Politecnico di Torino

Master of Science in Automotive Engineering
Accademic Year 2023/2024
Graduation Session March/April 2024

Simplified Finite Element model of the Solution Space for Frontal Crash Analysis for Uncertainty Propagation

Supervisor:
Prof. Alessandro Scattina
Co-supervisors:
Prof. Dr.-Ing. habil Fabian Duddeck
Paolo Ascia, M.Sc.

Candidate:
Michele Morese

Abstract

Passenger safety is a key aspect during the development of a new vehicle. However, designing for safety is a very complicated task that results in high costs. The extensive use of finite element (FE) simulations helps accelerate development and contain the overall costs. Nevertheless, recent developments in data science can further help obtain the desired information at a fraction of the current costs.

In this thesis, I propose a method to test a component in the early stage of development. This method considers the effects of development uncertainties to study how they affect the behavior of a component. For this purpose, I propose a simplified FE model that couples 1D elements with the full mesh of a component to be studied. Surrogate models are then used to predict the behavior of the other components. The solution space method is used in combination with the surrogate models to characterize the performance of the 1D element under the development uncertainties. To propagate the uncertainty, I finally run multiple samples. This confirms the capability of the method I propose to capture how a component is affected by the variations of the design parameters of the other components. This information provides an indication of the robustness of the design of the studied component.

My method offers a tool for engineers to quickly test early-stage designs of a component, identifying which could work in the full model. This strategy reduces the cost of the overall development.

Acknowledgements

I want to dedicate a few words to those who made this path possible. First of all, my thankfulness and gratitude to my supervisors, prof. Alessandro Scattina and prof. Fabian Duddeck, for giving me the opportunity to gain experience in a new and wonderful environment. These months in Germany helped me to mature.
A special thanks goes to Paolo Ascia for bearing with me in the darkest and most difficult moments. Your emotional and intellectual support was fundamental for the conclusion of this work.

Un ringraziamento speciale va alla mia famiglia ed amici.
Grazie a mio padre e mia madre, per esserci sempre stato nei momenti felici, tristi e soprattutto i più noiosi. È bello sapere che c'è sempre qualcuno per te.
Grazie a mio fratello Gaetano, che non ha mai mancato un'occasione per rinfacciarmi i 30 mancati e i 23 sofferti. Hai reso questi anni leggeri e divertenti, tranne quando vincono i T1.
Grazie a mia nonna per l'amore incondizionato che mi ha dato in tutti questi anni, e per tutte le chiamate negli orari peggiori.
Grazie a Dino e Maria Teresa, che mi hanno accolto e trattato come un nipote in questi anni a Torino.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Vehicle safety represents a key factor in the development of automobiles in the modern era. It is approximated that 1.19 million people die each year due to road traffic crashes. Additionally, 20 to 50 million people suffer non-fatal injuries, with many facing disabilities after the crash [1]. These data highlight the importance of the focus on further improvements in road and vehicle safety. Designing a frontal crash structure, however, is a costly procedure, both in terms of time and computational resources. A common workflow is to design independently each component. The safety standards are verified at the component level by performing a traditional drop tower test. These tests allow to capture the dynamic effects of a crash. However, as the components are tested independently, a traditional drop tower test is not able to highlight the interactions between components in a crash.

The cost of the development of a new crash structure can be substantial. The solution space method introduced by Zimmermann et al. [2] offers a process to reduce the development cost. The methodology proposed allows to define a set of feasible possible designs for each component of the structure. The application in the automotive field is based on the Geometry Space Models (GSM) and Deformation Space Models (DSM) introduced by Fender et al. [3]. The paper by Lange et al. [4] presents an iterative process to evaluate the DSM of any complex structure, allowing the introduction of connecting elements between load paths or inclined elements in general. The thesis by Chaudhry Taimoor Niaz [5] explores a mathematical model adaptation for the connecting elements. This model allows us to link the solution space method with a Deformation Space Model that includes connecting elements. These studies focus on the performance of the entire crash structure. Ascia et al. [6] present a new vehicle model adapted to the analysis of the effect of each component on a component represented with a 2D mesh. The methodology applied takes advantage of surrogate models to predict the behavior of 1D components. This process allows the reduction of the computational cost of each FE simulation.

In this thesis, I implement the DSM proposed by Chaudhry Taimoor Niaz [5], which is composed of 2 parallel load paths and a connecting element. A lumped mass is implemented in the model. The relative solution spaces are computed taking advantage of the python script developed by Paolo Ascia. The performance of the simplified 1D FE model is verified, paying specific attention to data that confirm the satisfaction of the imposed constraints. The model is then coupled with the 2D

mesh of one component. The component defined with a 2D mesh is designed based on an early assumption of the performance desired. The effect on the solution space of the system is then evaluated. As a last step, the uncertainty on the designs of each component of the structure is propagated to evaluate their effect on the 2D mesh implemented.
The goal of this workflow is to provide a methodology suitable to fully analyze the performance of an early-phase design. This methodology allows the study of the robustness of a design, providing information on the dependency of the performance of a component on the changing designs of other components in the entire system.

The thesis is structured as follows:

- **Chapter 2** provides an overview of the Aims and Objectives of the thesis;
- **Chapter 3** presents an overview of the relevant research papers. Studies regarding the solution space method and its application in the automotive field are reviewed. Relevant research on uncertainty quantification and the development of surrogate models are presented;
- **Chapter 4** presents the methodology applied to implement a simplified 1D Finite Element model and evaluate the relative solution space. The model is then coupled with the 2D mesh of one component. A surrogate model is designed to predict the behavior of 1D components. The uncertainties of the 1D components are propagated and their effect is tested on the 2D mesh;
- **Chapter 5** discuss the results obtained. The solution space corridors are shown, and relevant data to ensure the satisfaction of the imposed constraints is presented. The performance of the surrogate models, the design of component 2, and the effect of the uncertainties on the 2D mesh are discussed.
- **Chapter 6** offers a detailed analysis of the key choices adopted for the implementation of the model. A further analysis of the results previously presented is provided.
- **Chapter 7** offers an overview of the key findings of the work presented and suggests relevant topics for future research.

Chapter 2

Aims and Objectives

In the first stages of development of new vehicles, it is important to meet the desired design targets in a vast range of variations of the parameters of a component, without immediately fixing their design or testing multiple different combinations. This is possible through the solution space method. However, variations in a component design affect how other components in the system behave, differences that are not highlighted in a conventional drop tower test.
The goal of this thesis is to create a simplified FE model and test the effect of the variations of the parameters of each component, effect measured through the forces acting on a selected component. The work is divided into different objectives:

1. Create a new Geometry Space Model (GSM) and the relative Deformation Space Model (DSM);
2. Adapt an existing mathematical model to our DSM, and solve the linear system of equations to obtain the solution space corridors;
3. Create a simplified 1D FE model in LS-Dyna;
4. Verify the accuracy of the mathematical model implemented on the 1D FE model with an FE simulation of a frontal crash. The performance of the structure in compression is studied;
5. Replace one monodimensional component in our model with a 2D mesh;
6. Create a surrogate model of each component, using various drop tower test results as training data;
7. Fix an early design for the component with the 2D mesh, and modify the relative solution space model;
8. Evaluate the effect of the variations of the parameters of each component on the component studied.

Chapter 3

Literature review

3.1 Solution space method

The main concern during the design of a new vehicle is the safety of the occupants. The focus of the design strategy is to increase the protection of the occupants to reduce the number of fatal and serious injuries. Frontal crashes account for the majority of injuries and fatalities. However, designing for safety is a very complicated task that results in high costs. Designing and testing different solutions, aiming to obtain the best performance, leads to time loss due to the sensitivity of the components to small variations in the entire system. Designs that would be ideal in the initial stages of development could return poor performance in later stages. This causes continuous redesigns of components and extensive testing. An alternative to this methodology was first proposed by Zimmermann et al. [2], introducing the solution space method. The solution space method is based on identifying a wide range of feasible designs. Compared to optimization methods that search for the best performance, the goal of the solution space method is to obtain robust designs. Each design is identified by a vector $\mathrm{x} = (\mathrm{x}_1, \mathrm{x}_2, ..., \mathrm{x}_\mathrm{p})$, where p is the number of design parameters. The set of all possible designs defines the design space. The solution space method proposes to first subdivide the entire design space into good and bad regions. The good regions contain the set of design points that fulfill a set of imposed constraints. Then, the goal is to find the solution box with the largest volume, with the following characteristics:

- The required system performance is guaranteed if the design of each component is within the solution box;
- Each component is decoupled from each other. The permissible range of each component is independent of the behavior of other components. This characteristic allows the modification of the design of each component without compromising the entire system.

Figure 3.1 shows the subdivision of the design space for a generic problem. The goal of the solution space method is to define the solution box with the largest volume.

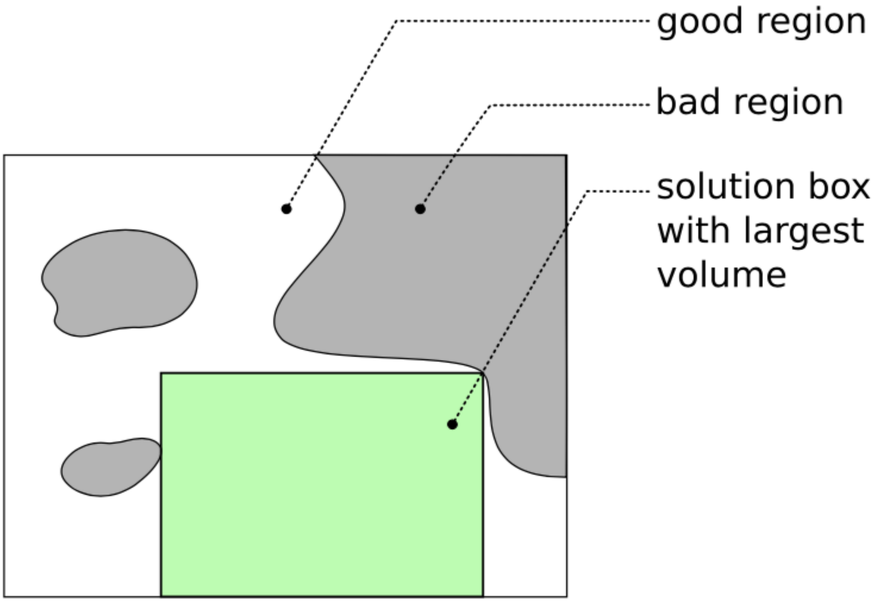


Figure 3.1: Solution box for a general problem [2]

The solution box is derived through an iterative process, proposed by Zimmermann et al. [2]. The solution box is first enlarged as much as possible while maintaining the fractions of bad design points below a specified threshold. Then, all bad regions are removed, based on the results of a few sampling points selected with a Monte Carlo algorithm. Figure 3.2 shows the iterative process implemented to define the solution box.

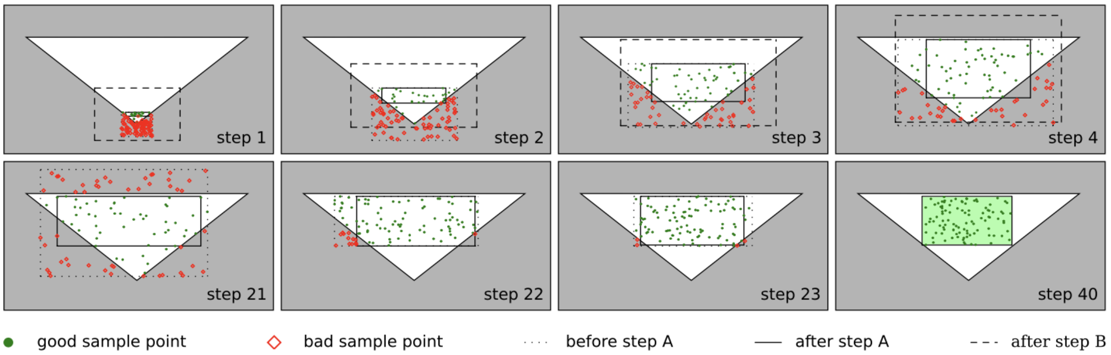


Figure 3.2: Evolution of the candidate box in phase I (steps 1–20) and phase II (steps 21–40). The result is shown at step 40. [2]

Geometry Space Model and Deformation Space Model The Geometry Space Model (GSM) and Deformation Space Model (DSM) representation of the crash structure was first presented by Fender et al. [3]. The GSM represents the actual geometry of the front structure. Each component is represented by a deformable part and an undeformable part. This representation allows to reproduce the behavior of a real component. It is assumed that a real component would have a specific maximum deformation length and act like a rigid body when fully compressed. Lumped masses can be included between different components. The model proposed by Fender et al. [3] is composed of only parallel load paths. The load paths are defined by the crash load case considered. The load paths are defined at the beginning of the design phase. The GSM is divided into multiple sections, based on the order of deformation. Each section can contain multiple components and each component can be divided into different sections. This condition indicates each section must deform its entire length before the next section can start deforming. Components parts that deform simultaneously are grouped in the same section. However, in the GSM is possible that those parts do not lie in parallel.

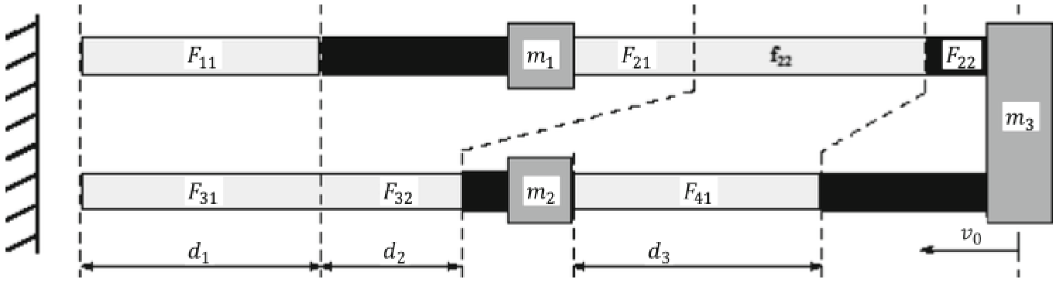


Figure 3.3: Geometry Space Model with 4 components, 3 different sections and lumped masses. [3]

The Deformation Space Model (DSM) is derived directly from the GSM. The non-deformarble parts and gaps are removed. Each deformable section is evaluated simulating a frontal crash. A new section starts when one component is deformed for its entire length. The sections that deform simultaneously lie in parallel. Figure 3.4 shows the DSM derived from the previous GSM.

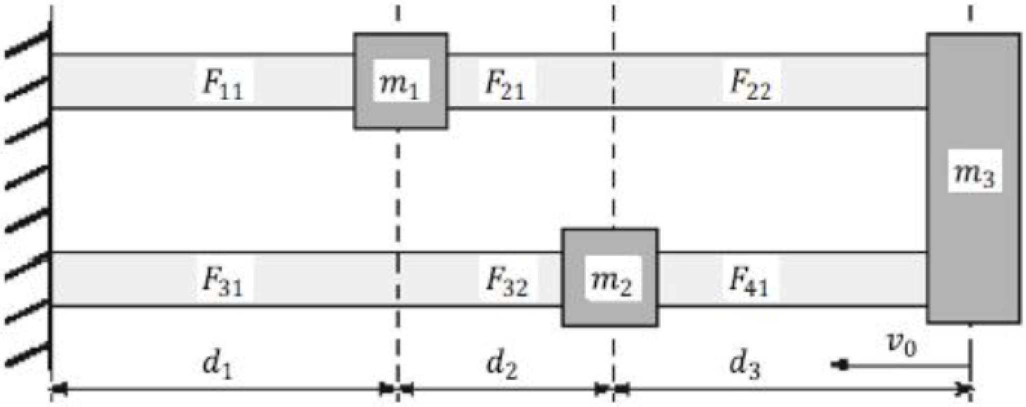


Figure 3.4: Deformation Space Model of the system. Component parts that deform simultaneously lie in parallel, undeformable parts are removed. [3]

This methodology was further analyzed by Lange et al. [4], presenting a more complex model. Lange et al. first introduced components that connect different load paths. Moreover, it presents a step-by-step methodology to derive the DSM from its related GSM.

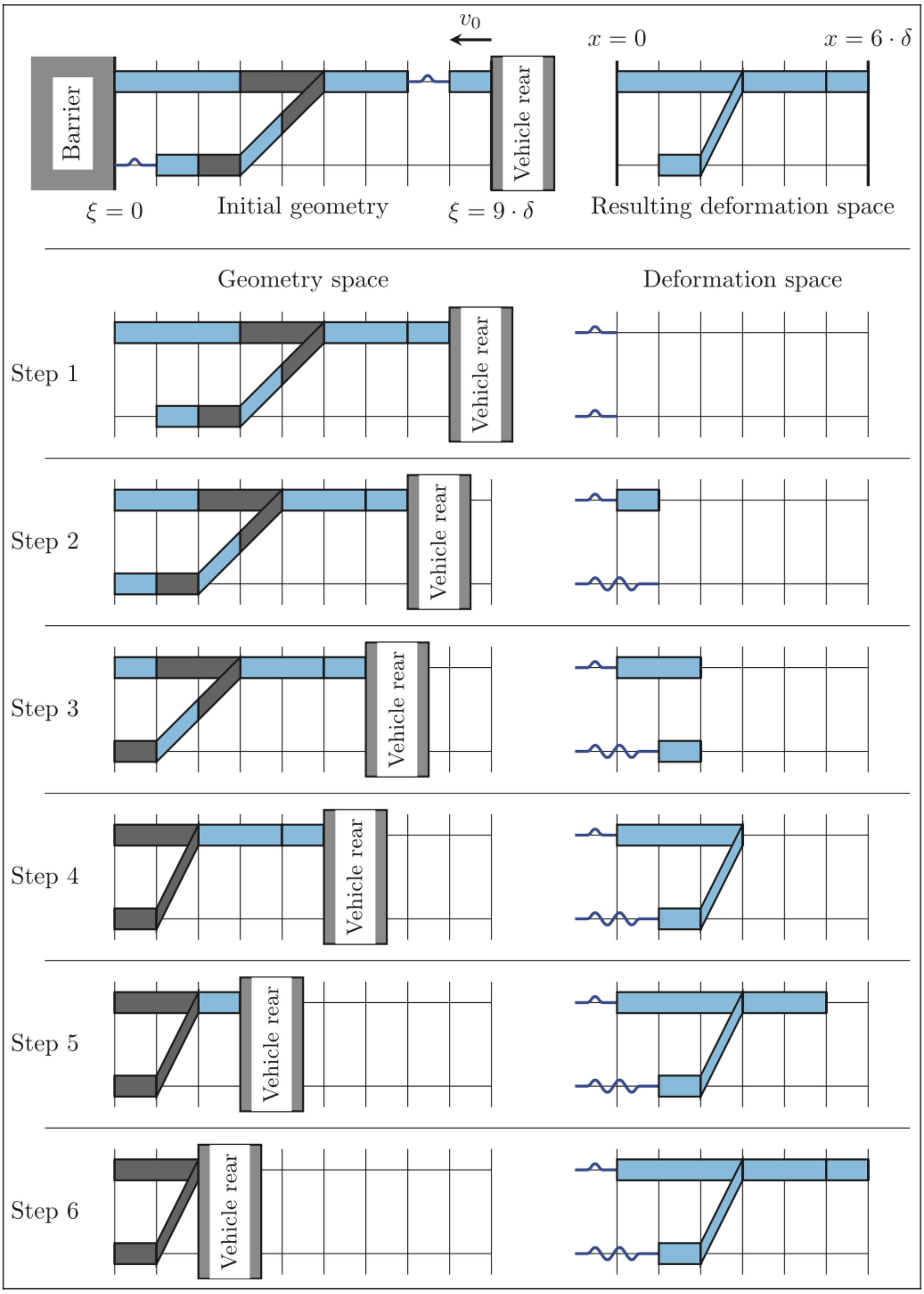


Figure 3.5: Deformation progress in both GSM (left) and DSM (right) [4]

Figure 3.5 shows the process followed to derive the DSM of the relative GSM, which includes 2 parallel load paths and 1 connecting element. The DSM is derived by simulating a frontal crash. The gaps in the structure, included due to the available free crash space, are removed in the first step. Each section is defined by the component parts that deform simultaneously. A section ends when one component has completely deformed or when a new component starts deforming. In step 2 of Figure 3.5, it is possible to note how a new section is defined due to the initial deformation of the component in the lower load path. In step 3, instead, a new section is defined as the same component has completed its deformation. The DSM is completely defined when each component has deformed for their entire length.

Mathematical model The iterative process of the solution space method initially explored by Zimmermann et al. [2] requires a full FE simulation for each sampling point analyzed. This approach is computationally and time-expensive. A new methodology was first analyzed by Fender et al. [3], and later by Daub et al. [7]. Fender proposes to simplify the solution space problem to a set of linear equations. These equations are derived from the requirements of the USNCAP crash load case [8]. The approach proposed is based on three constraints on the system:

- Energy absorption: the kinetic energy of the impact needs to be entirely absorbed by the structure. This results in a complete stop of the vehicle during the deformation. It must be true that sum of the energy that can be absorbed by each component is higher than the kinetic energy of the impact:

  $$\sum_{i=0}^{n} \int_{L_i} F_i(u_i) du_i \leq -\frac{1}{2} m v_0^2 \tag{3.1}$$

  The absorbed energy is evaluated as the sum of the integral of the crushing force $\mathrm{F_i}$ over the deformation $\mathrm{u_i}$, where i indicates the i-th component of the structure.

- Maximum acceleration: a critical acceleration has to be imposed to limit serious crash-related injuries. Specifically, the sum of the crashing forces of the M simultaneously deforming components over the acting mass m^*:

  $$\frac{1}{m_i^*} \sum_{j=1}^{M} F_{i,j} \leq a_{c,u} \tag{3.2}$$

- Progressive order of deformation: The front of the car must be the first part to deform in a crash. It must be true that the collapse force of each component is higher than the force at any point of deformation of a previously deforming component:

  $$F_i \leq F_{i+1} \tag{3.3}$$

  This constraint is extremely important for low-speed crashes, as the front part of the crash structure can be more easily replaced.

This set of equations returns an optimization problem related to the dimensions of the solution box. Fender et al. [3] propose to solve the optimization problem using the objective function:

$$\Phi = \Delta F_{\omega,min} = min_i \left(\frac{F_{u,i} - F_{l,i}}{\omega_i} \right) \tag{3.4}$$

where F_u and F_l indicate the upper and lower limits of the force interval related to the component i. The variable ω_i represents the weight assigned at each individual interval. This variable is fundamental as an increase of one force interval always causes a decrease in the other intervals. Setting up the weight of each interval permits to define the most important design parameters in the structure. Given the set of equations and the objective function provided, it is possible to derive a linear system of equations to solve the solution space problem:

$$\begin{bmatrix} 0 & \frac{d_1}{m_1^*} & \frac{d_2}{m_2^*} & \frac{d_3}{m_3^*} & 0 & 0 & 0 \\ \omega_1^* & -1 & 0 & 0 & 1 & 0 & 0 \\ \omega_2^* & 0 & -1 & 0 & 0 & 1 & 0 \\ \omega_3^* & 0 & 0 & -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & \frac{1}{m_1^*} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{m_2^*} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{m_3^*} \end{bmatrix} \begin{bmatrix} \Delta F_\omega \\ F_{l,1}^* \\ F_{l,2}^* \\ F_{l,3}^* \\ F_{u,1}^* \\ F_{u,2}^* \\ F_{u,3}^* \end{bmatrix} = \begin{bmatrix} \frac{v_0^2}{2} \\ 0 \\ 0 \\ 0 \\ a_c \\ a_c \\ a_c \end{bmatrix} \tag{3.5}$$

The system of equations 3.5 is referred to the Deformation Space Model presented in Figure 3.4 The mathematical model proposed by Fender et al. [3] is based on parallel and independent load paths. In this thesis, we will focus on the model proposed by Chaudhry Taimoor Niaz [5]. In the new model, an angular correction is considered to allow the implementation of a connecting element between two load paths. The modified mathematical model with the angular correction is presented in Section 4.1.

