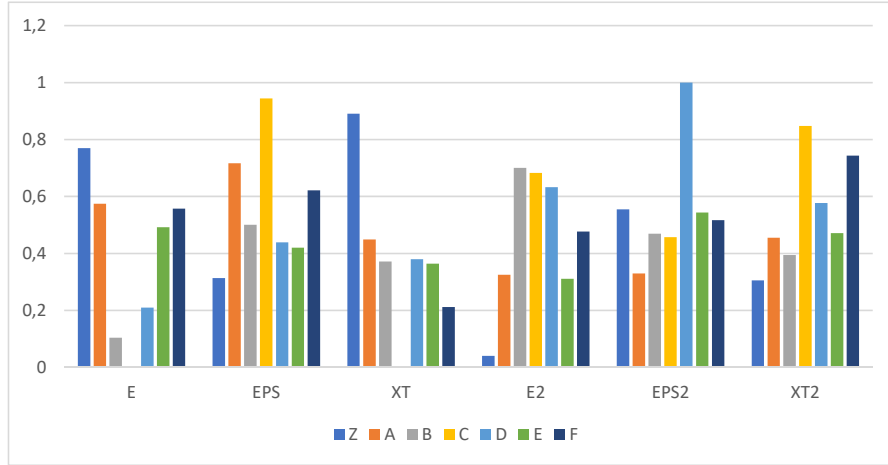


Figure 7.3: Optimal parameters (normalized to their initial ranges) obtained from configurations A to F and Z.

7.1.3 Conclusion

The main conclusion that can be gathered from previous results is that different meta-modeling and optimization techniques lead to different optimized parameters. However, for parameter estimation problems, the main objective is to find one set of optimal parameters which give good results when compared with experimental data and this is the case for almost all configurations used.

In the following, comments will be done on single options of the optimization procedure, based on the obtained results:

- Metamodel and Sampling: it seems that both polynomial metamodels and RBF networks lead to good results when coupled with their default sampling technique, i.e. D-optimal designs for polynomials and Space Filling designs for neural networks. The most efficient techniques were the linear polynomial and the radial basis functions, since the quadratic one needs too much simulations to be built. It is important to keep in mind that with sequential optimizations the actual optimum might not occur after the last iteration, as it has been the case for several configurations.
- Curve matching composite: the Partial Curve Mapping algorithm has produced better results compared to the Mean Square Error composite, although the latter one is less computationally expensive. For what concerns the number of points used to compute the composite, it is evident that too much points lead to more computational effort, without increasing the accuracy of the result. Therefore a relatively small number of points should be used.
- Optimization algorithm: both hybrid algorithms ASA/LFOP and GA/LFOP lead to good results in terms of accuracy, however GA is more time consuming than ASA, therefore the last one should be used.

It is important to emphasize that results of a parameter identification problems always depend on the specific problem analysed, on the number of parameters and on their baseline values and initial ranges. A complete study of optimization strategies and various LS-OPT settings should involve many problems (for example several load cases), and lots of different optimization runs are needed in order to deduce general results, independent from the specific problem that has to be solved.

7.2 Optimization of MAT54

All the following results (including those on MAT58) were obtained by performing a Metamodel-based Optimization with Sequential with Domain Reduction strategy (10 iterations in total). A linear polynomial metamodel was used to approximate histories in conjunction with D-optimal sampling (with the number of points varying based on the number of parameters). Partial Curve Mapping with 25 points was used to compute the curve matching composite and ASA/LFOP was chosen as the optimization algorithm.

Considering that many experimental curves were available for each load case, it was decided to consider only one of them (per load case) to compute the curve matching composite. In particular, the blue curves in Figure 5.6 and Figure 5.8 were used for the 0° and 90° load case on the Type I specimen, while the black curve in Figure 5.12 was used for validation in the 0° load case on Type II specimen.

7.2.1 Single layer model

Main load case

In the first stage of the optimization of MAT54, it was chosen to model the specimen as a single layer with equal properties along the longitudinal and transverse directions. Therefore, the Young's moduli in both directions were equal. Also, the shear moduli

in the three coordinate planes were assumed to be coincident. In summary, only three independent elastic constants were considered for the optimization, namely E , ν and G . In order to model the failure, parameters XT and $DFAILT$ were considered for the optimization. As already described in Section 6.3.2, XT is the longitudinal tensile strength and $DFAILT$ is the maximum strain that an element can undergo before being deleted from the mesh.

The first optimization run was then performed on parameters listed in Table 7.3. The analysis was done on the Type I specimen with fibers in the $0^\circ/90^\circ$ directions. Due to numerical instability, many of the simulations failed during the optimization process, which stopped before finishing iteration 8 since there were not enough usable simulations to build the linear metamodel. The optimal curve of iteration 7, which is shown in Figure 7.4, is not too far from the experimental curve, but it can certainly be improved.

Instead of re-running the optimization process with more simulations per iteration, it was decided to remove $DFAILT$ from the variable set to be optimized and to fix it to a high constant value ($DFAILT=10.0$). This drastically reduced the number of failed runs and also decreased the average time of each simulation, thus speeding up the entire optimization procedure. Due to the behaviour of MAT54 when $DFAILT$ has a high value (see Section 6.3.2), the experimental curve needed to be modified in order to help the curve mapping algorithm to capture the failure point. In particular, the last experimental point was removed and the second-last force value was repeated constantly up to the final displacement of 3mm. The modified experimental curve and the optimized simulated curve are shown in Figure 7.5 together with the Curve Matching value.

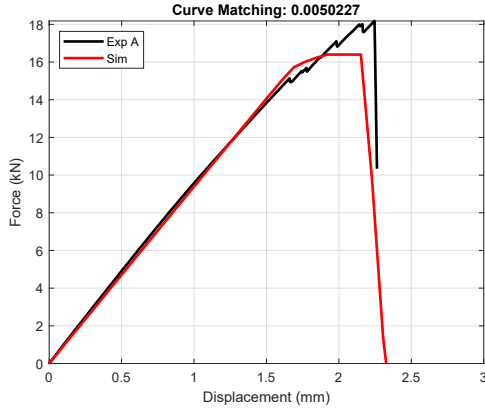


Figure 7.4: Optimized curve using both XT and $DFAILT$ in the variable set.

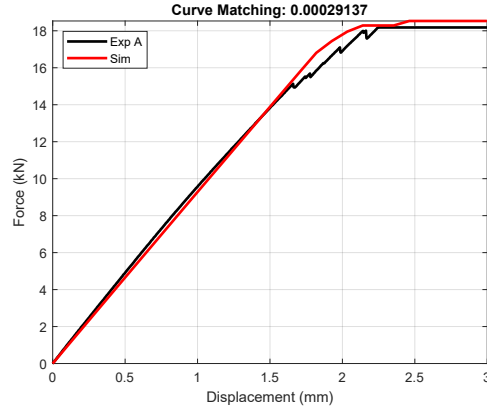


Figure 7.5: Optimized curve using only XT in the variable set, while fixing $DFAILT$ to a high constant value.

The optimized parameters for the first and second optimization are shown in Table 7.3. Recalling that the Young's modulus computed from experimental curves is around 6.2GPa for this load case, it can be noticed that the second optimization slightly underestimates it. This is due to the fact that the material model is perfectly linear until failure, while the experimental curve presents a small curvature even in the first part.

Par.	Baseline	Min	Max	1st Opt.	2nd Opt.
E	6E4	1E4	1E5	5.18E4	5.07E4
PR	0.100	0.010	0.500	0.318	0.096
G	5E3	5E2	1E4	6.43E3	3.39E3
XT	700	100	2000	792	735
DFAILT	0.008	0.001	1	0.133	–

Table 7.3: Parameters used for the first and second optimization runs on MAT54.

