Master of Science Thesis

Crashworthiness Analysis of an Aircraft Subfloor Structure using Artificial Neural Networks

Joost Waaijer

Delft University of Technology

TU Delft

Master of Science Thesis

Crashworthiness Analysis of an Aircraft Subfloor Structure using Artificial Neural Networks

by

Joost Waaijer

Supervisor:	Prof. C. Bisagni
Thesis Committee:	Dr. C.D. Rans
	Dr. D.M.J. Peeters
	Dr. J. Paz Mendez
Institution:	Delft University of Technology
Place:	Faculty of Aerospace Engineering, Delft
Department:	Aerospace Structures and Materials
Project Duration:	January, 2022 - December, 2022

Cover Image: Aircraft Flying in the Sunset by Gerhard Gellinger
The work in this thesis was supported by the Clean Sky 2 STUNNING project. Their cooperation is gratefully acknowledged.

Preface

This thesis marks the end of my incredible journey at this faculty. Along the way I learned an awful lot about engineering but also about myself. I will carry the highs and the lows I have experienced with me forever. There are a number of people I would like to show my gratitude to.

First and most important, I would like to thank my parents, for supporting me in every way they could. Then my supervisors for this thesis, Prof. Chiara Bisagni and Dr. Javier Paz Mendez, thank you for your guidance for the past year. Lastly, to all family and friends, thank you for keeping me sane throughout this year.

Joost Waaijer
Delft, December 2022

Abstract

The goal of this thesis is to build an artificial neural network(ANN) surrogate model, that predicts the crashworthiness performance of a structure. The structure used in this research is a thermoplastic fibre-reinforced composite aircraft subfloor section which is part of the SmarT multifUNctioNal and INteGrated thermoplastic fuselage (STUNNING) fuselage demonstrator section. The structure will be analysed numerically, by making use of the finite element software LS-Dyna by Ansys. Within this research, a surrogate model is defined as a model that has been fed data from a computationally expensive model, yet is computationally considerably more inexpensive to run to obtain similar results as the expensive model. The main methodology to do so in this research is to build a two-dimensional design space, which is sampled using optimal latin hypercube sampling. By using an existing model for the STUNNING fuselage subfloor section and by automatic generation of geometry changes of the structure through a Python script, input files can be created according to the sampled design space. The simulations are then run on a high performance cluster. Once the simulations are run the results can be fed to an artificial neural network that uses this data to predict crashworthiness performance of the structure within the design space.

To build towards the goal of the thesis, a small comparative study has been performed on material models MAT054, MAT058, and MAT 261 in LS-Dyna. These material models are based on different failure criteria, and thus have different performance in coupon simulations. Using these simulations as a basis, a simple design space was created, sampled, and modelled so that a first ANN surrogate model could be built that could predict the failure displacement and the ultimate load of a composite coupon in tension. It was found that for a small dataset (<10 samples), the neural network could easily fit to the data, and could predict accurately on samples within the design space that were withheld from the network during training.

Then, the STUNNING subfloor section model was introduced and a number of models that section this structure are investigated. The main purpose of this investigation is to find out if the model can be reduced in size without changing the resulting failure mode. When a satisfactory reduced model was found, the research carried on by doing a sensitivity analysis on a proposed two-dimensional design space, and similarly a sensitivity analysis on the parallelization settings. From this study, a second failure mode of the model was discovered and the effect of the parallelization settings was quantified. From here the limits of the design space could be formed and sampled. Finally, different variations of ANN surrogate models were defined, trained, and tested. In the end, a root mean square discrepancy of around 10% was achieved during testing of the network. This means that the methodology could prove useful in design applications, depending on the design stage and needed accuracy.

Contents

Nomenclature

Abbreviations

Abbreviation	Definition
AT	Acceleration Time
AI	Artificial Intelligence
ANN	Artificial Neural Network(s)
CFE	Crush Force Efficiency
DL	Deep Learning
EASA	European Union Aviation Safety Agency
FAA	Federal Aviation Administration
FEA	Finite Element Analysis
FEM	Finite Element Model or Finite Element Method
GeLu	Gaussian error Linear Unit
ML	Machine Learning
MSE	Mean square Error
NASA	National Aeronautics and Space Administration
OLH	Optimal Latin Hypercube
OLHS	Optimal Latin Hypercube Sampling
RBF	Radial Basis Function
ReLu	Rectified Linear Unit
RMS(E)	Root Mean Square (Error)
SEA	Specific Energy Absorption
SGD	Stochastic Gradient Descent
TFD	Target Force Displacement Response

Symbols

Symbol	Definition	Unit
a	activation value: the output of a single artificial neuron.	[-]
E	Energy absorbed	[J]
ϵ	strain	[-]
u	displacement	[m]
F_{max}	Maximum force on a specimen or part	[N]
F_{mean}	Mean force on a specimen or part during crushing process.	[N]
m_{tot}	Total Mass	[kg]
V	Velocity	[ft/s] or [m/s]
SEA	Specific Energy Absorption	[J/kg]
z	total input of a single artificial neuron.	[-]

List of Figures

List of Tables

Introduction

This report contains a thesis on crashworthiness development of regional passenger aircraft fuselages. The crashworthiness of an aircraft, defined as the ability to protect an occupant in the event of a crash is a key aspect in the pursuit of safety in air travel. Tests from the Federal Aviation Agency and National Aeronautics and Space Administration (NASA) have determined for different aircraft types the vertical descent velocity upon impact which is still considered survivable for every occupant [1]. Regulations, therefore, dictate that for these velocities, an aircraft designer must show that at these velocities, ranging from 22 to 30 ft/s, the occupant has every reasonable chance to survive a crash [2]. Due to the highly nonlinear behavior of a structure subjected to a heavy impact or crash, the development of a crashworthy structure is achieved through expensive real-life testing and Finite Element Analysis (FEA). This is why the development and crashworthiness certification of the fuselage is a cumbersome process.

This thesis and following research will attempt to develop a surrogate model that is able to help speed up the optimization of the crashworthiness capabilities of a fuselage section. This surrogate model will be in the form of an artificial neural network (ANN). The ANN is a model that passes its input through a set of artificial neurons, which map the input to an output. By feeding the ANN data from FEA simulations, the neurons can be trained to map the structural variables of the structure to the expected crashworthiness performance. They are proven to be able to mimic heavy nonlinear functions, which makes the ANN a promising surrogate model. An important note is that the crashworthiness will be assessed globally, meaning that the research will be focused on the global response of a fuselage section, rather than the response of individual parts.

The content of this thesis is structured into 6 main chapters. First of all, in Chapter 1.1, the factors that define and drive aircraft crashworthiness are investigated. Then the regulations and certification process for crashworthiness is investigated, giving an overview of the most important requirements and certification methods used in practice today. Next in section 1.2, the basics of machine learning and deep learning, which are the large study areas to which the ANN belongs, are laid out with a focus on ANN modeling for the intended application of crashworthiness. In section 1.3, crashworthiness optimization problems that were solved using ANN surrogate models in existing literature are highlighted, the complete optimization process is analyzed and some optimization algorithms are investigated which are of interest for this research. Then in chapter 2, the finite element solver, LS-Dyna is introduced. This software is used throughout the entirety of the thesis. Within chapter 2 different material models available for fibre reinforced plastic composite in LS-Dyna are investigated. The first few models are created in form of coupons, and subsequently an impact analysis is done on a composite tube. Then in chapter 3, the first ANN surrogate model is created, which is trained on data gathered from coupon modeling in LS-Dyna. Then in 4 a large fuselage subfloor model is introduced called the SmarT multifUNctioNal and INteGrated thermoplastic fuselage (STUNNING) fuselage demonstrator section. In this chapter a couple of ideas are worked out, that can potentially reduce the model size while maintaining the ability to model the global crashworthiness behavior closely to the original model. To build a surrogate model, a design space needs to explored and defined. This will be done in chapter 5. Aditionally in this chapter, sensitivity analysis will be performed on the design space dimensions and other parameters. Lastly, in chapter 6, the methodology for building the surrogate models is introduced, and the results are examined.

1
Literature Study

1.1. Aircraft Crashworthiness, Certification, and Regulations

This chapter will provide an overview of the current state-of-the-art in the research field of aircraft crashworthiness. The focus of this chapter is mostly on regional passenger aircraft. The goal of the research that follows this literature study is to build a surrogate model that can predict structural behavior of aircraft fuselages as stated in the introduction of this report. To do so, an understanding is required of what the criteria are and how crashworthiness is assessed qualitatively. The aim of this chapter is to obtain a technical framework of what parts of the aircraft fuselage are of interest to assess through a surrogate model. More generally, the aim is to find what kind of parameters and structural responses are investigated in the industry.

A basis of what defines crashworthiness and how is it measured as a property will first be established in section 1.1.1. Then in section 1.1.2, the requirements/regulations which apply to the crashworthiness of an aircraft are more deeply examined. Next, in section 1.1.3, the common practices used in the industry to show compliance, also known as the certification process, to these regulations are investigated by making use of existing literature on this matter. Lastly, to fulfill the aim of this chapter as stated before a number of key observations are made in section 1.1.4.

1.1.1. Aircraft Crashworthiness

An aircraft's crashworthiness is measured by the ability to protect its occupants from injury in a crash event [3]. During impact of the aircraft, the occupant is protected by dissipating the structures' kinetic energy, as well as controlling the peak acceleration experienced by the occupant. An unfortunate fact when dealing with crashworthiness development is that no structure exists that can protect against all aircraft crashes seen in history due to the extremely violent accelerations involved. A lot of research has been done before the 1960s, summarized in 1959 by Eiband [4], which have tried to define the acceleration limits that the human body can withstand. The graph found in [5] (Figure 1.1) is a result of this research, and it shows that the maximum vertical acceleration without injury for humans is around 18G for a duration no longer than 0.03 seconds. Based on such results the industry has set out a range of crashes that are considered survivable and for which an aircraft manufacturer must show it can offer the necessary protection.

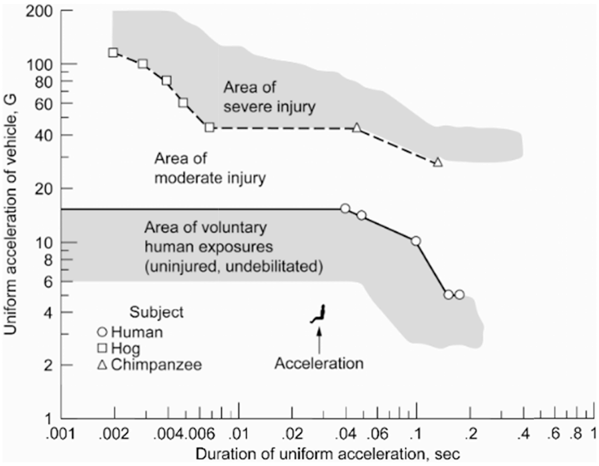
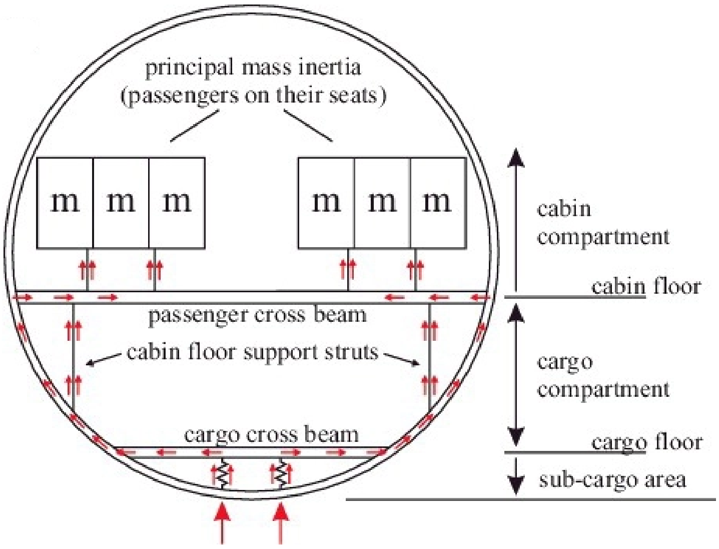
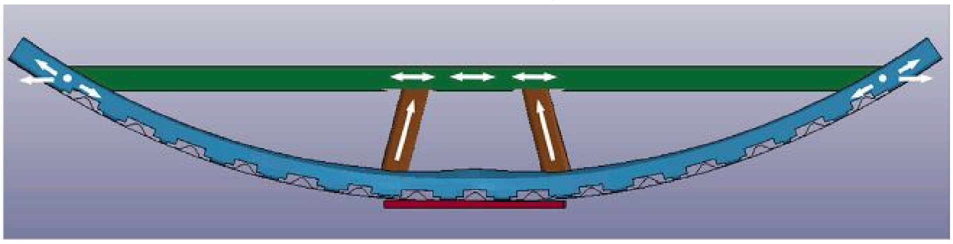


Figure 1.1: Acceleration tolerance graph found in research by Lawrence et al. [5] based on [4].

Naturally, this means that the most important sub-substructure of the aircraft in crashworthiness is the fuselage, containing the cabin and cargo holds/fuel tanks depicted in Figure 1.2a. The structure must not only protect the occupant during the crash but also from dangers that arise after the crash such as fire. Furthermore, every occupant needs to be able to escape the aircraft following the impact. There are a large number of safety measures and regulations in place which are dedicated to these aspects, these will be reviewed in more detail in section 1.1.2. This section will give an overview of the structural considerations, crash kinematics, and performance indicators that are used to assess the quality of the crashworthiness of a fuselage.



(a) A generic layout of a fuselage section. The arrows show the load path through the section when the impact starts. Image taken from [6].



(b) A more detailed view of the sub-cargo area. Image taken from [7].

Figure 1.2: The layout and load transfer of the most relevant structural elements in crashworthiness design

Aircraft Crash Kinematics

The section view of the fuselage in Figure 1.2a shows how the load is transferred through the structure on initial contact with the ground during a crash event. There are two possible resulting crash kinematics when a fuselage section makes impact [6]. They are called the 'unrolling' and the 'flattening' kinematics. They are depicted in Figure 1.3. Both crash kinematics are characterized by a number of phases as seen in the figure. They start off similarly with the crushing of the sub-cargo area. A general layout of this area is depicted in Figure 1.2b. The diagonally placed columns seen in figure 1.2b, often referred to as the energy absorbers, and the horizontal cargo cross beam that holds the cargo floor absorb energy in this first phase of the crash. At the end of this phase, the whole structure beneath the cross cargo beam compacts. This compaction will result in the failure of the cross cargo beam causing the frames to bend inward or it will cause the frames to bend parallel to the impact surface causing the fuselage to flatten. Lastly, for both the unrolling and the flattening crash kinematics, the support strut will start to be loaded and crushed (the support struts are indicated in figure 1.2a). The fuselage section design can be shaped such that one of these kinematic systems can be favored, however, identical fuselage sections have shown both kinematics in drop tests. This is also due to the presence or absence of cargo during the drop test. According to [8], the flattening kinematics are more likely to occur in a drop test when cargo is present, due to the earlier failure of the sub-cargo cross beams causing a decrease in stiffness laterally.

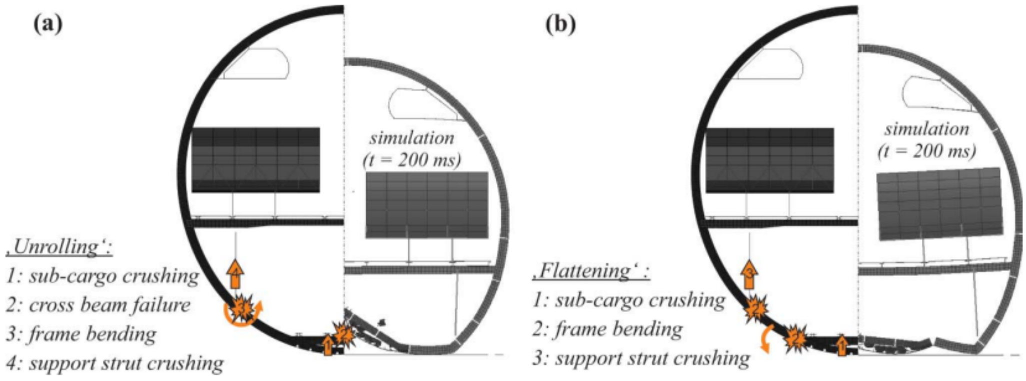


Figure 1.3: The unrolling crash kinematics (a) and the flattening crash kinematics (b) [6].

The sub-cargo area indicated in figure 1.2b plays an important role in the crashworthiness characteristics of the aircraft as it is the first part of the structure to collapse when the aircraft makes contact in a crash. When the crushing load on the diagonal columns seen in figure 1.2b is too high, the resulting deceleration becomes hazardous to the occupants and other structural components of the aircraft [9]. Vice versa, when the crushing load on the columns is too low the structure will not dissipate enough energy and the column will collapse too soon. This section of the fuselage is therefore the primary area to be investigated in the research that follows this literature study.

Crashworthiness Objective Functions

Throughout the previous section, it has become clear that there are multiple performance measures to assess the quality of a fuselage in crashworthiness. These have been more or less subjective so far, in that there was no quantification of these measures. The crashworthiness performance needs to be quantified, such that different designs can be compared to each other and the overall performance can be pushed in a positive direction. This section will look at the most commonly used performance measures, called the objective functions in optimization problems, and explain their significance.