3.2 Uncertainty Quantification

During the development of a new vehicle, the uncertainties in the design of each component need to be considered. As expressed previously, the goal of the solution space method is to obtain a robust design. The robustness of a component is defined by its sensitivity to the variable conditions of the system. The uncertainty in the engineering field is classified as aleatory and epistemic uncertainty, depending on the nature of the uncertainty. Uncertainties are referred to as aleatory if it is caused by variability of fluctuations of the system. This type of uncertainty is irreducible and is described by probabilistic models. Epistemic uncertainties are, instead, caused by a lack of or imprecise knowledge. This kind of uncertainty is reducible, by improving the model used to describe the phenomenon. A case study of the quantification of uncertainties in the early stage of development was presented by Ascia et al. [9, 6]. In this thesis, we focus on the epistemic uncertainties derived from the lack of information on the characteristics of each component of the model. We will use surrogate models to describe the behavior of each component, and we will propagate the uncertainties of the design of each component to analyze the robustness of our design.

3.2.1 Surrogate Model

An analysis of the robustness of the design of one component, based on the uncertainty of the system, requires extensive FE testing. An accurate simulation model requires high computational resources and is typically time-consuming. Surrogate models are introduced to reduce the time and computational cost of this procedure. A surrogate model is an engineering method implemented when an outcome cannot be measured or computed, so an approximated mathematical model is used. Surrogate models approximate the target function as closely as possible, in order to predict the behavior of the model while requiring less computational resources. The surrogate model is built on data obtained from high-fidelity simulations. Fender et al. [10] proposed a substitute model to predict the force-deformation curve of a component under compression. The substitute model is built starting from a mathematical description of the physics involved. The unknown parameters are then determined through an optimization process. Common surrogate models are radial basis models and kriging [11].

Radial Basis Models The radial basis function (RBF) surrogate model represents the interpolating function as a linear combination of basis functions, one for each training point. The basis function depends on the distance between the prediction and the training point. The coefficients of the basis functions are computed during the training stage. The RBF model is defined by the equation:

$$y = p(x)w_p + \sum_{i}^{nt} \phi(x, xt_i)w_r \tag{3.6}$$

where $\mathbf{x}$ is the input vector, xt_i is the input vector for the i-th training point, $\mathbf{p(x)}$ is the polynomial basis function, $\phi(x, xt_i$ is the radial basis function, w_p is the vector of polynomial coefficients, and w_r is the vector of radial basis function coefficients. The RBF is able to fit many different functions, but it returns a poor performance for a Design of Experiments (DoE) analysis.

Kriging (KRG) model Kriging is an interpolating model that combines a known function f(x) and a stochastic process Z(x):

$$y = f(x) + Z(x) \tag{3.7}$$

"$Z(x)$ is a realization of a stochastic process with mean zero and spatial covariance function given by:

$$cov[Z(x^i), Z(x^j)] = \sigma^2 R(x^i, x^j) \tag{3.8}$$

where σ^2 is the process variance, and R is the correlation." [11]
The Kriging model has a good performance working with less than 50 variables. In the case of a single design variable, it reduces to a simple line-fitting model, as

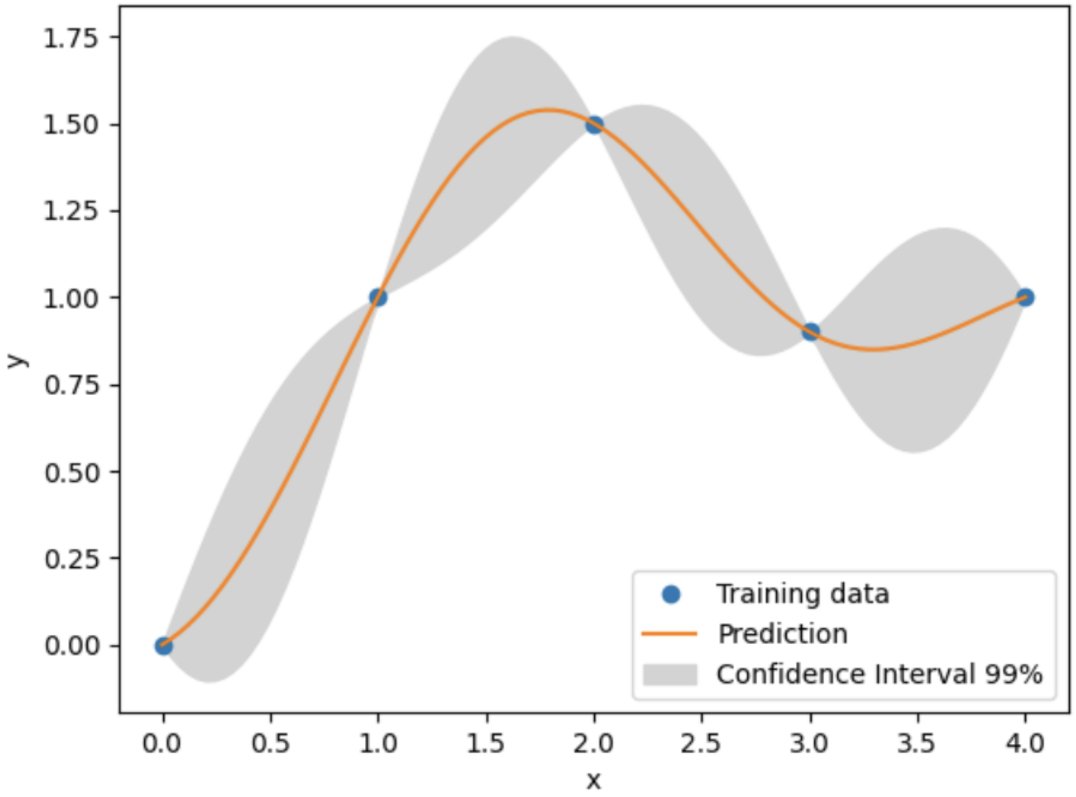
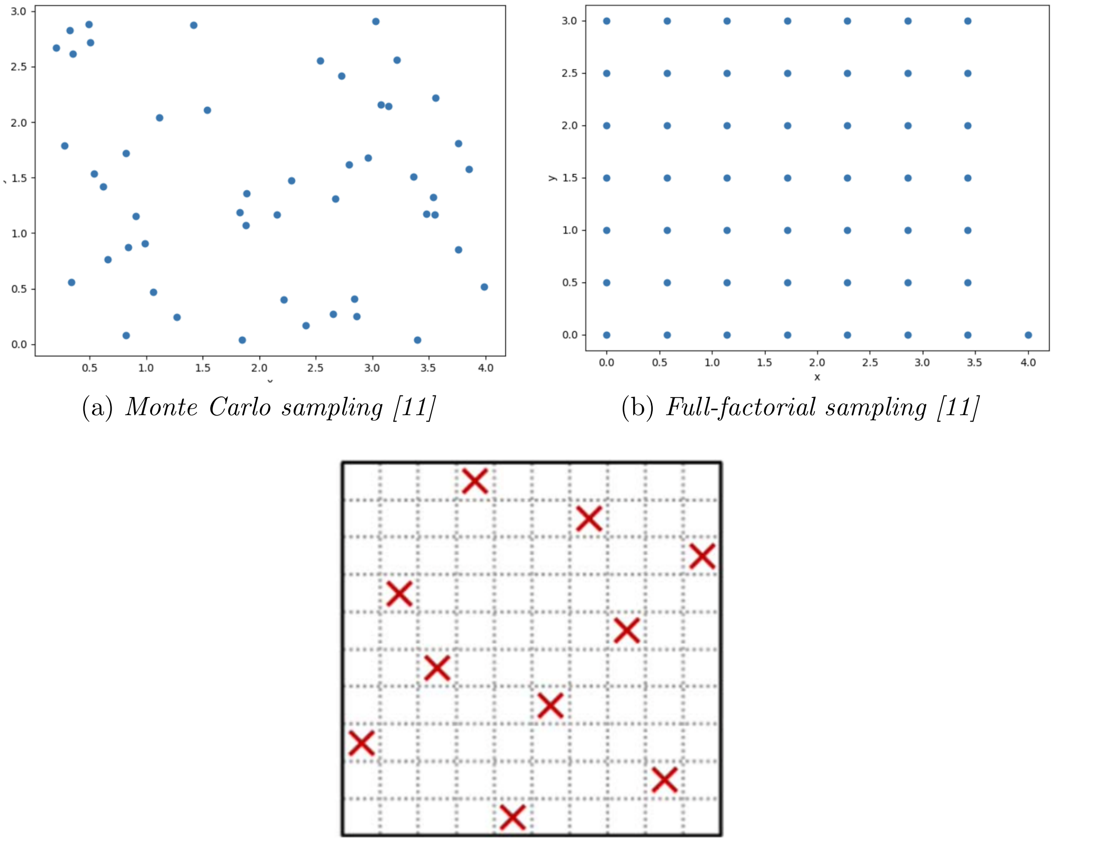


Figure 3.6: Line-fitting model obtained through Kriging [11]

shown in Figure 3.6. In this thesis, we focus on building a surrogate model based on data collected from FE drop-tower tests. Considering that each component will be defined by 2 to 4 design variables, the Kriging model is optimal for the case study.

Sampling strategy The main challenge in surrogate model training is the definition of the training samples. The accuracy of the model highly depends on the location and the number of samples in the design space. Komeilizadeh et al. [12] presented a comparative study between different sampling strategies. The sampling strategies can be divided into two categories, one-shot and sequential sampling. One-shot sampling strategies are based on a one-time definition of the location of the sample points. The most common strategies are (Quasi) Monte Carlo ((Q)MC) and full-factorial. Sequential sampling refers to methods in which the selection of sample points depends on the information derived from previous samples. The most common strategy is Latin-Hypercube sampling (LHS). Figure 3.7 shows an example of sampling using these three different strategies.



(a) *Monte Carlo sampling [11]*

(b) *Full-factorial sampling [11]*

(c) *Latin-Hypercube sampling [13]*

Figure 3.7: Different sampling strategy to define the training points.

The MC sampling strategy is based on a pseudo-random choice of samples in the design space. The simplicity of the approach leads to a high popularity of the method. However, a strategy of this type could lead to a low uniformity of the sample distribution in the design space. More efficient methods, the Quasi Monte Carlo (QMC) strategies, were developed The QMC sampling strategies use a low-discrepancy sequence to guarantee a faster rate of convergence. Full-factorial strategies create samples equidistant in the design space in each dimension. This method guarantees a uniform distribution of the sample points. However, the absence of randomness in the location of the samples could cause a loss of information on the component behavior. The Latin-Hypercube strategy offers a chess-like approach to the choice of the sample points. Each dimension of design space is first divided into N sections of equal probability, with N being the number of samples required. For a case study with d dimension, the design space is divided into N^d cells. N cells are randomly selected, limiting the number of samples in each section to one. Referring to Figure 3.7, this strategy can be compared to a chess board in which N rooks (sample points) are not threatening each other.

Training and validation strategy Validation of the surrogate model is a key factor in identifying the capability of the substitute model to predict the behavior of the component. A common strategy for validation is the subdivision of the entire data set into two branches, a training set and a test set. This strategy is possible with a large data set. In engineering problems, obtaining data from simulations could be proved to be computationally and time-expensive. Validation strategies suitable for small data sets have been proposed, under the name of k-fold cross-validation. K-fold cross-validation proposes to split the data set into k consecutive folds. Each fold is then used once as validation while the k-1 remaining folds constitute the training set. Leave-one-out is a particular case of k-fold cross-validation, with the parameter k equal to 1. Cross-validation returns a biased estimate of the performance of the model due to the reduced training data set. The performance of the model is estimated on a training set of $100\frac{(k-1}{k}\%$ of the available data, rather than on 100%. So, a model with its performance estimated through k-fold cross-validation, and then trained on all of the data for operational use, will perform slightly better than the cross-validation estimate suggests. Leave-one-out cross-validation is approximately unbiased due to the small difference between the training set used in each fold and the entire data set [14]. A decrease in the parameter k provides a better estimation of the model performance. However, a low value of k means a higher number of iterations required for the estimation of the performance. So, the cross-validation can be computationally costly. It is fundamental to balance the increased bias, due to the difference in the dimension of the training set, and the computational cost required by each iteration.

Optimization strategy Studying the effects of the uncertainties in the design of a frontal crash structure introduces an optimization problem. The optimization normally requires many evaluations of a black-box function, which connects the inputs of our system (the design parameters of the structure) with the desired output (force-deformation curve or energy absorbed by a component). Each evaluation, in the engineering field, requires an expensive FE simulation. While the introduced surrogate models can partially relieve the high computational cost required, an optimization of the output will still present a challenge. This is due to the difficulty of fine-tuning the parameters of the approximating function. So, optimization methods based on machine learning algorithms have been presented. The use of machine learning algorithms allows an automatic procedure to define the hyper-parameters of the optimizer. An optimization method based on machine learning can be implemented through [15], based on Bayesian Optimization. The Bayesian Optimization is a common algorithm thanks to its performance in numerous challenging benchmark functions.

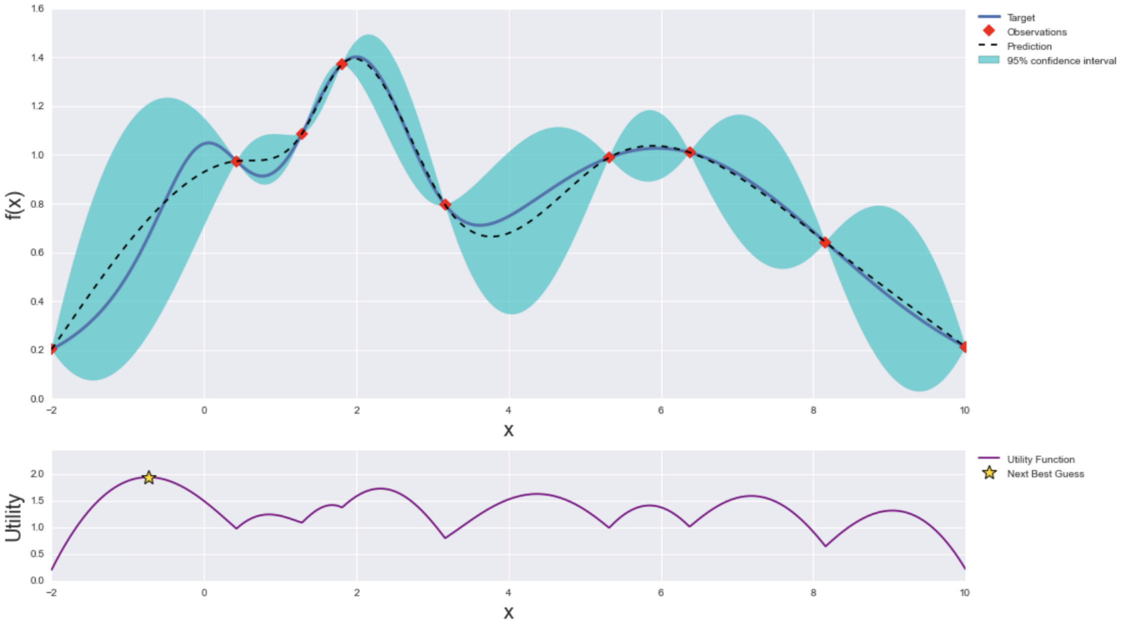


Figure 3.8: Gaussian Process (GP) and utility function after 9 steps [15]

Bayesian Optimization [16] works assuming that the unknown function can be sampled from a Gaussian process. The algorithm constructs a posterior distribution of functions that best describes the function to optimize. The algorithm identifies the regions of the parameters space that are worth exploring, based on the results of each observation point. At each iteration, the algorithm decides between exploiting the results obtained and exploring new regions, based on the information on the target function. The Gaussian Process is fitted to the sample points at each iteration, and the posterior distribution is used to identify the next sample point. This process allows the minimization of the sample points required to optimize the target function.

Chapter 4

Methodology

In this section, the methodology followed to study the effect of the uncertainties in a frontal crash structure on an early design of one component is presented. It is first introduced a simplified 1D FE model of a frontal crash structure, composed of 2 parallel load paths and a connecting element. Then, one component is replaced with a 2D mesh, and the process followed to measure the effect of the uncertainties in terms of force and deformation measured is presented. A few examples of keyword files used to implement the model in LS-Dyna and the code written to run the uncertainty quantification are provided in the Appendix.

4.1 FE Model

Creating a simplified 1D Finite Element model that can be used for Solution Space analysis requires multiple steps, from the generation of a new GSM and the relative DSM to the computation of the Solution Space and its implementation.

This thesis focuses on the implementation of a simplified model on a software for FE Simulation, LS-Dyna. It aims to gradually reduce the initial assumptions and perform an Uncertainty Quantification analysis on the model. An existing GSM and DSM, and the methodology to compute the relative Solution Spaces [5], will be slightly modified and used for the following analysis.

Figure 4.1 shows the existing Geometry Space Model used as a starting point for the following work. The model represents the crash-relevant component of the Honda Accord. Only the left side of the frontal crash structure is considered. Each component is divided into two sections, an undeformable part and a deformable one. A connecting element is present between the two longitudinal beams, and a lumped mass m_1 is added between components 5 and 6. A brief explanation of the mathematical model for the Solution Space method will be presented in this section. For a more in-depth analysis, refer to the Chaudhry Taimoor Niaz thesis [5].

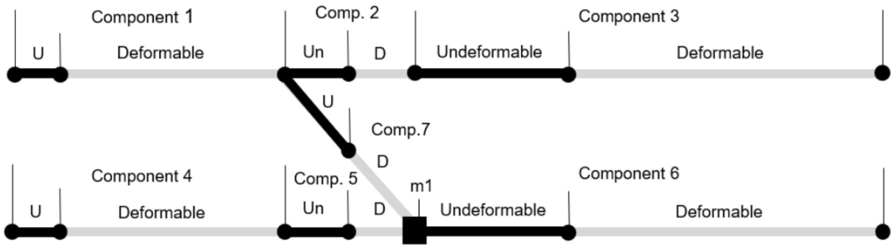


Figure 4.1: Existing Geometry Space Model [5]

Mathematical Model. The mathematical model of the components, fundamental to derive the Solution Space of our system, is built while keeping into account the three constraints as stated in [2, 3, 17]. The approach stated by [3], which aims to obtain a linear set of equations, is used. The constraints imposed on the system are as follows, as already presented in Section 3.1:

- Minimum energy absorption: the main objective of the front structure of a vehicle is to protect the passengers by absorbing the kinetic energy of a crash in its entirety.
- Maximum acceleration: a critical acceleration has to be imposed to limit serious crash-related injuries. A cautious threshold of 0.3 mm/ms^2 is applied for this model.
- Progressive order of deformation: the front of the car must be the first part to deform in a crash. This criterion is imposed to increase the repairability of the vehicle in case of a low-speed crash.

In addition to the requirements just presented, particular attention must be paid to component 7. The solution space method allows the study of a linear system derived from different load paths. However, the mathematical model proposed by Fender et al. [3] is based on parallel load paths. For this reason, the implementation of component 7, which connects the two load paths, needs to be revised. We use the analytical approach proposed by Chaudhry Taimoor Niaz [5] in his thesis work. An angular correction approach is followed, creating a third independent load path. This correction aims to define the horizontal force of component 7 with a simple trigonometric formulation:

$$F_{hor} = \cos\alpha \cdot F_7 \tag{4.1}$$

The constraints applied to our system are arranged in an equation of the form:

$$Ax < b \tag{4.2}$$

The matrix A is composed of m rows and n columns of coefficients, derived from the equations of the constraints. Equation 4.3 shows the expanded matrices, with the coefficients referred to the existing GSM presented in Figure 4.1. Each row indicates an equation needed to apply the constraints on the system, while each column refers to a different section of the model.

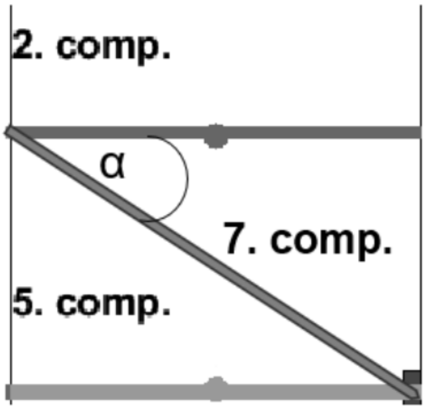


Figure 4.2: Component 7 connecting the load paths, and the correcting angle [5]

$$\begin{bmatrix} \frac{-s_1}{m_t} & \frac{-s_2}{m_t} & \frac{-s_3}{m_2} & \frac{-s_4}{m_t} & \frac{-s_5}{m_t} & \frac{-s_6}{m_2} & \frac{-\alpha \cdot s_7}{m_t} \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & \alpha \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & \text{-}\alpha \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & -\beta & 0 & 0 & \gamma & -1 & \alpha \cdot \gamma \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \\ F_5 \\ F_6 \\ F_7 \end{bmatrix} = \begin{bmatrix} \frac{-v_0^2}{2} \\ (m_t) \cdot a_c \\ (m_t) \cdot a_c \\ (m_2) \cdot a_c \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (4.3)$$

Equation 4.3 shows the effect of the lumped mass between components 5 and 6, causing a variation of the acting mass in each section. The value α implements the angular correction for component 7. The values β and γ, instead, are mass correction factors introduced due to the discrete mass point between components 5 and 6. A detailed analysis of the equations required to derive the matrices showed in 4.3 is provided by [5].

Setting up the FE Model. The next step is to implement the model in a software suitable for Finite Element simulations, LS-Dyna, through keyword files. This type of file allows one to quickly modify any characteristic of the model in a notepad, avoiding the loss of time caused by interacting with the software interface. The components are defined by their nodes, the material of the undeformable and deformable parts, and the boundary conditions.

Figure 4.3 shows the FE model built into the software. Few alterations have been implemented with respect to the existing model presented previously: the connection of component 7 on the lower load path has been moved between components 4 and 5, avoiding a direct connection to the lumped mass, and the upper load path has been shortened, to recover the inclination of component 7 that has been lost with the first adjustment.

Each component is composed of a deformable part, highlighted in red and defined by the material card MAT 068 Non-Linear-Plastic-dicrete-beam, and an undeformable part, identified by the black sections and characterized by the material MAT66 that simulates a spring with high translational stiffness. An exception is seen in component 7, which contains a single deformable section. This solution is implemented to avoid undesirable global bending. A more in-depth analysis of the phenomenon can be found in the Discussions section 6.1.