Validation on other load cases

In order to assess the validity of the optimal parameters of the second optimization, they were used to run simulations on other load cases, namely the tensile test on Type I specimen with 45° fibers and the tensile test on Type II specimen with 0° fibers.

Results of the former load case are shown in Figure 7.6. The first linear part of the experimental curve is perfectly fitted by the simulated curve, thus validating the accuracy of the elastic constants. However, the model is not capable of capturing the plastic behaviour of the material. This is actually a limitation of material model MAT54, which is perfectly linear until failure, at least when using a single layer model.

Results of the second validation load case are shown in Figure 7.7. In this case, the simulated curve is completely different (even in the elastic part) from the experimental one, suggesting that the single layer model is not able to reproduce the behaviour of specimens with different layer compositions, as it is the case for Type I and Type II specimens used in this study (see Section 5.2.2).

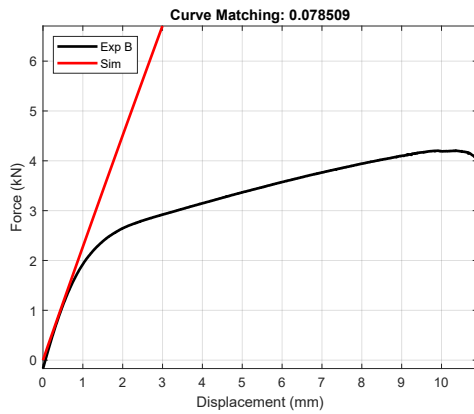


Figure 7.6: Validation of single layer MAT54 on the 45° test.

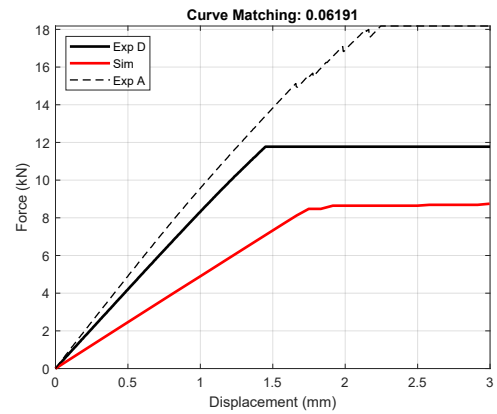


Figure 7.7: Validation of single layer MAT54 on the 6-layer (Type II) specimen.

7.2.2 Multiple layer model

First load case

In order to obtain a model that works well independently from the layer composition of the specimens, it seems necessary to model layers separately, which can be done using the `*PART_COMPOSITE` keyword as described previously in Section 6.3.4. The Type I specimen was modeled as a laminated composite made of 4 distinct layers. By following the specifications given by the manufacturer on the density of single layers, it was chosen to model each layer using MAT54 with equal properties along the longitudinal and transverse directions. Since there are two distinct type of layers (internal and external), two different material cards were defined in the LS-DYNA input file. Optimization was performed using 8 parameters in totals, which are listed in Table 7.4.

The optimized curve of the third optimization on MAT54, shown in Figure 7.8, is not very dissimilar from the previous run, although the matching is slightly better in the latter case: the curve matching composites have the same order of magnitude, but the current one is half the previous one. Optimal parameters, listed in the last column of Table 7.4, can not be compared with previous results due to the different modeling approach in this case. It is interesting to notice that $E2 > E$ and $XT2 > XT$, which seems correct since the internal layers of the specimen have a higher fiber density than the external ones.

Par.	Baseline	Min	Max	3rd Opt.
E	6E4	1E4	1E5	2.95E4
PR	0.100	0.010	0.500	0.215
G	5E3	5E2	1E4	4.65E3
XT	700	100	2000	546
E2	6E4	1E4	1E5	7.15E4
PR2	0.100	0.010	0.500	0.400
G2	5E3	5E2	1E4	2.54E3
XT2	700	100	2000	916

Table 7.4: Parameters used for the third optimization run on MAT54. The first half characterizes the two external layers, while the second half characterizes the internal ones.

Validation on other load cases

As in the single layer model, also for the multiple layer one two extra simulations were carried out to validate optimal parameters. The results of the 45° tensile test are shown in Figure 7.9 and are identical to the single layer case. This confirms that MAT54 is not capable of approximating the real plastic behaviour of the material even with multiple

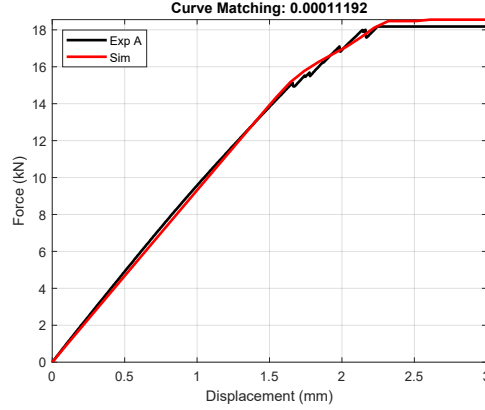


Figure 7.8: Optimized curve of multiple layer MAT54.

layer modeling¹.

Results of the tensile test on the Type II specimen are shown in Figure 7.10. In this case, the layered model is able to reproduce the experimental curve almost perfectly. It is important to notice that the strength variability observed in the experiments is not considered in the model, so that other experimental curves might be slightly less accurate after failure. However, the elastic part of the curve is always perfectly matched, confirming that the layered model can be used to model sheets with an arbitrary number of layers, as long as all layers are of the two types considered here.

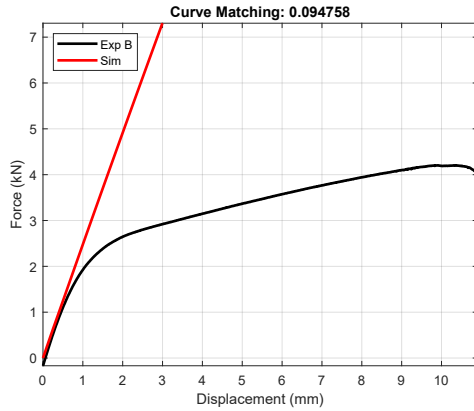


Figure 7.9: Validation of multiple layer MAT54 on the 45° test.

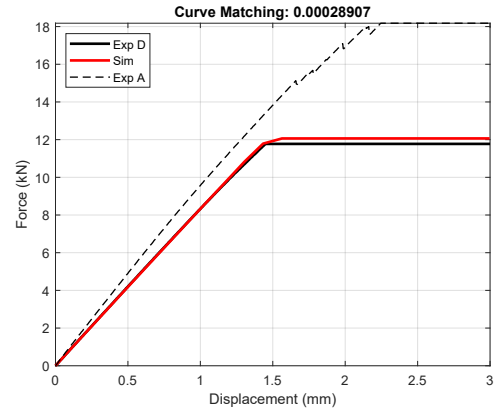


Figure 7.10: Validation of multiple layer MAT54 on the 6-layer (Type II) specimen.

It is important to notice that the specimen does not actually fail in the simulation

¹It might be possible to improve the behaviour of MAT54 in the 45° load case by considering additional failure parameters in order to accurately model the response of each layer and their interaction. However this should be investigated in depth.

since DFAILT is set to a high value, as already explained in Section 6.3.2. However it is possible to see when and where the specimen might have failed by looking at the strain distribution, which is shown in Figures 7.11 and 7.12 for the two specimens. Interestingly, the direction and location of the fracture are in accordance with the experimental results, confirming that the model MAT54 has the potential of correctly modeling not only the elastic behaviour but also the failure of the material when the load is applied parallel to one of the fiber directions.