Specific Energy Absorption

The energy absorption was already shortly mentioned in section 1.1.1. To calculate the specific energy absorption (SEA), the energy absorption E must be calculated first. The energy absorption is defined as:

$$E = \int^{L} F dx \tag{1.1}$$

Where F is the load on the structure which is assumed to be a continuous function and L is the crushing distance. The SEA can now be calculated by dividing the energy absorption with the mass of the structure as seen in equation 1.2 [10].

$$SEA = \frac{E}{m_{\text{tot}}} = \frac{\int^{L} F dx}{m_{\text{tot}}} \tag{1.2}$$

Where E is the energy absorbed, F is the load on the structure, L is the crushing distance and m_{tot} is the total mass of the structure.

Mean and Maximum Crushing Force

The mean and maximum force which the structure experiences are both of high importance. The mean force is a measure for how well the structure absorbs the energy of the impact [10]. The maximum force is often a measure for the maximum acceleration that the structure experiences. Since the human body is limited to a maximum tolerable acceleration, this criteria is naturally of importance. Both these indicators are a basis to start off with when designing any structure that needs to withstand crushing.

Crush Force efficiency (CFE)

The CFE crashworthiness indicator is defined as follows [10]:

$$\text{CFE} = \frac{F_{\text{mean}}}{F_{\text{max}}} \tag{1.3}$$

Where $F_{\text{mean}}, F_{\text{max}}$ is the mean and maximum force on the structure during a impact event. It is a measure of how smoothly the structure can decelerate or mitigate the impact, which is important for occupant survivability [10]. Since $F_{\text{mean}} < F_{\text{max}}$ per definition the ideal CFE is a value close to but never larger than 1.

Target Force-Displacement response (TFD) method

TFD is a method that uses the Force-Displacement curve of a structure as optimization objective [11]. To measure the performance some ideal TFD response must be defined from design requirements and or past data. By using an analytical relationship, the force-displacement curve can be determined from the acceleration time (AT) curve. This curve graphs the acceleration of the structure with respect to time. This is done in [11] using a method called the Equivalent Dynamic Load (EDL) method. An example of a target AT curve and an actual AT curve (derived from FEA) is shown in Figure 1.4. As can be seen from the figure, the target AT curve is simply a constant acceleration through time. The objective function is then defined as the difference between the target AT curve and the actual curve.

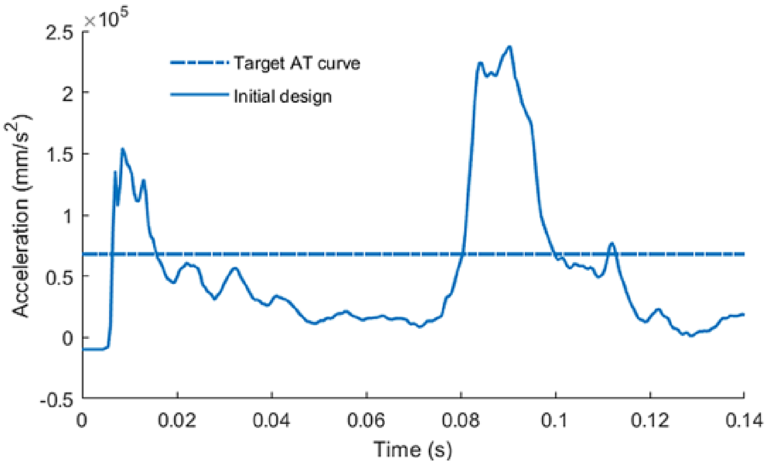


Figure 1.4: A target acceleration-time (AT) curve and an actual AT curve from some initial design in [11]. Noteworthy is the second peak in acceleration at around 0.08s in the initial design. This is an unwanted feature of a protective crash structure. This graph is taken from [11].

Intrusion

Intrusion measures the inward deformation at a certain location or area of a structure. A range of different measurement points within the structure can be assigned for this criteria. The intrusion velocity is also of importance, as the human body experiences a higher load when the velocity of the intrusion is increased [12].

1.1.2. Regulations

In the aeronautical industry, crashworthiness certification is mostly focused on situations in which some sort of controlled emergency landing can still be performed. In these situations, the landing is considered to be survivable by all occupants without the need to create a protective structure that would be too heavy to fly. A lot of injuries/fatalities can be prevented by having strict regulations concerning the aircraft cabin itself. The Federal Aviation Administration (FAA) and the European Union Aviation Safety Agency (EASA) have set strict requirements and rules on the dangers that precede or follow the aircraft's impact. The main areas of concern according to the FAA cabin interiors crashworthiness handbook [13] are:

- Protection of the occupants during the crash.
- Minimizing risk of hazards such as smoke/fire.
- Rapid evacuation of all occupants in the aircraft.

Protection of the occupants during the crash consists of proper use of the seats, seatbelts, and retention of significant mass that could possibly injure an occupant. The seats have to prevent exceeding the load caused by the accelerations of the crash in various body parts and specific bones within the occupant from a certain threshold value. Minimization of fire is mostly dependent on the secondary structure of the aircraft, ensuring no escape of oxygen from the oxygen supply, proper sealing of any flammable fluids, and using fireproof materials within the interiors of the cabin. Furthermore, the rapid evacuation is ensured by many systems of which some are probably well known to anyone who has ever flown commercially. These are the emergency exits, but also the cabin lights and emergency exit signs. Some lesser-known regulations are regarding the seats, they have to stay attached at all attachment points to the floor of the cabin.

For fuselage sections of the aircraft there are no specific rules set by the regulatory bodies but rather a set of requirements [8]. They are in line with the main areas of concern for the cabin listed above. They are presented in the Federal Aviation Regulations (FAR) 25 [2]. Two important sections are §25.561 and §25.562. Regulation §25.561 states that the occupant must be given every reasonable chance to survive in a minor crash. Regulation §25.562 describes the requirements on the dynamic load conditions of emergency landings. For airplane designs with a significant amount of carbon fiber reinforced plastics(CFRP) in their material usage, the FAA has issued Special Conditions(SC) reports for the Boeing 787-8 (25-362-SC) and the A350-900 (25-537-SC) [14, 15]. Four important factors for crash survivability are identified by the FAA which are listed below. Some additional guidelines for composite-based aircraft are added to these factors in the SC documents which are summarized below each factor.

1. **Retention of items of Mass**

   It must be shown that the airframe and floor structure deformation at the interface of the structure to seats, overhead bins, and other items of mass is comparable to those of previously certificated aircraft in the same size category. Furthermore, the attachment points need to be designed for the static emergency-landing loads that are experienced when the occupant experiences the ultimate inertia force relative to the surrounding structure acting separately of (i)3.0g upward,(ii)9.0g forward,(iii) 3.0g sideward on the airframe and 4.0g on the seats and their attachments(iv) 6.0g downward,(v) 1.5g rearward if the load factors at the attachment points are comparable to previously certificated similar aircraft.

2. **Maintaining acceptable acceleration and loads experienced by the occupants**

   The vertical acceleration experienced at the seat/floor interface must be comparable to levels expected for a previously certificated aircraft in the same size category or according to FAR §25.562(b) [2]. Here two dynamic impacts need to be considered with a minimum peak acceleration of the cabin floor of 14g and 16g respectively. The difference in these tests is the assumed attitude of the aircraft when the impact occurs.

3. **Maintaining a survivable volume**

   The structural deformation can not exceed a level where the living space of the occupant is affecting the survivability significantly.

4. **Maintaining occupant emergency egress paths**

   The egress path following a crash event is of similar suitability as a comparable aircraft in the same size category.

The vertical descent velocity upon impact, also called impact velocity, at which an aircraft needs to comply with these four criteria, depends on the size of the aircraft. Investigation performed by the National Aeronautics and Space Administration (NASA) reported in a document by the Transport Aircraft Crashworthiness and Ditching Working Group contains different impact velocities for different aircraft categories. The different categories and their respective impact velocity are found in Table 1.1 [1].

Table 1.1: Vertical descent velocity upon impact at which different category aircraft must comply to to ensure an equivalent level of crashworthiness according to NASA [1].

Aircraft Fuselage type	Impact Velocity
Small Narrow-body	6.7 m/s (22 ft/s)
Narrow-body	7.9 m/s (26 ft/s)
Wide-body	9.14 m/s (30 ft/s)

Furthermore testing performed by the FAA and NASA also has shown that occupant survivability is maintained for both Narrow- and wide-body aircraft during crash events of up to 30 ft/s impact velocity.

1.1.3. Means of Compliance (Certification)

Currently, aircraft's crashworthiness behavior is evaluated and certified using a combination of numerical methods, mostly in the form of Finite Element Modelling (F.E.M.), and impact tests. Development and evaluation of the design is often done on different scales of magnitude according to the building block approach reported in the FAA's advisory Circular AC20-107B [16] report for composite aircraft structures. It is visualised in Figure 1.5. The building block approach is a series of tests and numerical simulations on different scales, starting on a small scale with a large number of tests, and then decreasing the number of tests as the scale of the structure increases. The advantage of this technique is the cost efficiency because testing on larger parts of the structure requires more resources. Another advantage is the robust way in which a structure can be proven to comply with regulations. By following the building block approach a manufacturer can show it understands the structural behavior of its product by using knowledge and results from smaller scale parts to demonstrate reliable performance of the components prior to the testing of the components themselves [16]. The use of the building block approach is merely a recommendation and not a mandatory methodology to achieve a design that passes the certification tests and simulations.

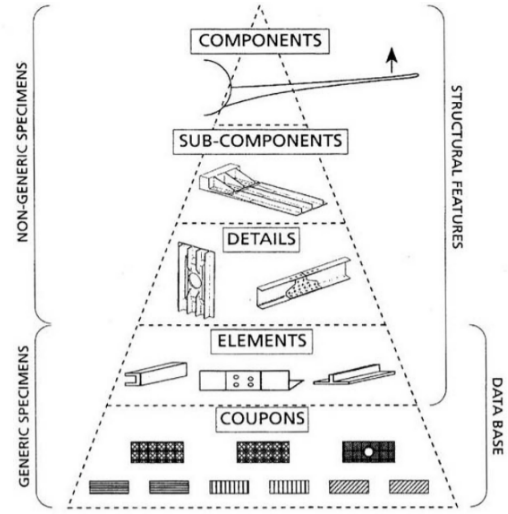


Figure 1.5: The building block according to FAA AC 20–107B [16].

As stated in AC 20-107B, the evaluation and procedures necessary for showing compliance with the requirements of an airworthy aircraft are highly dependent on specific factors such as the application, criticality to flight safety, expected service usage, manufacturing processes, and so on. Therefore all proposed means of compliance methods are indefinite in the sense that they do not state exact amounts of accuracy or required tests. Nevertheless, they provide a framework of what kind of factors and scenarios an aircraft designer should include in their analysis and design. Mou et al. [8] have made an overview of research and requirements of the different layers of the building block.

Coupons
The purpose of coupon testing and modeling is to understand the basic characteristics of the material. Additionally, for crashworthiness purposes, a strong comprehension of the failure modes and failure behavior of joints is crucial [8]. Another significant part of coupon testing is the manufacturing margins and documenting the scatter in material quality [16]. This is of importance even after certification, as it is necessary to show that batch productions do not have significant changes in material properties. The material data used in certification could also be used if a procurement method has been altered, to ensure that the material is still of acceptable quality.

Elements
An element is defined by AC20-107B [16] as a generic component that is part of a more complex structural member. The element design is where the job of the structural engineer becomes more significant. Although there are many different types of elements, the focus in this research will be on columns. For crashworthiness, columns are tested in their energy absorption capability, deformation, and failure modes [8]. Noteworthy is the change in geometry from closed section columns to open section columns when changing from a metal-based design towards a composite design. The reason for this is the buckling behavioral difference between the two materials. As can be seen in Figure 1.6[8], when an open section metal column is compressed, it will reach a buckling point after which its resistance and energy absorption ability are extremely negatively affected. The global buckling failure mode can be prevented by weakening or changing the geometry of the column such that global buckling is no longer the dominating failure mode. These changes in geometry are known as triggers, which will not be further discussed in this research.

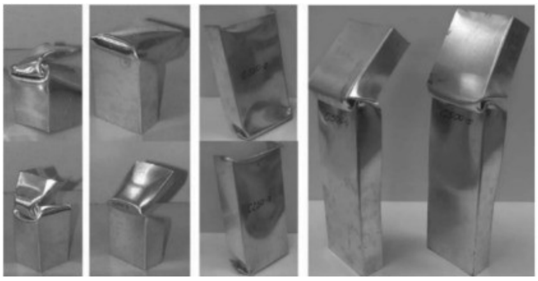


Figure 1.6: Global Buckling of a metal C shaped column [8].

For fiber reinforced plastic laminates, the crushing behavior of a C-shaped column is very different and more favorable with the right layup [8]. This is due to the fact that a composite column will fail in local buckling rather than global buckling. This local buckling causes a favorable folding failure mechanism which can be seen in Figure 1.7 [17]. Here local buckling failure causes the flanges of the column to fold, leading to a good amount of energy absorption and a higher and constant mean force(F_{mean}) during axial crushing [17]. Additionally, the crushing behavior is stable when the maximum force in the column F_{max}, does not exceed the global buckling force of the column [17].

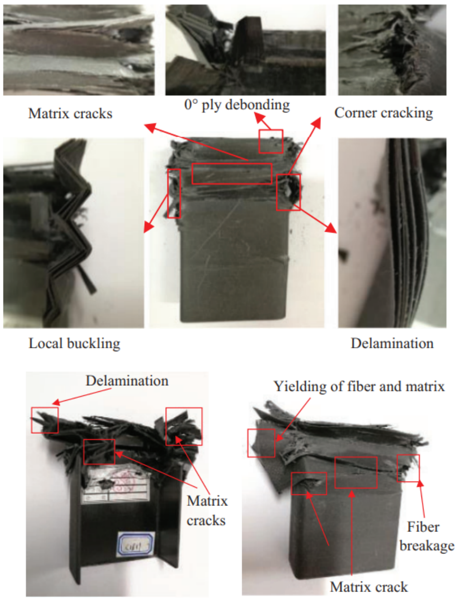


Figure 1.7: failure of a composite C shaped column [8].

Details

The detail level contains slightly more expensive and complex parts of the aircraft structure such as sub-floor webs and fuselage skin panels [8]. A small assembly procedure is often part of creating a detail and a detail is a non-generic specimen. More important for crashworthiness is the sub-floor column design depicted in Figure 1.8. This detail is part of the bottom part of the fuselage of an aircraft, the frame indicated in the figure is connected to the skin of the fuselage, and the floor beam holds the cargo or payload of the aircraft. Identification of the damage sequence and the evaluation of the energy-absorbing characteristics of the columns is a crucial part of this certification phase[8]. Different types of solutions exist for the geometry of the energy absorbing columns such as half tube columns [18], C shaped columns, or full tube columns with crush caps which causes debris to fill the column as the column is crushed, changing the strength of the tube during impact [9].

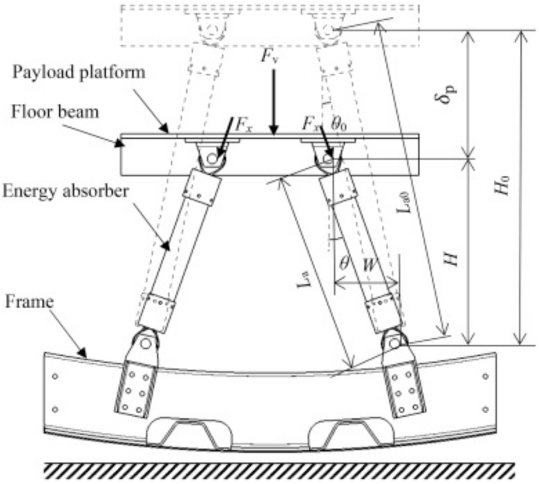
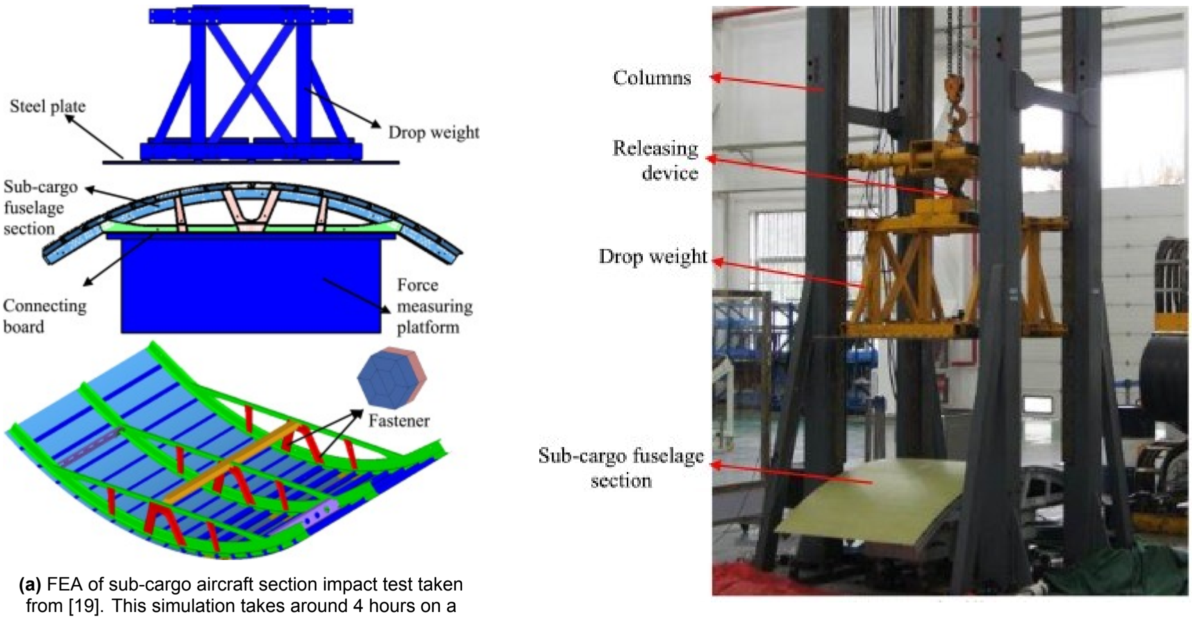


Figure 1.8: Schematic of sub-floor columns taken from [9].