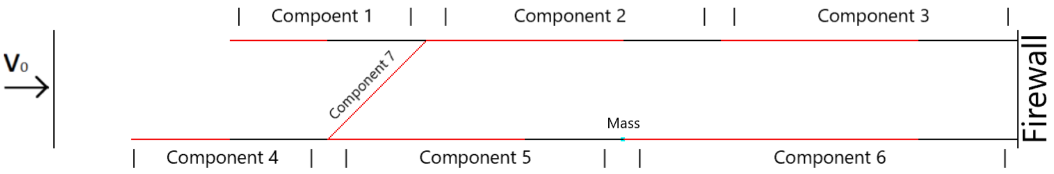


Figure 4.3: FE model implemented in LS-Dyna

The model is composed of 3 different load paths, following the methodology proposed by [5]:

- the 1st load path includes components 1, 2, and 3. This load path refers to the upper longitudinal beam;
- the 2nd load path includes components 4, 5, and 6. This load path refers to the lower longitudinal beam;
- the 3rd load path includes component 7. This load path refers only to the connecting element.

To define the different sections of the model, 3 different nodes have been assigned to each component, while only 2 for component 7. Table 4.1 shows the different nodes that define the model, while Tables 4.2, 4.3 highlights their position. In each component, the nodes n_1 and n_2 delimit the deformable part, while nodes n_2 and n_3 define the undeformable part. The absence of n_3 in component 7 reminds us that the component is defined by a single deformable section. Table 4.4 includes the position of the nodes 9000001 and 9000002. The node 9000001 indicates the starting location of the moving rigid wall, while node 9000002 refers to the position of the firewall, on which the structure is constrained.

Table 4.1: Node ID of each component

Component ID	1	2	3	4	5	6	7
$\mathbf{n_1}$	1000000	1000002	2000003	4000000	4000002	5000003	4000002
$\mathbf{n_2}$	1000001	2000002	3000002	4000001	5000003	6000002	1000002
$\mathbf{n_3}$	1000002	2000003	3000003	4000002	5000003	6000003	

Table 4.2: Node coordinates for components in 1st load path

Node ID	1000000	1000001	1000002	2000001	2000002	2000003	3000001	3000002	3000003
x	-1600.0	-1400.0	-1200.0	-1000.0	-800.0	-600.0	-400.0	-200.0	0.0
y	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
z	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Table 4.3: Node coordinates for components in 2nd load path

Node ID	4000000	4000001	4000002	5000001	5000002	5000003	6000001	6000002	6000003	6000004
x	-1800.0	-1600.0	-1400.0	-1200.0	-1000.0	-800.0	-600.0	-400.0	-200.0	0.0
y	-200.0	-200.0	-200.0	-200.0	-200.0	-200.0	-200.0	-200.0	-200.0	-200.0
z	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

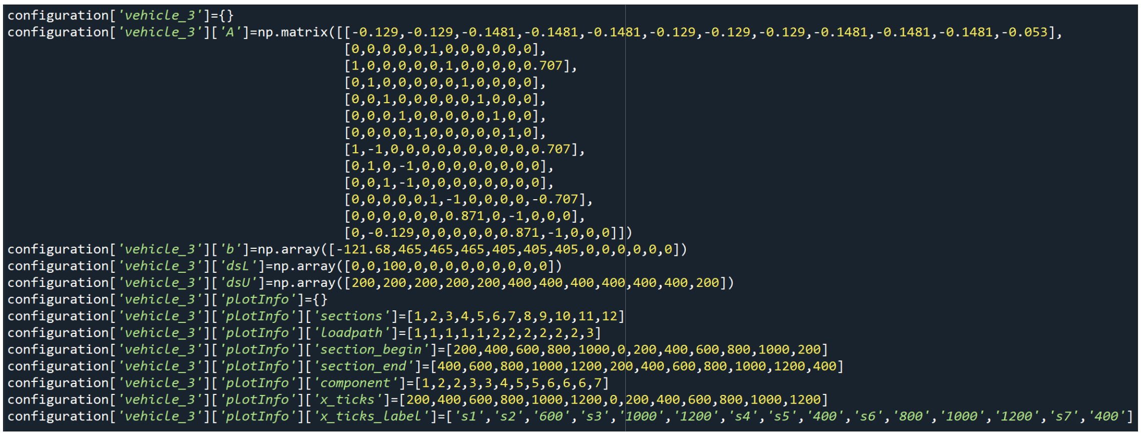
Table 4.4: Node coordinates for rigid wall (9000001) and firewall (9000002)

Node ID	9000001	9000002
x	-1956.0	0.0
y	-200.0	0.0
z	0.0	0.0

The model implemented is composed of 12 different deformable sections:

- 1 section each in components 1, 4 and 7;
- 2 sections in components 2, 3 and 5;
- 3 sections in component 6.

Referring to the mathematical model in matricial form, applied to the model studied, we obtain a 13x12 A matrix. Figure 4.4 shows the coefficients of the matrix A and vector b relative to the FE model implemented in LS-Dyna.



```
configuration['vehicle_3']={}
configuration['vehicle_3']['A']=np.matrix([[-0.129,-0.129,-0.1481,-0.1481,-0.1481,-0.129,-0.129,-0.129,-0.1481,-0.1481,-0.1481,-0.053],
                                           [0,0,0,0,0,1,0,0,0,0,0,0],
                                           [1,0,0,0,0,0,1,0,0,0,0,0.707],
                                           [0,1,0,0,0,0,0,1,0,0,0,0],
                                           [0,0,1,0,0,0,0,0,1,0,0,0],
                                           [0,0,0,1,0,0,0,0,0,1,0,0],
                                           [0,0,0,0,1,0,0,0,0,0,1,0],
                                           [1,-1,0,0,0,0,0,0,0,0,0,0.707],
                                           [0,1,0,-1,0,0,0,0,0,0,0,0],
                                           [0,0,1,-1,0,0,0,0,0,0,0,0],
                                           [0,0,0,0,0,1,-1,0,0,0,0,-0.707],
                                           [0,0,0,0,0,0,0.871,0,-1,0,0,0],
                                           [0,-0.129,0,0,0,0,0,0.871,-1,0,0,0]])
configuration['vehicle_3']['b']=np.array([-121.68,465,465,465,405,405,405,0,0,0,0,0,0])
configuration['vehicle_3']['dsL']=np.array([0,0,100,0,0,0,0,0,0,0,0,0])
configuration['vehicle_3']['dsU']=np.array([200,200,200,200,200,400,400,400,400,400,400,200])
configuration['vehicle_3']['plotInfo']={}
configuration['vehicle_3']['plotInfo']['sections']=[1,2,3,4,5,6,7,8,9,10,11,12]
configuration['vehicle_3']['plotInfo']['loadpath']=[1,1,1,1,1,2,2,2,2,2,2,3]
configuration['vehicle_3']['plotInfo']['section_begin']=[200,400,600,800,1000,0,200,400,600,800,1000,200]
configuration['vehicle_3']['plotInfo']['section_end']=[400,600,800,1000,1200,200,400,600,800,1000,1200,400]
configuration['vehicle_3']['plotInfo']['component']=[1,2,2,3,3,4,5,5,6,6,6,7]
configuration['vehicle_3']['plotInfo']['x_ticks']=[200,400,600,800,1000,1200,0,200,400,600,800,1000,1200]
configuration['vehicle_3']['plotInfo']['x_ticks_label']=['s1','s2','600','s3','1000','1200','s4','s5','400','s6','800','1000','1200','s7','400']
```

Figure 4.4: Configuration of the model implemented in Python

Other relevant pieces of information provided by the configuration are the limits imposed on the solution space corridors, expressed in kN. The limits of the solution space corridors are chosen arbitrarily, based on the results expected. Due to the nature of the model, higher forces are expected on the 2nd load path, up to double the values measured on the 1st load path. Tables 4.5 and 4.6 summarise the relevant information provided by the configuration. Additionally, the initial velocity of the rigid wall and the value of the lumped mass and of the rigid wall are provided. The limits of the design space parameters are provided referring to the different load paths.

Table 4.5: Limit of the design space parameters

Load path	1st	2nd	3rd
Upper Limit [kN]	200.0	400.0	200.0
Lower Limit [kN]	0.0	0.0	0.0

Table 4.6: Model parameters

Quantity	v_0	a_c	m_1	m_2	s
Value	15.6 mm/ms	0.3 mm/ms^2	1350 kg	200 kg	200 mm

The linear system can be implemented in a Python script, written by Ascia Paolo, that allows the resolution of the system and the derivation of the corridors of each section of the model.

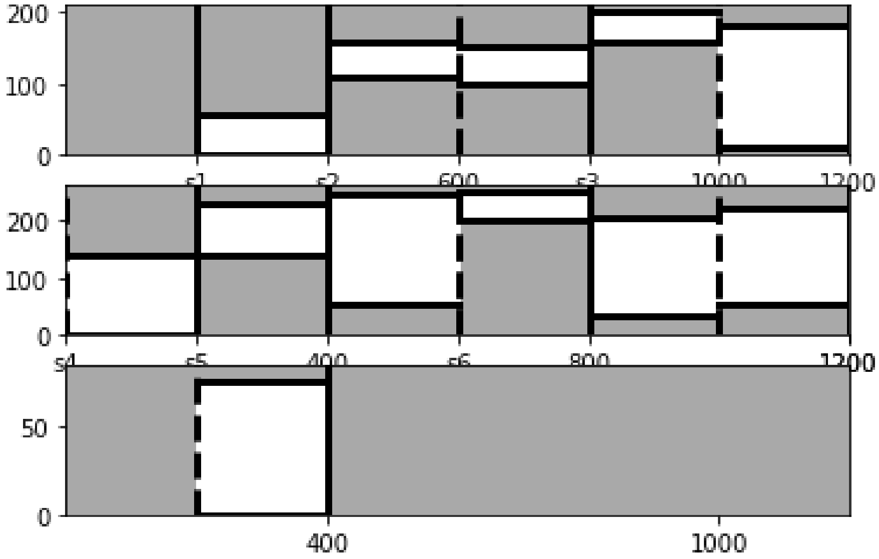


Figure 4.5: Solution Space corridors of the model. Each graph refers to one load path, from 1st to 3rd (up to bottom).

Table 4.7: Solution space corridor relative to 1st load path

Section	1	2	3	4	5
Component	1	2	2	3	3
Upper limit [kN]	57.13	158.12	152.03	200.00	181.13
Lower limit [kN]	0.00	110.03	100.00	158.13	10.25

Table 4.8: Solution space corridor relative to 2nd load path

Section	6	7	8	9	10	11
Component	4	5	5	6	6	6
Upper limit [kN]	138.83	230.68	246.98	252.96	205.00	223.86
Lower limit [kN]	0.00	138.83	55.65	200.93	34.11	52.97

Table 4.9: Solution space corridor relative to 3rd load path

Section	12
Component	7
Upper limit [kN]	78.80
Lower limit [kN]	0.00

Figure 4.5 is divided into three different plots, one for each load path considered. The x-axis of the plot represents the deformation of each load path in millimeters, while the y-axis the force values measured in kN. The white box represents the solution space for each section, indicating the range of acceptable forces on the component. The limits of the solution space corridors are summarized in Tables 4.7, 4.8 and 4.9.

The energy absorbed by each component can be imposed through the material keyword. Figure 4.6 and 4.7 shows an example of the force-deformation curve imposed on one component, and how it is implemented in the keyword file.

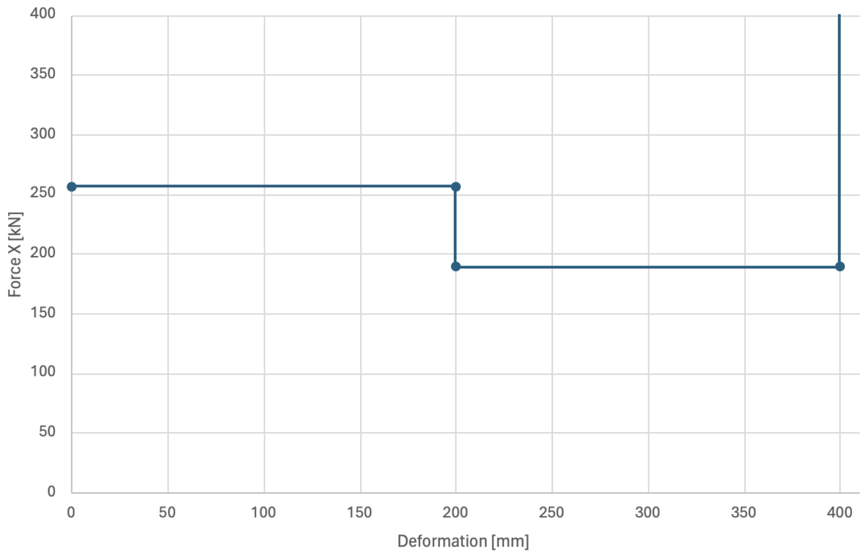
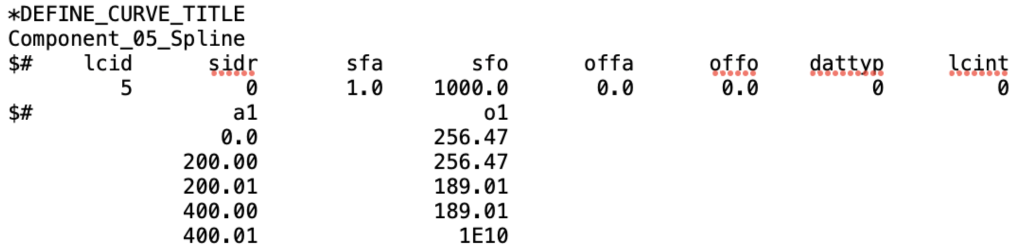


Figure 4.6: Force-deformation curve imposed on component 5



```
*DEFINE_CURVE_TITLE
Component_05_Spline
$#    lcid      sidr       sfa       sfo      offa      offo    dattyp     lcint
         5         0       1.0    1000.0       0.0       0.0         0         0
$#                a1                  o1
                 0.0              256.47
              200.00              256.47
              200.01              189.01
              400.00              189.01
              400.01                1E10
```

Figure 4.7: Material keyword referred to component 5 implemented in LS-Dyna

It is important to highlight a characteristic of the force-deformation curve shown. We impose a limit on the deformation of the component by requesting a force of several orders of magnitude, not achievable in a real scenario, to obtain higher deformations than the one desired by the model. This limit is imposed in order to give information to the software on how the component behaves after it is completely compressed. A more detailed analysis of the phenomenon, and the reasons why this limit has to be imposed, is provided in Section 6.2.

The model is tested by implementing in the material card the lower limit of each corridor, to verify that the model is able to absorb the kinetic energy of the impact. This constraint is verified if the rigid wall velocity is 0 m/s at the end of the impact. An additional test is performed imposing the upper limit of the solution space corridors, verifying that the critical acceleration is not exceeded during the impact. Figures 4.8 and 4.9 show the rigid wall velocity and acceleration during the impact. The rigid wall velocity reaches 0 m/s, indicating that the kinetic energy of the impact has been absorbed. However, the velocity stabilizes at a negative velocity of -0.2 m/s. This effect is caused by the relaxation of the crash structure, which pushes the rigid wall away from the vehicle. This phenomenon reproduces the small bouncing effect of a real vehicle after an impact. The acceleration registers a peak higher of 0.3 mm/ms^s for a short period of time but stabilizes to values close to the critical acceleration.

Additional results obtained in the simulations are discussed in Section 5.1 to better understand the behavior of the model.

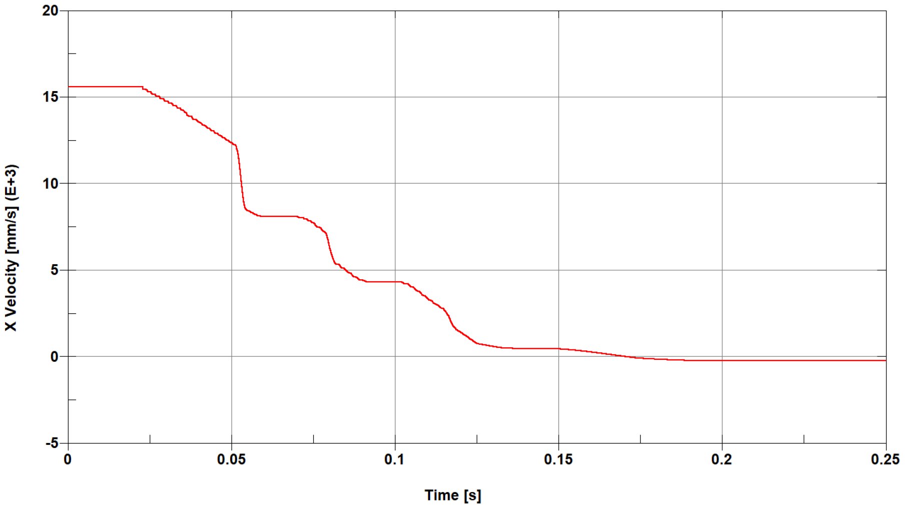


Figure 4.8: Rigid wall velocity during the impact

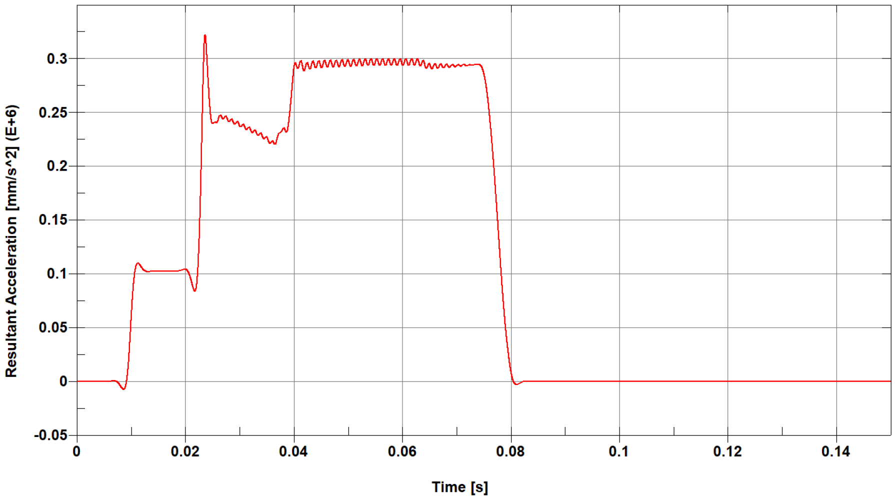


Figure 4.9: Acceleration-time curve during the impact

4.2 Uncertainty Quantification

The model shown in the previous section is based on multiple assumptions that allow the implementation of each component as a 1D element. However, the simplified model is not able to perfectly replicate the behavior of the components in a real environment and the effect that each component has on the entire system. In this section, we will describe how we can measure the effect that different designs of the components have on the system. In order to perform this analysis, we will first replace one component in the model with a 2D mesh. Then, we will describe the behavior of each 1D component with a surrogate model. The surrogate model will be able to connect the design characteristics of the component to the force absorbed and the resultant deformation. We will fix the design of the component studied, analyzing the effect of the design choices on the solution space of the model. In the end, we will propagate the design uncertainty of the system and analyze the robustness of the design chosen. The analysis will be performed by studying the range of variation of the force measured on the component studied.

4.2.1 Implementing the 2D mesh of a component

The simplified 1D FE model just presented can be expanded by implementing the 2D mesh of a component. This is useful for analyzing the behavior of a real component in the system by measuring the compression forces and understanding the effect of the entire system on it. Figure 4.10 shows an example of a component implemented in LS-Dyna. The component is defined by a linear order mesh with quadrilateral elements. This type of element is chosen thanks to its performance compared to triangular elements. Each color identifies a different section, reflecting the simplified 1D model implemented previously. The differentiation of the sections allows setting a different thickness in each section, obtaining a variable capability to absorb energy along the component. The component is characterized by a rectangular cross-section of dimension 120x70 mm.

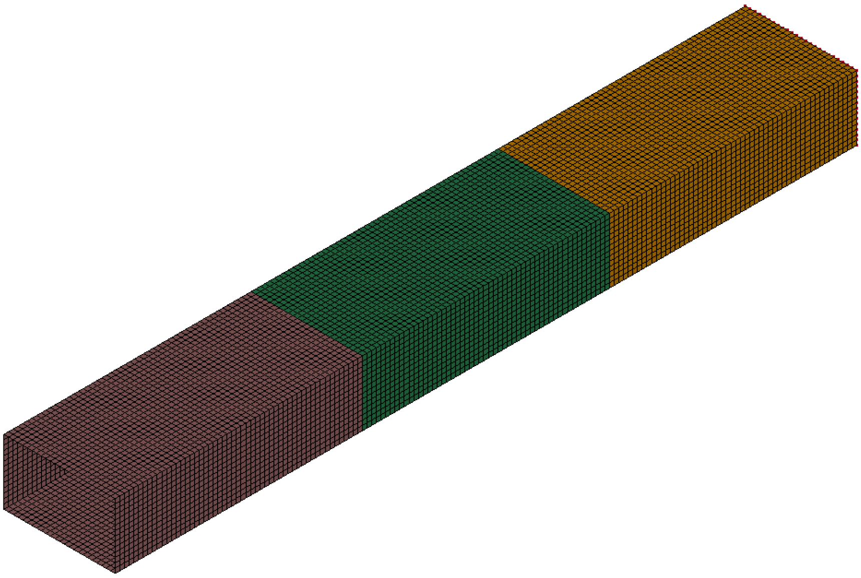


Figure 4.10: Implementation of a 2D mesh of component 6

The capability of the component to absorb energy is defined by the material, implemented through the material card MAT 024 Mat-Piecewise-Linear-Plasticity. This keyword defines an elastoplastic material ”with an arbitrary stress as a function of strain curve and arbitrary strain rate dependency” [18]. Figure 4.11 shows the stress-strain curve implemented in the model, that will be applied to every 3D component used for the following analysis.

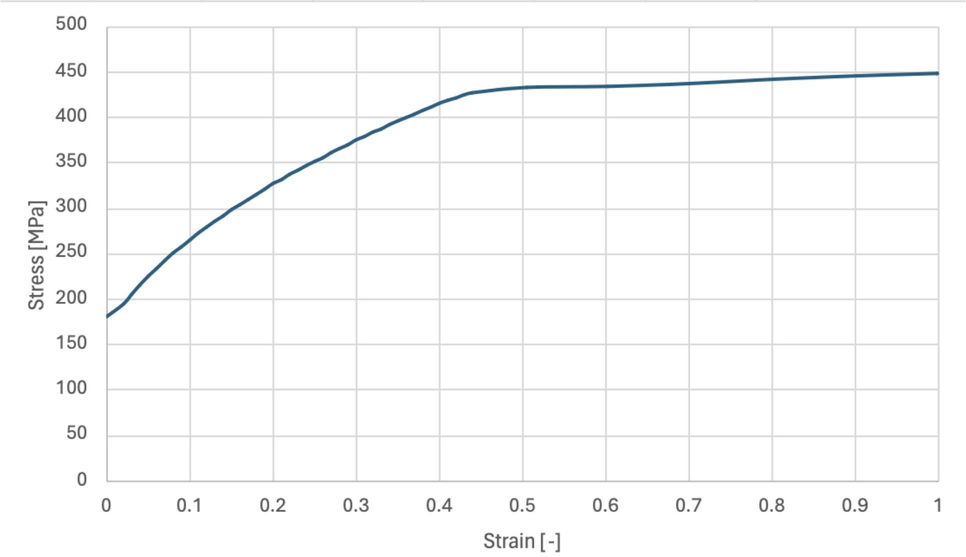


Figure 4.11: Stress-strain curve implemented in the material card

The component is connected to the model through a rigid connection, applied with the keyword ”Constrained nodal rigid body”, as shown in Figure 4.12. The keyword imposes a master-slave connection between the node that delimits the 1D component and each node of the mesh selected in the node-set.