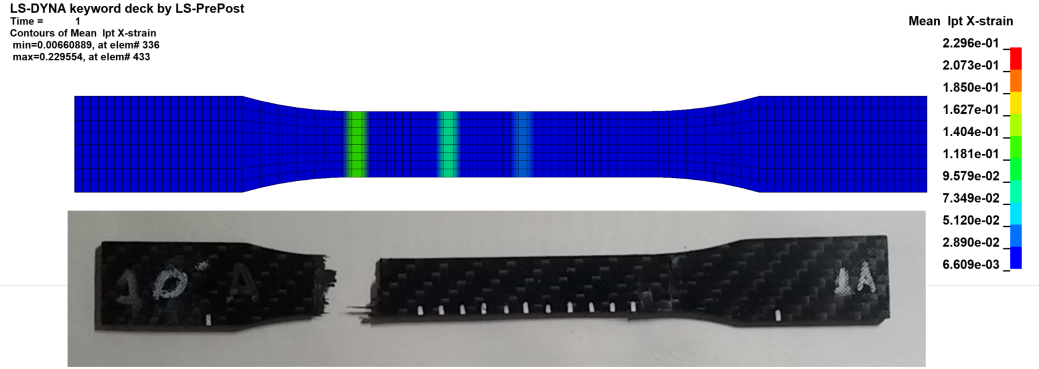


Figure 7.11: Longitudinal strain distribution of the Type I specimen with failed specimen of group A for comparison.

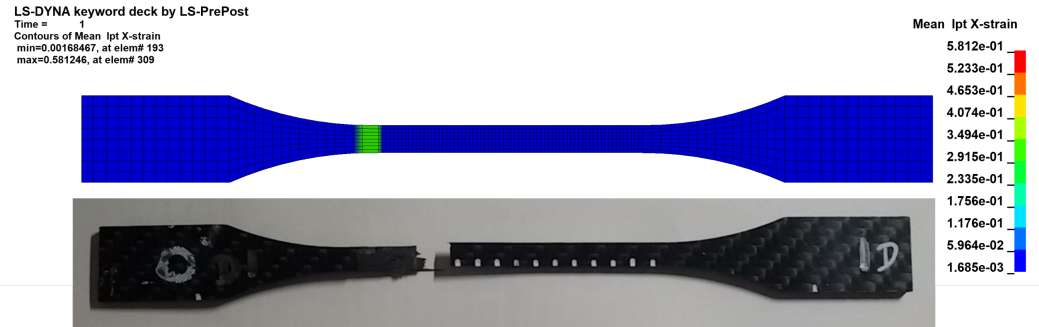


Figure 7.12: Longitudinal strain distribution of the Type II specimen with failed specimen of group D for comparison.

7.3 Optimization of MAT58

Considering the results obtained from the optimization of material MAT54, the first simulation on MAT58 was done using the optimized elastic constants E , ν , G and the optimized strength X_T (for both internal and external layers) as in Table 7.4. Other parameters were fixed to some non-optimized value and a multiple layer model was used. The result of this simulation (Figure 7.13) confirms once again the accuracy of the elastic constants, but it is evident that parameters related to failure have to be optimized.

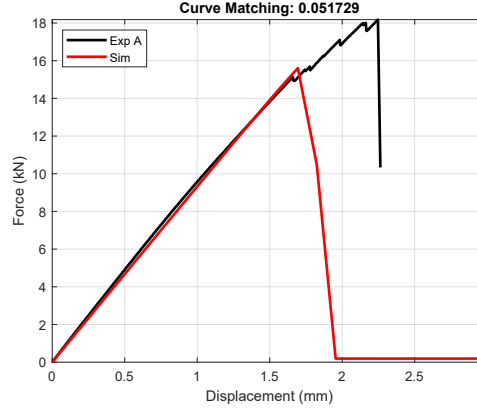


Figure 7.13: Validation of the optimized elastic constants of MAT54 on MAT58.

With the information obtained until this point, it was decided to perform optimization of MAT58 under the following assumptions:

- the material was always modeled using multiple layers;
- elastic constants of single layers were assumed to be equal to those previously obtained. Therefore, they were not included in the variable set for the following optimizations. This significantly reduced the number of parameters to be optimized and also the cost of the optimization.

The parameters that were chosen for the optimization are all related to failure and can be divided into two groups:

1. the first group includes EPS, XT, EPS2 and XT2 (4 parameters) which govern failure when load is applied in the longitudinal (or transverse) direction;
2. the second group includes TAU, GAMMA, GMS, SC, TAU2, GAMMA2, GMS2 and SC2 (8 parameters) which describe the plastic behaviour and the failure for shear strain and shear stress.

Details on the meaning of each parameter are given in Section 6.3.3. In total, 12 parameters have to be optimized and two different approaches can be followed². In the first one, two distinct optimization runs are performed for each group of parameters. The advantage of this approach is that a smaller number of simulation is carried out, but it is valid only as long as the two set of parameters are completely independent from each other, which is

²Actually, a third approach, involving Multilevel Optimization, is possible. In Multilevel Optimization two LS-OPT procedures are created: the first "outer" level optimizes the first set of parameters, while the second "inner" level optimizes the second one. These two optimizations are performed with alternating iterations, so that at each iteration one level uses the optimal parameters of the previous other level. The total cost of the optimization is the same as the two-steps approach, but the results are more likely to be similar to those obtained with one-step optimization.

rarely the case. In the second one, all parameters are optimized at the same time in a single optimization run. This could lead to better results since more simulations are performed than the first case and interactions between all parameters are considered. Both approaches have been studied in order to understand which is more effective and they are described in detail in the following sections.

7.3.1 Two-steps optimization

Main load cases

The first step of the optimization on MAT58 was done using 4 parameters, listed in Table 7.5. The 0° load case was considered since the chosen parameters govern the behaviour of the material in the fiber direction. The optimal curve, shown in Figure 7.14, is quite close to the experimental one and the material is able to capture the slight non-linearity in the curve. It is important to notice that the optimal curve was obtained at iteration 5, after which the computed curve matching value began to increase instead of decreasing.

Optimal parameters obtained in the first step were then set to constants and the second optimization was carried out on the other 8 parameters, listed in Table 7.6, using the 45° load case. The optimized curve is shown in Figure 7.15. The material model has been able to perfectly reproduce the first linear elastic part of the curve as well as the plastic part. This is mainly due to the presence of **GAMMAS** and **TAUs** parameters in the variable set. The last part of the curve is not fitted well, probably because of the parameters used in the optimization. It is possible (and should be investigated) that even the failure of the specimen could be modeled by introducing other parameters of MAT58 in the optimization.

The total number of simulations per iteration performed for the two-step optimization was 22, since 8 and 14 designs were used respectively for the first and second steps for each iteration.

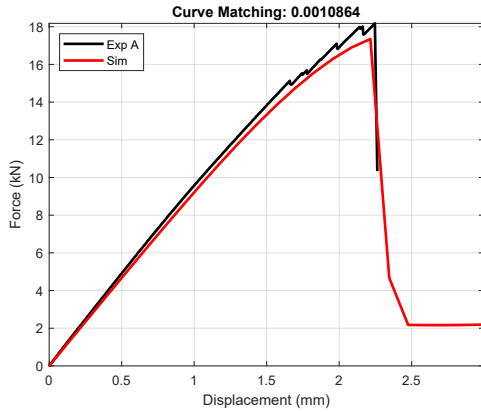


Figure 7.14: First optimized curve of the two-steps optimization. Notice that the optimum was reached after iteration 5 (out of 10).