Sub-components

Sub-components are non-generic specimens of a larger size than details. When zooming out further from the energy absorbing columns of the sub-floor, the design process for crashworthiness continues by looking at the sub-cargo fuselage section. Here, the crashworthiness behavior needs to be considered as a series of events in sequence. To analyse the crash mechanics, impact tests are performed such as the one in Figure 1.9b [19]. The results of this test are then used to verify the FEA in Figure 1.9a.



(a) FEA of sub-cargo aircraft section impact test taken from [19]. This simulation takes around 4 hours on a reasonably standard machine (2.1 GHz 8 core processor).

(b) Impact test of the same sub-cargo aircraft section [19].

Figure 1.9: Crashworthiness of a sub-cargo aircraft section with inverted drop testing validation [19].

When the FEA is further verified based on the impact tests, it becomes possible to further identify failure modes of different components such as in figure 1.10. By identifying the order in which components are affected, and by obtaining acceleration responses of the sub-cargo section through the verified FEM, the designer can start to construct an appropriate response of the section.

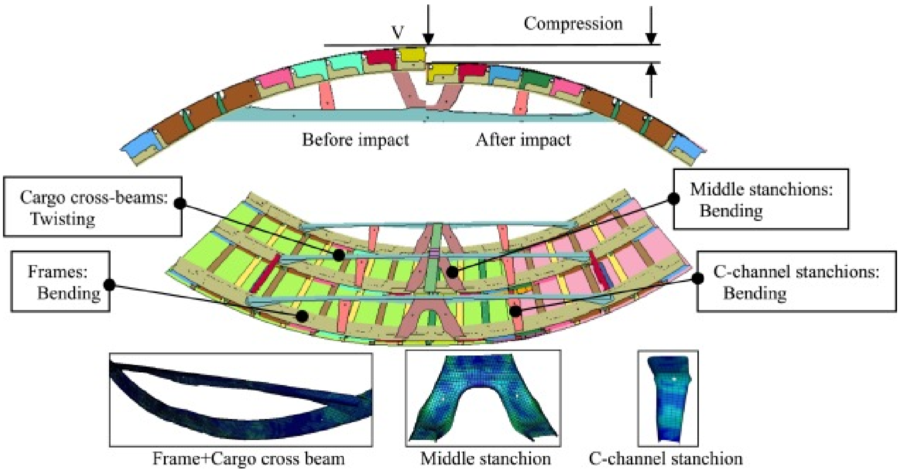
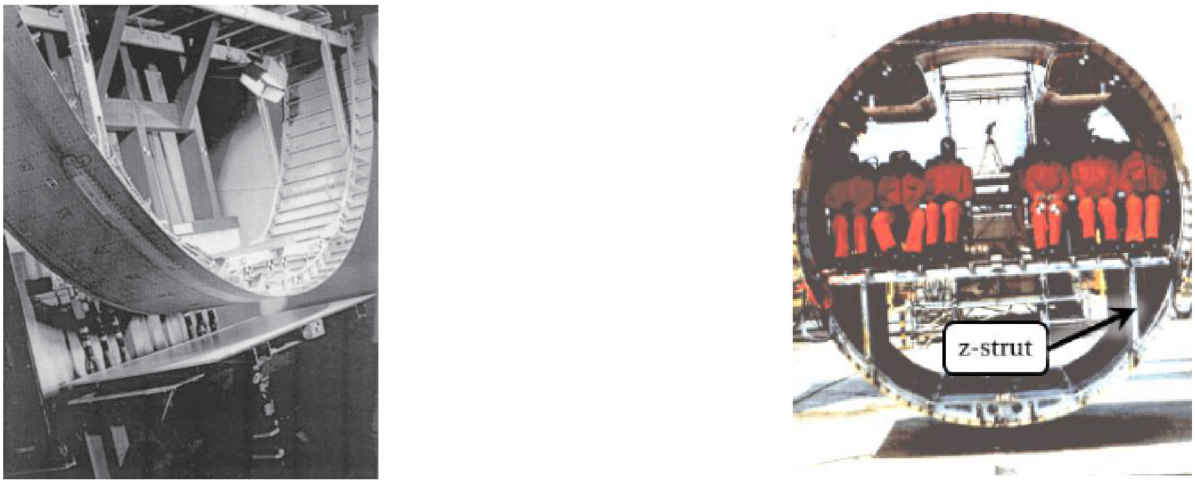


Figure 1.10: Results of the FEM simulation of the sub-cargo section of the aircraft [19].

Components

Last in the crashworthiness development process is the fuselage section, most commonly tested by using a drop test. As mentioned in section 1.1.2, the fuselage needs to withstand a crash with a vertical impact velocity of 9.14 m/s (30 ft/s). Therefore, for certification, it is best to conduct a drop test with this impact velocity as a result. However for the FEM verification process, the devastating failure of the section at this impact velocity makes it difficult to use the results of the drop test for the verification process. For this reason, impact/drop tests are also performed at smaller impact velocities of around 6.1-6.7 m/s [8]. Additionally, there are two types of specimens possible, the full fuselage section that will be placed in the actual aircraft, or a specimen that is only a part of the section's actual length. When using a partial section, often the cabin floor with seats are not present in the specimen and can be replaced with an equivalent mass. An example for each type of fuselage specimen section is given for the Airbus A320 [20, 21] in Figure 1.11.



(a) A shortened fuselage section placed on a impact test bench.

(b) A complete fuselage section including test dummies to be dropped

Figure 1.11: Different types of fuselage section for crashworthiness testing.Fuselage section of two frames of an A320 (a) [20] and a complete fuselage section prepared for a drop test (b)[21].

Case Study: Fokker F28 Full-Scale Analysis

In 2020 NASA published the results of a full-scale crash test of a Fokker F28 Fellowship Aircraft, along with an LS-Dyna simulation that attempts to mimic these test results [22]. The research that follows this literature study will focus on analysing of regional aircraft, similar to the F28. Before the full-scale analysis was performed, three fuselage sections, depicted in figure 1.12 were drop tested.

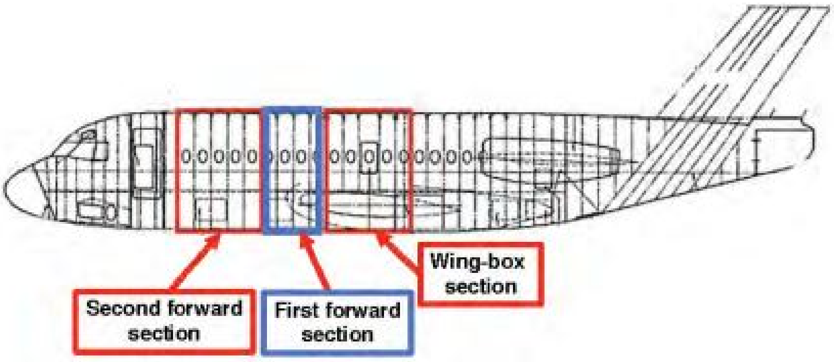


Figure 1.12: Different sections of the Fokker F28 fuselage that were selected for drop tests [22].

The individual sections are pictured before and after the drop tests in figure 1.13. In reference to the research that will follow this literature study, a number of interesting aspects will be highlighted. First of all, when looking at figure 1.13, it becomes clear that the geometry and function of the sub cargo section of the fuselage differentiates along the length of the complete fuselage. In figure 1.13c, the difference is more apparent, as that is the section where the wing is connected, resulting in an entirely different sub-cargo structure. In figure 1.13a and 1.13b, the structure is similar, however, in (a) there is no cargo present at the time of the drop test where as in (b) there is. The resulting failure mode changes from unrolling to flattening failure kinematics as discussed previously in section 1.1.1. This means that per aircraft model there are multiple sections of which the crashworthiness performance needs to be investigated and certified. The failure of the sections are interdependent, meaning that the acceleration response of one sections depends on the structural properties of the others. The research that follows this literature study will constrain itself to one type of fuselage section to fit the time frame of the research. Throughout the research in [22], an important goal was to validate the FEM of the fuselage by comparing the acceleration response curve of the drop test to the model in different locations within the fuselage. This reinforces the idea of using the Force-Displacement method discussed in section 1.1.1.

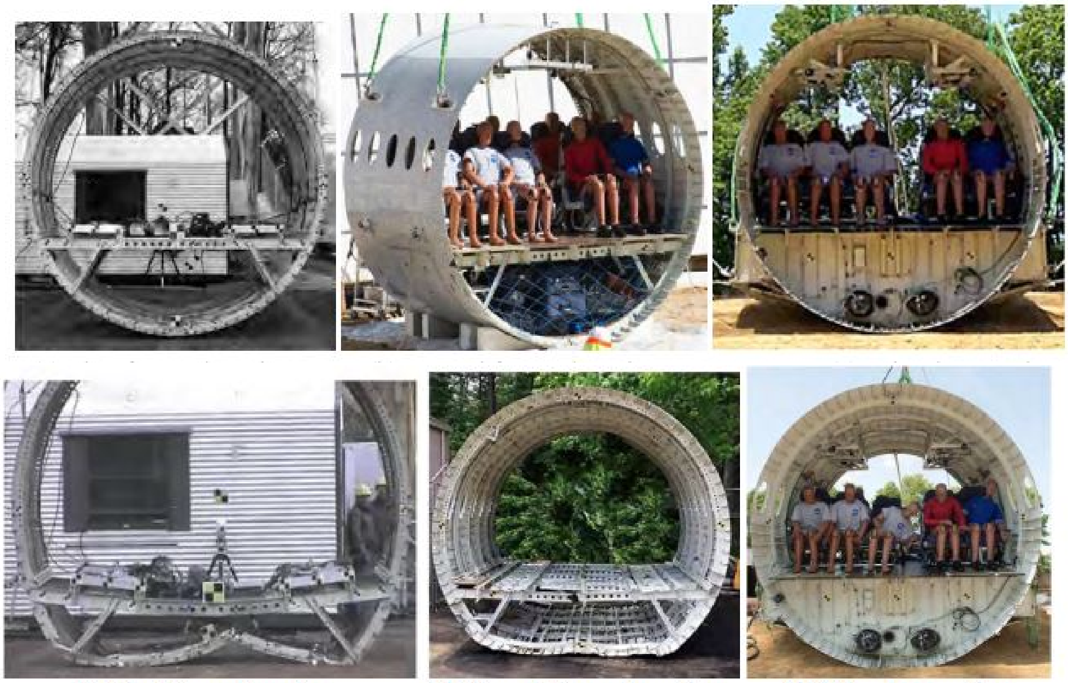
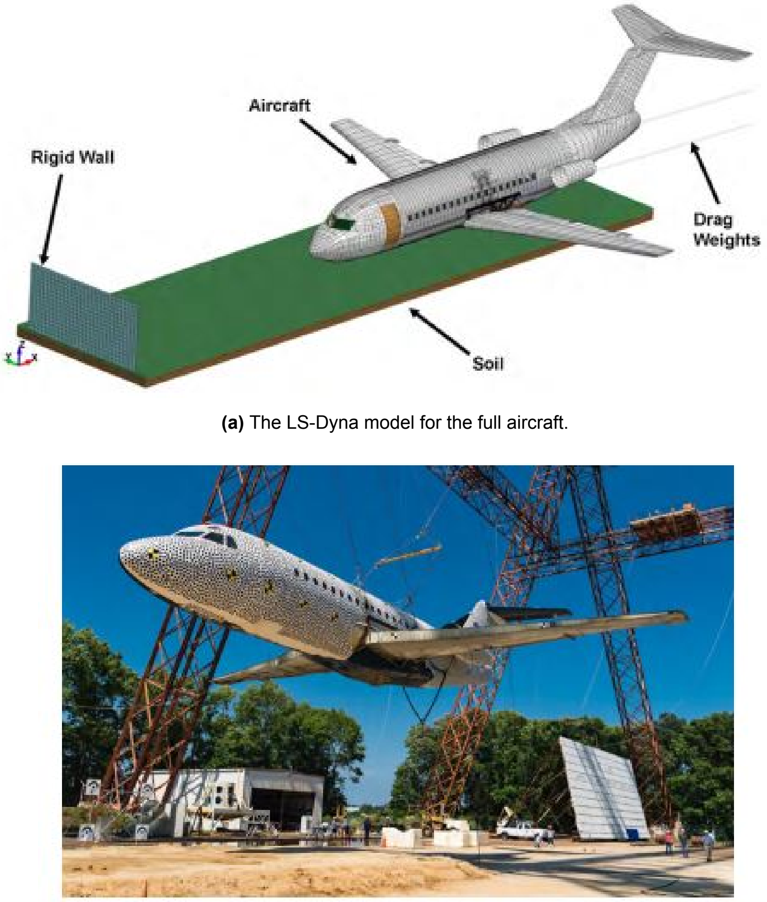

(a) First forward section. (b) Second forward section. (c) Wing-box section.

Figure 1.13: Fuselage sections of the Fokker F28 indicated in figure 1.12 drop tested by NASA [22].

After the individual drop tests on the fuselage sections were performed, the research was scaled up to a full F28 aircraft drop test. The aircraft was modified slightly, for example, the engines were removed, and also parts of the interior. The dropped F28 is found in figure 1.14. Along with it, an image of the FEM build in LS-Dyna is depicted.



(a) The LS-Dyna model for the full aircraft.

(b) The full F28 structure used for the drop test, which was slightly modified and painted with black dots for collecting photogrammetric data.

Figure 1.14: The drop test and finite element model of the entire F28 which crashed into a soil bed at a vertical velocity of 9.69 m/s [22].

The acceleration response of the pilot seat can be found in figure 1.15. When looking at the figure it becomes clear that a dedicated FEA still shows discrepancies when comparing to real life test data. Nonetheless, the authors of [22] still conclude that the FEA is predicting the duration of the response and the overall shape well. With this is meant that even though certain acceleration peaks are under/over predicted, they are still predicted to happen at the correct time. Also, the time it takes for the aircraft to become stationary is compared. The authors conclude when looking at the pilot seat location response specifically that the model 'showed excellent comparison with the test' [22]. For the research that follows this literature study this is a relevant observation, as it gives a benchmark of what the state-of-the-art of FEA can offer and what the industry deems sufficient. It can also act as a guideline for assessing the performance of a surrogate model that predicts structural responses of fuselage sections.

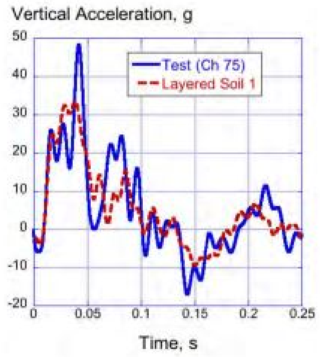


Figure 1.15: The vertical acceleration response of the pilot seat of the LS-Dyna model and of the full aircraft drop test. Here the term ”layered Soil 1” refers to the FEA and Test (Ch 75) to the drop test.

1.1.4. Key Observations

In this section, the most important aspects and takeaways for the thesis work that follows this literature study will be highlighted. First of all, in section 1.1.1 the general layout of an aircraft fuselage that will be investigated for the upcoming research has been explained, and also the general crash kinematics that were observed in drop tests and experimental work has been laid out. Here, the structural design and behavior of the sub-cargo area is very critical in delivering a survivable impact response and is therefore the primary focus of the upcoming research. Furthermore, it became clear that a number of parameters, such as the presence of cargo and characteristics of the cargo can have an effect on the resulting failure of the crash absorbing structure. This adds a level of complexity to the required analysis which does not fit in the time frame of the research that follows this report. Therefore, the presence of cargo loads will be neglected going forward. Section 1.1.1 also contains an overview of crashworthiness performance indicators. These indicators are all viable candidates for use in the surrogate model that will be built following this literature study. The most advanced and preferred indicator would be the force-displacement (FD) response. If this response can be predicted accurately by the surrogate model, it would mean that the other indicators such as maximum and mean crushing force could be easily deduced as well. However, the prediction of an accurate force-displacement response is a complicated task, and thus making this a goal is not the first priority of this research. The research will start off with the prediction of the SEA, mean and maximum crushing force through a surrogate model, and depending on those results can be expanded towards the FD response. This strategy will not add any FEM computation cost, since the FD response can simply be extracted from any performed simulation and added to the training data at any time. Lastly, the intrusion performance indicator will be omitted in the research that follows this study. This is because the research will be focused on the analysis of the crash absorbing structure below the cargo hold. In this area of the fuselage, intrusion does not harm any occupant within the aircraft.