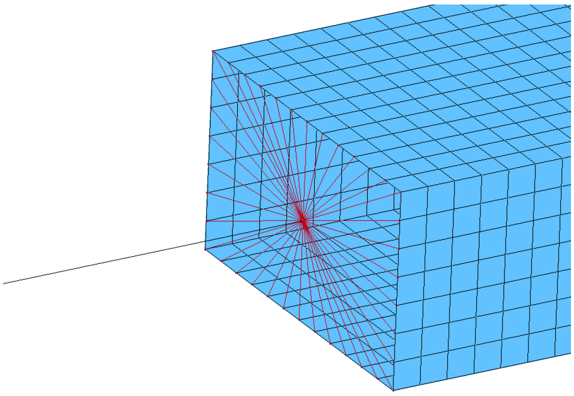


Figure 4.12: Rigid connection between the 3D component and the 1D FE model

The connection between the 2D component and the monodimensional elements is verified by running an FE simulation. Figure 4.14 shows the model during the impact at different time steps. In the simulation performed, we tested the connection between the 2D mesh of component 6 and the undeformable section of component 5. The purpose of the simulation is that monodimensional elements can correctly transfer the forces to the mesh. Component 6 correctly deforms, reaching a maximum deformation of around 600 mm and leaving an undeformable section of almost 200 mm. However, it is possible to highlight that not every monodimensional component is fully compressed. While it can be partially attributed to a not-optimized design of component 6, this phenomenon is caused by the force-deformation curves implemented in the 1D elements. Differently from the curve shown in section 4.1

a force-deformation curve without the implementation of the limit on the deformation, shown in Figure 4.13, is tested. This type of curve caused an interpenetration between the different elements of the components. However, as the goal of the simulation is to simply verify the connection between the 2D component and the 1D elements, the result is considered positive.

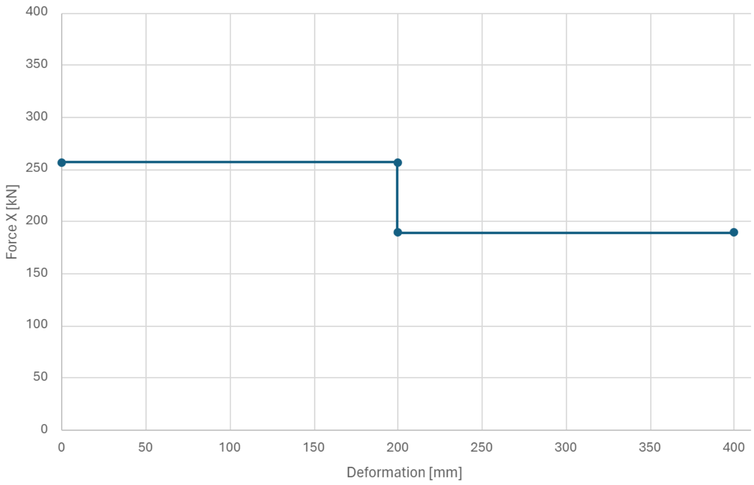
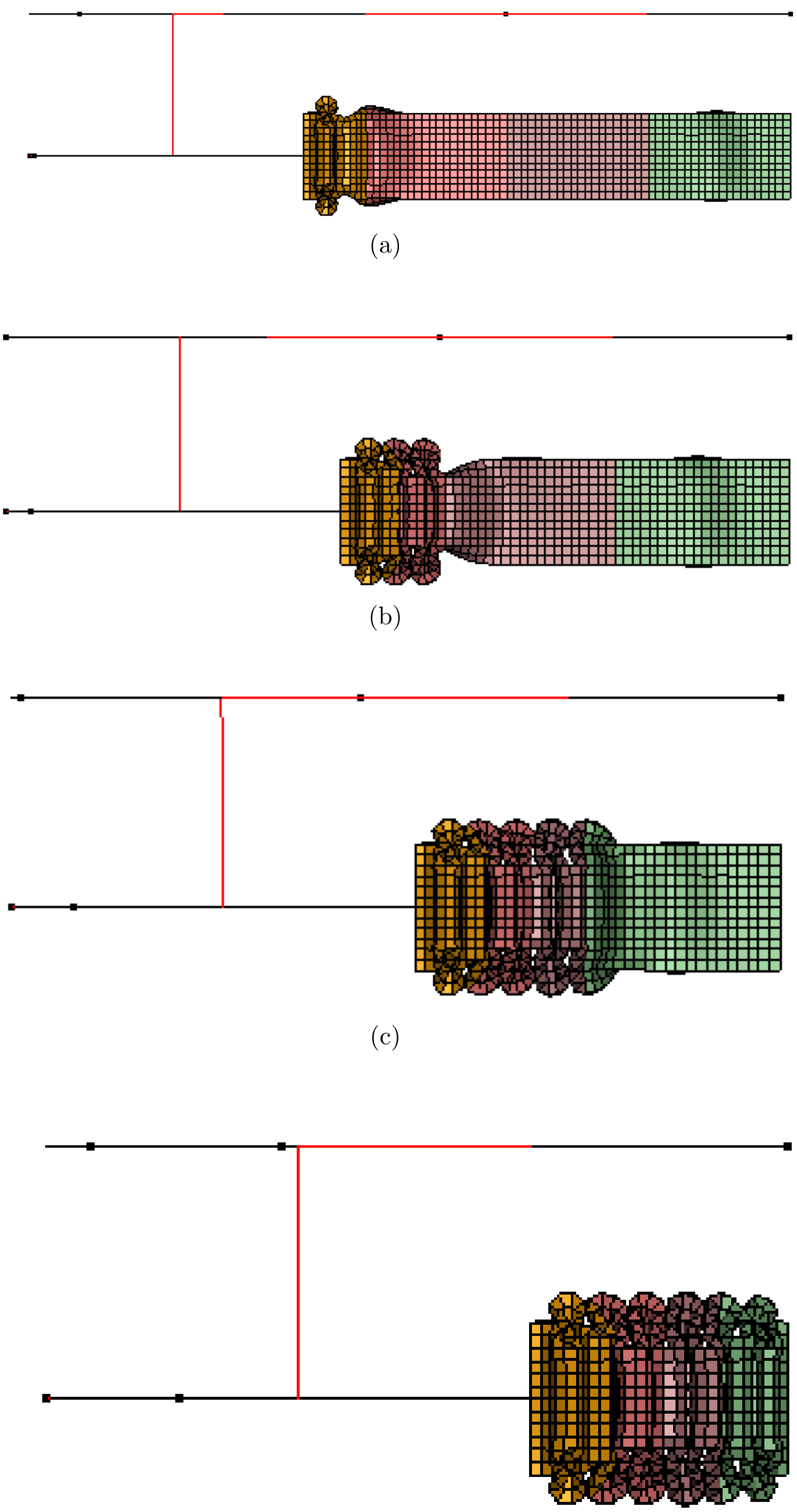


Figure 4.13: Example of force-deformation curve without imposing the limit on the maximum deformation.



(a)

(b)

(c)

(d)

Figure 4.14: Frontal impact simulation of the FE model with the 2D mesh of component 6.

4.2.2 Creating a surrogate model

In the simplified 1D FE model presented in section 4.1, the behavior of the components was reduced to a constant force absorbed during the entire length of each section. This assumption can be reduced by introducing surrogate models. The surrogate model aims to predict the force-deformation curve relative to the components. The model can be trained on data from drop tower tests performed individually on each component in a FE simulation. The procedure of the drop tower test is presented by van Mierlo et al. [19]. These tests consist of a compression due to the impact of a rigid wall, that simulates the mass dropped in a real test. Figure 4.15 shows an example of a drop tower test implemented in LS-Dyna.

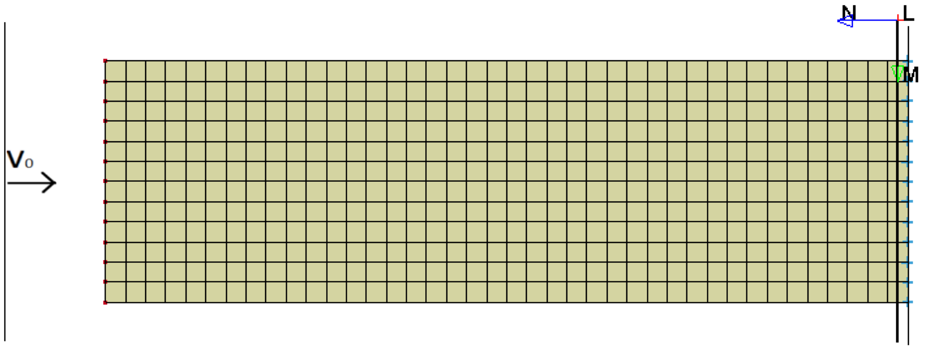


Figure 4.15: Drop tower test performed on component 1

One edge of the component is fixed. This constraint on every DOF is applied on each node of the edge. The energy absorbed by the component can be chosen by variating the mass and velocity of the rigid wall. The compression forces are measured at a cross-section set 10 mm from the component base. As we expect the measured force to act mainly along the axis of the component, we measure only the force contribution on the x-axis. Figure 4.16 shows the constraints applied on the nodes and the cross-section used to measure the force. The opposite edge of the component is connected to a node with a rigid connection, applied with the keyword ”Constrained modal rigid body”. The node is created at the intersection between the frontal plane of the component and its axis. The central node has 5 DOF constrained, allowing only translation along the x-axis. Figure 4.17 shows the rigid connection and the central node implemented to apply the constraints (Node 1100000 for component 1).

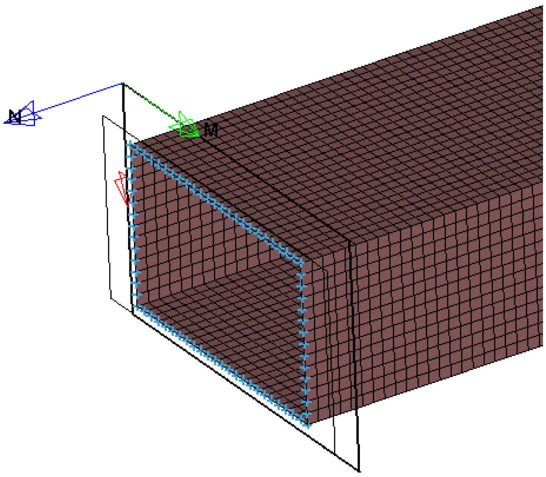


Figure 4.16: Cross section for force measurement and constraints on each node of the edge of the component

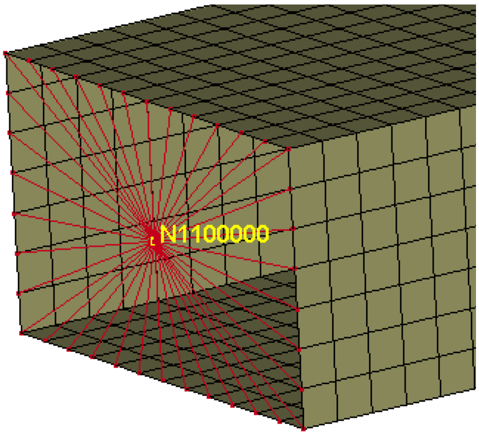


Figure 4.17: Rigid connection between the 3D component and the 1D FE model

Component 7 specifics. Component 7 needs particular attention due to the nature of its deformation. As the component faces not only compression forces but also tangential forces that allow its axis to rotate in the global reference frame, we need to measure both the x and y forces and evaluate the total force at each timestep. The model used for the drop tower test presents also many differences, to reflect its behavior in the entire system. Figure 6.3 shows the model implemented for the test on component 7. In this test, the rigid wall collides with a longitudinal beam composed of 2 undeformable sections (black segments) and 1 deformable section (red segment). The energy absorbed by the monodimensional deformable section is imposed to 0 J, in order to avoid any effect on the component. Its length is equal to the distance on the x-axis between the center node of each edge of component 7, 200 mm.

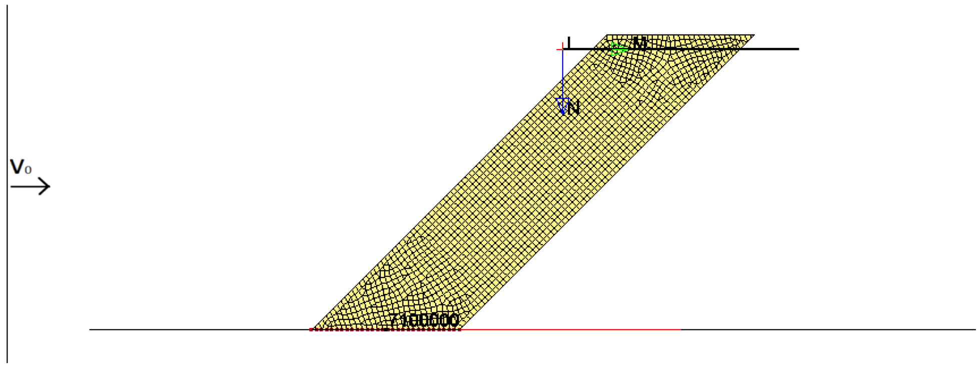


Figure 4.18: Drop tower test performed on component 7

The length of the component at each time step is measured through the Pythagorean theorem, and shown in the following equation:

$$l_7 = \sqrt{(200 - d_{7100000})^2 + 200^2} \tag{4.4}$$

where l_7 is the length of component 7, $d_{71000000}$ is the deformation of the central node along the x-axis and 200 mm is the distance between the two edges of the component. Following this equation, the deformation of component 7 is evaluated as:

$$d_7 = l_{in} - l_7 \tag{4.5}$$

where d_7 is the deformation of the component and $\mathrm{l_{in}}$ is the initial length of component 7, 282.84 mm.
Similar to the deformation, the force along the axis of deformation has to be evaluated at each time step from the x and y forces measured at the cross-section. The contribution of each force to the compression of the component varies at each time step. As a first step, we need to compute the angle between the axis of the component and the cross-section, already shown in Figure 4.2 and evaluated as:

$$\alpha = \arctan(\frac{200}{200 - d_{7100000}}) \tag{4.6}$$

Then, the axial force can be computed as the sum of the contribution of the x and y forces:

$$F = F_x \cdot \sin\alpha + F_y \cdot \cos\alpha \tag{4.7}$$

Generating the surrogate model. The inputs of the drop tower test are the mass of the rigid wall and the thickness of each section of the component, while the velocity of the rigid wall is kept constant at 5.56 m/s for components 1 and 7, and 11.1 m/s for components 3 and 6. The range of the DoE on the mass of the rigid wall is chosen to reflect the limits imposed on the design space shown in Table 4.5. The limits in the thickness of each section, instead, are imposed by the available manufacturing technology [6]. The input is randomized in the selected range through a DoE, implemented with the library SciPy [20]. The sampling strategy applied for generating the DoE is the Latin hypercube sampling (LHS), allowing the reduction of the number of samples needed. Table 4.10 shows the range of variations of the parameters and the number of points of the DoE relative to each component. Figure 4.19 shows the DoE obtained through an LHS strategy for the surrogate models of component 1. The DoE referred to other components are not shown due to the higher number of dimensions, making their representation difficult in this thesis.

Table 4.10: DoE for drop tower tests

Component	1	3	6	7
Velocity of rigid wall[m/s]	5.56	11.1	11.1	5.56
Mass [ton]	[0.013, 5.184]	[0.006, 2.593]	[0.009, 3.889]	[0.005, 1.072]
Thickness section 1 [mm]	[0.7, 4.5]	[0.7, 4.5]	[0.7, 4.5]	[0.7, 4.5]
Thickness section 2 [mm]	-	[0.7, 4.5]	[0.7, 4.5]	-
Thickness section 3 [mm]	-	-	[0.7, 4.5]	-
Number of points	60	60	80	60

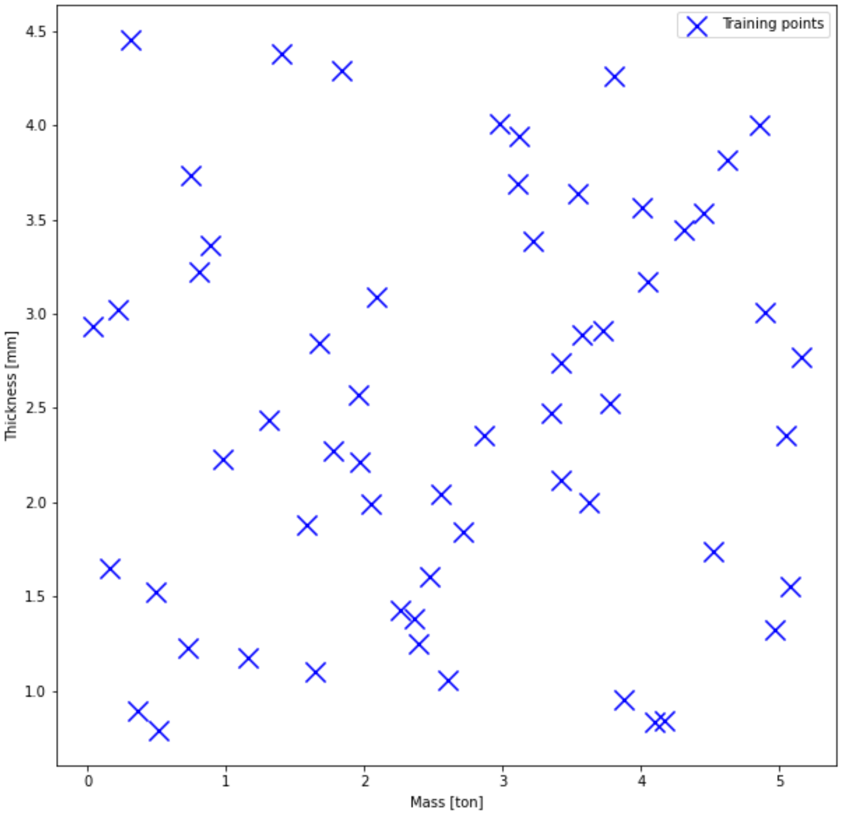


Figure 4.19: DoE for the surrogate models of component 1.

The surrogate model is trained only on components 1, 3, 6, and 7 as the components with the same number of sections are equivalent. The output provided by the drop tower tests consists of two vectors, the force absorbed and the deformation. The two vectors are combined to obtain a force-deformation curve, which is interpolated with a b-spline through the Python library SciPy [20]. The interpolation is performed on a different number of points, based on the complexity of the component, and so of the force-deformation curve generated:

- 15 points for component 7;
- 20 points for components 1 and 3;
- 25 points for component 6.

The b-spline was chosen as interpolating curve due to the parameters that are generated as outputs, the knots and the coefficients. The knots provide information on the deformation of the components, while the coefficients provide information on the curve dependent on the deformation. Using the information provided by the knots, it is possible to quickly identify the inputs that generate the desired deformation. Figure 4.20 shows a force-deformation curve obtained with an FE simulation of a drop tower test on component 1 and the interpolating curve used to train the surrogate models. The crashing force has been filtered to reduce the number of interpolating points needed to generate the curve, resulting in a reduced number of knots and coefficients of the b-spline. This solution can be implemented as the reduction in energy absorbed is negligible.

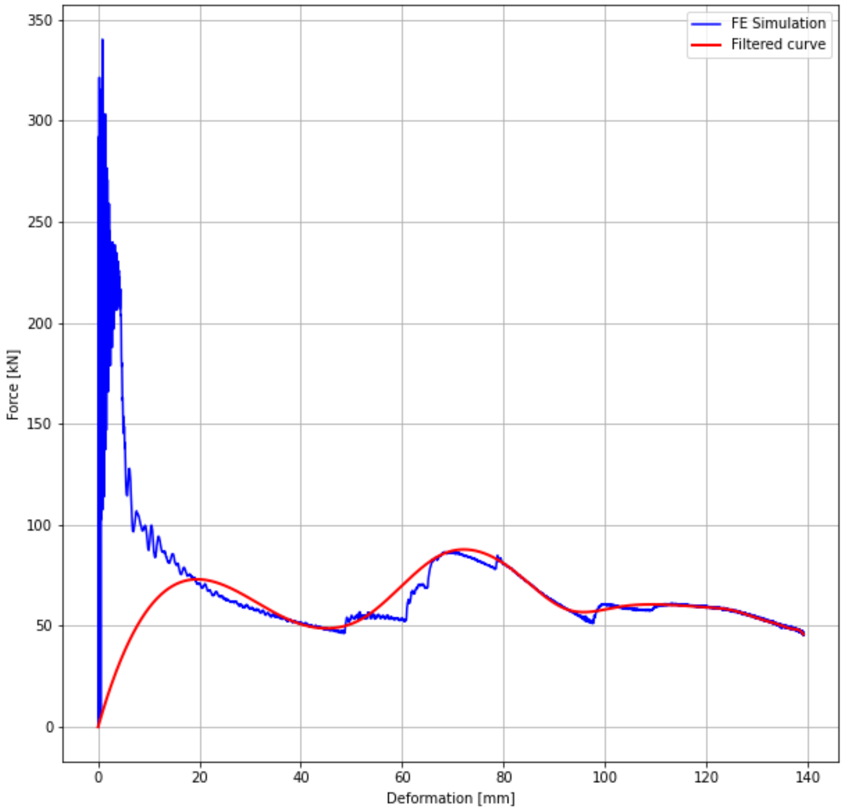


Figure 4.20: Comparison between the F-def curve obtained in FE simulation and the interpolating curve.

Two surrogate models for each component are trained on these sets of parameters, using the Kriging method implemented with the Python library SMT [11].

Training and validation of the surrogate models are performed using the Leave-One-Out cross-validation, implemented with the library scikit-learn [21]. This methodology allows us to iteratively subdivide the results obtained from the DoE samples in train and test sets. At each iteration, one sample is used to validate the model, while the other samples are used for training. The error of the model is computed at each iteration as the normalized root mean squared error (NRMSE) of the prediction. The global error of the model is evaluated as the average of the NRMSE obtained at each iteration:

$$NRMSE_j = \frac{y_{j,pred} - y_{j,act}}{max(y_{j,act})} \tag{4.8}$$

$$NRMSE = \frac{\sum_i^N NMRSE_j}{N} \tag{4.9}$$

where j indicates the current iteration, y_{pred} and y_{act} are the vectors of the predicted and actual coefficients or knots, and N is the number of iterations performed. As the methodology used for validation is leave-one-out, the number of iterations is equal to the number of sample points for the component. This methodology for training/validation is used due to the low sample size of the data set available. Table 4.11 shows the performance obtained by each surrogate model trained. The results are discussed in Section 5.2.