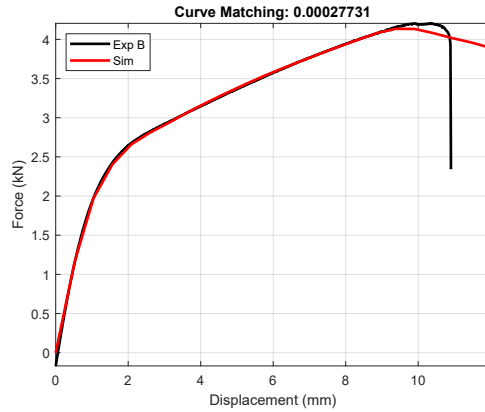


Figure 7.15: Second optimized curve of the two-steps optimization.

Par.	Baseline	Min	Max	1st Step Opt.
EPS	0.013	0.001	0.1	0.0511
XT	546	100	2000	973
EPS2	0.013	0.001	0.1	0.0164
XT2	916	100	2000	853

Table 7.5: Parameters used in the first step of the two-steps MAT58 optimization. Optimal values refer to iteration 5.

Par.	Baseline	Min	Max	2nd Step Opt.
GAMMA	0.013	0.001	0.100	0.0248
TAU	100	50	150	52
GMS	0.070	0.010	0.200	0.1639
SC	150	50	300	84
GAMMA2	0.013	0.001	0.100	0.0547
TAU2	100	50	150	53
GMS2	0.070	0.010	0.200	0.1815
SC2	150	50	300	91

Table 7.6: Parameters used in the second step of the two-steps MAT58 optimization.

Validation on other load cases

Since the variable set used in the first and second steps are not completely independent, it is important to see how the second optimization has modified the optimal curve for the $0^\circ/90^\circ$ load case. The modified simulated curve, shown in Figure 7.16, is slightly worse than the previously optimized one, suggesting that the one-step optimization might lead to better results.

Another validation run was performed on the Type II specimen and the result is shown in Figure 7.17. Also in this case, the fit is not as good as for MAT54, since the experimental curve is perfectly linear while the simulated curve slightly bends before failure.

The differences observed between MAT54 and MAT58 in these two validation runs are not really bad, considering the natural variability of the strength of the specimens observed in the experiments.

7.3.2 One-step optimization

Main load cases

In the one-step optimization all 12 parameters were optimized in the same process. In order to compare results of the two approaches, the same baseline values and initial ranges

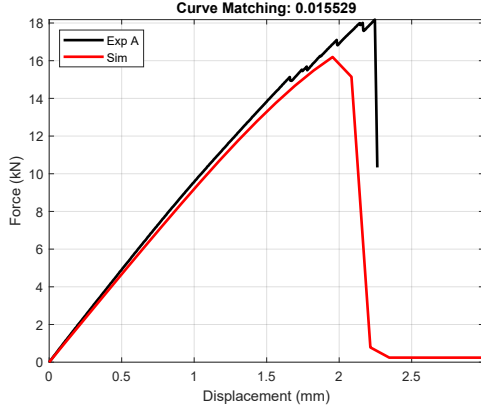


Figure 7.16: Validation of optimal parameters obtained from the second step on the first load case.

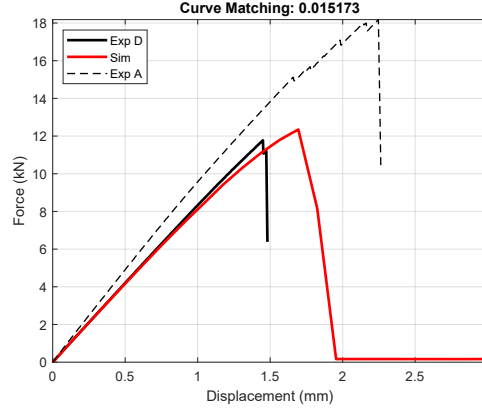


Figure 7.17: Validation of optimal parameters on the Type II specimen.

were considered for all parameters. The number of simulations per iteration was 40, which is much greater than the two-step optimization one. The results are shown in Figures 7.18 and 7.19 and should be compared with respective cases in Figures 7.16 and 7.15 which are already superimposed in current graphs for simplicity. It is evident that the one-step procedure is able to optimize all parameters better than the two-steps procedure, especially for the 0° load case. The 45° load case does not show big differences from the previous case in terms of accuracy of the simulated curve.

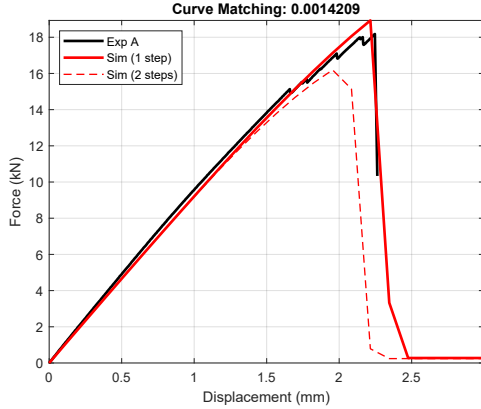


Figure 7.18: Optimized curve of the one-step optimization for the 0° load case.

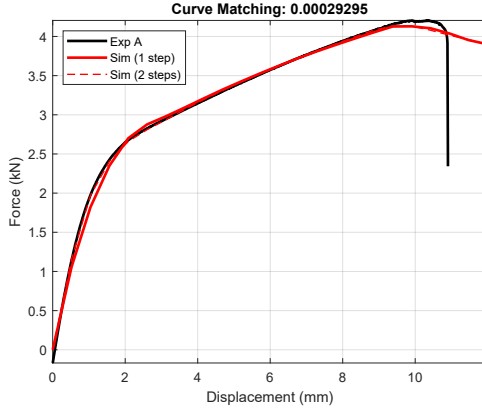


Figure 7.19: Optimized curve of the one-step optimization for the 45° load case.

The optimal parameters of the one-step optimization are reported in Table 7.7. Optimal parameters of the two-steps optimization are also listed for comparison. It is evident that the two approaches led to different optimal parameters, since many of them varied more than 10% with respect to their initial ranges (see Figure 7.22).

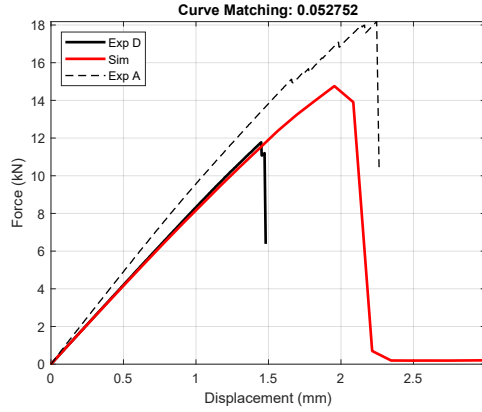


Figure 7.20: Validation of the one-step optimization on MAT58 for the Type II specimen.

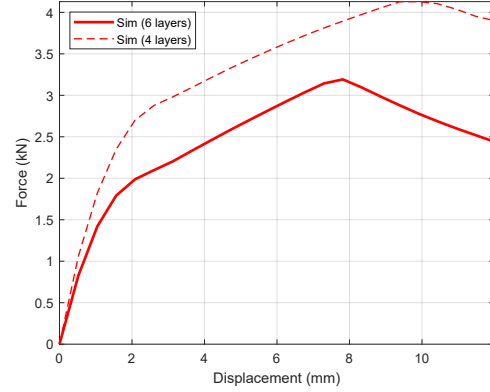


Figure 7.21: Simulated curve for the 45° load case on the Type II specimen (experimental curve not available).

Par.	2-step Opt.	1-step Opt.
EPS	0.0511	0.0631
XT	973	922
EPS2	0.0164	0.0211
XT2	853	1153
GAMMA	0.0248	0.0477
TAU	52	56
GMS	0.1639	0.1729
SC	84	57
GAMMA2	0.0547	0.0595
TAU2	53	60
GMS2	0.1815	0.1846
SC2	91	119

Table 7.7: Comparison between optimal parameters resulting from the one-step and the two step-optimizations.