Then in section 1.1.2 and 1.1.3 the regulations and certification of aircraft crashworthiness behavior were discussed to fully describe the problems and challenges faced in the industry. With this, the building block approach was introduced, which is a common technique to develop aircraft structures. By studying the different layers of the building block a good understanding has been obtained of the role and crush behavior on each scale of the crash absorbing structure. For the research that follows this literature study, inspiration will be taken from this approach. The surrogate model that will be built, will be first applied to a structure of a scale on the ’detail’ level described in section 1.1.3 before moving to the intended scale of the sub-cargo fuselage section. From the full-scale analysis of the Fokker F28 by NASA [22] in section 1.1.3 the current state-of-the-art of FEA was displayed against physical tests. This can help assess the surrogate model output qualitatively.

1.2. Designing Artificial Neural Networks to Predict Structural Behavior

Artificial Intelligence (AI), Machine Learning, Deep Learning, and (Artificial) Neural Networks (ANN) are all buzzwords since the second half of the 20th-century [23]. Although these words are sometimes used interchangeably, they do not define the exact same thing. The idea of creating artificial life or intelligence (AI) was already philosophized about before the computer was invented [24]. However, AI as we know it is essentially the ability of a machine to mimic or perform tasks that human beings can as well. An early application of this was for example the supercomputer that plays chess called Deep Blue by IBM which started development in 1985. This is not an example of Machine Learning however, which is a subset of Artificial Intelligence in which a machine requires data injection to improve predictions. This can, but does not have to involve the use of ANN. Deep Learning is a type of Machine Learning that depends heavily on the use of ANN and is, therefore, a subset of both Machine Learning and ANN. This explanation may be easier to understand when looking at the Venn diagram depicted in Figure 1.16 [25]. This chapter will give an overview of some basics of machine learning. Then, the workings and considerations behind the design of an ANN will be described.

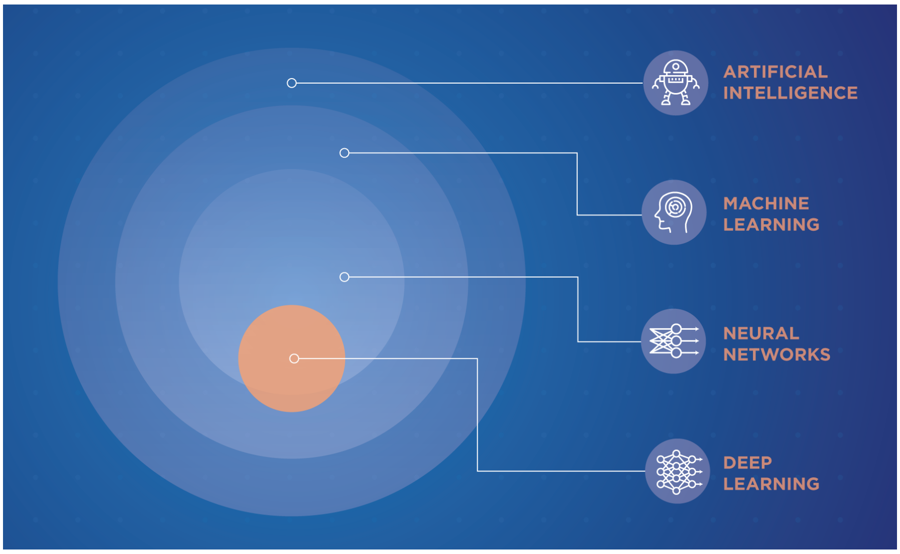


Figure 1.16: Artificial Intelligence and its subsets taken from [25].

1.2.1. Machine Learning

Before diving into Deep Learning, a general understanding of Machine Learning needs to be established. A Machine Learning algorithm is a learning algorithm, that is dependent on data input. The algorithm learns from an experience 'E' with respect to some task 'T', it uses a performance measure 'P' to determine whether the algorithm is getting better at task T as the experience E grows [24, 26]. There are a lot of Tasks, Performance measures, and Experiences, that an algorithm can possess. The ones that are important for this research will be highlighted.

- **Tasks**

  The task of the learning algorithm is the function that needs to be performed. The designer of the algorithm wants the algorithm to solve a problem by mapping a certain input to an output. Two important tasks are:

  1. **Classification**: Classification is a task in which the algorithm is given a list of classes. A class is one entity to which the input can belong. When the algorithm is given one or more inputs, it needs to give an estimate of what class these input(s) belong to. A common example of classification algorithms is image recognition programs. These programs are familiar with a set of objects (e.g. a car, a house, and a fish). Each object in this case is a class. The program is then given an image it has never seen before and needs to determine which object is presented in the image. The quality of the input can vary, the algorithm can still

try to establish a prediction if it is not given a clear image. In structural engineering practices, the classification task can be applied as well. For example, a classification algorithm can analyse the vibration response of a failed composite laminate and determine the failure mode of the laminate [27]. Here the list of failure modes consists of the different classes and the vibration response is the input.

2. **Regression**: A regression task is one where the algorithm does not have a certain class as output but rather a numerical value. The algorithm is given one or more input(s). It is very similar to any regression technique in classic Calculus. It can for example be used in economics for determining the change in the price of a certain item depending on some inputs[24].

- **Experience**

  In Machine Learning there are two main types of learning experiences, supervised and unsupervised. Unsupervised learning experiences are out of scope for this research. They are mainly used to let algorithms find patterns or clusters of a dataset [24].

  Supervised learning is different from unsupervised learning in that the algorithm is not only fed a dataset, but also the correct answer or target. This allows the algorithm to make a prediction based on the data and subsequently check the target. If the performance measure is unsatisfactory, the algorithm will have to adjust itself to gain performance.

- **Performance**

  The Performance of an algorithm needs to be expressed as a quantitative measure to allow the Machine Learning algorithm to evaluate itself. The parameter(s) used to measure the performance are highly dependent on the task that is being performed. In Supervised learning and Regression/Classification, the most common parameter is probably the accuracy/error. In classification, this would be a value between 0 and 1 that indicates the fraction of examples the algorithm predicts correctly [24]. Similarly, in regression, the error can be calculated by looking at the target and an error percentage can be obtained [24].

1.2.2. The Artificial Neural Network

The Artificial Neural Network (ANN) consists of many neurons which are grouped in layers. Each layer receives and passes on a signal or output. A neuron is connected to one or more neurons in the previous layer and to at least one or more in the coming layer. A neuron's output will depend on the input from the connected neurons in the previous layer. This idea is based on the way that the human brain performs. The connections between neurons and the way they react to the input they receive are dictated by the training process of the network. It could also be, that certain connections or information is purposely removed or cut away by the designer during training. This is called dropout and it is sometimes necessary to prevent the network from overfitting. An overfitting ANN is in short simply memorizing all case studies in their training, instead of finding useful patterns/generalizations with which it can predict case studies that have significant differences compared to the training. It is therefore important when training an ANN to use a training data set and a test data set. There is no significant difference between these two sets, only that the test data set has to be withheld from the ANN during training, so that the ANN can be tested on data of which the target is known later on. Opposite to overfitting is underfitting, when an ANN cannot find patterns or generalizations during training due to the complexity of the problem being too large for the network. Intuitively this is logical, as one can imagine that the more complex problems an ANN needs to solve, the more complex the ANN has to become.

To teach a network to perform a task, a training dataset is required to gain experience, and a performance measure is needed to evaluate the network's quality. But before that, a definition of a neuron needs to be established.

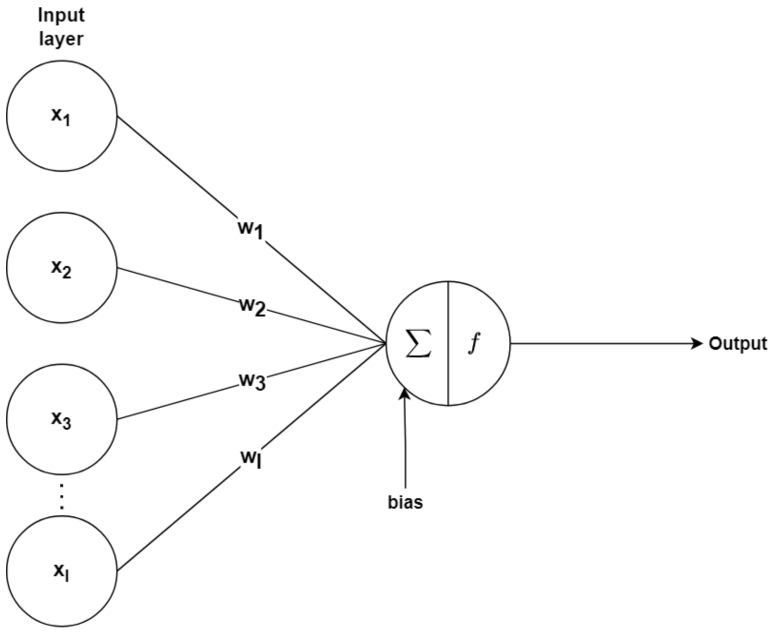


Figure 1.17: An input layer connected to an individual artificial neuron.

A single artificial neuron is depicted in Figure 1.17. This neuron receives input from the previous layer. In this case, this is the input layer, but the layer of which a neuron receives input can also simply be the neurons from the previous layer. The neuron will receive the output values, called the activation values, from the input layers which are $\{x_1, x_2, x_3....x_l\}$. There is also a weight assigned to each connection between the neurons which in vector form is $\{w_1, w_2, w_3....w_l\}$. Lastly, there is a bias value b, which is a numerical value that can be assigned to a neuron. The purpose of the bias is to add a degree of freedom to the ANN so that it can take an increased number of shapes. It allows a neuron to remain inactive (give an output of 0 or close to 0) if a certain threshold of the weighted sum is not reached. These three components are then combined to form the total input z of the neuron as follows:

$$z = \sum_{i=1}^{l}(w_i x_i) + b \tag{1.4}$$

The input z is then inserted into the activation function f which then produces the final output of a neuron, also known as the activation $'a'$ of the neuron.

$$a = f(z) = f(\sum_{i=1}^{l}(w_i x_i) + b) \tag{1.5}$$

The activation value is then passed onto the next layer of neurons.

The activation function of the neuron is of great importance for the network's ability to learn (quickly) and there does not exist a standard function that can be applied to every problem. It is therefore up to the designer of the ANN to choose the right activation function. The key aspects in choosing an activation function are the output range of the function, the derivative and for non-linear problems, a non-linear activation function is required. The derivative is important for the backpropagation method. Backpropagation is an important method that is used for learning. This method and the learning process will be explained in section 1.2.2. Some popular activation functions will now be listed below [24].

- **Rectified Linear Unit (ReLU)**: ReLU neurons are very popular in ANN applications presently [24]. The activation of the ReLU neurons is calculated as follows:

  $$f(z) = max(0, z) \tag{1.6}$$

  The function is non-linear and the derivative is equal to 1 when $x > 0$, equal to 0 for $x < 0$ and undefined at $x = 0$. z here is the input of the neuron, which is found by using equation 1.4. A disadvantage of this function is that if the input of the neuron is equal to a value far below zero, the neuron is inactive and sometimes stays inactive because the step size of weight changes during learning are too small to make the input switch to a positive value. This is known as the ’dying’ neuron problem. For sparse data applications, they are proven to work better than Sigmoid and Hyperbolic Tangent neurons [28].

- **(Logistic) Sigmoid**: Before the introduction of the ReLU function, the Sigmoid function was most popular [24]. With its output always lying in the interval $(0, 1)$, its a non-linear function with a well defined derivative which is continuous.

$$f(z) = \frac{1}{1 + e^{-z}} \tag{1.7}$$

- **Hyperbolic Tangent**: This function is fairly similar to the Sigmoid function. It resembles the Identity function slightly better in the sense that it crosses the origin and can be better trained because it is linear near the origin [24].

$$f(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}} = tanh(z) \tag{1.8}$$

- **Softmax**: The Softmax function is often used for the output layer of a classifier ANN. This is because this function yields a value between 0 and 1 and the sum of all the output activations of the Softmax layer is guaranteed to sum up to 1. Each Softmax neuron, therefore, is giving a probability for whether the input corresponds to a certain class.

$$f(z)_i = \frac{e^{z_i}}{\sum_j e^{z_j}} \tag{1.9}$$

- **Linear Activation Function (Identity)**: The Identity function or Linear Activation Function is only applicable in ANN when solving linear problems since the function itself is simply not transforming the input. Its range and derivative are quite trivial.

$$f(z) = z \tag{1.10}$$

When designing an ANN, there is not a perfect rule or guideline regarding the number of layers and neurons. There doesn’t exist a network that can solve every problem, and therefore it is up to the designer to create a network that produces useful results for the application. There are different families of neural network architectures that are specialized for certain applications. There are two main families of neural networks, Feedforward Neural Networks, and Recurrent Neural Networks. Recurrent Neural Networks are networks that are used for processing sequential data [24] (e.g. time series data). In this research, the data will not be sequential, and therefore the focus will be mainly on the Feedforward Neural Network.

Feedforward Neural Network

The Feedforward Neural Network is the most logical basis one can start off with. As stated in the name, the information given to the input layer is simply passed onto the layers, and there is no layer or neuron that receives feedback or information from neurons further down the ’stream’. The layers between the input and output layer are called the hidden layers, as seen in Figure 1.18. The dimensions of the hidden layers determine the so called width of the model. The number of hidden layers determines the depth of the network.

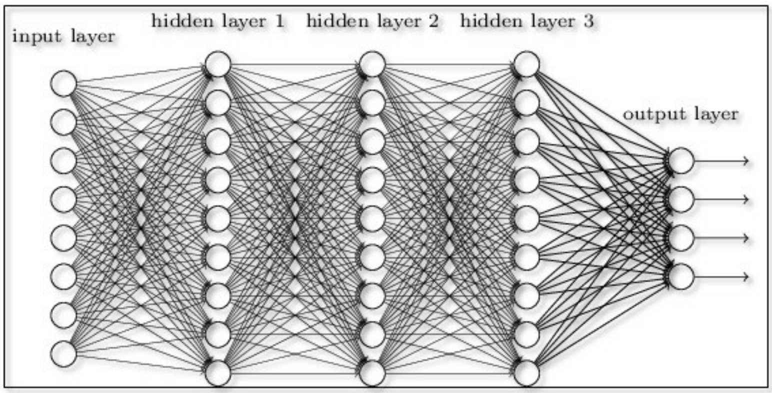


Figure 1.18: An example of a feedforward neural network architecture from [29]. Each circle represents a neuron, each column of neurons represents a layer.

Cost Function

The cost function is analogous to the performance measure described in section 1.2.1, and it is another important aspect of the design of the ANN. The cost is a measure of how accurately the ANN is doing its task. The cost function calculates this by comparing the output of the ANN and the target/label in the training data set. The learning algorithm will try to find a minimum of the cost. The cost function needs to calculate a numerical value for the cost. The function can differ depending on the task and application of the ANN. Some common cost functions are listed below.

- **Mean Squared Error & Mean Absolute Error**: These two cost functions are not well suited when combined with gradient-based learning networks according to [24]. In literature on crashworthiness optimization, seen in Chapter 1.3, this cost function is used however (e.g. in [30]). This contradiction may be caused by the difference in network tasks to which these cost functions are applied.
- **Cross-entropy**: This function significantly improves the learning ability of an ANN with activation functions such as the Sigmoid, which has a small slope in some regions. This is because the learning speed is related to the derivative of the activation function. So if the slope is small the learning is slow. This cost function cancels out the derivative of the activation function, and thus it learns quicker when the derivative of the activation function is small [31]:

  $$C = -\frac{1}{n}\sum_j (y_j \ln(a_j^L) + (1-y_j)\ln(1-a_j^L)) \tag{1.11}$$

  Here n is the amount of training examples in the dataset, $\sum_j$ is summing over all the output neurons, y_j is the target/label of the output neuron and a_j^L is the actual output of the neuron.

Gradient Descent Learning with Backpropagation

When training non-linear artificial neural networks, the cost function of the model is often non-convex. This means, that the function has multiple local minima. Thus, an ANN will not always be trained to produce the smallest error, because it does not find the global minimum of the cost function. However, this is often accepted by most ANN designers as it still provides a good solution within a reasonable time frame. The minimum that the learning algorithm finds depends on the initialization of the ANN. Most ANNs are trained using gradient descent-based techniques and that works as follows.