Table 4.11: Performance of surrogate models

Component	1	3	6	7
Knots NRMSE	0.226	0.244	0.060	0.279
Coefficient NRMSE	0.177	0.261	0.381	0.229

4.2.3 Fixing an early design for component 2

The component selected for the purpose of our analysis is the component 2. The choice made can be justified by the position and inclination of the connecting element in the model, component 7. Looking at the characteristics of the model, we expect that component 2 will be the element mainly affected by the deformation of the connecting element and by the forces transmitted by the same, representing so the best choice for the desired analysis. In the following, we present the Python code implemented to derive the optimal design for component 2:

```python
sampler = qmc.LatinHypercube(d=3)
test = sampler.random(n=10000)
l_bounds = [0.00648, 0.7, 0.7]
u_bounds = [2.59252, 4.5, 4.5]
test_scaled = qmc.scale(test, l_bounds, u_bounds)

y_c = skm_c.predict_values(test_scaled)
```

```
y_k = skm_k.predict_values(test_scaled)
min_en1 = 1E20
min_en2 = 1E20
min_en = 1E20
target_en1 = 22000000   %[N*mm]
target_en2 = 22000000   %[N*mm]
best_des = [0,0,0]
best_i = 0

for i in range(10000):
   if y_k[i][24] > 390 and y_k[i][24] < 410:
      bspl.t = y_k[i,:]
      bspl.c = y_c[i,:]
      if y_k[i,24] > 200:
          test1 = np.linspace(0,200,1000)
          test2 = np.linspace(200, y_k[i,24],1000)
          energy1 = simps(bspl(test1), test1)
          energy2 = simps(bspl(test2), test2)
      else:
          test1 = np.linspace(0,y_k[i,24],1000)
          energy1 = simps(bspl(test1), test1)
          energy2 = 0
      diff_en1 = abs(target_en1-energy1)
      diff_en2 = abs(target_en2-energy2)
      diff_en = diff_en1 + diff_en2
      if diff_en1 < min_en1 and diff_en2 < min_en2 and diff_en < min_en:
          best_des = test_scaled[i,:]
          best_en1 = diff_en1
          best_en2 = diff_en2
          min_en1 = diff_en1
          min_en2 = diff_en2
          min_en = diff_en
          best_i = i
```

The design is chosen by running 10000 samples with the surrogate models previously trained. The 10000 samples are selected with a Latin-Hypercube sampling strategy within the limits of the variables presented in Table 4.10. The energy absorption required by the component is 22 kJ and 26 kJ in the first and second sections respectively. This returns a constant force of 110 kN in the first section and 130 kN in the second section. Of the 10000 samples tested, we select those that return a deformation of 400 mm, within an error of 2.5%. For each feasible sample point, the force-deformation curve is generated. The energy absorbed by the component is evaluated as the integral of the curve. The design selected for component 2 is the one that returns the smallest difference between the evaluated absorbed energy and the requested one in each section. Figure 4.21 shows the force deformation curve of the design chosen.

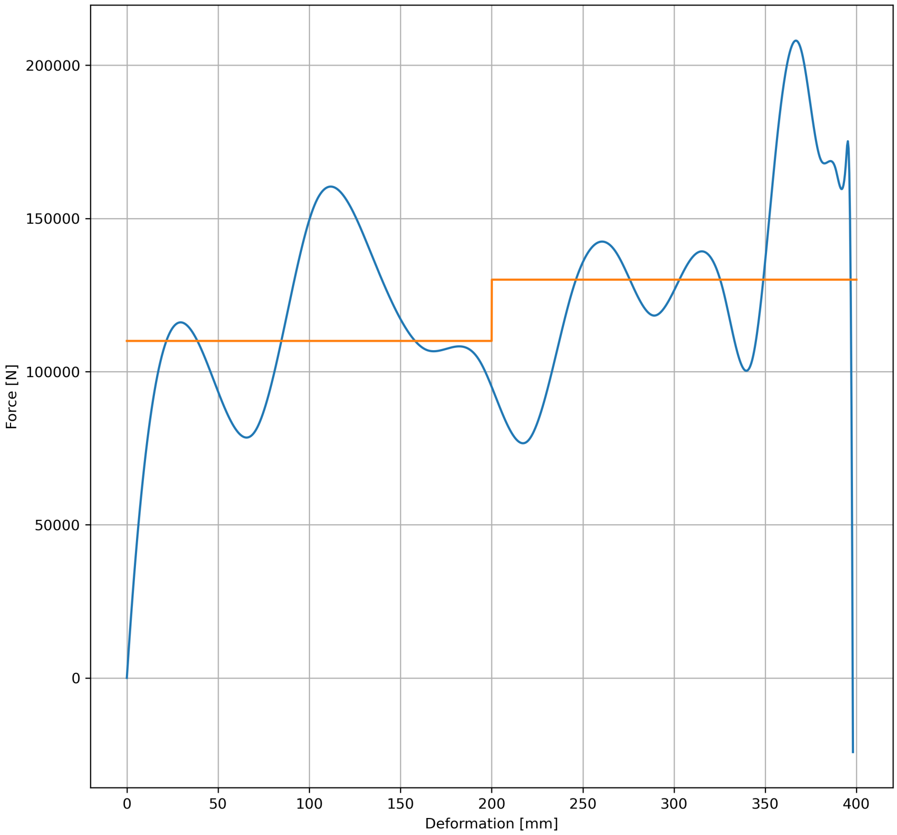


Figure 4.21: Force-deformation curve of the fixed design for component 2

Table 4.12 shows the design of component 2, in terms of the thickness of each section, selected to obtain a deformation of 400 mm and the difference in the energy absorption to the desired one.

Table 4.12: Characteristic of component 2

Section	1	2
Thickness [mm]	3.33	3.47
Energy required [J]	22000.0	26000.0
Energy difference [J]	233.20	0.60
Energy difference [%]	1.06	0.0027

4.2.4 Adapting the Solution Space model

Forcing the energy absorbed by component 2 has an effect on the Solution Space of the model. This effect can be easily understood by analyzing the equation 4.3 presented previously. The values F_2 and F_3 indicate the forces absorbed by component 2 in sections 1 and 2, respectively. Following the values of energy imposed in Table 4.12, we can evaluate that the force absorbed by each section is 110 kN for section

1 and 130 kN for section 2. Equation 4.3 showed the form of the matricial equation of the system considering the 12 variables. However, fixing the forces F$_2$ and F$_3$ allows us to reduce the number of variables to 10. This variation of the equation is implemented by eliminating the columns of matrix A relative to sections 2 and 3. The vector b is modified accordingly. Equation 4.10 shows the modified matricial equation implemented:

$$\begin{bmatrix} -0.129 & -0.1481 & -0.1481 & -0.129 & -0.129 & -0.129 & -0.1481 & -0.1481 & -0.1481 & -0.053 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0.707 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.707 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & -0.707 \\ 0 & 0 & 0 & 0 & 0.871 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.871 & -1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} F_1 \\ F_4 \\ F_5 \\ F_6 \\ F_7 \\ F_8 \\ F_9 \\ F_{10} \\ F_{11} \\ F_{12} \end{bmatrix} = \begin{bmatrix} -88.237 \\ 465.0 \\ 465.0 \\ 355.0 \\ 275.0 \\ 405.0 \\ 405.0 \\ 110.0 \\ -110.0 \\ -130.0 \\ 0.0 \\ 0.0 \\ 14.19 \end{bmatrix} \tag{4.10}$$

The matrix A and the vector b are implemented in the Python script previously introduced to derive the corridors of the remaining sections of the model. Figure 4.22 shows the adapted solution space corridors. We can highlight, compared to the solution space corridors evaluated for the entire model and shown in Figure 4.5, an increase in the corridor of the first section of component 3. Smaller variations can be also identified in the other components. Tables 4.13, 4.14 and 4.15 summarize the solution space corridors relative to each load path.

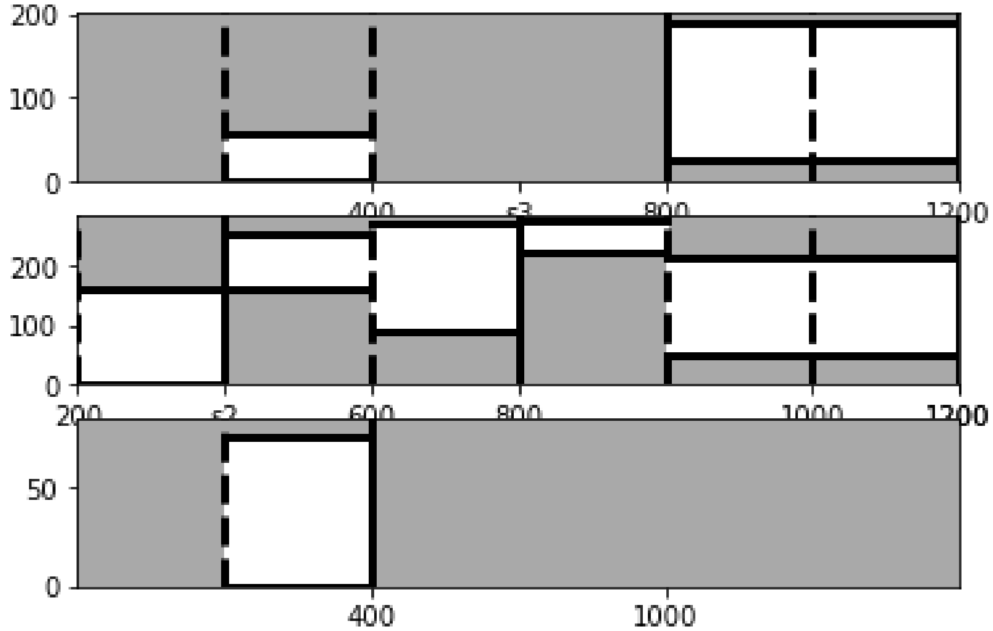


Figure 4.22: Solution Space corridors of the model

Table 4.13: Solution space corridor relative to 1st load path

Section	1	4	5
Component	1	3	3
Upper limit [kN]	57.12	190.56	190.56
Lower limit [kN]	0.00	27.44	27.44

Table 4.14: Solution space corridor relative to 2nd load path

Section	6	7	8	9	10	11
Component	4	5	5	6	6	6
Upper limit [kN]	158.50	255.50	271.79	275.00	214.43	214.43
Lower limit [kN]	0.00	158.51	89.20	222.55	51.32	51.31

Table 4.15: Solution space corridor relative to 3rd load path

Section	12
Component	7
Upper limit [kN]	74.78
Lower limit [kN]	0.00

4.2.5 Propagation of the uncertainties

The behavior of component 2 during a frontal crash is affected by the uncertainties in the model. In this section, we will present the methodology applied to uncertainties of the design parameters of each component to study their effect on component 2. We propose an energy approach for the uncertainty propagation. Figure 4.22 showed the solution space corridors of the model obtained after defining the design of component 2. We assign a value of 0 to the lower limit of each corridor, and 1 to the upper limit. The requested crashing force is evaluated as:

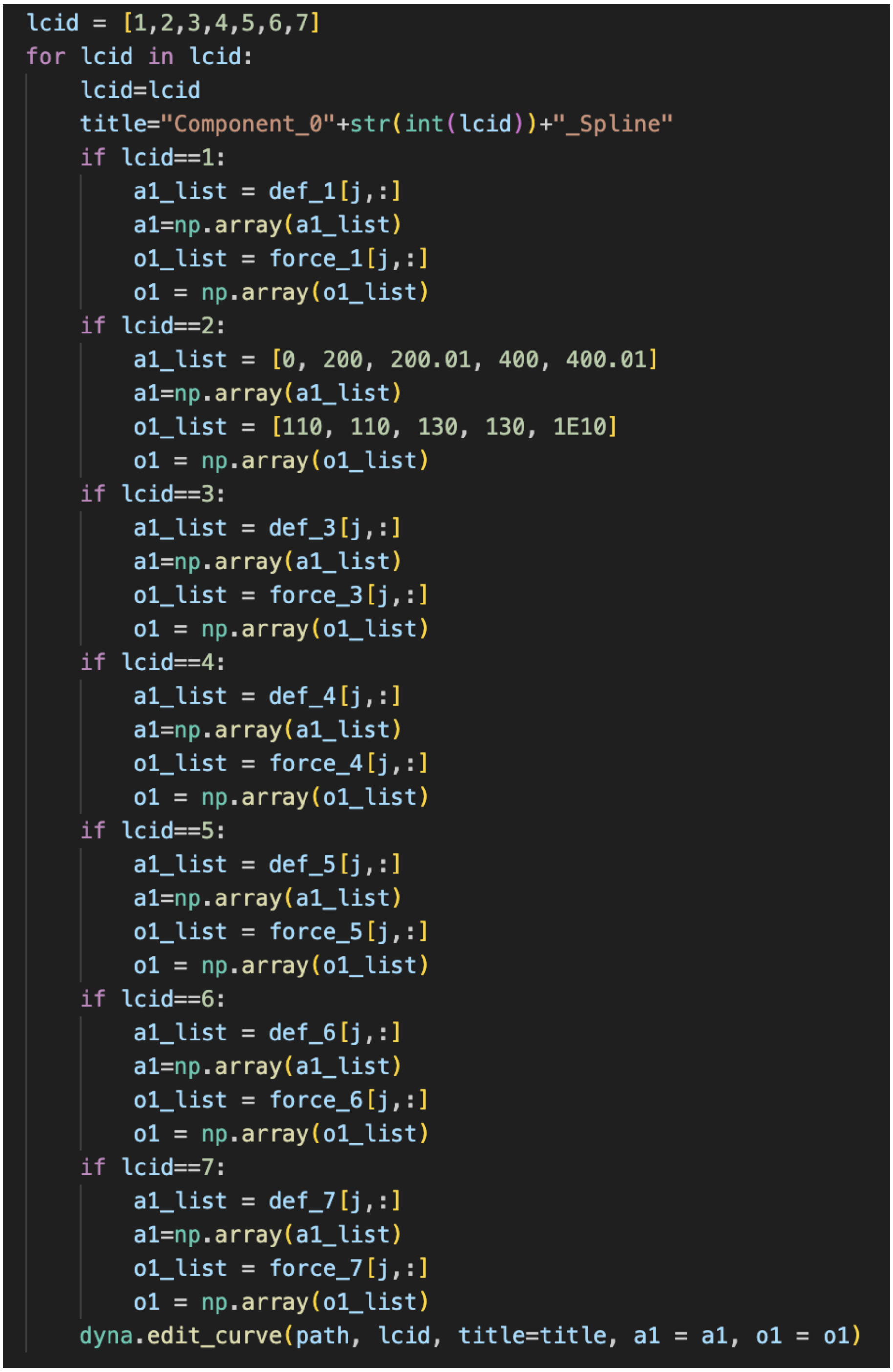
$$F_i = k \cdot (F_{i,u} - F_{i,l}) + F_{i,l} \tag{4.11}$$

where F_i is the crashing force of section i, $F_{i,u}$ and $F_{i,l}$ are the upper and lower limit of each corridor, and k is a parameter between 0 and 1. So, the energy absorption requested at each section is equal to:

$$E_i = F_i \cdot d_{max} \tag{4.12}$$

where E_i is the energy absorbed by section i, and d_{max} is the maximum deformation of the section (200 mm for each section of the two parallel load paths, 82.84 mm for component 7). To reduce the number of variables in the analysis, we consider for each sample point the same value of k for the different sections of a single component. This assumptions leads to a single variable for the 2 sections of components 3 and 4 and for the 3 sections of component 6. This reduces the problem to 6 variables for the propagation of the uncertainties.

An optimization method based on machine learning has been presented in Section 3.2.1. However, the computational resources required by the surrogate models caused a heavy slowdown of the FE simulations. For this reason, the parameters k relative to each component are defined by a DoE based on a Latin-Hypercube sampling strategy. Similarly to the methodology used to select the design of component 2, we find the design of each component that returns the smallest difference between the energy absorbed and the requested one. This process allowed us to evaluate the force-deformation curves a priori. Adopting this solution, each FE simulation can be run by implementing the curves previously obtained, without recalling the surrogate models at each iteration. The force-deformation curve of each component is implemented in LS-Dyna using the edit_curve python function shown in Figure 4.23. As expressed in Section 4.2.4, defining the design of one component affects the solution space of the other components of the model. In this thesis, we consider the variation of the solution space corridors negligible. This assumption is applied to simplify the method proposed, due to the high computational resources required by an iterative adaptation of the solution space and the reevaluation of the force-deformation curves.



```
lcid = [1,2,3,4,5,6,7]
for lcid in lcid:
    lcid=lcid
    title="Component_0"+str(int(lcid))+"_Spline"
    if lcid==1:
        a1_list = def_1[j,:]
        a1=np.array(a1_list)
        o1_list = force_1[j,:]
        o1 = np.array(o1_list)
    if lcid==2:
        a1_list = [0, 200, 200.01, 400, 400.01]
        a1=np.array(a1_list)
        o1_list = [110, 110, 130, 130, 1E10]
        o1 = np.array(o1_list)
    if lcid==3:
        a1_list = def_3[j,:]
        a1=np.array(a1_list)
        o1_list = force_3[j,:]
        o1 = np.array(o1_list)
    if lcid==4:
        a1_list = def_4[j,:]
        a1=np.array(a1_list)
        o1_list = force_4[j,:]
        o1 = np.array(o1_list)
    if lcid==5:
        a1_list = def_5[j,:]
        a1=np.array(a1_list)
        o1_list = force_5[j,:]
        o1 = np.array(o1_list)
    if lcid==6:
        a1_list = def_6[j,:]
        a1=np.array(a1_list)
        o1_list = force_6[j,:]
        o1 = np.array(o1_list)
    if lcid==7:
        a1_list = def_7[j,:]
        a1=np.array(a1_list)
        o1_list = force_7[j,:]
        o1 = np.array(o1_list)
    dyna.edit_curve(path, lcid, title=title, a1 = a1, o1 = o1)
```

Figure 4.23: Python script for the implementation of the force-deformation curve of each component in the material keyword file.

4.2.6 Evaluation of component 2 robustness

The uncertainties of the model are propagated considering 500 sample points in the design space presented in the previous section. Each sample point is tested with an FE simulation, using the MPP solver provided by LS-Dyna. The robustness of component 2 is verified by analyzing the crashing force dependent on the deformation and the maximum deformation of the component at each simulation.

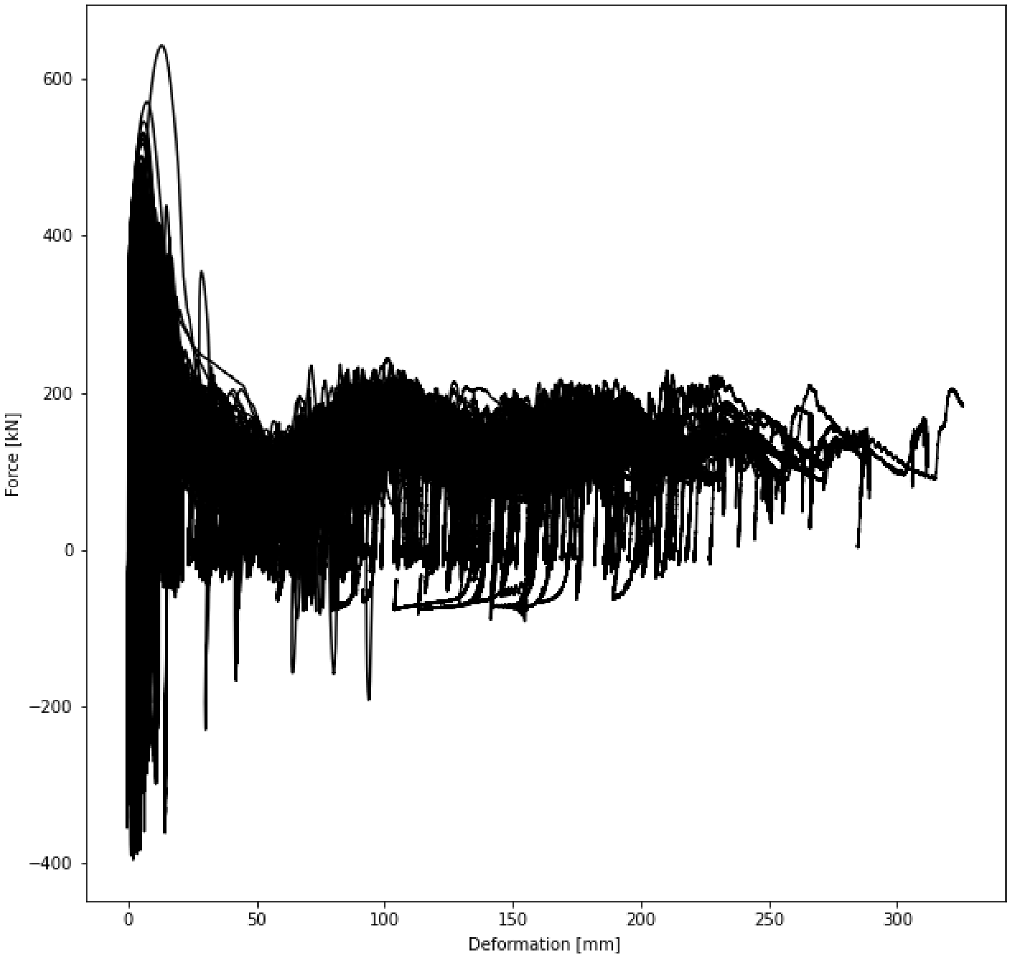


Figure 4.24: Force-deformation curve measure on component 2

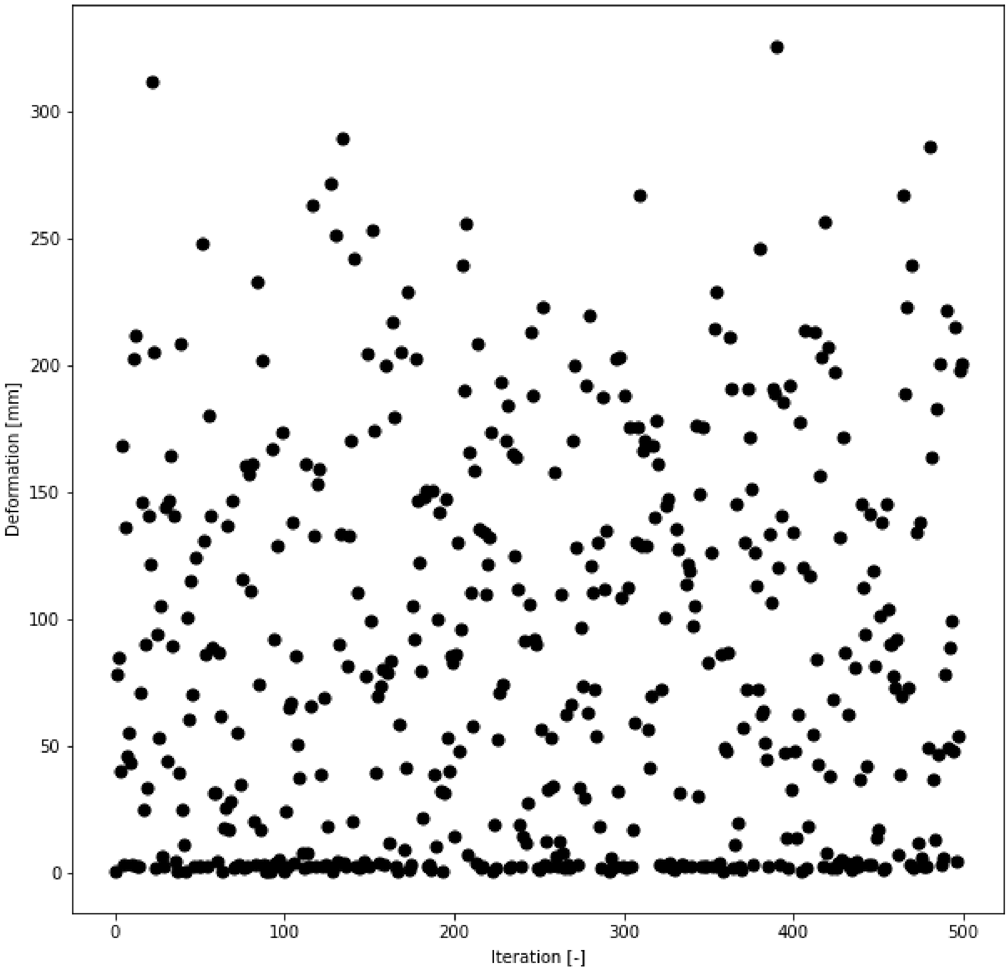


Figure 4.25: Maximum deformation measured at each simulation performed

The force and deformation of the component are analyzed independently due to the high range of variation measured on the deformation. The maximum deformation measured highlights the problems in the implementation of the 2D component in the model. The component has never reached the desired deformation of 400 mm. This behavior highlights high energy dissipation in the model. The Results and Discussion sections provide a more detailed analysis of the behavior of component 2.