Validation on other load cases

As in the previous cases, a final simulation was performed on the 6-layer specimen in order to validate the computed optimal parameters. The result of this simulation, shown in Figure 7.20, does not have a good match with the experimental curve, similarly to the same validation for the two-step optimization (Figure 7.17).

Finally, another simulation was performed for the 45° load case on the 6-layer specimen (Type II). The result of this simulation is shown in Figure 7.21, however it is not possible to assess the quality of the result since no experimental curve was available for this specific

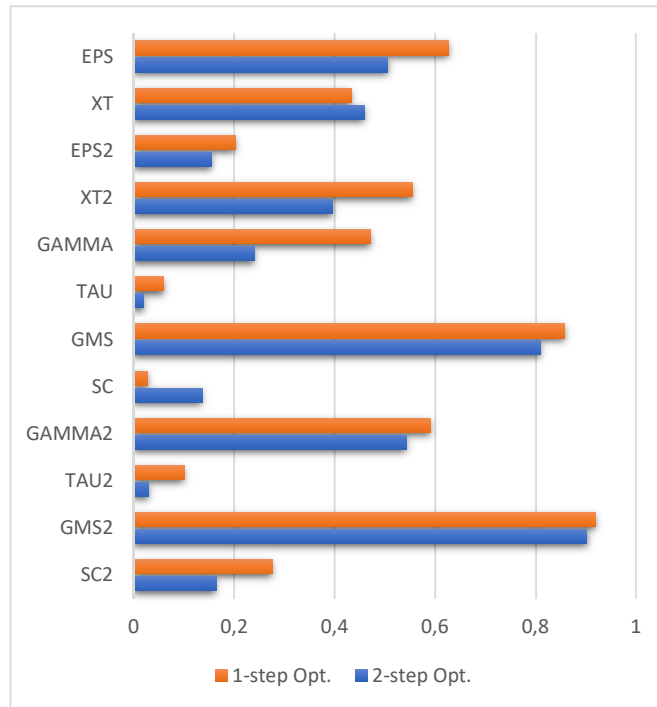


Figure 7.22: Graphical comparison of optimal parameters listed in Table 7.7, normalized to their initial ranges.

load case.

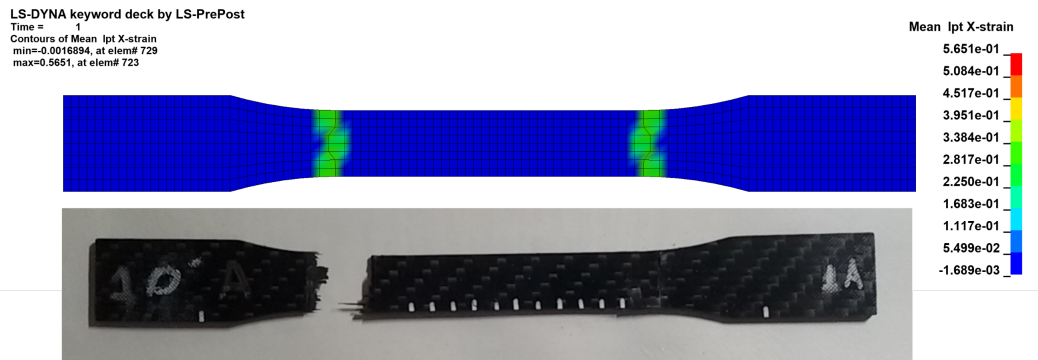


Figure 7.23: Longitudinal strain distribution of the Type I specimen for the 0° load case, with failed specimen of group A for comparison.

Similarly to the case of MAT54, the specimen modeled with MAT58 does not fail during the simulation (since $ERODS=0$). However it is useful to look at the strain plots to see where it would have failed. The 0° load case, shown in Figure 7.23, is very similar to the previous case (MAT54), with failure occurring at the ends of the gage section of the specimen. Contrarily, for the 45° load case in Figure 7.24, the plastic behaviour is noticeable with evident necking of the specimen at the center of the gage section, where the failure would

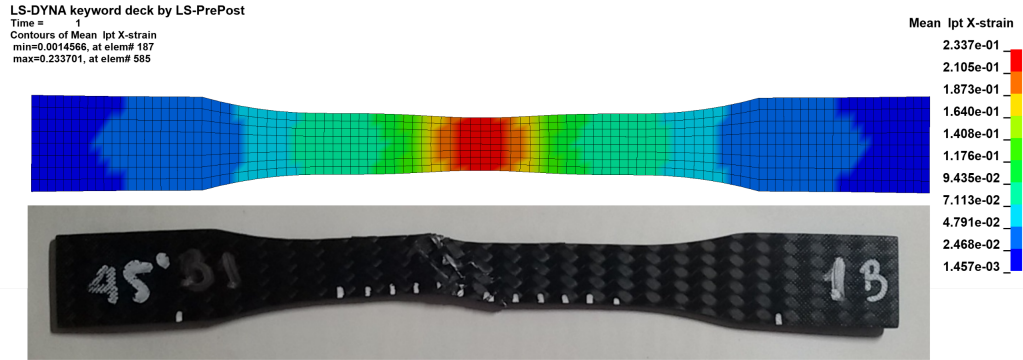


Figure 7.24: Longitudinal (with respect to the global system of reference) strain distribution of the Type I specimen for the 45° load case, with failed specimen of group B for comparison.

have occurred. However, there is no evidence in the simulated specimen of the fracture occurring along the fiber directions.

Chapter 8

Conclusion

In this thesis parameter estimation has been performed on a carbon-fiber reinforced epoxy resin using metamodel-based optimization techniques implemented in the software LS-OPT.

In a preliminary study, different LS-OPT settings have been used on the same problem in order to find the best option configuration. It has been found that the optimized parameters are very sensitive to the specific settings used, although most of the tested configurations produced really good results in terms of matching between experimental and simulated curves. Considering the obtained results, it was decided to perform successive optimization runs using a Metamodel-based Optimization with Sequential with Domain Reduction strategy. In particular, a linear polynomial metamodel with D-optimal sampling was used. The Partial Curve Mapping algorithm with a reduced number of points was used to match simulated and experimental curve and the hybrid algorithm ASA/LFOP was used to optimize the metamodel.

Parameter identification has been performed using two different material models, namely MAT54 and MAT58. From various optimization runs on MAT54, it has been deduced that this material model is appropriate for modeling the tested material only when the load is applied in the same direction as the carbon fibers. Both single layer model and a multiple layer one have been simulated. It was observed that both models can be optimized on longitudinal load cases, however the second one can be applied to layups with a generic number and disposition of layers.

The material model MAT58 was then considered because of its capability of reproducing the experimental behaviour of the material when the load is applied both in the 0° and in the 45° directions. In order to find optimal parameters for MAT58, two different approaches were studied, namely the one-step optimization and the two-steps optimization. It was found that the 45° load case is well optimized with both approaches, but the second one produces better results for the 0° load case, because more simulations are performed and interactions between all parameters are considered. In conclusion, MAT58 was capable of reproducing the behaviour of the tested material for both 0° and 45° load cases, although the matching for the 0° load case with different layers is slightly worse than the correspondent one for MAT54.

By comparing the deformed mesh and the failed specimen, it was observed that there are similarities in the position and type of failure between the experimental and simulated specimens. This suggests that both material models are capable of capturing the failure

behaviour of the material.

In conclusion, material models MAT54 and MAT58 are capable of reproducing the observed experimental behaviour of the material and the elastic properties of single layers of the material have been successfully determined thanks to the optimization procedure. Parameters related to failure have also been optimized, however they should be investigated more in depth.