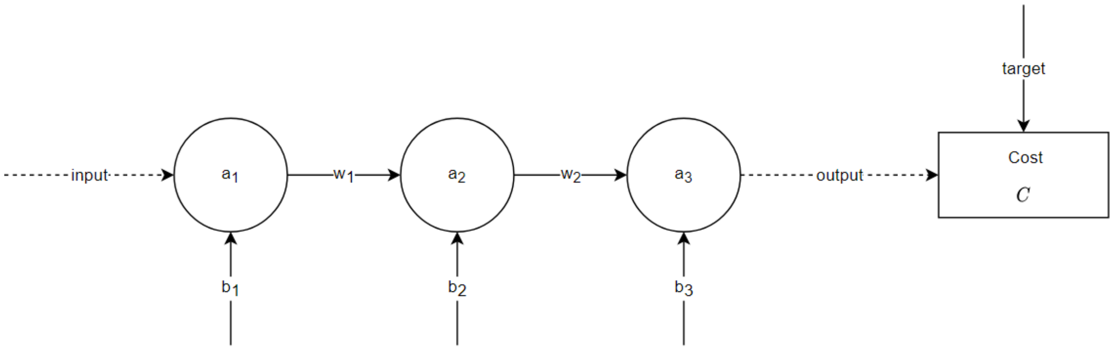


Figure 1.19: Three artificial neurons schematic

As one can imagine, the cost function is for most applications a non-convex function. The algorithm needs to find a minimum or saddle point, it will know when it has found one because there the gradient of the cost function is zero. To perform this gradient descent it is required to calculate the gradient with respect to every weight/bias in the ANN. Since there are many weights/biases which all have their partial derivative regarding to the cost, it would be incredibly time-consuming to evaluate each derivative numerically [24]. To solve this, most ANN learning algorithms use back-propagation to calculate the gradient of the cost function. Back-propagation uses the chain rule of calculus and has a specific order of operations that result in an efficient algorithm [24]. For example when considering the ANN consisting of three neurons directly connected in Figure 1.19, if one would like to compute the partial derivative of the cost with respect to w_2, $\frac{\delta C}{\delta w_2}$.

$$\frac{\delta C}{\delta w_2} = \frac{\delta C}{\delta a_3}\frac{\delta a_3}{\delta z_3}\frac{\delta z_3}{\delta w_2} \tag{1.12}$$

Where a_i is the activation output of neuron i ,w_i the weights,b_i the biases and z_i the weighted sum input as described in section 1.2.2. Similarly, it is possible to calculate the partial derivative w.r.t. the bias values (e.g. b_1, b_2, b_3). The weights and biases are then updated depending on the learning rate ϵ which is defined by the designer of the network in two ways. First, the initial learning rate is set which affects the learning process in the start phase, and secondly, an update rule for the learning rate is implemented, which is often done in the form of a decay rate that increases with the number of iterations such that the learning rate decreases as the final solution is approached. These derivatives calculated for the gradient descent method grow in size the further you move back in the network, and that becomes computationally expensive. For example, when the effect that w_1 has on the cost needs to be calculated the derivative becomes:

$$\frac{\delta C}{\delta w_1} = \frac{\delta C}{\delta a_3}\frac{\delta a_3}{\delta z_3}\frac{\delta z_3}{\delta a_2}\frac{\delta a_2}{\delta z_2}\frac{\delta z_2}{\delta w_1} \tag{1.13}$$

To generalize this algorithm to be programmed, the equations need to be rewritten such that the partial derivatives w.r.t. a weight/bias in layer l need to be calculated in terms of the derivatives of the next layer $l+1$. This allows the algorithm to move back into the network. They are derived in [31]. First the error δ^L of the output layer:

$$\delta^L = \frac{\delta C}{\delta a_j^L}\frac{\delta a_j^L}{\delta z_j^L} = \frac{\delta C}{\delta a_j^L} f'(z_j^L) \tag{1.14}$$

where f is the activation function, and $a_j^l = f(z_j^l)$ and thus $\frac{\delta a_j^L}{\delta z_j^L} = f'(z_j^L)$. Now an equation for the error of a neuron in terms of a neuron in the next layer is needed which is:

$$\delta_j^l = \sum_k w_{kj}^{l+1} \delta_k^{l+1} f'(z_j^l) \tag{1.15}$$

here j and k are integers that indicate certain neurons in the layers. With the two previous equations, it is possible to calculate the error in any layer, as there is an equation for the last layer and an equation which then can calculate the error in the layer before that and so on. It is then possible to find the derivatives that we are looking for to perform the gradient descent:

$$\frac{\delta C}{\delta b_j^l} = \delta_j^l \tag{1.16}$$

$$\frac{\delta C}{\delta w_{jk}^l} = a_k^{l-1} \delta_j^l \tag{1.17}$$

It is important to note that some partial derivatives calculated are used multiple times, especially when considering neural networks are much larger than the one in this example. One way of increasing computational performance is to memorize these partial derivatives or sub expressions [24]. Other types of elegant solutions to decrease the required computer memory and processing power to do back-propagation exist, however, this is out of scope for this literature research. Another method that is of importance to speed up the learning process is the so-called Stochastic Gradient Descent (SGD) method. in SGD, the large training dataset is divided into minibatches. They are often between 1 and 100 training examples large [24]. Instead of trying to compute the gradient over all the examples of the training dataset, SGD only computes the gradient over the minibatch. When the gradient descent has been completed on the minibatch, a new minibatch is formed with different training examples from the dataset. This allows for the weights and biases to be changed more often in the same span of time, as one iteration takes less long.

1.2.3. Key Aspects in ANN design

When designing or developing a neural network, there are some parameters that require trial and error and experience to initialize. For example, setting the initial learning rate is ”more of an art than a science” according to Goodfellow et al. [24]. Other aspects that influence the quality of an ANN and for which do not exist strict guidelines are listed and explained below.

- **Hyperparameters**: These are settings that must be determined before the learning process has started. As mentioned before, the learning rate is one of those parameters. The number of neurons and layers are also essential hyperparameters. Other hyperparameters are the activation functions of the neurons, batch sizes, and epochs. Most of these parameters are described in previous sections except the epochs. The Epoch parameter is an integer that determines how many times the complete training data set is passed through the ANN. Another unmentioned parameter is weight decay, which is a parameter that prevents overfitting. Goodfellow et al.[24] state that the tuning of the hyperparameters is mostly about finding the correct model capacity for the complexity of the task. Their informal definition of capacity is a measure of how many functions the model(=ANN) is able to fit. When the capacity is too large overfitting is more likely to occur and the opposite is true for underfitting. Goodfellow et al. provide a helpful look-up table (Table 1.2) to help with this issue.

Table 1.2: A general guidance table towards manually tuning hyperparameters for ANN taken from [24].

Hyperparameter	**Increases model capacity when**:	**Reason**
Number of hidden units	increased	Increasing the number of hidden units increases the representational capacity of the model.
learning rate	tuned optimally	An improper learning rate, too high or low, results in a model with low effective capacity.
Dropout rate	Decreased	Dropping units less often gives the units more opportunities to conspire with each other to fit the training data set.
Weight decay coefficient	Decreased	Decreasing this parameter frees the model parameters to become larger.

- **Data set quality and size**: data set quality is dependent on such aspects as noise, missing inputs, and wrong labels. Dataset size is of importance, if the set grows in size it becomes more computationally expensive to run it through the network, although it offers more opportunities for the network to learn. According to [24] it is best to gather more data when the network performs well on the training data set, but not on a test data set. Additionally, if the performance on the training dataset is poor, gathering more data will have little effect.
- **Network verification (debugging)**: Sometimes a network will miserably fail to learn anything. This can be caused by a multitude of problems or just a simple issue. It is sometimes difficult to know what to expect from an ANN, so it is important to build some tools that can help determine whether the ANN setup is correct and the learning strategy executed as meant to be.

1.2.4. Key Observations

Throughout this chapter baseline knowledge of machine learning has been obtained that offers a comprehensive overview of the workings of ANNs and additionally some methods and guidelines to properly design ANNs to perform well. In this section, a number of observations and interests that were identified and will possibly carry relevance for the research that follows this literature study are listed. First of all, the primary goal of the research that follows this literature study is to predict structural behavior or performance. To do so there are two ANN tasks that are of interest for this research. The most important is the regression task. This task allows structural parameters to be predicted as a scalar quantity. The other task is classification, which in the framework of this research can serve as a failure mode identifier. The classification task is not a primary goal of this research, but if distinct failure modes come forward during the research a possibility would be to build an ANN that can predict the failure mode a structure will display. Furthermore, in this chapter, a number of observations were already made. Such as the use of the feedforward neural network for the research to come. This is because the research that will follow this literature study aims to create datasets from Finite Element Modeling (FEM). This data will not be sequential and thus the use of recurrent neural networks is ruled out for this research. Another would be to use supervised learning, since unsupervised learning is not really aimed to solve the problems in this research, this choice is rather trivial.

Regarding the learning method itself, the literature seems to unanimously agree to use gradient descent learning with backpropagation. Since the results in all literature studied in this report seem satisfactory with this algorithm, and for lack of a proper contender in this area, the ANN that will be built following this literature study will use this algorithm too. Where the coming research will try to diversify is in the area of the network architecture, dataset size, and cost function. As already stated a controversy between literature was found in section 1.2.2 regarding the suitability of the Mean squared/ Mean absolute error cost functions. Furthermore, other literature such as [31] explains the potential of the Cross-entropy function. This function is not seen in literature on structural crashworthiness as much as the mean squared error, but this is discussed in Chapter 1.3. It is therefore a candidate for evaluation during the coming research. The architecture and dataset size will also be discussed in more detail in Chapter 1.3.

1.3. Modeling and Optimization of Crashworthiness Behavior of Structures

This chapter is a review of the state-of-the-art on optimization of crashworthiness of structures using artificial neural networks (ANN). The question here is how to use the ANN, described in chapter 1.2, to model the behavior of a structure enduring crush impact. This will be described in section 1.3.1. For the surrogate modeling, different solutions have been demonstrated to produce satisfactory results, so this chapter will not establish conclusive optimal strategies but rather serve as a summary of what has already been done.

1.3.1. Modeling Structural Crashworthiness Behavior

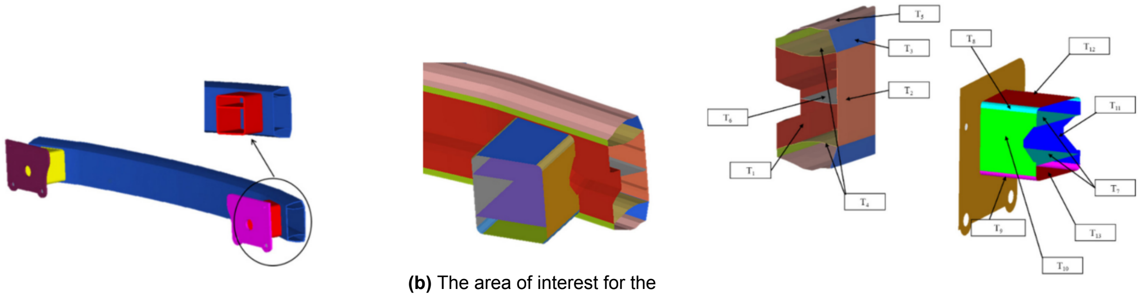
The current state-of-the-art of using ANN to model structural crashworthiness behavior will be described here. An ANN can be built with two different types of goals in mind, topology optimization [32, 33] and surrogate modeling. Topology optimization is a design strategy that forces some algorithm to build a geometry (from scratch), given some material properties and design loads (other constraints can be added as well). It uses a cost function to determine whether to change the distribution of material such that the cost is reduced [34]. For this research the interest lies more on improving an existing design with its many constraints, therefore this method would require incredibly complex definitions of the design space to achieve a viable design iteration.

Surrogate modeling (in structural applications) on the other hand, is about creating a computationally inexpensive model that uses data from a relatively much more expensive model. This inexpensive model will then, based on the data, try to predict at least one structural property of the same or slightly adjusted structure as the expensive model but requiring much less time/resources. The purpose of surrogate modeling is to consider many options in a design space in a short time frame, to allow the engineer to make a more informed decision on what the expensive model should be utilized for [35]. Naturally, the accuracy of the surrogate model is lower compared to the primary model but still needs to have significant accuracy to be of use. Surrogate models can also have other goals, such as approximating missing inputs and data mining [35]. With data mining is meant finding relations between certain (design) variables that are not known beforehand. This is somewhat relevant to this research, since ANN can work with missing inputs and data mining will in some way also be performed by the ANN when it is learning it is not a real priority or necessity.

The scale to which surrogate modeling is applied differs in literature. Most literature is more focused on individual structural elements, however, sub-assemblies or larger components are also considered. Existing literature will now be reviewed, where relevant/interesting aspects will be highlighted.

Vehicle Rear Bumper

First of all, C. li and I.Y. Kim [36] modeled the crashworthiness behavior of a vehicle rear bumper which is depicted in Figure 1.20. They found that against Response Surface and Kriging Method, the ANN produces a better response prediction to use in the optimization process. Here the ANN's output consisted of the following parameters: Weight, Peak force, Energy absorbed, and Intrusion (displacement). This was done using one ANN that has a neuron for each of these outputs in the output layer. The input variables were 13 different thicknesses of the bumper's different details which were constrained to a range (seen in Figure 1.20c).



(a) The complete bumper model

(b) The area of interest for the crashworthiness research

(c) The 13 design variables of the bumper

Figure 1.20: Crashworthiness optimization for a vehicle rear bumper, all images taken from [36].

To create a training dataset, the design space was sampled using the Latin Hypercube method. This method is seen quite often in similar literature, it is, therefore, worthwhile to include an explanation of this method in this section. According to [37], The Optimal Latin Hypercube Sampling (OLHS) concept divides the design space uniformly first, meaning that each design variable will have an equal amount of divisions between its minimum and maximum value. An example in Figure 1.21 (left) is given for a design space with two variables and three divisions giving nine sample points. If one would sample the design space in this manner, the number of sampling points would be equal to n^{var} in the general form where n is the number of divisions, and *var* is the number of variables. This approach would be highly problematic when dealing with problems that have a high amount of design variables. This would mean that for a design space containing 15 design variables for example, a minimum of two divisions per variable would result in 2^{15} data samples. Furthermore, in the example in the figure, with three levels per variable, you would only be able to fit a quadratic function to the points which would lead to incorrect results if the fitting function should be a higher-order function. An improvement would be therefore to sample the design space with a random sampling which is seen in Figure 1.21 (middle). In this case, there are nine levels for each variable with just nine sample points as before, however, the points are less distanced from each other and do not cover the same space. In Figure 1.21 (right) an Optimal Latin Hypercube (OLH) is shown. It has nine levels for each factor and the sampling points are evenly spaced, allowing to fit a highly nonlinear function. Thus OLH sampling carries both advantages of uniform sampling and random sampling. Additionally, a user of the algorithm would like to define the design space and the number of sample points beforehand. With this algorithm, it is only possible to choose the number of divisions. There is an addition to this algorithm that solves this issue which is found in [38]. In short, this is done by first generating an OLH that has at least the number of required points, but most likely more. Then the points furthest from the center of the design space are removed one by one until the desired number of points is reached. At last, the remaining points will be rescaled such that the sampled design space will span the same volume/area as before resizing [38].

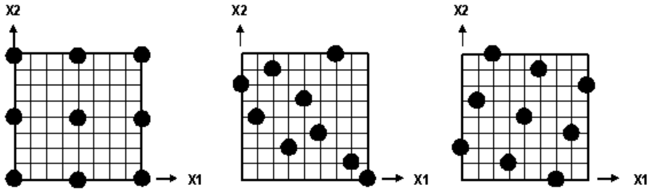
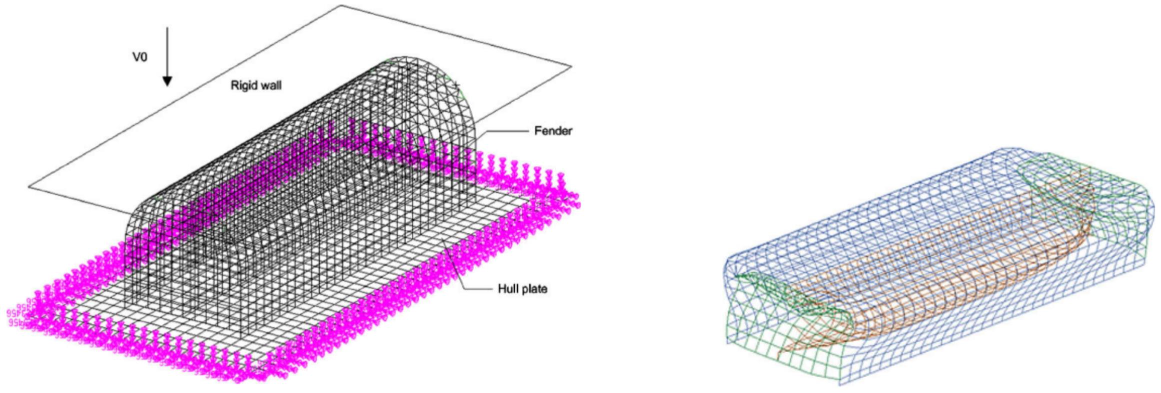


Figure 1.21: Optimal Latin Hypercube sampling for a design space with two variables x_1, x_2. Image taken from [37].

Ship Fender Structure

Next, the fender structure of a ship is investigated by Zhiyu Jiang and Mintong Gu [39], pictured in Figure 1.22. The fender is crushed more often than once in its service time. Additionally, it is not crucial to the survival of occupants of the ship. However, if the side structure of a ship is not well protected by the fender, the cost of repair/maintenance will increase rapidly.



(a) Finite element model of the fender structure for impact analysis.

(b) Deformed fender geometry

Figure 1.22: Design of a fender structure [39].