Chapter 5

Results

Following the procedure explained in section 4.1, we managed to create a simplified 1D FE model with a connecting element between the load paths. The solution space corridors are evaluated and the model is tested to verify the constraints applied are met.

The FE model is then expanded, coupling 1D elements with a 2D mesh of a component. We trained a surrogate model for each component to predict their behavior. The surrogate model is used to fix an initial design of component 2, affecting the solution space previously evaluated. As a last step, the surrogate models are used to propagate the uncertainty of each component of the model. The effect of the propagation is studied by analyzing the variation of the force-deformation curve measured on component 2.

5.1 Results of Solution Space method

In section 4.1 we presented the simplified 1D FE model studied, and how it is possible to describe different constraints applied through a matricial equation. The three performance requirements on the model are minimum energy absorption, maximum acceleration and progressive order of deformation. The equations are assembled into a linear system Ax<b, shown in Equation 4.2 and Figure 4.4. The system is solved through the Adaptive Solution Space Code in Python, programmed by Ascia, Paolo. Tables 5.1, 5.2 and 5.3 show the upper and lower limits of each section, divided between the three load paths.

Table 5.1: Solution space corridor relative to 1st load path

Section	1	2	3	4	5
Component	1	2	2	3	3
Upper limit [kN]	57.13	158.12	152.03	200.00	181.13
Lower limit [kN]	0.00	110.03	100.00	158.13	10.25

Table 5.2: Solution space corridor relative to 2nd load path

Section	6	7	8	9	10	11
Component	4	5	5	6	6	6
Upper limit [kN]	138.83	230.68	246.98	252.96	205.00	223.86
Lower limit [kN]	0.00	138.83	55.65	200.93	34.11	52.97

Table 5.3: Solution space corridor relative to 3rd load path

Section	12
Component	7
Upper limit [kN]	78.80
Lower limit [kN]	0.00

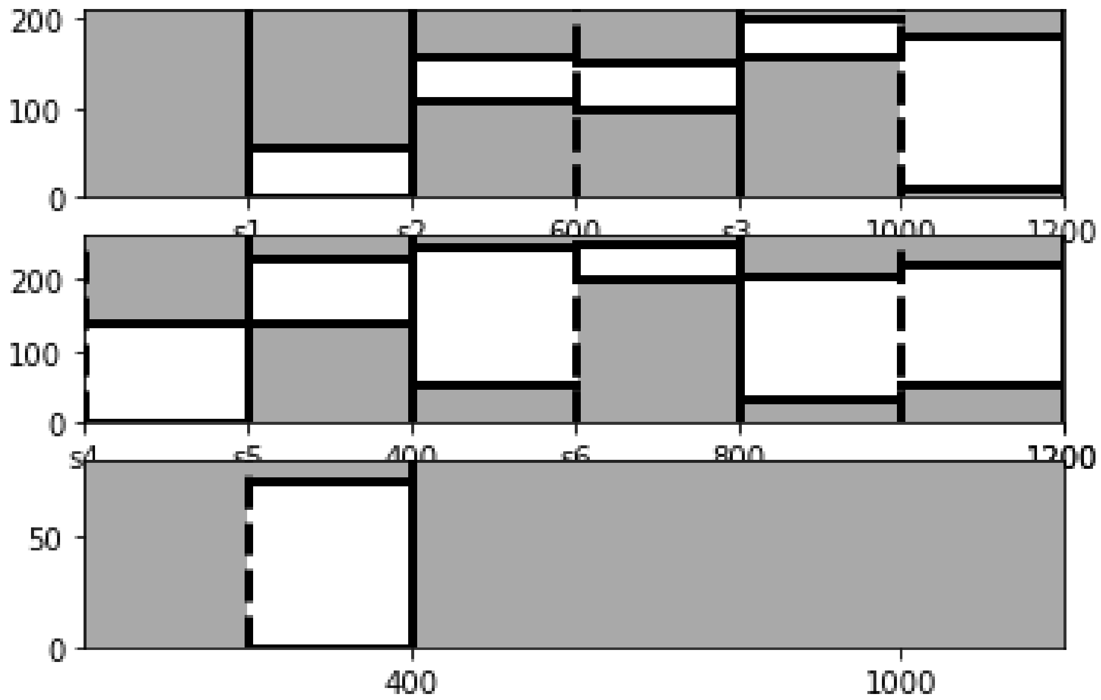


Figure 5.1: Solution Space Corridors

Studying the solution space corridors, we can note how the force absorbed in the first section of one component is higher than the one absorbed in each section of the component that precedes it. This increase in force is caused by the constraint on the progressive order of deformation. The lower limit of each corridor is implemented in the keyword file Material of the model. The model is impacted by a rigid wall with a mass of 1350 kg and an initial velocity of 15.6 m/s to verify that the solution implemented is able to absorb all the kinetic energy.

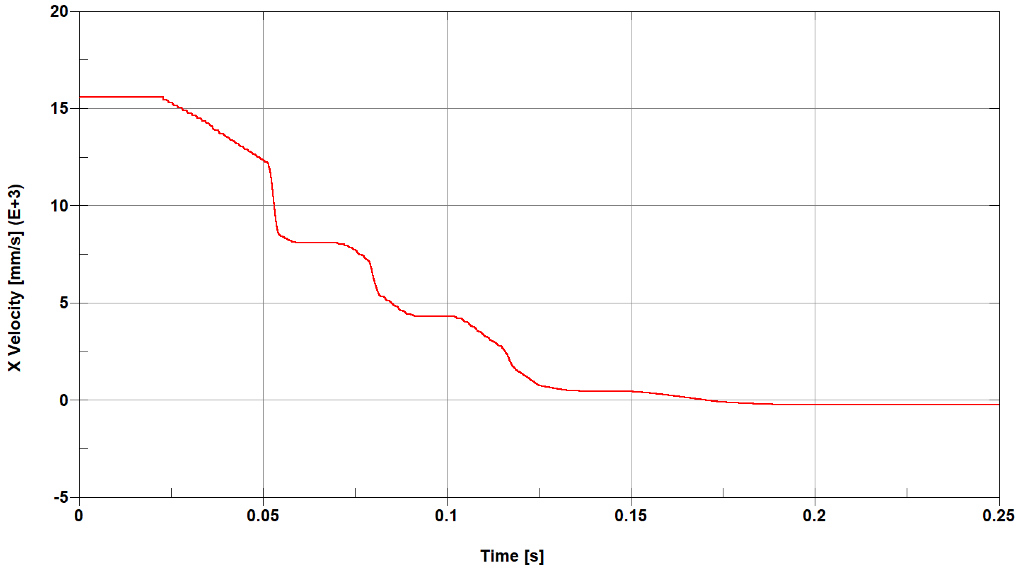


Figure 5.2: Rigid wall velocity during the impact

Figure 5.2 shows how the velocity of the rigid wall varies during the impact. We can see how the impacting object reaches a velocity of 0 m/s, proving that the model was able to absorb the kinetic energy of the frontal crash. At the end of the impact, the velocity of the rigid wall reaches small negative values. This phenomenon is caused by the relaxation of the compressed components that push away the rigid wall.

The progressive order of deformation is also verified by analyzing the displacement of the first node of each component. Component 7 is not analyzed as its deformation depends on the displacement of the connected load path. In particular, component 7 starts deforming when node 4100002, which refers to the start of the deformable part of component 5, registers a displacement. Figures 5.3 and 5.4 show that the constraint is correctly implemented in the model. The graphs shown also provide information on the deformation of the entire structure. Component 4 in the 2nd load path is the first deforming element, as the lower longitudinal beam is 200 mm longer than the upper longitudinal beam. Components 1 and 5 start deforming simultaneously, as expected by the definition of the solution space. In fact, those components share the same section in the first 200 mm of deformation. Component 6 starts deforming before component 3, as it is component by one more section: its deformation begins while component 2 is still deforming and it reached the section 3. This data, however, is impossible to be visualized in the graphs shown. An analysis of the deformations of each component during the impact would be needed.

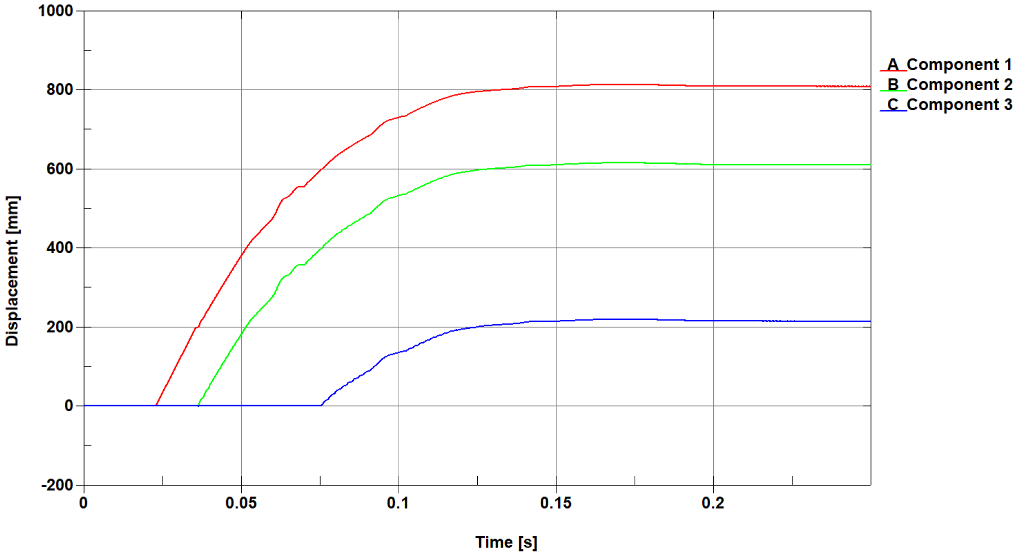


Figure 5.3: Order of deformation of 1st load path

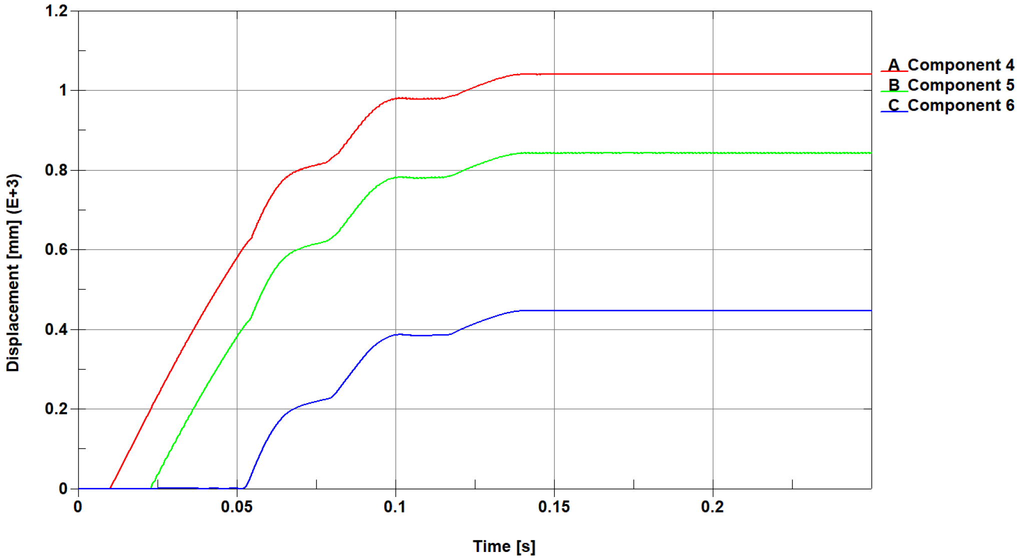


Figure 5.4: Order of deformation of 2nd load path

A second simulation is performed by implementing in the material card the upper limit of the solution space corridors. With this test, we aim to verify that the maximum acceleration does not exceed the limit imposed. Figure 5.5 shows that the maximum acceleration reached is slightly lower than the limit imposed. The limit is exceeded after 0.025s. However, as the limit is not respected for a small period of time, less than 1/100 of a second, we can consider the result positive.

As demonstrated by the results obtained, all constraints imposed are respected. We can consider the mathematical model proposed by Chaudhry Taimoor Niaz [5] and the implementation presented in this thesis successful.

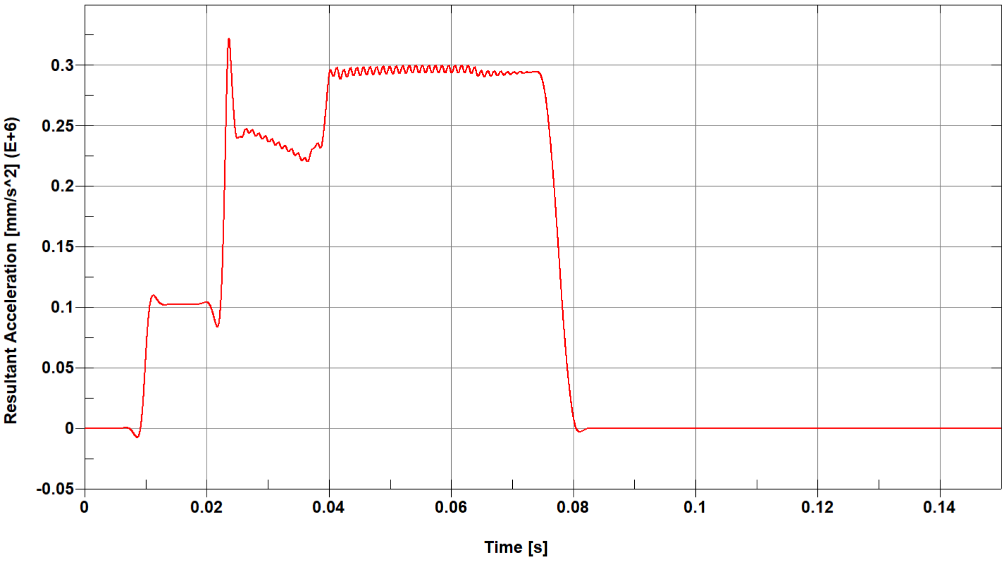


Figure 5.5: Acceleration-time curve during the impact

5.2 Performance of surrogate models

The surrogate models have been trained on a DoE based on LHS strategy, shown in Table 4.10. Two surrogate models are trained for the force-deformation curve of each component:

- One surrogate model for the knots of the curve, with information on the deformation;
- One surrogate model for the coefficients of the curve.

The performance of the surrogate models are tested with the leave-one-out strategy presented in Section 3.2.1, and the error are evaluated with the normalized root mean squared shown in equations 4.8 and 4.9. The following table shows the errors evaluated for each surrogate:

Table 5.4: Performance of surrogate models

Component	1	3	6	7
Knots NRMSE	0.226	0.244	0.060	0.279
Coefficient NRMSE	0.177	0.261	0.381	0.229

Using the NRMSE, we expect an error between 0.2 and 0.5 for a well-trained model. In our study, we obtained an error below 0.4 for each surrogate model. The surrogate model relative to the knots of component 6 performs particularly well, with an error of only 0.06. This is caused by the increased number of sample points used to train component 6. This choice was necessary due to the complexity of the component, as shown by the performance of the surrogate related to the coefficients of the curve. However, the error on the knots showed a slower increase compared to the complexity of the components. This trend leads to a high decrease of the knots NRMSE for component 6.

5.3 Design of component 2

The design of component 2 is chosen following the methodology presented in Section 4.2.3, as the design that returns through the surrogate models the minimum difference between the energy absorbed and the energy requested. Table 5.5 shows the parameters of the designed component. Figure 5.6 shows the force deformation curve of the design chosen.

Table 5.5: Characteristic of component 2

Section	1	2
Thickness [mm]	3.33	3.47
Energy required [J]	22000.0	26000.0
Energy difference [J]	233.20	0.60
Energy difference [%]	1.06	0.0027

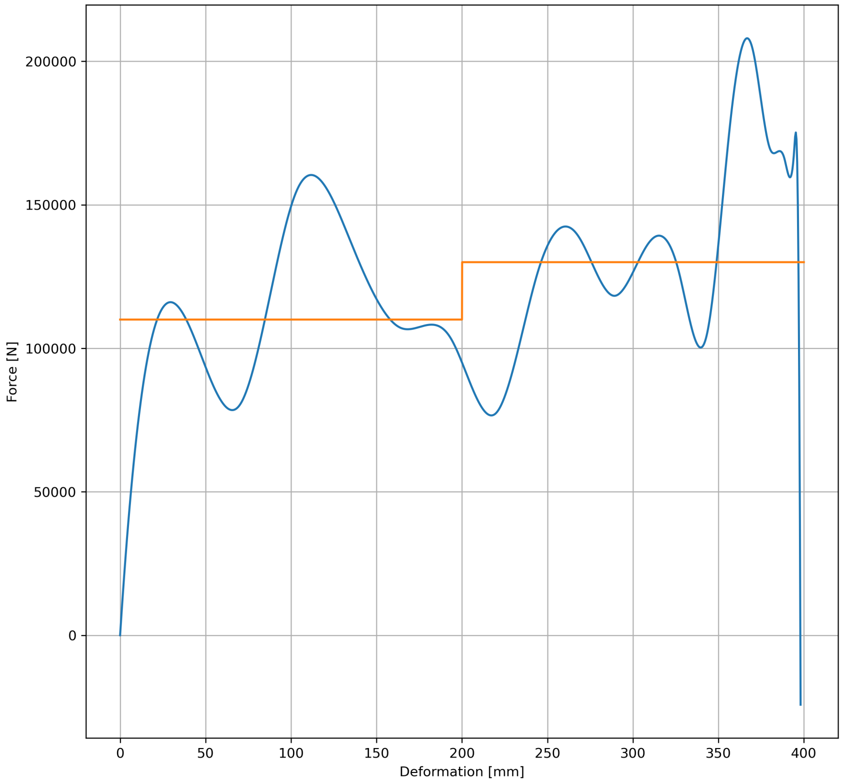


Figure 5.6: Force-deformation curve of the fixed design for component 2

Choosing the design of one component affect the solution space corridors of the model, effect shown in Figure 4.22. Tables 5.6, 5.7 and 5.8 show the upper and lower limits of the corridors of each load path.

Table 5.6: Solution space corridor relative to 1st load path

Section	1	4	5
Component	1	3	3
Upper limit [kN]	57.12	190.56	190.56
Lower limit [kN]	0.00	27.44	27.44

Table 5.7: Solution space corridor relative to 2nd load path

Section	6	7	8	9	10	11
Component	4	5	5	6	6	6
Upper limit [kN]	158.50	255.50	271.79	275.00	214.43	214.43
Lower limit [kN]	0.00	158.51	89.20	222.55	51.32	51.31

Table 5.8: Solution space corridor relative to 3rd load path

Section	12
Component	7
Upper limit [kN]	74.78
Lower limit [kN]	0.00

The design has been verified by running an FE simulation of the drop tower test. Figure 5.7 shows the force-deformation curve obtained in the simulation. The results obtained in the drop tower test return a smaller deformation of 382.85 mm. This variation indicates an error of 3.8% compared to the deformation of 398.08 mm obtained with the surrogate model. The energy absorbed by the component in the simulation is 49000 J. The energy absorbed by the force-deformation curve obtained with the surrogate model is 47800 J. These results return an error of 2.45% between the predicted result and the simulation. This data is shown in Table 5.9. Nevertheless, we can identify some differences in the forces evaluated in Figures 5.6 and 5.7. The biggest difference is shown in the peak force, reached in the simulation at the start of the deformation. This crashing force has been filtered in the trained surrogate model, and it caused smaller variations in the force evaluated in the rest of the curve. Even though the energy absorbed is correctly predicted, as shown in Table 5.9, the absence of the crashing force in the surrogate models prediction caused further problems in the simulations of the entire system, due to the non-compliance of the order of deformation. This phenomenon is further analyzed in Section 6.3.

Table 5.9: Comparison between simulation and surrogate model results of the component 2 design

	Simulation	Surrogate model	Error
Maximum deformation [mm]	382.85	398.08	3.8 %
Energy absorbed [J]	49000	47800	2.45 %

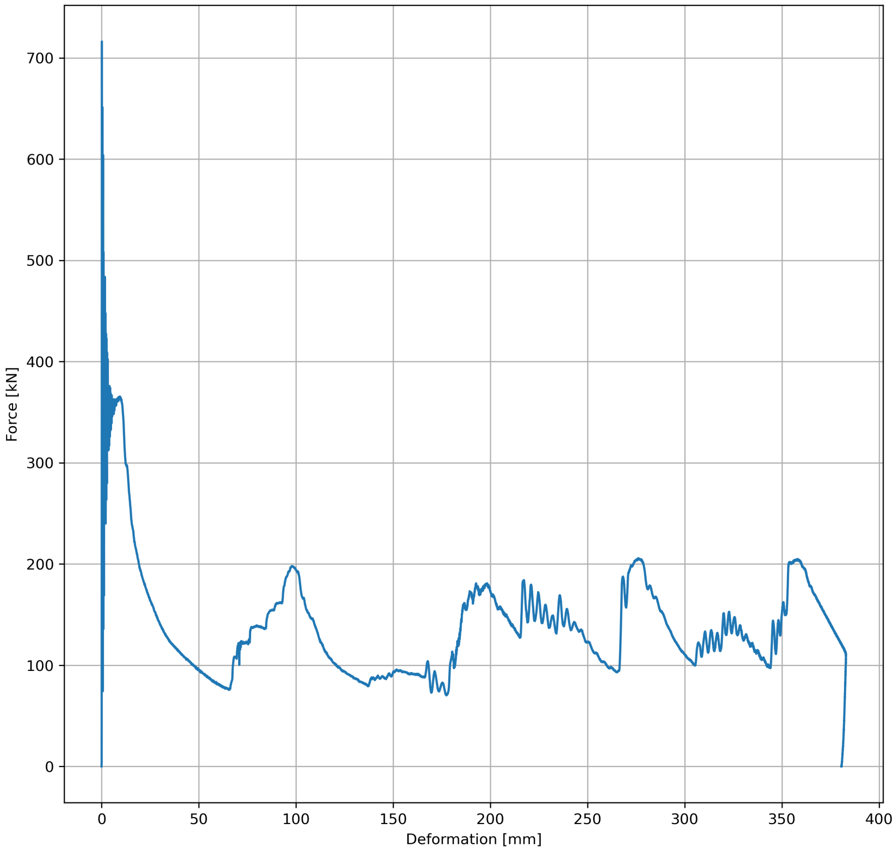


Figure 5.7: Force-deformation curve of the design chosen for component 2 obtained in FE simulation

5.4 Robustness of component 2

The uncertainty in the design characteristics of the frontal crash structure is propagated by modifying the energy absorbed by each component. The methodology applied to obtain the force-deformation curves imposed on each component has been presented in Section 4.2.5. A DoE based on Latin-Hypercube sampling strategy has been generated with 500 sample points. Each sample point, which refers to one force-deformation curve for each component, has been tested with an FE simulation. The force and deformation of component 2 have been measured to analyze the effects of the uncertainties. Figure 5.8 shows the maximum deformation measured on the component. It can be highlighted that the deformations registered on component 2 are lower than the desired value of 400 mm. This behavior is caused by an increase in the energy absorbed by each component that constitutes the structure.