Future works on the subject might concern further validation of the optimized parameters of MAT54 and MAT58 on different load cases, involving different loads (for example bending or compression loads) or complex geometries. Moreover, the failure and damage behaviour of the material could be investigated in order to accurately model it, using parameters and criteria already implemented in MAT54 and MAT58 or using other material models. Parameter identification could also be performed on tested specimen with uni-directional fibers, for which there were not sufficient valid experimental results.

Appendix A

Failed specimens

In this Appendix photos of all specimens are displayed after they have been tested. For those who have failed, a close view of the fracture is shown.

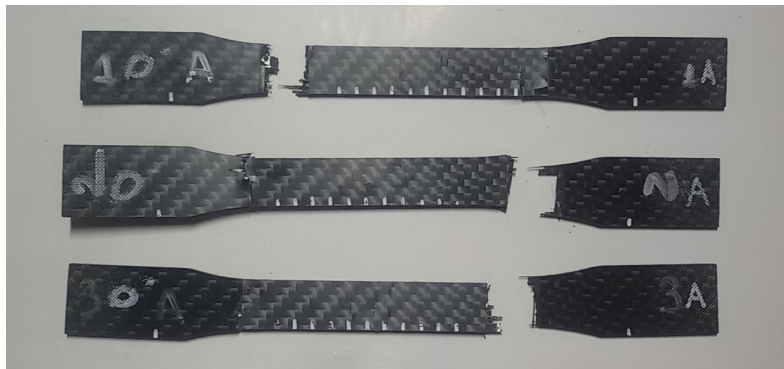


Figure A.1: Failed specimens of group A.

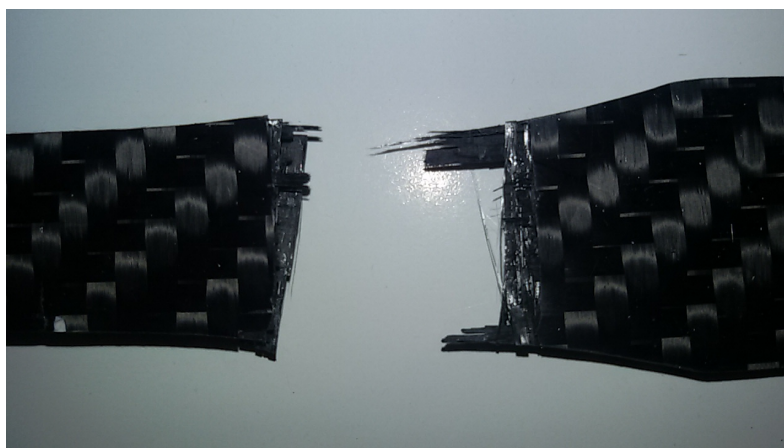


Figure A.2: Close view of fracture for specimens of groups A.



Figure A.3: Failed specimens of group B.



Figure A.4: Close view of fracture for specimens of groups B.



Figure A.5: Failed specimens of group C.

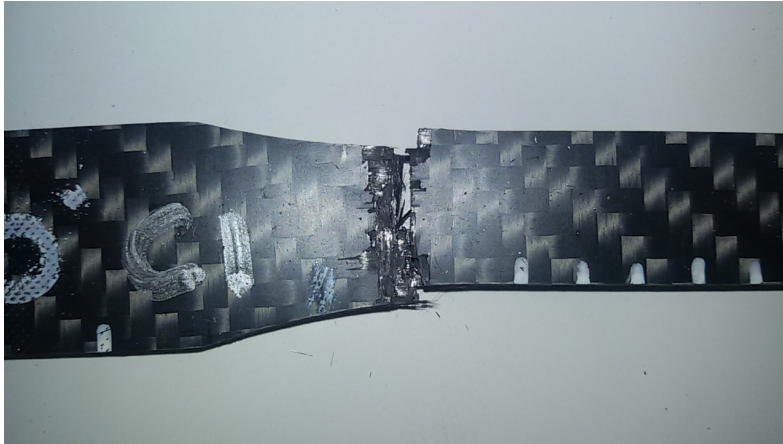


Figure A.6: Close view of fracture for specimens of groups C.

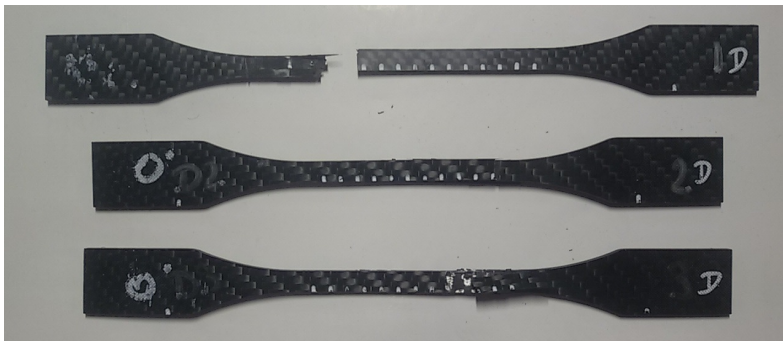


Figure A.7: Failed specimens of group D.



Figure A.8: Close view of fracture for specimens of groups D.



Figure A.9: Specimens of group E after testing. None of the specimens has failed in the gage section.

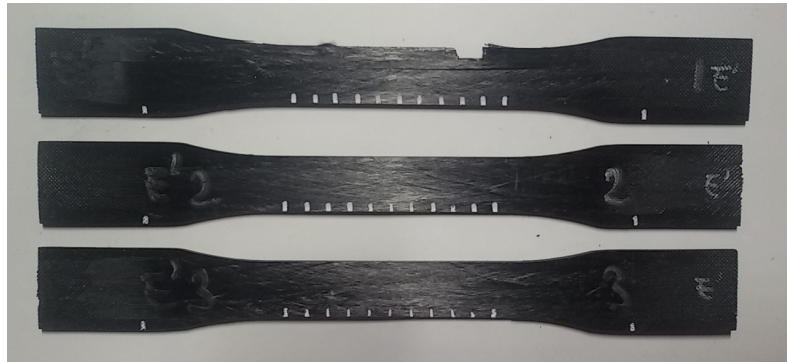


Figure A.10: Specimens of group E'. Only the first specimen has failed in the gage section.

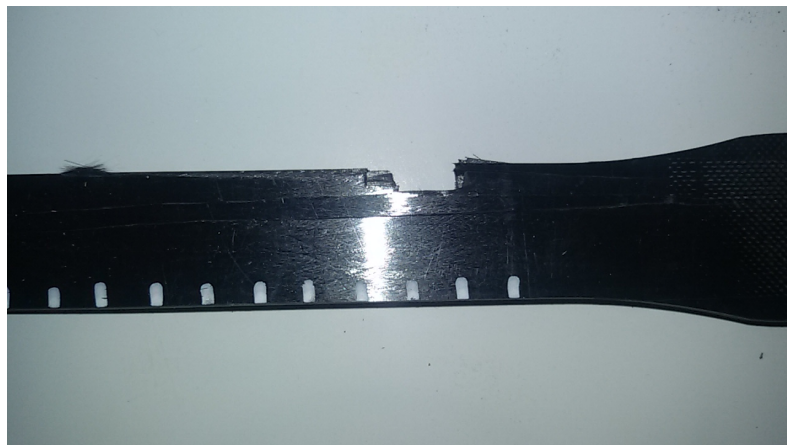


Figure A.11: Close view of fracture for specimens of groups E'.

Appendix B

LS-DYNA input files

In this Appendix, some relevant parts of the LS-DYNA input files are reported. It was not possible to include the entire files because of their excessive length.

The first three pieces of code that are presented were used to model the single layer and the multiple layer materials.