To assess the crashworthiness quality of the fender design, Jiang and Gu propose two indicators, the maximum force and the Specific Energy Absorption(SEA). SEA here is defined as:

$$SEA = \frac{EnergyAbsorbed}{\Delta l \cdot \textit{StructuralWeight}} \tag{1.18}$$

Where Δl is defined as the intrusion of the hitter [39]. Which is different from the usual definition of SEA:

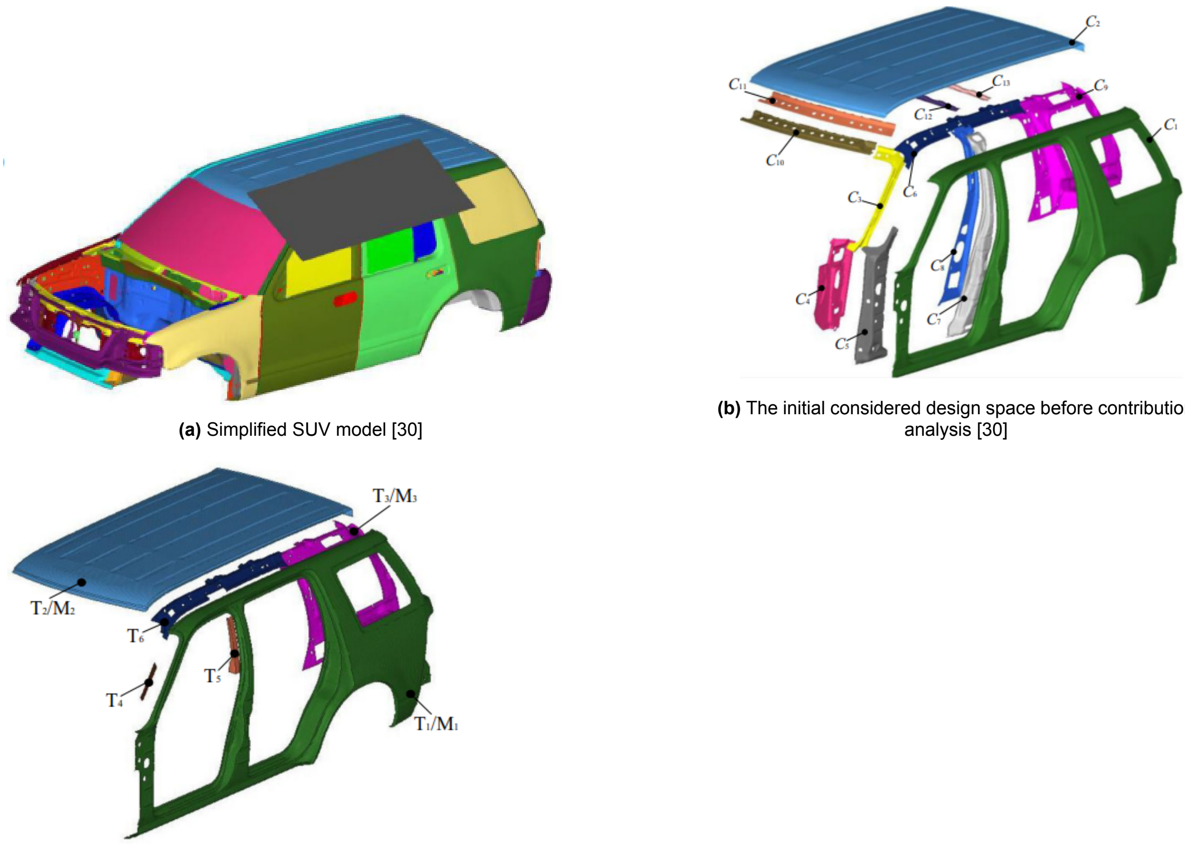
$$SEA = \frac{EnergyAbsorbed}{StructuralWeight} \tag{1.19}$$

It is arguably also possible to see this as an equivalent of three crashworthiness indicators (ordinary SEA, intrusion, maximum force). Jiang et al. use three thicknesses and the height of the fender as design space, creating a design space with four variables. Additionally, each variable is divided into four levels, creating 4^4 or 256 design points. They are the input of the ANN which is trained using backpropagation and gradient descent. More precisely, they use the scaled conjugate gradient training algorithm. The output of the ANN is the two crashworthiness indicators mentioned above. Furthermore, the architecture of the ANN has 2 hidden layers besides the input and output layer. 196 sets of training data are used from FEA and an additional 60 are used for testing the ANN. The training sets were evenly distributed in the design domain. Using trial and error, the architecture is chosen to be 4/21/21/2 (input/hidden/hidden/output layer) and the activation functions are, hyperbolic tangent(x2), Sigmoid, and then linear(Identity) function. from the test data set of 60 simulations, a maximum percentage error of around 6% is obtained. Although this result looks very promising, one should not forget that this was obtained by running a total of 256 FE simulations, which for larger applications would become extremely computationally expensive. For the cost function, a squared error is used. For optimization, multi-objective optimization is performed. This is done using NSGA-II. Sensitivity analysis is performed using Monte Carlo(MC) simulation and a single objective Generic Algorithm(GA).

SUV Rollover

Jin et al. [30] propose the use of a radial basis function (RBF) neural network to optimize the rollover crashworthiness of SUV's. The RBF neural network uses the Gaussian activation function within the hidden layer:

$$\varphi_i = \exp\left(-\frac{1}{2\sigma^2}\left\|\mathbf{X} - S_i\right\|^2\right) \quad i = 1, 2, \ldots, n_m \tag{1.20}$$



(a) Simplified SUV model [30]

(b) The initial considered design space before contribution analysis [30]

(c) The final design variables of the SUV body [30].

Figure 1.23: Crashworthiness rollover optimization research performed by Jin et al.[30]

Here, S_i is the Gaussian center, σ is the variance, and m is the number of neurons in the hidden layer. S_i is set using OLH sampling of the sampling space and the variance σ is set pre-training [30]. The network is trained using backpropagation and gradient descent. Then new pairs of learning rate η and variance σ are formed using a genetic algorithm and the network is trained again. After 50 iterations of 20 pairs per iteration, the optimal pair is chosen. This training algorithm combined with the genetic algorithm increased the accuracy of the RBF neural networks by 4.3-6.6% compared to the RBF network that was trained without the genetic algorithm. Furthermore, it reduced the maximum error and the root mean square error with a range of 23.5-42.1%. Seven thicknesses are selected as design variables from 13 by doing 27 FE simulations and subsequently looking at the change of contact force that the structure can withstand as function of each thickness. The same is done for the weight. The thicknesses that show significant contribution to the contact force and weight of the body are chosen as a design variable. Furthermore, three different materials are selected for optimization of different components. The output of the RBF network is the weight, the maximum contact force, and torsion frequency of the SUV body. The frequency is important for reasons of ride comfort, so it is not directly related to crashworthiness performance. Compared to the previously discussed research on the crashworthiness of a fender structure [39], this research by Jin et al.[30] shows that satisfactory results for the ANN surrogate model can be obtained with a much smaller number of FE simulations. It also illustrates the fact that large crashworthy structures often have other functionality requirements, in this case providing ride comfort, which need to be considered while optimization of crashworthiness is performed.

Helicopter Subfloor Structure

To optimize the subfloor of a helicopter structure for crashworthiness, Lanzi et al. [40] propose a framework that decomposes the complete structure into different elements, and then by building an ANN for each element and making use of symmetry, the structural optimization problem is simplified.

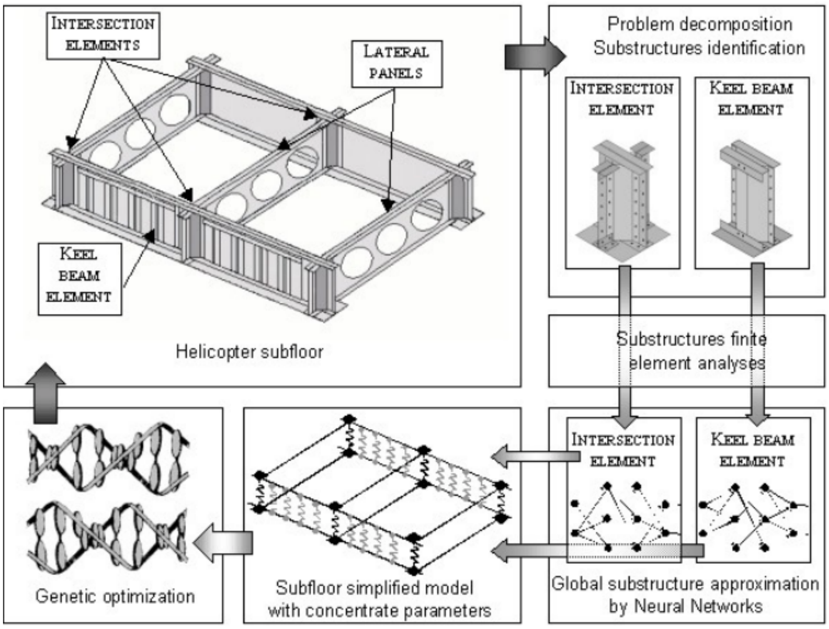


Figure 1.24: The optimization strategy used by Lanzi et al. to optimize the structure of a helicopter subfloor [40].

The ANN system has multiple outputs, the load-time curve, maximum force, and mean force. The system consists of multiple independent networks, working in parallel. The cost function used in the network is the RMS function. The structure is decomposed into two substructures, the amount of design variables per substructure is four and three variables. Contrary to the previously discussed optimization problems, these variables are not just defining thicknesses, but also the number of rivets, width, length, and positioning of reinforcing smaller elements. The ANN system is compared to the FEM analysis, and the accuracy of the outputs is in a satisfactory error range.

Component Design

Multiple research papers were published on aerospace related design and optimization of smaller components which are part of a larger crash structure. A lot of research has been done, for example, on composite tubes being crushed in axial direction [41, 42]. An interesting parameterization study in the experimental investigation performed by Laban et al. [41] on the performance of the ANN versus the architecture of the hidden layers. The outcome of this study is that two hidden layers give the best MSE accuracy compared to a one hidden layer architecture, but the difference is not that significant. Other than that this research offers no useful insight to the current research because a dataset of over 45000 data sets was used to train the ANN, which is a too excessive amount for any feasible study on larger structures. The research of Wang et al. [42] on the mechanical properties of braided-textile reinforced tubes includes the use of an interesting empirical formula that could serve as a starting point for the architecture of the ANN when using two hidden layers.

$$L = \sqrt{m+n} + a \tag{1.21}$$

Where m and n are the number of nodes in the input layer and the output layer respectively. a is an integer between 1 and 10 [42].

1.3.2. Key Observations

In the presented research in section 1.3.1, ANN surrogate models were developed for crashworthiness applications of different structural magnitudes. Already highlighted in the chapter was the varying dataset size used in these papers. Naturally, for a larger design space more data samples are required, this is to allow the ANN to learn the effect of each dimension in the design space. the complexity of the structure is suspected to have a role in this as well. An aspect that was missing and will be investigated in the research that follows this literature study is the dataset size versus performance analysis. This is done in chapter 6. The reviewed literature generally does not examine quantitatively or qualitatively to what extent the number of data samples can be changed in order to increase or decrease accuracy of the ANN. Knowledge on this, even if not very sophisticated, could be valuable for any further research. When looking at the architecture of the ANNs seen in literature, they are generally one or two hidden layer networks. The activation functions used for the layers differ per research. It is impossible to compare these different strategies as candidates for the upcoming research, but the intention is to start with trying similar networks as seen in the literature of this chapter and propose new ANN architectures from there.

2

LS-Dyna Modelling

The work of this thesis is for a large proportion supported by the Finite Element Modeling (FEM) software called LS-Dyna. The purpose of this chapter is to gain a more thorough understanding of how the software works. A bit more information about the software is given in section 2.1. A goal of this thesis is to model unidirectional fibre-reinforced thermoplastic composite laminates in LS-Dyna. Therefore, three different material models that are capable of doing this are investigated in section 2.2. These material models are tested by applying them to a coupon in section 2.3. Lastly, in section 2.4 one material model is applied to a hollow tube. The tube is subjected to a crushing force by modelling a rigid plate that impacts the tube in the axial direction.

2.1. Introduction to the LS-Dyna Solver

LS-Dyna is a non-linear transient dynamic Finite Element Analysis (FEA) solver by Ansys specifically developed for crashworthiness analysis of structures. The solver uses an explicit time integration algorithm (an implicit version exists as well but is not commonly used). The explicit method offers quick computation and is conditionally stable. The time step needs to be lower than a critical value to allow good solutions. The critical time step is dependent on the wave propagation velocity of the material because the time step needs to be less than the length of time it takes to travel through the material from one node to the next at the speed of sound.

The solver has its own input file format, known as the keyword file. Keywords are entities that define different aspects of the finite element model such as the nodes, materials, boundary conditions etc. The keyword file does not contain any history of how the model was made, therefore there is no undoing for most changes that are applied to the model.

2.2. Material Models

The material models in LS-Dyna are often referred to as material cards. The material cards hold options and fields to fill in parameters of the material model associated with the material card. The database of Ansys [43] offers information on every card available in LS-Dyna and describes the general purpose of the card. For this research the aim is to choose a material model/card that satisfies a number of requirements listed below:

1. The material model must be able to model the behavior of unidirectional fibre-reinforced thermoplastic composite laminates. Within this research, every layer of the laminate is assumed to have the same material properties. The only variable between layers is the angle and location within the ply stack.
2. The material will be loaded until failure, and therefore needs failure criteria along with the defined behavior beyond the elastic region.
3. The material card needs to be applicable to shell elements within LS-Dyna.

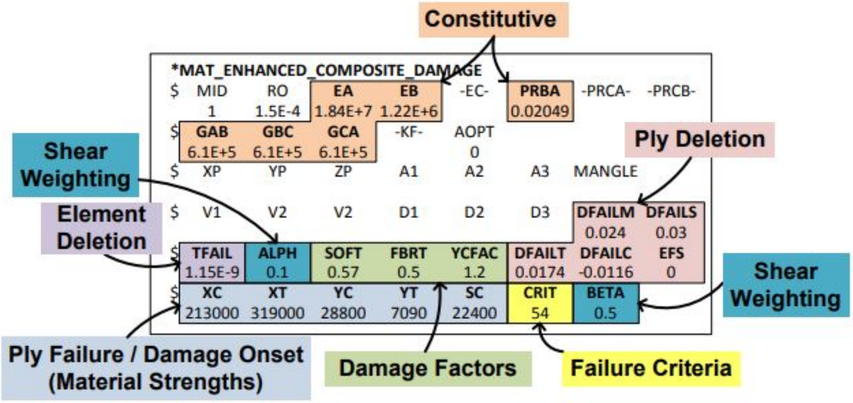
Based on these requirements there are a number of candidates found in [43] that are listed in Table 2.1 below.

Table 2.1: Candidate material cards in LS-Dyna to be used for this research.

Material card	Model Name	Description
MAT054 -MAT055	MAT_ENHANCED_COMPOSITE_DAMAGE	An enhanced composite model that can assume different failure criteria. The default option is Chang-Chang failure and the alternative is Tsai-Wu.
MAT058	MAT_LAMINATED_COMPOSITE_FABRIC	A model based on the Hashin failure criteria.
MAT261	MAT_LAMINATED_FRACTURE_DAIMLER_PINHO	Pinho et al. [44, 45]

2.2.1. Mat054/055

This material card, depicted in Figure 2.1 is characterized by the constitutive parameters, which primarily define the elastic region of the material. Once the plastic region is reached, the damage and the failure criteria are determined using the damage factors and damage onset properties.



```
*MAT_ENHANCED_COMPOSITE_DAMAGE
$  MID      RO       EA       EB      -EC-     PRBA    -PRCA-  -PRCB-
    1     1.5E-4  1.84E+7  1.22E+6            0.02049
$  GAB      GBC      GCA     -KF-     AOPT
 6.1E+5   6.1E+5   6.1E+5             0
$  XP       YP       ZP       A1       A2       A3     MANGLE
$  V1       V2       V2       D1       D2       D3     DFAILM  DFAILS
                                                        0.024   0.03
$ TFAIL     ALPH     SOFT     FBRT    YCFAC   DFAILT   DFAILC   EFS
 1.15E-9    0.1      0.57     0.5      1.2    0.0174  -0.0116   0
$  XC       XT       YC       YT       SC      CRIT     BETA
 213000   319000   28800    7090    22400     54       0.5
```

Figure 2.1: An image indicating the parameter groups of the MAT054 material card [46].