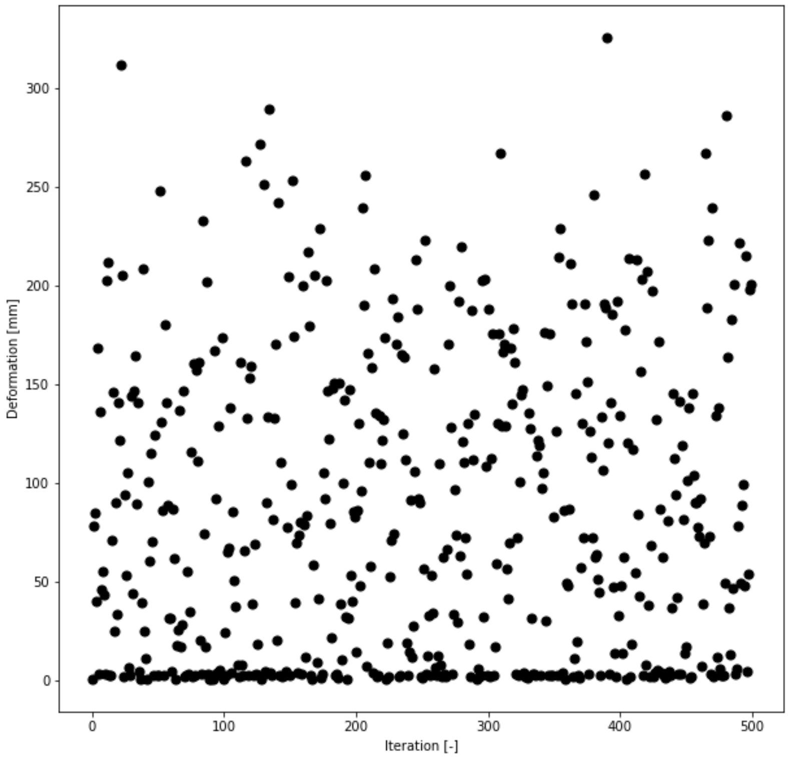


Figure 5.8: Maximum deformation measured at each simulation performed

A simulation with the minimum energy absorbed imposed on each component has been run to verify the discussion presented. Figure 5.9 shows the force-deformation curve measured on component 2 in this simulation. Even at the lowest energy absorbed by the structure during the impact, and so at the maximum absorption required by component 2, the deformation measured is much lower than the 400 mm desired in the design phase. However, the deformation measured in the uncertainty quantification analysis is, on average, lower than the 265 mm measured in the following simulation. This result supports the analysis of the reduced deformation registered due to the increase of the energy absorbed by the other components in the model. Additionally, we can highlight that component 2 does not deform in multiple simulations performed. While this effect is partially caused by the higher energy absorbed by the other components of the structure, the main cause of this effect is the non-compliance with the order of deformation, as shown in Figure 5.10. Component 2 is the last deforming element of the structure, due to the high crashing force required. This effect causes a dissipation of the kinetic energy due to the deformation of the following components 3 and 6, which should start deforming after component 2. As the energy absorbed by these components is higher than the minimum value required, the remaining kinetic energy in the impact is not high enough to guarantee the maximum deformation. In multiple simulations, the remaining kinetic energy is not high enough to provide the necessary crashing force.

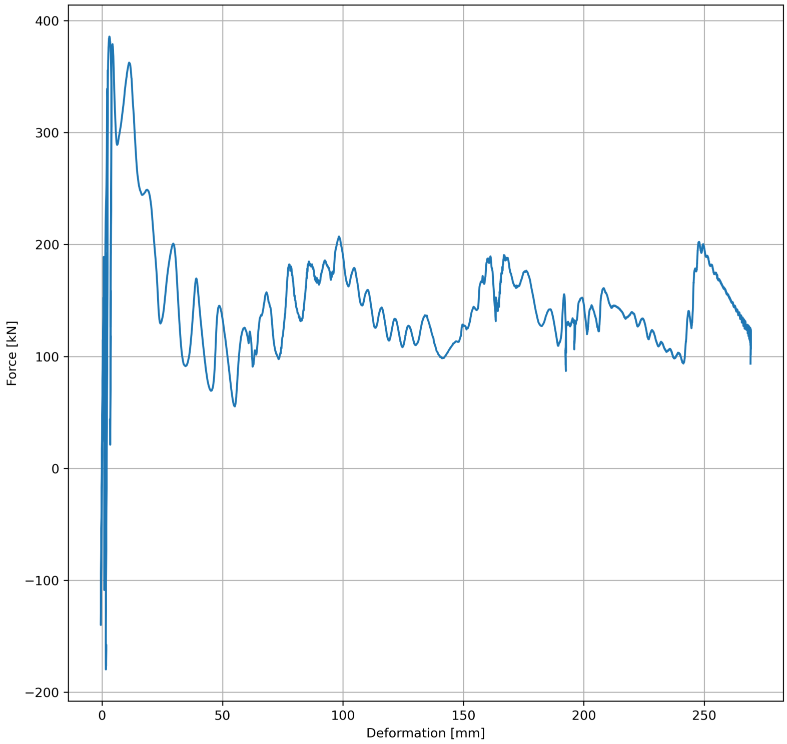


Figure 5.9: Force deformation curve at minimum energy absorbed by the structure

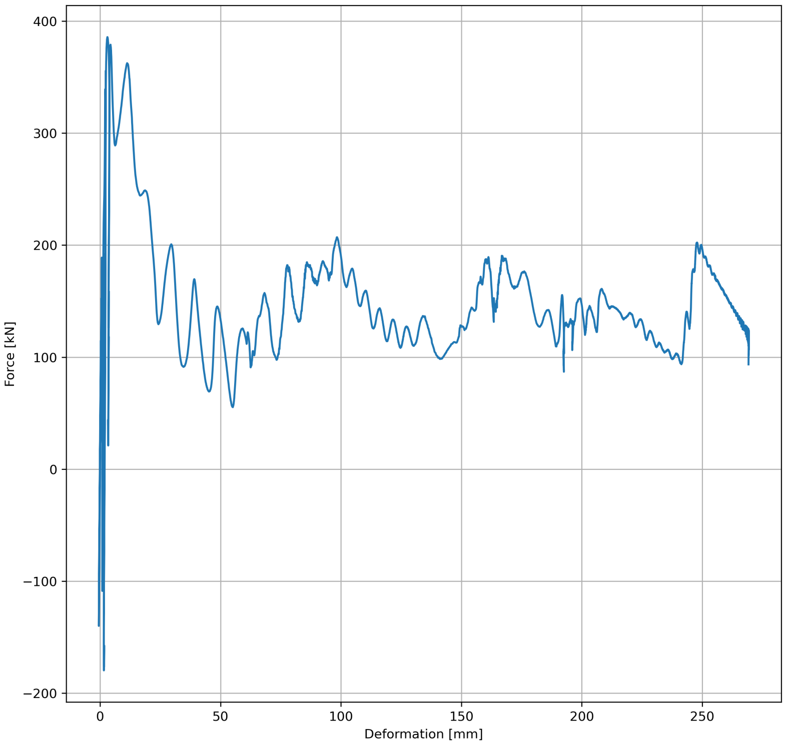


Figure 5.10: Force deformation curve at minimum energy absorbed by the structure

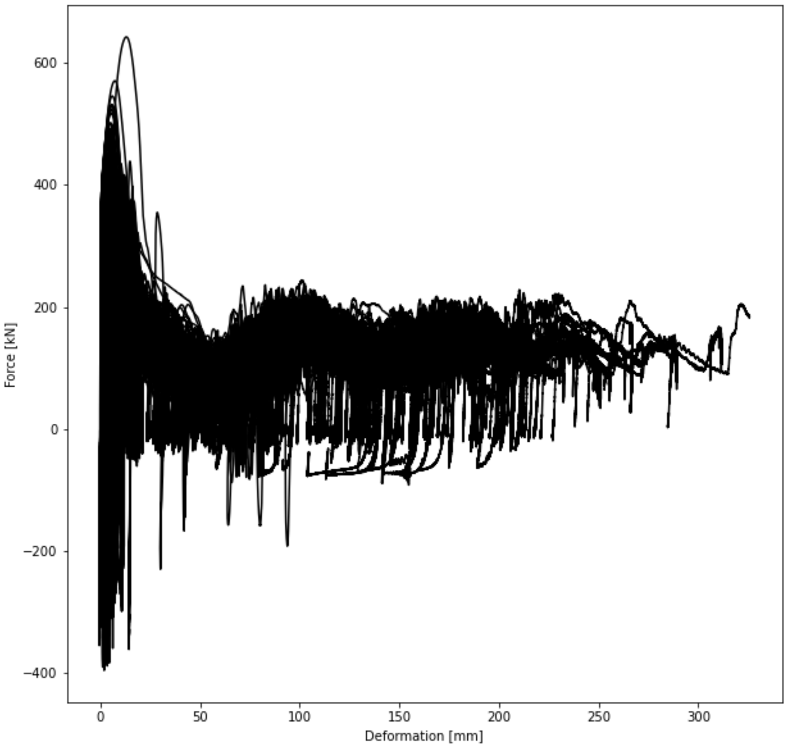


Figure 5.11: Force-deformation curve measure on component 2

The force-deformation curves measured in the 500 simulations performed are then analyzed. Figure 5.11 shows the results obtained in the uncertainty quantifications analysis. We can highlight a peak force registered in the first millimeters of deformation of the component. This behavior is caused by the first impact of the rigid wall with the component. The peak force could cause a peak acceleration in the impact that overcomes the imposed limit of 300 mm/s^2. However, we expect that the limit will be surpassed over a short period of time and can be neglected. For this reason, the peak force has been filtered in the force-deformation curve on which the surrogate models have been trained. The force registered reaches values of around 200 kN, higher than the maximum force allowed in the solution space corridors presented in the previous section. As the crashing force is higher than the maximum allowed, the order of deformation of the structure is not followed. This behavior could be the cause of the lower deformation registered. Component 3 starts deforming before component 2 finishes its deformation. The forces are transmitted to the following components, causing a lower deformation than the one imposed in the GSM.

Chapter 6

Discussion

6.1 Behavior of connecting element

The FE model featuring the connecting element originally implemented by Chaudhry Taimoor Niaz [5] presented various problems with the component. Figure 6.1 shows an example of the behavior of the connecting element obtained in the previous model.

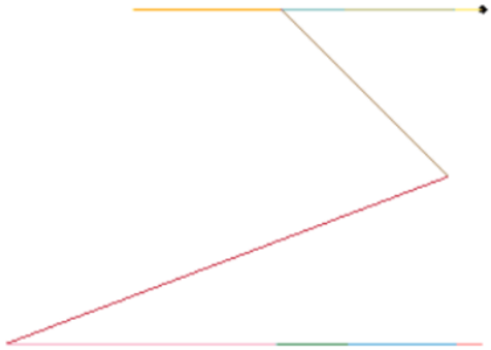


Figure 6.1: Global bending of connecting element [5]

During a frontal crash simulation, the element was not able to work only in compression, but global bending could be observed. The global bending of the component caused undesired peaks in the acceleration measured, and a general poor performance of the model. To solve this problem, we replaced the connecting element characterized by two different sections, one deformable and the other undeformable, with a component made up of a single deformable section. The maximum deformation of the component, equal to 82.84 mm, is forced by the geometrical characteristics of the model. However, the LS-Dyna software needs a complete force-deformation curve of the component, to understand how the element behaves in case of a higher deformation. For this reason, the limit is imposed by requesting a force of several orders of magnitude for deformations higher than 82.84 mm. These values are not achievable in a real scenario, meaning that the second part of the deformable section acts like an undeformable component. Figure 6.2 shows an example of a force-deformation curve implemented in the material card to simulate the behavior described.

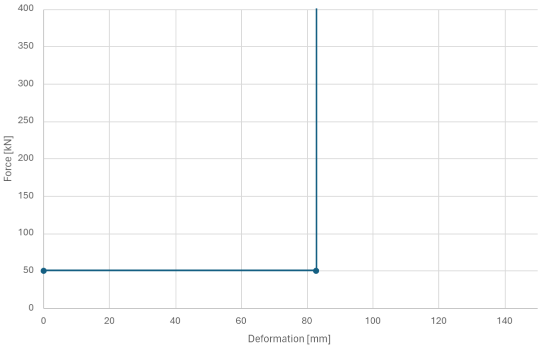
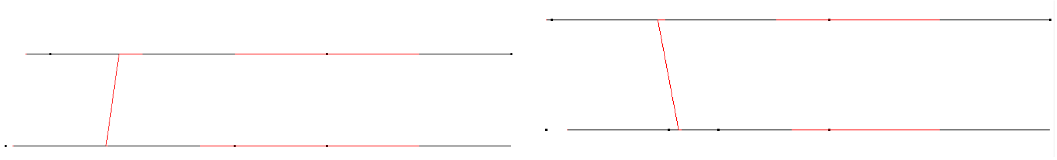


Figure 6.2: Force-deformation curve of connecting element

Testing the model with this force-deformation curve implemented, we can note how the connecting element works only in compression in the local reference frame. The component is subject also to tangential forces, which cause the rotation of its axis in the global reference frame. The rotation of the axis is necessary for a correct representation of the component 7 behavior in a real crash due to the higher deformations in the 2nd load path, compared to those imposed in the 1st load path. This behavior allows for a correct absorption of the kinetic energy and transmission of forces between the load paths. Figure 6.3 shows different moments of a crash test, demonstrating the deformation of the connecting element just described.



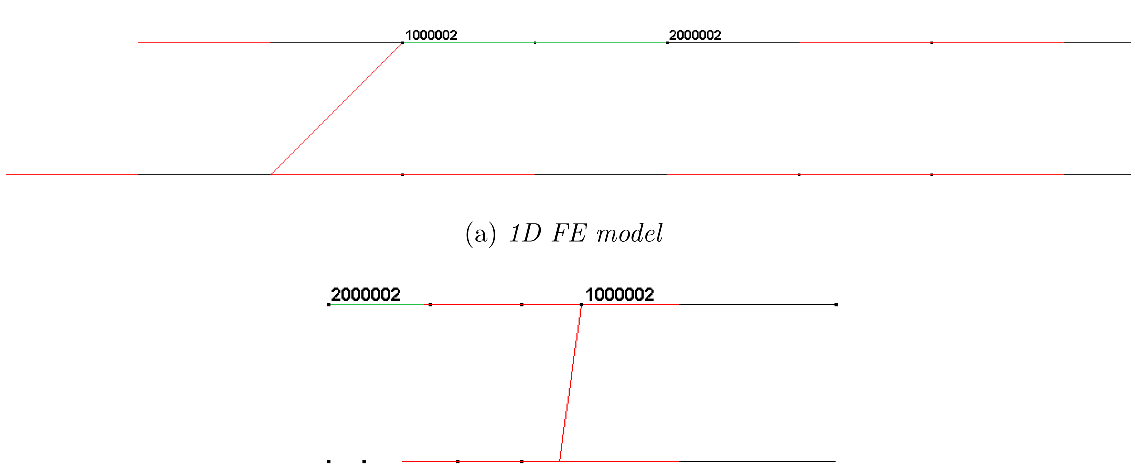
(a) *Initial impact*

(b) *Crash test completed*

Figure 6.3: Deformation of connecting element at different moments of the simulation

6.2 Limit on the deformation of each component

In Section 4.1, Figure 4.6 showed how the same limit discussed in the previous section was implemented in the force-deformation curve of each component. The implementation of this limit can be easily understood by analyzing the result of a simulation in which the limit is not imposed. Figure 6.4 shows the deformation obtained by the deformation of the model, in particular highlighting component 2.

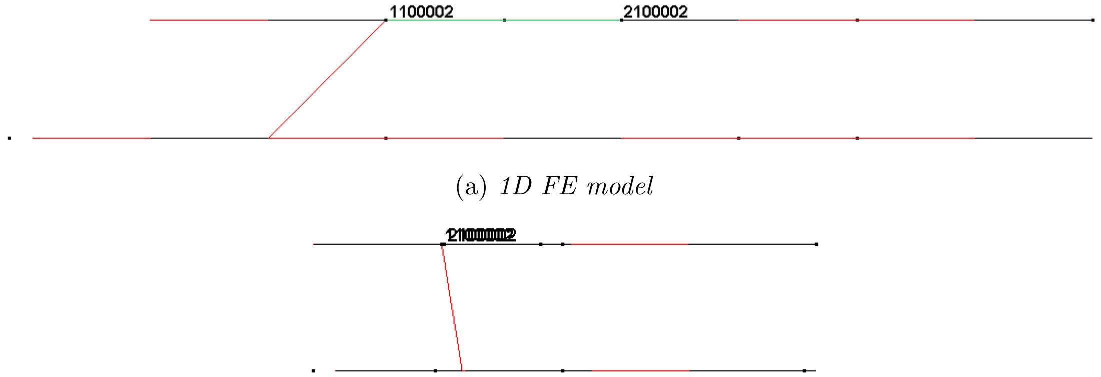


(a) *1D FE model*

(b) *Model compressed*

Figure 6.4: Deformation of the 1D FE model without limit on the deformation of each component

The deformable part of component 2 is highlighted in green. We can note that the LS-Dyna software is not able to understand how the components behave when the deformation is higher than the one provided in the force-deformation curve. This phenomenon is highlighted by the position of the nodes 1000002 and 2000002, which delimits the deformable part of component 2. The nodes "switch places", indicating that the deformation obtained is much higher than the length of the component. Due to this error in the simulation, is fundamental to implement a limit on the maximum deformation in the material keyword of each component. Figure 6.5 shows the same simulation performed, implementing the limit on the deformation of each component. The results show that the deformation of the highlighted component 2 is fixed to a maximum of 400 mm.



(a) *1D FE model*

(b) *Model compressed*

Figure 6.5: Deformation of the 1D FE model with limit on the deformation of each component implemented

6.3 Deformation of component 2

The results presented in Section 5.4 show that the deformation measured on the component reaches values much lower than the desired deformation of 400 mm. In Section 5.3, we showed a comparison between the results obtained with the use of the surrogate models and the ones obtained with an FE simulation of the drop tower test. The deformation presented a difference of only 3.8%, demonstrating the validity of the surrogate models implemented.
To better clarify the results obtained, we perform an FE simulation of the full model imposing on each component the lower limit of the solution space corridors shown in Section 5.3. The displacement of the two nodes that delimit the component is analyzed in Figure 6.6. The result shows that component 3, which follows the analyzed mesh, deforms before component 2. This behavior supports the discussion presented in Section 5.3, and is caused by the higher forces measured on the component.

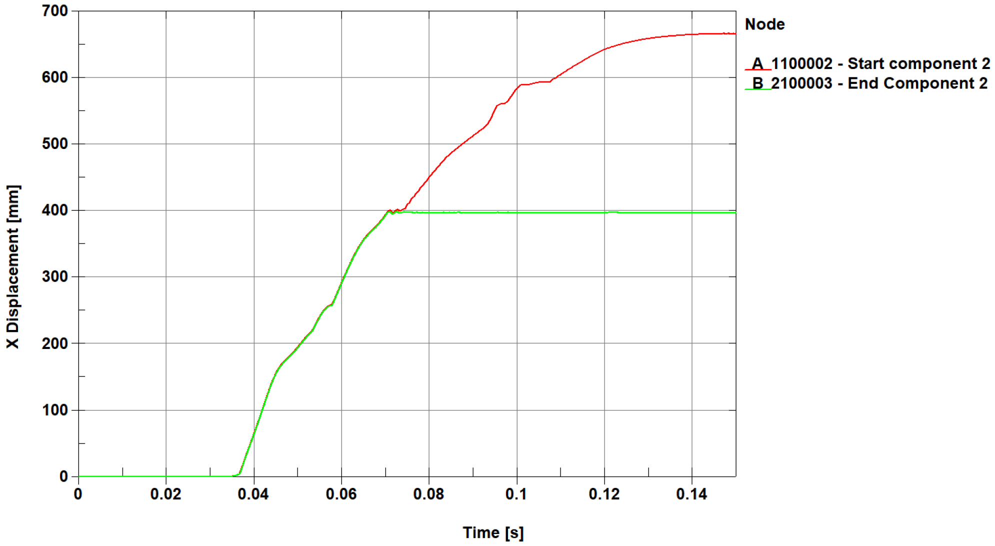


Figure 6.6: Displacement of nodes 1100002 and 2100003, which delimit component 2

Figure 6.7 shows the force-deformation curve measured on component 2. Imposing the lower limit of the corridors on each component, and so forcing the highest energy absorbed on component 2, we obtain a maximum deformation of 269.22 mm. We can highlight a peak force of 390 kN at the start of the deformation. These results could be the main cause behind the non-compliance of the order of deformation constraint.

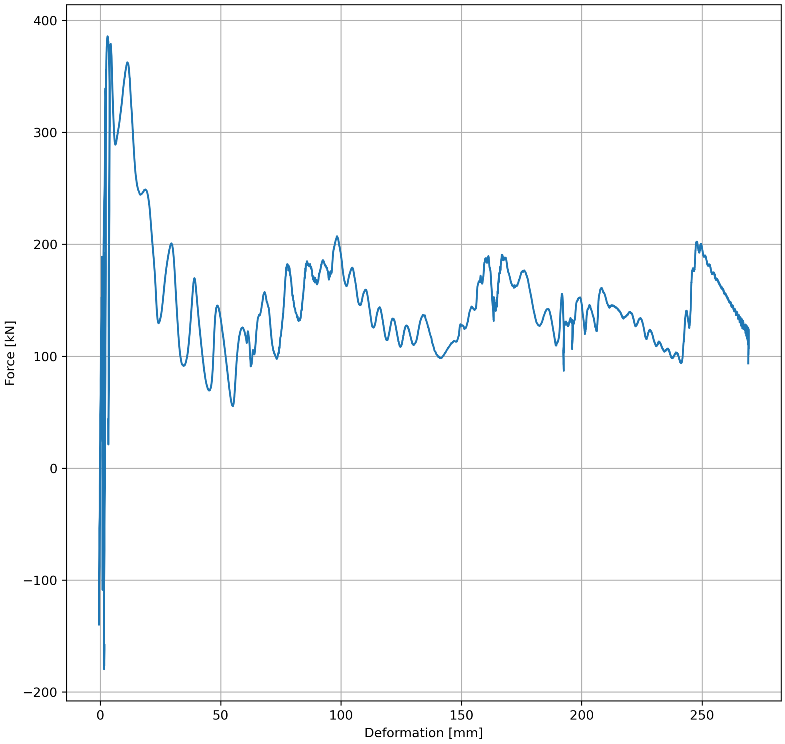


Figure 6.7: Force deformation curve at minimum energy absorbed by the structure

This behavior suggests that the problem arises in the implementation of the component in the entire model, Additionally, we can compute the energy absorbed by the component as the integral of the force-deformation curve, obtaining a result of 40600 J. The energy absorbed is 7400 J less than the desired one. This value suggests that there is a transmission of forces in the entire structure that was not considered in the mathematical model of the solution space method, or an effect of component 7 on the structure that is not captured by a conventional drop tower test.