- Single layer model

*PART								
\$#	pid	secid	mid	eosid	hgid	grav	adpopt	tmid
	40	40	1	0	0	0	0	0
*SECTION_SHELL								
\$#	secid	elform	shrf	nip	propt	qr/irid	icomp	setyp
	40	16	1.0	5	1.0	0	0	1
\$#	t1	t2	t3	t4	nloc	marea	idof	edgset
	2.0	2.0	2.0	2.0	0.0	0.0	0.0	0

- Multiple layer model (4 layers)

*PART_COMPOSITE								
\$#	pid	elform	shrf	nloc	marea	hgid	adpopt	ithelfrm
	40	16	0.0	0.0	0.0	0	0	0
\$#	mid1	thick1	b1	tmid1	mid2	thick2	b2	tmid2
	1	0.50	0.0	0	2	0.50	0.0	0
	2	0.50	0.0	0	1	0.50	0.0	0

- Multiple layer model (6 layers)

*PART_COMPOSITE								
\$#	pid	elform	shrf	nloc	marea	hgid	adpopt	ithelfrm
	40	16	0.0	0.0	0.0	0	0	0
\$#	mid1	thick1	b1	tmid1	mid2	thick2	b2	tmid2
	1	0.50	0.0	0	2	0.50	0.0	0
	2	0.50	0.0	0	2	0.50	0.0	0
	2	0.50	0.0	0	1	0.50	0.0	0

The following pieces of code concern the definition of the material model and were used for most of the simulations in Chapters 6 and 7. In order to model the 0° load case it suffices to define $a1=1$, $a2=a3=0$, while the 45° load case is defined by specifying $a1=a2=1$ and $a3=0$.

- **Material model MAT54**

*MAT_ENHANCED_COMPOSITE_DAMAGE								
\$#	mid	ro	ea	eb	(ec)	prba	(prca)	(prcb)
		11.60000E-9	&E	&E	0.0	&PR	0.0	0.0
\$#	gab	gbc	gca	(kf)	aopt	2way		
	&G	&G	&G	0.0	2.0	0.0		
\$#	xp	yp	zp	a1	a2	a3	mangle	
	0.0	0.0	0.0	1.0	0.0	0.0	0.0	
\$#	v1	v2	v3	d1	d2	d3	dfailm	dfails
	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
\$#	tfail	alph	soft	fbrt	ycfac	dfailt	dfailc	efs
	0.0	0.0	0.0	0.0	0.0	10	-10	0.0
\$#	xc	xt	yc	yt	sc	crit	beta	
	&XT				54.0	0.0		
\$#	pel	epsf	epsr	tsmd	soft2			
	100.0	0.0	0.0	0.0	0.0			
\$#	slimt1	slimc1	slimt2	slimc2	slims	ncyred	softg	

- **Material model MAT58**

*MAT_LAMINATED_COMPOSITE_FABRIC								
\$#	mid	ro	ea	eb	(ec)	prba	tau1	gamma1
		11.60000E-9	&E	&E	0.0	&PR	&TAU	&GAMMA
\$#	gab	gbc	gca	slimt1	slimc1	slimt2	slimc2	slims
	&G	&G	&G	0.01		0.01		1.0
\$#	aopt	tsize	erods	soft	fs	epsf	epsr	tsmd
	2.0			0.0	-1.0			
\$#	xp	yp	zp	a1	a2	a3	prca	prcb
	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
\$#	v1	v2	v3	d1	d2	d3	beta	
	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
\$#	e11c	e11t	e22c	e22t	gms			
	&EPS	&EPS	&EPS	&EPS	&GMS			
\$#	xc	xt	yc	yt	sc			
	&XT	&XT	&XT	&XT	&SC			

For completeness, also control, database and boundary condition cards are reported below, since they were the same for all simulations (except for the nodes identification numbers).

- **Control cards**

*CONTROL_IMPLICIT_AUTO								
\$#	iauto	iteopt	itewin	dtmin	dtmax	dtexp	kfail	kcycle

```

      1      200      0      0.0      0.05      0.0      0      0
*CONTROL_IMPLICIT_GENERAL
$# imflag      dt0      imform      nsbs      igs      cnstn      form      zero_v
      1      0.01      2      0      2      0      0      0
*CONTROL_IMPLICIT_SOLUTION
$# nsolvr      ilimit      maxref      dctl      ectol      rctl      lstol      abstol
      12      0      0      0.0      0.0      0.0      0.01.0000E-10
$# dnorm      diverg      istif      nlprint      nlnorm      d3itctl      cpchk
      2      1      0      0      2      0      0
$# arcctl      arcdir      arclen      arcmt      arcdmp      arcpsi      arcalf      arctim
      0      0      0.0      1      2      0      0      0
$# lsmtl      lsdir      irad      srad      awgt      sred
      4      2      0.0      0.0      0.0      0.0
*CONTROL_IMPLICIT_SOLVER
$# lsolvr      lprint      negev      order      drcm      drcprm      autospc      autotol
      4      0      2      0      4      0.0      1      0.0
$# lcpack      mtxdmp
      2      0
*CONTROL_SHELL
$# wrpang      esort      irnxx      istupl      theory      bwc      miter      proj
      0.0      0      0      0      2      2      1      0
$# rotascl      intgrd      lamsht      cstyp6      thshel
      1.0      0      0      1      0
$# psstupl      sidt4tu      cntco      itsflg      irquad      w-mode      stretch      icrq
      0      0      0      0      2      0.0      0.0      0
$# nfail1      nfail4      psnfail      keepcs      delfr      drcpsid      drcprm      intperr
      0      0      0      0      0      0      1.0      0
*CONTROL_TERMINATION
$# endtim      endcyc      dtmin      endeng      endmas      nosol
      1.0      0      0.0      0.01.000000E8      0

```

- Database cards

```

*DATABASE_BNDOUT
$#      dt      binary      lcur      ioopt
      1.00000E-4      2      0      1
*DATABASE_ELOUT
$#      dt      binary      lcur      ioopt      option1      option2      option3      option4
1.00000E-4      2      0      1      0      0      0      0
*DATABASE_GLSTAT
$#      dt      binary      lcur      ioopt
1.00000E-4      2      0      1
*DATABASE_NODFOR
$#      dt      binary      lcur      ioopt
1.00000E-4      2      0      1
*DATABASE_NODOUT
$#      dt      binary      lcur      ioopt      option1      option2
1.00000E-4      2      0      1      0.0      0
*DATABASE_SECFORC
$#      dt      binary      lcur      ioopt

```

```

1.00000E-4      2      0      1
*DATABASE_SPCFORC
$#      dt      binary      lcur      ioopt
1.00000E-4      2      0      1
*DATABASE_BINARY_D3PLOT
$#      dt      lcdt      beam      npltc      psetid
      0.01      0      0      0      0
*DATABASE_EXTENT_BINARY
$#  neiph      neips      maxint      strflg      sigflg      epsflg      rltflg      engflg
      0      0      3      1      1      1      1      1
$#  cmpflg      ieverp      beamip      dcomp      shge      stssz      n3thdt      ialemat
      0      0      0      1      1      1      2      1
*DATABASE_CROSS_SECTION_PLANE_ID
$#      csid                                     title
      1
$#      psid      xct      yct      zct      xch      ych      zch      radius
      0      0.0      -6.5      1.0      8.25      -6.5      1.0      0.0
$#      xhev      yhev      zhev      lenl      lenm      id      itype
      0.0      0.0      0.0      0.0      0.0      0      0
*DATABASE_HISTORY_NODE
$#      id1      id2      id3      id4      id5      id6      id7      id8
      929      1514      314      0      0      0      0      0