The failure is implemented through the Chang-Chang failure criteria or the Tsai-Wu. This can be changed through the CRIT parameter in the keyword definition. The Chang failure criteria have four failure modes, tensile/compressive fiber/matrix failure. For the tensile fiber mode the criterion is:

$$\sigma_{aa} > 0 \Rightarrow e_f^2 = \left(\frac{\sigma_{aa}}{X_t}\right)^2 + \beta\left(\frac{\sigma_{ab}}{S_c}\right) - 1 \begin{cases} \geq 0 & \text{failed} \\ < 0 & \text{elastic} \end{cases} \\ E_a = E_b = G_{ab} = v_{ba} = v_{ab} = 0 \tag{2.1}$$

where a, b, c refer to a,b, and c-direction respectively, e_i is a scalar that indicates whether the material is failing or in elastic loading. ν_{ij} indicates the Poisson ratio, G_{ij} the shear modulus, E_i the Young's modulus in direction i. σ indicates the stress. X_t and S_c are the strength of the material in longitudinal tension and shear. β is a weighting factor between 0 and 1 that controls the shear term in the tensile fiber mode. The compressive fiber mode is then:

$$\sigma_{aa} < 0 \Rightarrow e_c^2 = \left(\frac{\sigma_{aa}}{X_c}\right)^2 - 1 \quad \begin{cases} \geq 0 & \text{failed} \\ < 0 & \text{elastic} \end{cases} \\ E_a = v_{ba} = v_{ab} = 0 \tag{2.2}$$

The equation for the tensile matrix failure mode:

$$\sigma_{bb} > 0 \Rightarrow e_m^2 = \left(\frac{\sigma_{bb}}{Y_t}\right)^2 + \left(\frac{\sigma_{ab}}{S_c}\right)^2 - 1 \begin{cases} \geq 0 & \text{failed} \\ < 0 & \text{elastic} \end{cases} \\ E_b = v_{ba} = 0 \Rightarrow G_{ab} = 0 \tag{2.3}$$

Similarly, for the compressive matrix failure mode:

$$\begin{gathered}\sigma_{bb} < 0 \Rightarrow e_d^2 = \left(\tfrac{\sigma_{bb}}{2S_c}\right)^2 + \left[\left(\tfrac{Y_c}{2S_c}\right)^2 - 1\right]\tfrac{\sigma_{bb}}{Y_c} + \left(\tfrac{\sigma_{ab}}{S_c}\right)^2 - 1 \begin{cases} \geq 0 & \text{failed} \\ < 0 & \text{elastic} \end{cases} \\ E_b = v_{ba} = v_{ab} = 0 \Rightarrow G_{ab} = 0 \\ X_c = 2Y_c, \text{ for } 50\% \text{ fiber volume}\end{gathered} \tag{2.4}$$

When Tsai-wu is activated the material card becomes MAT055, which changes one aspect of the failure criteria. The failure mode for tensile and compressive matrix failure is checked using the following equation:

$$e_{\text{md}}^2 = \frac{\sigma_{bb}^2}{Y_c Y_t} + \left(\frac{\sigma_{ab}}{S_c}\right)^2 + \frac{(Y_c - Y_t)\,\sigma_{bb}}{Y_c Y_t} - 1 \quad \begin{cases} \geq 0 & \text{failed} \\ < 0 & \text{elastic} \end{cases} \tag{2.5}$$

The fibre-based failure modes are the same as the Chang-Chang model.

2.2.2. MAT058: Laminated Composite Fabric

This material card is using the Hashin failure criteria. The Hashin failure criteria are similar to the Chang-Chang failure criteria. Four equations are tested for failure for every load step in the material. Some variations of the Hashin failure criteria exist, so for clarity, the ones that are used by LS-Dyna are listed below, they are taken from [47].

If $\sigma_{11} \geq 0$:

$$e_m^2 = \left(\tfrac{\sigma_{11}}{X_t}\right)^2 - 1 \quad \begin{cases} \geq 0 & \text{failed} \\ < 0 & \text{elastic} \end{cases} \tag{2.6}$$

If $\sigma_{11} < 0$:

$$e_c^2 = \left(\tfrac{\sigma_{11}}{X_c}\right)^2 - 1 \quad \begin{cases} \geq 0 & \text{failed} \\ < 0 & \text{elastic} \end{cases} \tag{2.7}$$

If $\sigma_{22} \geq 0$:

$$e_{\text{m}}^2 = \left(\frac{\sigma_{22}}{\text{Y}_{\text{t}}}\right)^2 + \left(\frac{\tau}{\text{S}_{\text{c}}}\right)^2 - 1 \begin{cases} \geq 0 \text{ failed} \\ < 0 \text{ elastic} \end{cases} \tag{2.8}$$

If $\sigma_{22} < 0$:

$$e_d^2 = \left(\frac{\sigma_{22}}{\text{Y}_{\text{c}}}\right)^2 + \left(\frac{\tau}{\text{S}_{\text{c}}}\right)^2 - 1 \begin{cases} \geq 0 \text{ failed} \\ < 0 \text{ elastic} \end{cases} \tag{2.9}$$

In these equations, σ_{11} is the stress in fibre direction, and σ_{22} is the stress perpendicular to the fibre direction. Similarly to MAT058, X_t,X_c,Y_t,Y_c, and S_c are the strength in longitudinal direction in tension and compression, the strength in the lateral direction in tension and compression and the shear strength respectively.

2.2.3. MAT261: Laminated Fracture Daimler Pinho

This material model as mentioned in Table 4.1 is based on the work of Pinho et al [44, 45]. This material is a physics-based failure model. This model addresses a simplicity that exists in the two previously described material models. In post-failure of the MAT 54-55/58 the elastic properties of the material are reduced to zero in a fixed number of time steps[44]. That is an assumption that is not based on any physical equation, which is where the Daimler Pinho model differs. This model uses experimental data to determine the damage onset of the material. Furthermore, the model also contains a system of equations that puts emphasis on fibre-kinking failure. The equations that are implemented for the finite element model are found in [44] and the equations that determine the failure criteria are presented below.

For fibre tensile failure the following equation is used:

$$f_{\text{ft}} = \frac{\sigma_a}{X_{\text{t}}} = 1 \tag{2.10}$$

where σ_a is the stress along fibre direction, and X_t the tensile strength(axial)

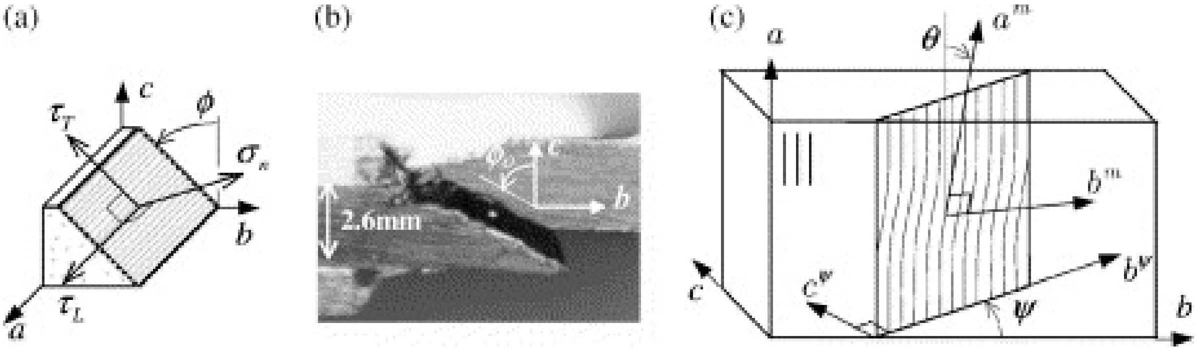


Figure 2.2: Images taken from [44]. In (a) the traction components of the fracture plane are shown. in (b) the failure angle for pure compression in the b direction is shown. Lastly, in (c) the orientation of a kink band for fibre kink failure definitions.

For the matrix failure, the traction components(σ_n, τ_T, and τ_L) defined in Figure 2.2a are used. The angle ϕ defined in figure 2.2a is not constant, but rather several trial angles are used in the range of [0,π] to compute the failure criterion[44].
For matrix compression:

$$f_{\text{mat}} = \left(\frac{\tau_{\text{T}}}{S_{\text{T}} - \mu_{\text{T}}\sigma_{\text{n}}}\right)^2 + \left(\frac{\tau_{\text{L}}}{S_{\text{L}} - \mu_{\text{L}}\sigma_{\text{n}}}\right)^2 = 1 \Leftarrow \sigma_{\text{n}} < 0 \tag{2.11}$$

Where μ_T and μ_L are the transverse and longitudinal friction-like parameters respectively. Similarly,S_L and S_T are the longitudinal/transverse shear strengths. The only independent variables that need to be set are the compressive strength Y_c, the longitudinal shear strength S_L, and the fracture plane angle (ϕ_0) that occurs for pure compression (pictured in 2.2b). The recommended value for ϕ_0 is $53°$.

$$\mu_{\text{T}} = -\frac{1}{\tan(2\phi_o)}, \quad S_{\text{T}} = \frac{Y_{\text{c}}}{2\tan(\phi_o)}, \quad \mu_{\text{L}} = S_{\text{L}}\frac{\mu_{\text{T}}}{S_{\text{T}}} \tag{2.12}$$

For matrix tension:

$$f_{\text{mat}} = \left(\frac{\sigma_{\text{n}}}{Y_{\text{t}}}\right)^2 + \left(\frac{\tau_{\text{T}}}{S_{\text{T}}}\right)^2 + \left(\frac{\tau_{\text{L}}}{S_{\text{L}}}\right)^2 = 1 \Leftarrow \sigma_{\text{n}} \geq 0 \tag{2.13}$$

where Y_t is defined as the transverse tensile strength.
For the fibre kink failure:

$$f_{\text{kink}} = \begin{cases} \left(\frac{\tau_{\text{T}}}{S_{\text{T}} - \mu_{\text{T}}\sigma_{\text{n}}}\right)^2 + \left(\frac{\tau_{\text{L}}}{S_{\text{L}} - \mu_{\text{L}}\sigma_{\text{n}}}\right)^2 = 1 \Leftarrow \sigma_{\text{n}} \leq 0 \\ \left(\frac{\sigma_{\text{n}}}{Y_{\text{t}}}\right)^2 + \left(\frac{\tau_{\text{T}}}{S_{\text{T}}}\right)^2 + \left(\frac{\tau_{\text{L}}}{S_{\text{L}}}\right)^2 = 1 \Leftarrow \sigma_{\text{n}} > 0 \end{cases} \tag{2.14}$$

2.3. Material Coupon modeling

Now that the candidate material models are discussed, they will be applied to a rectangular coupon. The coupon is clamped at one side and tensioned/compressed on the other by setting a fixed displacement according to the American Society for Testing and Materials (ASTM) test method D3039/D3039M-08(Tensile) and D3410/D3410M-16 (Compressive).

Table 2.2: ASTM standards for coupon tensile and compression tests.

Loading: Tension (T)/ Compression (C)	**Layup**	**Width [mm]**	**Length [mm]**	**Thickness [mm]**	**Tab Length [mm]**
T	0 deg unidirectional	15	250	1	55
T	90 deg unidirectional	25	175	2	25
T	Balanced/ Symmetric/ Random-discontinuous	25	250	2.5	Unspecified
C	0 deg unidirectional	10	140-155	See Eq. 2.15	65
C	90 deg unidirectional	25	140-155	See Eq. 2.15	65
C	Specially Orthotropic	25	140-155	See Eq. 2.15	65

Minimum thickness of the compression coupons is determined using the following equation:

$$h \geq \frac{l_g}{0.9069\sqrt{\left(1-\frac{1.2F^{cu}}{G_{xz}}\right)\left(\frac{E^c}{F^{cu}}\right)}} \tag{2.15}$$

Where $E^c, F^{cu}, G_{xz}, h,$ and l_g are the longitudinal elastic modulus, the ultimate compressive stress, through-thickness shear modulus, specimen thickness, and length of gage section respectively. To determine some standard properties to apply for a composite material, an existing material is used. For this chapter, standard properties are taken from a composite material of the world-wide failure exercise [48]. This is the IM7/8552 Fiber/Matrix Epoxy combination with a fibre volume fraction of 60% and a fibre diameter of 4.5 μm.

2.3.1. Material properties

In this section, the material properties used for the coupon modelling are presented. The mechanical properties found in [48] are used again for every model. They can be found in Table 2.3 below. This Table also contains the non-physical parameter quantities for MAT054/55 and MAT 058. The non-physical parameters are used to overcome difficulties with simulations, or for tuning the behavior slightly. An important parameter for example in crush simulations is the SOFT parameter. This parameter determines how much surrounding elements are weakened once an element fails. This allows a stable crush front to form. The material cards in LS-Dyna contain more options/parameters to set, but these are unreported in this section because they are left on the default setting/quantity.

Table 2.3: Material properties used for coupons MAT054/055 and MAT058.

Property	**LS-Dyna abbreviation**	**Quantity**
Density	RO	1.59e-6 kg/mm^3
Young's modulus (Longitudinal direction)	EA	165000 MPa
Young's modulus (Transverse direction)	EB	9000 MPa
Young's modulus (Normal direction)	EC	0 MPa
Poisson's ratio (ν_{ba})	PRBA	0.0185
Shear modulus (G_{ba})	GAB	5600 MPa
Shear modulus (G_{bc})	GBC	2800 MPa
Shear modulus (G_{ca})	GCA	5600 MPa
Factor to determine the minimum stress limit after stress maximum for:	SLIM__	see following 5 rows
Fiber tension	SLIMT1	0.01
Matrix tension	SLIMT2	0.1
Fiber compression	SLIMC1	0.375
Matrix compression	SLIMC2	1
shear	SLIMS	1
Time step for automatic element deletion	TSIZE/TFAIL	1e-7 s
Maximum effective strain for element layer failure.	ERODS	-0.55 [-]
Softening reduction factor for strength in the crashfront	SOFT	0.57
Failure Surface Type	FS	-1
Strain at longitudinal compressive strength	E11C	0.01551 [-]
Strain at longitudinal tensile strength	E11T	0.011 [-]
Strain at transverse compressive strength	E22C	0.032 [-]
Strain at transverse tensile strength	E22T	0.0081 [-]
Strain at shear strength	GMS	0.05 [-]
Longitudinal compressive strength.	XC	1590 MPa
Longitudinal tensile strength.	XT	2560 MPa
Transverse compressive strength	YC	185 MPa
Transverse tensile strength	YT	73 MPa
Shear strength	SC	90 MPa

The material properties in Table 2.3 that are used for the Daimler Pinho model as well with a few annotations found in Table 2.4. These properties are found in [44].

Table 2.4: Additional properties set for MAT 261.

Property	**LS-Dyna Abbreviation**	**Quantity**
Poisson Ratio (ν_{ca})	PRCA	0.0185
Poisson Ratio (ν_{cb})	PRCB	0.0218
Young's modulus (Normal Direction)	EC	9000 MPa

2.3.2. Coupon Model Details

An analysis is done on tension coupons with a symmetric layup (row 3 of Table 2.2). The coupon pictured in Figure 2.3 will be pulled in tension with a speed of 0.15 mm/s. The three different material models presented in section 2.2 (MAT 054/55, MAT058, MAT261) are applied and thus 3 different

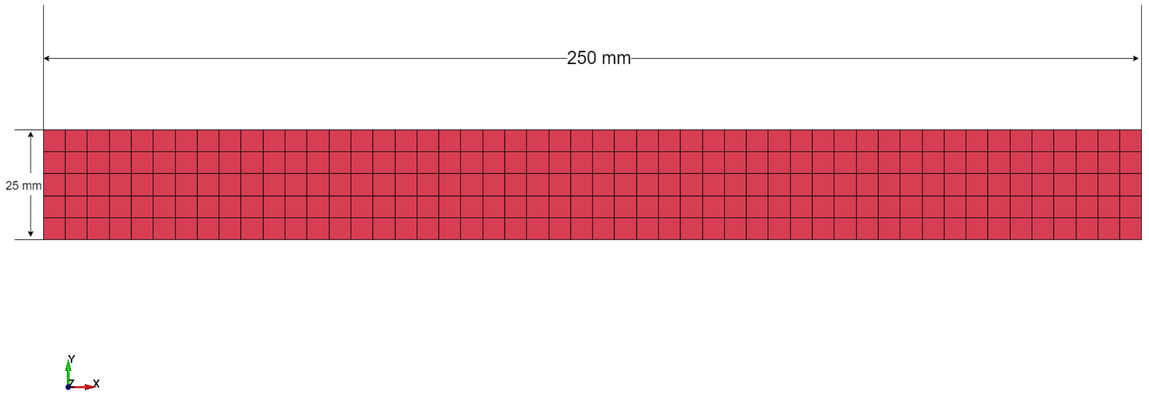


Figure 2.3: Dimensions of the coupon, the size of each shell element is 5. The coupon is fully clamped 55m from the left hand side of the coupon pictured. 55mm from the right hand side the coupons degrees of freedom are fully clamped expect the horizontal direction.

simulations are performed. The layup that is chosen is $[0_2/\pm 45/90_2]_s$. The coupon is given the dimensions in Table 2.2 for balanced/symmetric layups with a total coupon thickness of 2.5 mm. The tab length is set at 55 mm on each side. One tab consists of a fixed nodeset and the other side of the coupon, where the load is applied, is fixed in all directions/rotations except the loading direction. The tab length is not reported in the table, however since the tab lengths are decided upon based on slipping of the tensile machine grip, it is not an extremely relevant parameter for this analysis, for there is no slipping possible in this simulation. Each of the 12 layers has the same ply thickness of 0.2083 mm. in LS-Dyna a couple of keyword cards need to be set. They are explained below.