Chapter 7

Conclusions and Outlook

The aim of this project is to define a workflow to quantify the effects of the design uncertainties of the components in a frontal crash structure. The model used for the analysis is based on the structure proposed by [5], in which two parallel load paths are connected by an additional element. Few differences in the position and inclination of the connecting element are introduced. The proposed mathematical model, based on the constraints by [3] and the angular correction by [5], is verified through FE simulations at the upper and lower limits of the solution space corridors. In these simulations, for simplicity, the force is considered constant along the entire deformation of each section. Then, the 2D mesh of one component is introduced in the model, defined by quadrilateral elements. The 2D mesh replaces the component on which a higher influence of the connecting element is expected. The assumption of a constant force is removed. The force-deformation curve of each 1D component is evaluated using surrogate models, trained on multiple drop tower tests performed in an FE environment. The performance of the surrogate models is verified through a leave-one-out cross-validation strategy. Surrogate models are also used to define an early design of the 3D component, imposing the desired deformation and energy absorption. The solution space corridors are recomputed, assessing the effect of the fixed design. Finally, 500 simulations on the simplified FE model coupled with the 2D mesh of one component are performed, modifying the force-deformation curve of each monodimensional component. The effect of the uncertainties of the structure is verified in terms of force and maximum deformation measured on the component studied.

The key findings from our research are:

- It is possible to introduce a connecting element in an FE model that works only in compression in the local reference frame. The proposed method introduces a limit on the deformation of the component, implemented through a high slope of the force-deformation curve. With this solution, the component can be modeled with a single deformable section, avoiding the global bending displayed in [5].

- The same limit on the deformation needs to be introduced in the force-deformation curve of each deformable section. This solution is necessary due to the LS-Dyna solver, which fills in the missing information on the crashing force at higher deformations. The LS-Dyna allowed the interpenetration between the components.

- The current training strategy for the surrogate models allowed the use of a low number of sample points. This choice returned an impressive result in the prediction of the deformation of the components but resulted in a poor performance in the force prediction. The error on the energy absorbed, however, is comparable to the one measured on the deformation. The method proposed is acceptable for an approach based on the energy absorbed but needs to be revised in case a higher accuracy on the forces is required.

- The current surrogate models are trained on filtered force-deformation curves. This solution allowed faster training, due to the decreased number of coefficients, but lost information on the peak force registered at the initial contact. This assumption caused poor performance of the full model, due to the non-compliance of the order of deformation.

- The maximum deformations registered on the component studied are much lower than the desired value of 400 mm. While this behavior could be caused by the non-compliance of the order of deformation, we also measured a high dispersion of energy. This could be caused by the methodology used to implement the limit on the deformation of 1D components. Additionally, the interactions between the connecting element and the two load paths were not considered in the surrogate model training.

The main objective of our project is accomplished. A solution to implement a connecting element in a simplified 1D Finite Element model is introduced, which allows to study more complex structures in future research. Additionally, the introduction of surrogate models to predict the behavior of the components returns good results in terms of deformation measured and energy absorbed, allowing for a reduction of time and computational costs required for a full model simulation. However, the addition of the connecting element introduced difficulties in the uncertainty quantification of the model.

The results obtained in this thesis offer the following insights for future research:

- A limit on the deformation of the component needs to be introduced for a correct interpretation by the LS-Dyna solver. However, further studies on the possible energy dissipation caused are required.

- Introducing the limit on the deformation of the components, it may be possible to further simplify the 1D FE model, removing the undeformable section and modeling each component as a single deformable element.

- The filtered force-deformation curves used to train the surrogate models caused the non-compliance of the order of deformation constraint in the full model. We suggest investigating a better methodology for the interpolation of the simulation curves, which allows the reduction of the number of coefficients of the curve while maintaining the information regarding the crashing force of the components.

- The surrogate models are trained on data from drop tower tests. This methodology neglects the interactions between the components. While we consider this assumption possible for consecutive components, one should look into the effect of the connection of the connecting element with the two load paths.

The thesis proposes a methodology to identify the effects of the uncertainties in the design characteristics of a complex frontal crash structure, introducing an additional connecting element between two parallel load paths. The methodology described offers a detailed procedure to generate a GSM and the relative DSM, implement the early design of one component in an FE software, derive the adapted solution space corridors, and train surrogate models to reduce the computational costs of a full FE simulation. The methodology proposed is sufficient to design complex structures in simplified 1D FE models. The advantages of the solution regarding the 1D model, and the limitations of the combined model, are identified and discussed. Possible solutions to the limits of the 1D model combined with the 2D mesh of one component are proposed. By utilizing the proposed methodology, complex FE models can be used to accelerate the design process of new crash structures for crashworthiness.

Appendix

Appendix

A.1 Simplified 1D FE model

The following section presents two keyword files implemented for the study of the simplified 1D FE model. The first keyword file presented is the implementation of the 1D components in LS-Dyna, using component 1 as an example. The different keywords are introduced with the syntax ∗*KEYWORD*. In the file presented, the character ∗ is missing due to the limit of the text editor. The ∗*PART* keyword is used to define the 2 different segments of the component. ∗*ELEMENT_BEAM* defines the nodes that delimit each section. ∗*DEFINE_CURVE_TITLE* is used to implement the desired force-deformation curve. At the end of each part, the material card is defined: NON_LINEAR_PLASTIC for the deformable part, and LINEAR_PLASTIC for the undeformable.

Component 01

```
$# Unit System: ton, mm, sec, N, MPa, mJ
$###############################################################################
$# File: Component 01                                                          #
$###############################################################################
$
KEYWORD
$
$###############################################################################
$
$# Definition - Component_01_Deformable
$##########
PART
$#                                                                         title
Component_01_Deformable
$#     pid     secid       mid     eosid      hgid      grav    adpopt      tmid
   1000001   1000001   1000001
SECTION_BEAM_TITLE
Component_01
$#   secid    elform      shrf   qr/irid       cst     scoor       nsm     naupd
   1000001         6       1.0         1         0       2.0       0.0         0
$#     vol      iner       cid        ca    offset     rrcon     srcon     trcon
    2.51E5
ELEMENT_BEAM
$#   eid     pid      n1      n2      n3     rt1     rr1     rt2     rr2   local
 1000001 1000001 1100000 1100001       0       0       0       0       0       1
MAT_NONLINEAR_PLASTIC_DISCRETE_BEAM_TITLE
Component_01_Deformable
```

```
$#     mid        ro       tkr       tks       tkt       rkr       rks       rkt
   1000001  7.89E-09  2.07E+05
$#     tdr       tds       tdt       rdr       rds       rdt

$#   lcpdr     lcpds     lcpdt     lcpmr     lcpms     lcpmt
         1
$#  ffailr    ffails    ffailt    mfailr    mfails    mfailt

$#  ufailr    ufails    ufailt    tfailr    tfails    tfailt

$#     for       fos       fot       mor       mos       mot

DEFINE_CURVE_TITLE
Component_01_Spline
$#    lcid      sidr       sfa       sfo      offa      offo    dattyp     lcint
         1         0     1.000  1000.000     0.000     0.000         0         0
$#                a1                  o1
            0.000000                30.0
          200.000000                30.0
          200.100000      1000000.000000
$###############################################################################
$
$# Definition - Component_01_Undeformable
$##########
PART
$#                                                                         title
Component_01_Undeformable
$#     pid     secid       mid     eosid      hgid      grav    adpopt      tmid
   1000002   1000002   1000002
SECTION_BEAM_TITLE
Component_01_Undeformable
$#   secid    elform      shrf   qr/irid       cst     scoor       nsm     naupd
   1000002         6       1.0         1         0       2.0       0.0         0
$#     vol      iner       cid        ca    offset     rrcon     srcon     trcon
    2.51E5
ELEMENT_BEAM
$#   eid     pid      n1      n2      n3     rt1     rr1     rt2     rr2   local
 1000002 1000002 1100001 1100002
MAT_LINEAR_ELASTIC_DISCRETE_BEAM_TITLE
Component_01_Undeformable
$#     mid        ro       tkr       tks       tkt       rkr       rks       rkt
   1000002  7.89E-09  2.07E+09       0.0       0.0       0.0       0.0       0.0
$#     tdr       tds       tdt       rdr       rds       rdt
       0.0       0.0       0.0       0.0       0.0       0.0
$#     for       fos       fot       mor       mos       mot
       0.0       0.0       0.0       0.0       0.0       0.0
$##########
END
```

The second keyword file is used to implement the boundary conditions and define the kinetic energy. The keywords $*SET\_NODE\_LIST\_TITLE$ and $*BOUNDARY\_SPC\_SET$ are used to group a few nodes in a list and reduce the degrees of freedom of the desired node group. The keyword $*RIGID\_PLANAR\_MOVING\_FORCE\_ID$

implements the starting position of a moving rigid wall, with the selected mass and velocity. These two parameters allow the definition of the kinetic energy of the impact.

Boundary Conditions

```
$
KEYWORD
$
SET_NODE_LIST_TITLE
NODESET_SPC_111111
$#     sid       da1       da2       da3       da4    solver
         1       0.0       0.0       0.0       0.0MECH
$#    nid1      nid2      nid3      nid4      nid5      nid6      nid7      nid8
   3100003   6100004
SET_NODE_LIST_TITLE
NODESET_SPC_011111
$#     sid       da1       da2       da3       da4    solver
         2       0.0       0.0       0.0       0.0MECH
$#    nid1      nid2      nid3      nid4      nid5      nid6      nid7      nid8
   1100000   1100001   1100002   2100001   2100002   2100003   3100001   3100002
   4100000   4100001   4100002   5100001   5100002   5100003   6100001   6100002
   6100003
BOUNDARY_SPC_SET
$#    nsid       cid      dofx      dofy      dofz     dofrx     dofry     dofrz
         1         0         1         1         1         1         1         1
BOUNDARY_SPC_SET
$#    nsid       cid      dofx      dofy      dofz     dofrx     dofry     dofrz
         2         0         0         1         1         1         1         1
RIGIDWALL_PLANAR_MOVING_FORCES_ID
$#      id                                                                 title
   9000001CrashPlate
$#    nsid    nsidex     boxid    offset     birth     death     rwksf
         0         0         0       0.0       0.0   1.00E20       1.0
$#      xt        yt        zt        xh        yh        zh      fric      wvel
   -1956.0       0.0       0.0   -1955.0       0.0       0.0       0.0       0.0
$#    mass        v0
      1.35   15600.0
$#    soft      ssid        n1        n2        n3        n4
         0         0   9000001         0         0         0
RIGIDWALL_PLANAR_FORCES_ID
$#      id                                                                 title
   9000002Firewall
$#    nsid    nsidex     boxid    offset     birth     death     rwksf
         0         0         0       0.0       0.01.00000E20       1.0
$#      xt        yt        zt        xh        yh        zh      fric      wvel
       0.0       0.0       0.0      -1.0       0.0       0.0       0.0       0.0
$#    soft      ssid        n1        n2        n3        n4
         0         0   9000002         0         0         0
END
```

A.2 Uncertainty Quantification

The following section presents a keyword file and two Python scripts implemented for the study of the uncertainties of the model. The keyword file presented concerns the definition of the 2D mesh of component 2 and the implementation of the rigid connection between the 2D mesh and the 1D components in LS-Dyna. The two different sections are defined with the keyword *$*SECTION\_SHELL\_TITLE$*, which allows us to define the thickness of each section.
$*DATABASE\_CROSS\_SECTION\_PLANE\_ID$ implements the cross-section necessary for the measurement of the force acting on component 2. The keyword *$*CONSTRAINED\_NODAL\_RIGID\_BODY$* is used to rigidly connect one node of the precedent or following 1D component with every node on that plane, as shown in Figure 4.12.

Component 2 mesh

```
$###############################################################################
$# File: Component_2.k                                                         #
$###############################################################################
$
KEYWORD
$###############################################################################
$# Part Cards
$##########
PART
$#                                                                         title
Section_1
$#     pid     secid       mid     eosid      hgid      grav    adpopt      tmid
   2000001   2000001   2100001
SECTION_SHELL_TITLE
Section_1
$#   secid    elform      shrf       nip     propt   qr/irid     icomp     setyp
   2000001         2       1.0         2       1.0         0         0         1
$#      t1        t2        t3        t4      nloc     marea      idof    edgset
 3.3238611 3.3238611 3.3238611 3.3238611       0.0       0.0       0.0         0
$##########
PART
$#                                                                         title
Section_2
$#     pid     secid       mid     eosid      hgid      grav    adpopt      tmid
   2000002   2000002   2100001
SECTION_SHELL_TITLE
Section_2
$#   secid    elform      shrf       nip     propt   qr/irid     icomp     setyp
   2000002         2       1.0         2       1.0         0         0         1
$#      t1        t2        t3        t4      nloc     marea      idof    edgset
 3.4669875 3.4669875 3.4669875 3.4669875       0.0       0.0       0.0         0
$###############################################################################
$# For Output
$##########
SET_PART_LIST_TITLE
Component_2
$#     sid       da1       da2       da3       da4    solver
   2000001       0.0       0.0       0.0       0.0MECH
```

```
$#    pid1      pid2      pid3      pid4      pid5      pid6      pid7      pid8
   2000001   2000002
DATABASE_CROSS_SECTION_PLANE_ID
$#    csid                                                                 title
   2000001Force Measure
$#    psid       xct       yct       zct       xch       ych       zch    radius
   2000001    -610.0     -80.0      50.0    -608.0     -80.0      50.0       0.0
$#    xhev      yhev      zhev      lenl      lenm        id     itype
    -610.0     -80.0      46.0     100.0     160.0         0         0
$##########
CONSTRAINED_NODAL_RIGID_BODY
$#     pid       cid      nsid     pnode      iprt    drflag    rrflag
   2000006         0   2000001
SET_NODE_LIST_TITLE
Front
$#     sid       da1       da2       da3       da4    solver       its         -
   2000001       0.0       0.0       0.0       0.0MECH      1
$#    nid1      nid2      nid3      nid4      nid5      nid6      nid7      nid8
   1100002   2000001   2000002   2000011   2000012   2002433   2002434   2002435
   2002436   2002437   2002438   2002439   2002440   2002441   2002442   2002443
   2002444   2002445   2002446   2002447   2002448   2003234   2003235   2003236
   2003237   2003238   2003239   2003240   2003241   2003242   2003681   2003682
   2003683   2003684   2003685   2003686   2003687   2003688   2003689   2003690
   2003691   2003692   2003693   2003694   2003695   2003696   2004369   2004370
   2004371   2004372   2004373   2004374   2004375   2004376   2004377         0
$##########
CONSTRAINED_NODAL_RIGID_BODY
$#     pid       cid      nsid     pnode      iprt    drflag    rrflag
   2000007         0   2000002
SET_NODE_LIST_TITLE
Rear
$#     sid       da1       da2       da3       da4    solver       its         -
   2000002       0.0       0.0       0.0       0.0MECH      1
$#    nid1      nid2      nid3      nid4      nid5      nid6      nid7      nid8
   2100003   2000005   2000006   2000007   2000008   2000117   2000118   2000119
   2000120   2000121   2000122   2000123   2000124   2000125   2000126   2000127
   2000128   2000129   2000130   2000131   2000132   2000805   2000806   2000807
   2000808   2000809   2000810   2000811   2000812   2000813   2001245   2001246
   2001247   2001248   2001249   2001250   2001251   2001252   2001253   2001254
   2001255   2001256   2001257   2001258   2001259   2001260   2002002   2002003
   2002004   2002005   2002006   2002007   2002008   2002009   2002010         0
END
```

The first Python script allows the variation of the configuration of the vehicle, based on the desired forces applied on each section of component 2. The script used is specific to this study and would need a few adjustments to be applied to other components or configurations.

```python
import numpy as np
import csv

def vehicleConfigurationModifier(o1_fixed):

    #########
    # o1_fixed = vector with lower and upper limit of the corridors of each section:
    #            [dsL1, dsU1, dsL2, dsU2]
    #########

    with open('vehicleConfiguration.py', 'r') as file:
        config = file.readlines()

        with open('A_Matrix.csv','r') as A_matrix:
            lookupA = r"configuration['vehicle_4']['A']="
            for num_A, line in enumerate(config, 1):
                if lookupA in line:
                    print('found at line:', num_A)
                    break
            reader = csv.reader(A_matrix)
            data = list(reader)
            A_mat = np.matrix(data,dtype='float64')
            b_vector = open('b_Matrix.csv','r')
            readerb = csv.reader(b_vector)
            datab = list(readerb)
            b_vec = np.array(datab,dtype='float64')
            lookupB = r"configuration['vehicle_4']['b']="
            for num_B, line in enumerate(config, 1):
                if lookupB in line:
                    print('found at line:', num_B)
                    break
            A_col = A_mat[:,0].reshape(-1,1)
            A_col_ar = np.array(A_col,dtype='float64')
            o1 = float(o1_fixed[0])
            b_fix = np.multiply(A_col_ar, o1)
            b_vec = b_vec - b_fix
            b_vec = b_vec.T
            b_str='np.array(['+str(float(b_vec[0,0]))+',...)'
            new_B = str(lookupB) + str(b_str) + '\n'

            config[num_B-1] = new_B

    with open('vehicleConfiguration.py', 'w') as file:
        file.writelines(config)
        file.close()
```

The second Python script presented allows the training and validation of the surrogate models using a leave-one-out cross-validation. In the example presented, we train the surrogate model on the coefficients relative to component 1.

```python
nrmse = np.zeros(60)
loo = LeaveOneOut()
for train,test in loo.split(DOE_t):
    DOE_test = DOE_t[test,:]
    DOE_train = np.delete(DOE_t, test, 0)
    PAR_test = C_test[test,:]
    PAR_train = np.delete(C_test,test,0)
    ndim = 2
    C = KRG(theta0=[1e-2]*ndim,print_prediction = False)
    C.set_training_values(DOE_train,PAR_train)
    C.train()
    y = C.predict_values(DOE_test)
    nrmse[test] = mean_squared_error(y, PAR_test, squared=False)/np.max(PAR_test)
print('Coefficient RMSE:' +str(np.mean(nrmse)))
```

Bibliography

[1] World Health Organization. *Road traffic injuries.* https://www.who.int/newsroom/fact-sheets/detail/road-traffic-injuries, Last accessed on 2024-01-14. 2023.

[2] Markus Zimmermann and Johannes Edler von Hoessle. “Computing solution spaces for robust design”. In: *International Journal for Numerical Methods in Engineering* 94.3 (2013), pp. 290–307.

[3] Johannes Fender, Fabian Duddeck, and Markus Zimmermann. “Direct computation of solution spaces”. In: *Structural and Multidisciplinary Optimization* 55 (2017), pp. 1787–1796.

[4] Volker A Lange et al. “Early phase modeling of frontal impacts for crashworthiness: from lumped mass–spring models to deformation space models”. In: *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of automobile engineering* 233.12 (2019), pp. 3000–3015.

[5] Chaudhry Taimoor Niaz. “Improvement of a Simplified Car Model for Crash Performance Analysis in Early Design Stages”. Master’s thesis. München, BY: Technische Universität München, July 2023.

[6] Paolo Ascia, Matthias GR Faes, and Fabian Duddeck. “Quantifying the epistemic uncertainties linked to the development process of an early-stage component for crashworthiness”. In: ().

[7] Marco Daub, Ferdinand Wöhr, Markus Zimmermann, et al. “Optimizing distributed design processes for flexibility and cost”. In: *DS 103: Proceedings of the 22nd International DSM Conference (DSM 2020), MIT, Cambridge, Massachusetts, October 13th-15th 2020.* 2020, pp. 1–10.

[8] Lawrence L Hershman. “The US new car assessment program (NCAP): Past, present and future”. In: (2001).

[9] Paolo Ascia et al. “A finite element model of component solution space for early stage design uncertainties”. In: (2022).

[10] Johannes Fender, Fabian Duddeck, and Markus Zimmermann. “On the calibration of simplified vehicle crash models”. In: *Structural and Multidisciplinary Optimization* 49 (2014), pp. 455–469.

[11] P. Saves et al. “SMT 2.0: A Surrogate Modeling Toolbox with a focus on Hierarchical and Mixed Variables Gaussian Processes”. In: *Advances in Engineering Sofware* 188 (2024), p. 103571. DOI: `https://doi.org/10.1016/j.advengsoft.2023.103571`.

[12] Koushyar Komeilizadeh, Arne Kaps, and Fabian Duddeck. “Isovolumetric adaptations to space-filling design of experiments”. In: *Optimization and Engineering* 24.2 (2023), pp. 1267–1288.

[13] Robin Preece and Jovica V Milanović. “Efficient estimation of the probability of small-disturbance instability of large uncertain power systems”. In: *IEEE Transactions on Power Systems* 31.2 (2015), pp. 1063–1072.

[14] Peter A Lachenbruch and M Ray Mickey. “Estimation of error rates in discriminant analysis”. In: *Technometrics* 10.1 (1968), pp. 1–11.

[15] Fernando Nogueira. *Bayesian Optimization: Open source constrained global optimization tool for Python.* 2014–. URL: `https://github.com/bayesian-optimization/BayesianOptimization`.

[16] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. “Practical bayesian optimization of machine learning algorithms”. In: *Advances in neural information processing systems* 25 (2012).

[17] Lailong Song, Johannes Fender, and Fabian Duddeck. “A semi-analytical approach to identify solution spaces for crashworthiness in vehicle architectures”. In: *24th International Technical Conference on the Enhanced Safety of Vehicles (ESV).* 2015, pp. 8–12.

[18] An ANSYS Company Livermone Software Technology (LST). *LS-DYNA® KEYWORD USER’S MANUAL VOLUME II Material Models.* English. Livermone Software Technology (LST). Sept. 27, 2021. 1993 pp.

[19] Conradus van Mierlo et al. “Component-level impact performance assessment under spatially uncertain boundary conditions”. In: *Proceedings of International Conference on Uncertainty in Structural Dynamics.* International Conference on Uncertainty in Structural Dynamics. 2020, pp. 1–15.

[20] Pauli Virtanen et al. “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python”. In: *Nature Methods* 17 (2020), pp. 261–272. DOI: `10.1038/s41592-019-0686-2`. URL: `https://doi.org/10.1038/s41592-019-0686-2`.

[21] F. Pedregosa et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.