```

• Boundary conditions, curves and node sets

```

*BOUNDARY_PRESCRIBED_MOTION_SET_ID
$#      id                                     heading
      2
$#      nsid      dof      vad      lcid      sf      vid      death      birth
      2      1      2      1      1.0      01.00000E28      0.0
*BOUNDARY_SPC_SET_ID
$#      id                                     heading
      1
$#      nsid      cid      dofx      dofy      dofz      dofrx      dofry      dofrz
      1      0      1      1      1      1      1      1
*SET_NODE_LIST_TITLE
Fixed
$#      sid      da1      da2      da3      da4      solver
      1      0.0      0.0      0.0      0.0MECH/CESE/
$#      nid1      nid2      nid3      nid4      nid5      nid6      nid7      nid8
      620      621      622      623      624      625      626      683
      909      0      0      0      0      0      0      0
*BOUNDARY_SPC_SET
$#      nsid      cid      dofx      dofy      dofz      dofrx      dofry      dofrz
      8      0      0      1      1      1      1      1
*SET_NODE_LIST_TITLE
NODESET(SPC) 8
$#      sid      da1      da2      da3      da4      solver
      8      0.0      0.0      0.0      0.0MECH
$#      nid1      nid2      nid3      nid4      nid5      nid6      nid7      nid8

```

```

311      312      313      314      315      316      317      490
680      0      0      0      0      0      0      0
*DEFINE_CURVE_TITLE
Displacement_curve
$#  lcid      sidr      sfa      sfo      offa      offo      dattyp      lcint
      1      0      1.0      1.0      0.0      0.0      0      0
$#      a1      o1
      0.0      0.0
      1.001      3.0
*SET_NODE_LIST_TITLE
Motion
$#  sid      da1      da2      da3      da4      solver
      2      0.0      0.0      0.0      0.0MECH/CESE/
$#  nid1      nid2      nid3      nid4      nid5      nid6      nid7      nid8
      311      312      313      314      315      316      317      490
      680      0      0      0      0      0      0      0
*SET_NODE_LIST_TITLE
Displacement_Set
$#  sid      da1      da2      da3      da4      solver
      6      0.0      0.0      0.0      0.0MECH/CESE/
$#  nid1      nid2      nid3      nid4      nid5      nid6      nid7      nid8
      929      1514      0      0      0      0      0      0
*SET_NODE_LIST_TITLE
Cross_Section_Nodes
$#  sid      da1      da2      da3      da4      solver
      7      0.0      0.0      0.0      0.0MECH/CESE/
$#  nid1      nid2      nid3      nid4      nid5      nid6      nid7      nid8
      565      573      581      589      597      605      613      755
      770      0      0      0      0      0      0      0
*SET_SHELL_LIST_TITLE
Cross_Section_Shell
$#  sid      da1      da2      da3      da4
      8      0.0      0.0      0.0      0.0
$#  eid1      eid2      eid3      eid4      eid5      eid6      eid7      eid8
      549      558      567      576      585      594      603      612

```

Bibliography

- [1] ASTM International, *Designation D638-14, Standard Test Method for Tensile Properties of Plastics*.
- [2] D. J. Benson, N. Stander, M. R. Jensen, K. J. Craig, *On the Application of LS-OPT to Identify Non-Linear Material Model in LS-DYNA*, 7th International LS-DYNA Users Conference, 2002.
- [3] S. Dong, A. Sheldon, K. Carney, *Modeling of Carbon-Fiber-Reinforced Polymer (CFRP) Composites in LS-DYNA with Optimization of Material and Failure Parameters in LS-OPT*, 15th International LS-DYNA Users Conference, 2018.
- [4] N. Van Dorsselaer, V. Lapoujade, G. Nahas, B. Ciree, F. Tarallo, J.-M. Rambach, *LS-OPT Parameters Identification on Concrete Sample Tests for an Impact Simulation on Concrete Slab*, 9th European LS-DYNA Conference, 2013.
- [5] R. Dotoli, M. Rizzo, *Fabric impact drop tests: numerical simulations using the LS-DYNA micromechanical approach and experimental characterization*, International CAE Conference, 2014.
- [6] D. Dubois, J. Forsberg, *Using LS-OPT for Parameter Identification and MAT_FABRIC with FORM=-14*, 9th European LS-DYNA Conference, 2013.
- [7] A. A. Giunta, V. Balabanov, D. Haim, B. Grossman, W. H. Mason, L. T. Watson, R. T. Haftka, *Wing Design for a High-Speed Civil Transport using a Design of Experiments Methodology*, American Institute of Aeronautics and Astronautics, 1996.
- [8] R. Jin, W. Chen, T. W. Simpson, *Comparative studies of metamodeling techniques under multiple criteria*, Struct. Multidisc. Optim., 23, 1-13, 2001.
- [9] Livermore Software Technology Corporation (LSTC), *LS-OPT User's Manual Version 5.2*, 2015.
- [10] Livermore Software Technology Corporation (LSTC), *LS-DYNA Keyword User's Manual*, Vol. I, 2017.
- [11] Livermore Software Technology Corporation (LSTC), *LS-DYNA Keyword User's Manual*, Vol. II, 2017.
- [12] Livermore Software Technology Corporation (LSTC), *LS-DYNA Theory Manual*, 2017.
- [13] T. Phongsai, W. Julsri, B. Chongthairungruang, S. Suranuntchai, S. Jirathearant, V. Uthaisangsuk, *Identification of Material Parameters for Springback Prediction using Cyclic Tension-Compression Test*, Songklanakarin Journ. Sci. Technol., 38, 485-493, 2016.
- [14] M. Scapin, L. Peroni, M. Peroni, *Parameters identification in strain-rate and thermal sensitive visco-plastic material model for an alumina dispersion strengthened copper*,

- International Journal of Impact Engineering, 40-41, 58-67, 2012.
- [15] K. Schweizerhof, K. Weimar, Th. Münz, Th. Rottner, *Crashworthiness Analysis with Enhanced Composite Material Models in LS-DYNA – Merits and Limits*, LS-DYNA World Conference, 1998.
 - [16] A. J. Smola, B. Schölkopf, *A tutorial on support vector regression*, Statistics and Computing, 14(3), 199-222, 2004.
 - [17] J. A. Snyman, *A new and dynamic method for unconstrained minimization*, Appl. Math. Modelling, 6, 449-462, 1982.
 - [18] J. A. Snyman, *The LFOPC Leap-Frog Algorithm for Constrained Optimization*, Computer and Mathematics with Applications, 40, 1085-1096, 2000.
 - [19] I. M. Sobol, *Sensitivity Estimates for Nonlinear Mathematical Models*, MMCE, 1(4), 407-414, 1993.
 - [20] N. Stander, K. J. Craig, H. Müllerschön, R. Reichert, *Material Identification in Structural Optimization Using Response Surfaces*, Structural and Multidisciplinary Optimization, 29, 93-102, 2005.
 - [21] N. Stander, K. J. Craig, *On the robustness of a simple domain reduction scheme for simulation-based optimization*, Eng. Comput., 19(4), 431-450, 2002.
 - [22] N. Stander, R. Reichert, T. Frank, *Optimization of nonlinear dynamical problems using successive linear approximations in LS-OPT*, 8th Symposium on Multidisciplinary Analysis and Optimization, 2000.
 - [23] L.Y. Wang, L. Li, *Parameter Identification of GTN Model Using Response Surface Methodology for High-Strength Steel BR1500HS*, Journ. of Materials Eng. and Performance, 26, 3831-3838, 2017.
 - [24] K. Witowski, M. Feucht, N. Stander, *An Effective Curve Matching Metric for Parameter Identification using Partial Mapping*, 8th European LS-DYNA Users Conference, 2011.