- The *CONTROL_SHELL card needs to be invoked, and the LAMSHT flag needs to be set to 1 to activate laminated shell theory.
- The *CONTROL_TERMINATION card is also needed to set an end time for the simulation, the only requirement is that the coupon needs to be pulled until failure. It is found that a termination time of 20 seconds is adequate for this.
- in *DATABASE_ASCII_option the BNDOUT flag is turned on. This will let LS-Dyna know to store the forces on the nodes so that the total resultant force can be extracted.
- The *SECTION_SHELL keyword is introduced, in this keyword the layup is defined along with the thickness of the shell equal to 2.5 mm. Furthermore, the element type ELFORM is set at type 2. These are Belytschko-Tsay elements. The number of integration points, NIP, is set to 12. For laminated shells, this tells LS-Dyna that there are 12 plies.

For the analysis it would be helpful to obtain the strain in the material. Considering the boundary conditions of the coupon, the strain is calculated as follows:

$$\epsilon_{xx} = \frac{u_x}{l_c}, \text{ \& } l_c = l_{\text{tot}} - 2l_{\text{tab}} \tag{2.16}$$

Where ϵ_{xx} is the strain in the loading direction, u_x the displacement in loading direction, l_c is the initial length of the coupon that carries load, l_{tab} is the length of one tab, and l_{tot} is the total length of the coupon.

2.3.3. Results of Coupon modelling

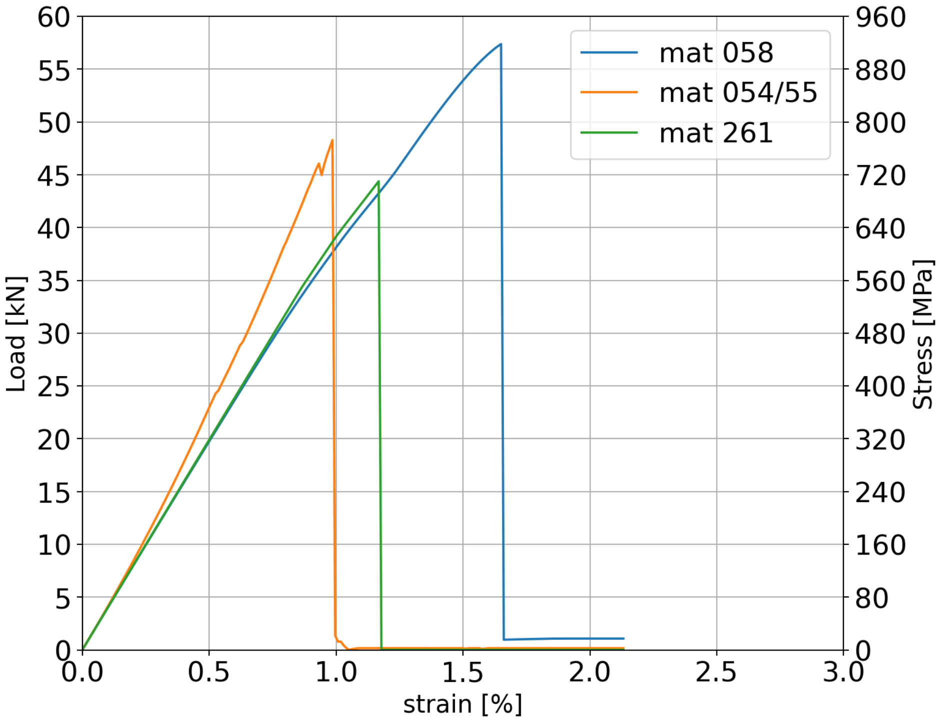


Figure 2.4: Different material models applied on the same coupon in LS-Dyna.

From the results in Figure 2.4, it can be seen that MAT054 and MAT261 show very similar behavior. The ultimate load and the failure strain is roughly similar. Interestingly however, the stiffness of MAT054 is slightly higher, while the stiffness of the other two models are almost equal. Given the similar approach MAT054 and MAT058 have, one would expect that the elastic behavior would be the same. The fact that MAT058 shows a higher ultimate load is not surprising. The other models take into account the shear stress in tensile failure, while the coupon using MAT058 does not (equation 2.6).

A primary interest for this thesis is the computing time of these material model simulations. A numerically efficient material model offers a great advantage for building training datasets containing many simulations. To compare the numerical efficiency, all coupon simulations are performed on a single CPU core. The coupon simulation using MAT 261 took 15 hours 29 minutes and 8 seconds. MAT 54 took 5 hours 37 minutes 9 seconds. MAT058 took 7 hours 56 minutes and 30 seconds. When comparing these, it becomes clear that the use of MAT261 is unbeneficial for this thesis, and it will therefore no longer be considered for any subsequent models. The fact that MAT261 is expected, as it became clear when the material model was explained that the algorithm checks the failure on each time step for multiple fracture angles, and it also introduces more calculations for fibre-kinking failure.

2.4. Composite Tube modelling

The last analysis in this chapter is the composite tube model, which is impacted with a rigid plate. The model is presented in Figure 2.5. The cross section of the tube is seen in Figure 2.6. It has a size of 50 by 50 mm and a length of 200mm. The radius of the corners is 6.4 mm. The thickness of the tube is equal to 2.16 mm. These dimensions are taken from the Oak Ridge National Laboratory (ORNL)[49]. Within the ORNL database, they are identified as 46B and 47B specimens, and these tubes were tested by ORNL, this allows a comparison between the LS-Dyna simulation and experimental data [49] which can be found at the end of this section. The tubes in the experiment had a bevel trigger similar to the one depicted in Figure 2.7. This trigger is slightly simplified in the LS-Dyna model, where it has no gradual change in thickness in Z-direction but rather a constant one. The bevel trigger allows for a stable and beneficiary crush front, optimizing the energy that can be absorbed by the tube.

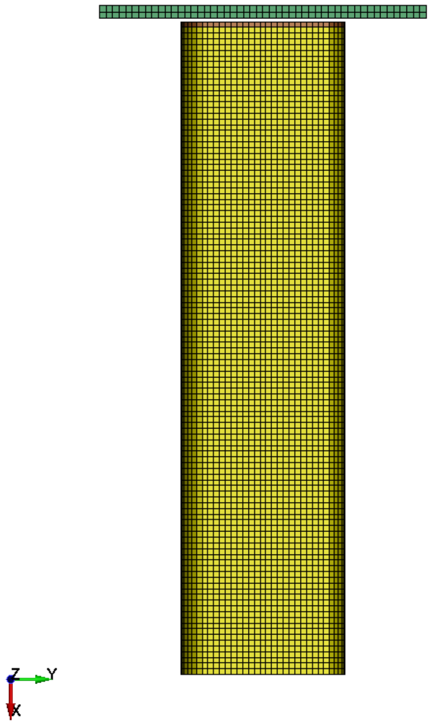


Figure 2.5: Top view of the composite tube and the crushing plate.

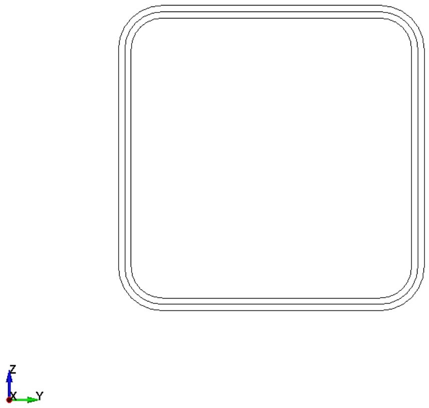


Figure 2.6: Crossection view of the tube. The center line is the shape of the shell. The outer edges form the real dimensions of the tube. Since the shell has edges of length 50 mm, the real outer dimensions of the tube are 52.16 mm.

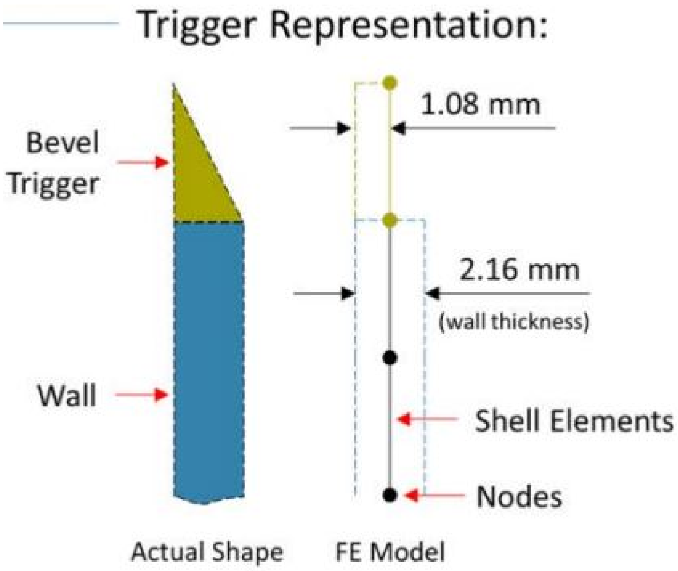


Figure 2.7: Depiction of the bevel trigger in the actual test specimen (left) and the simplified form that is used for the LS-Dyna model (right).Image found in [50].

2.4.1. Model details

The model is meshed with a constant element size of 1.75 mm in the tube and the crushing plate consists of elements of 2 mm and is a square of 100x100 mm with a thickness of 4 mm. The tube is modelled with Belytschko-Tsay elements shell elements, which is achieved by setting the ELFORM=2 *SECTION_SHELL keyword card. The layup of the tube is $[0_2/\pm45/90_2]_s$ with a ply thickness of 0.18 mm. The layup of the bevel is $[0_2/\pm45/90_2]$, with the same ply thickness so that the total thickness is 1.08 mm or half the tube thickness as seen in Figure 2.7. The plate is modelled with 8-point hexahedron solid elements. There are a total of 12312 shell elements and 5000 solid elements in the model.

Three boundary conditions are applied. The bottom of the tube in Figure 2.5 is fixed in all degrees of freedom. Secondly, The crushing plate is fixed in all degrees of freedom except the X-direction displacement. Lastly, the plate has a constant velocity of 5.5 m/s.

The contact behavior is introduced through the *CONTACT_AUTOMATIC_SINGLE_SURFACE card. Here the settings are mostly kept to default except for the Static coefficient of friction (FS) and Dynamic coefficient of friction (FD), both set at 0.2. Furthermore, the viscous damping coefficient in percent of critical (VDC) is set at 20 to achieve a stable solution.

A number of *CONTROL keyword cards are added to the model. An important CONTROL setting is in the *CONTROL_SHELL card, where the LAMSHT flag needs to be set equal to 1 to ensure laminated shell theory is applied in the model. Also, the *CONTROL_HOURGLASS keyword is used with IHQ=4 which is a stiffness-based hourglass control method. The *CONTROL_ACCURACY keyword is used to set the Invariant Node Numbering (INN) on by setting INN=4. This is recommended by the LS-Dyna manual [51] because it prevents over-rotation of the material coordination by hourglass modes and it stabilizes the calculations. Lastly in the *CONTROL_TERMINATION keyword card, the termination time is set at 0.015 s.

2.4.2. Tube Model Results

The results of experiments 46B and 47B as well as the results from the LS-Dyna model are presented in Figure 2.8. To achieve the Force-Displacement data from the LS-Dyna model, an SAE type 1000 filter is used. This filter is a double low pass Butterworth filter with zero phase shift. This is a very common filter to use for filtering reaction forces in post-processing [52]. It is used more often in this research because, without the use of the filter, the force graph is not readable due to high-frequency oscillations and also not in the slightest comparable to physical test data.

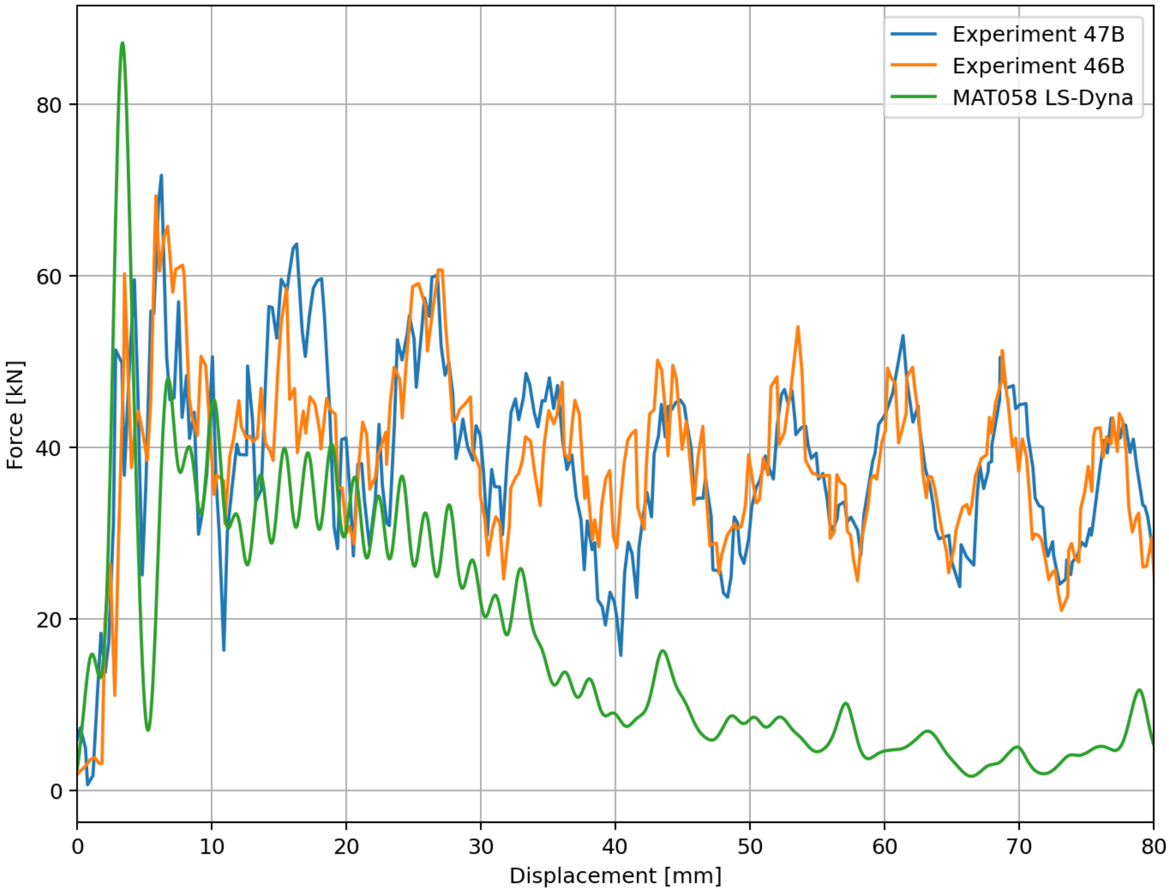


Figure 2.8: Crushing force of the tubes, 47B and 46B are based on tests, and MAT058 is the LS-Dyna model.

Comparing the graphs, initially, the experimental data and the model seem to show a similar resistance. The peak force in the model is slightly higher but on average the work done by the crushing plate is comparable in quantity to the experimental data. However, after a displacement of 10 mm the modelled tube's stiffness seems to collapse faster. A possible solution for this would be changing the SOFT parameter. As stated before this parameter weakens the elements surrounding a failed element.

Tweaking this parameter may allow the model of the tube to fail more gradually. There is no intention to carry this analysis any further, as it served merely as an exercise for learning the features of LS-Dyna in pre- and post-processing. nor does it hold significance to the rest of the thesis, this analysis is abandoned from here.

3 Coupon failure artificial neural network surrogate model

This Chapter contains the work on the first Artificial Neural Networks (ANNs) built for the thesis. The purpose of this analysis was to serve as a stepping stone for the work in chapter 6. The goal is to build a Neural network that can predict the ultimate load and the displacement of the loaded tab at the ultimate load of a composite specimen given the thickness and a layup. The chapter is split into three sections. These sections contain the LS-Dyna modeling, the Artificial Neural Network (ANN) building procedure, and the result respectively.

3.1. LS-Dyna Modelling and Simulation

Using the MAT058 material card in LS-Dyna, nine different coupon variations were built. Every coupon was of size 250x25 mm, and the element size was set at 5 mm (see Figure 3.1). The tab length was set at 55mm, which is the largest tab length for tension coupons in Table 2.2. Three layups were created, and three distinct thicknesses were determined by creating a linear space between 0 and 2.5 mm. To achieve this, the ply thickness was increased so that the layup would be constant. The assumption made here is that the properties of the material remain the same for different ply thicknesses. The coupons are loaded in tension using a tab length of 55mm on each side. This means that the left side in Figure 3.1 is fully fixed in all degrees of freedom. The right side is fixed in all degrees of freedom except the Horizontal direction. The material properties in table 2.3 and other various settings were derived or directly taken from the work of in Chapter 2.3. For example, the displacement loading is set at 0.15 mm/s. For more details on the model, see section 2.3.2. Table 1 offers a complete overview of the training inputs of the ANN (first two columns) and their corresponding output (training/test data). The data is extracted through a python script that reads a .txt file containing the XY data that can be seen in Figure 3.2. For every layup tested, it can be seen that the ultimate load accumulates at the same displacement. Since this is a displacement controlled experiment this is expected, because the tension increases as the thickness increases. The ultimate load seems to increase semi-linearly as the thickness of the coupons is increased. The 0.8 mm coupon with a layup of $[0/\pm 45/90]_s$ shows a different failure behaviour to the rest of the coupons. Furthermore, for this layup, once a ply fails the load drops and builds up again, Whereas in the other two layups all plies seem to fail simultaneously causing the force to drop to zero after the ultimate load has been reached.

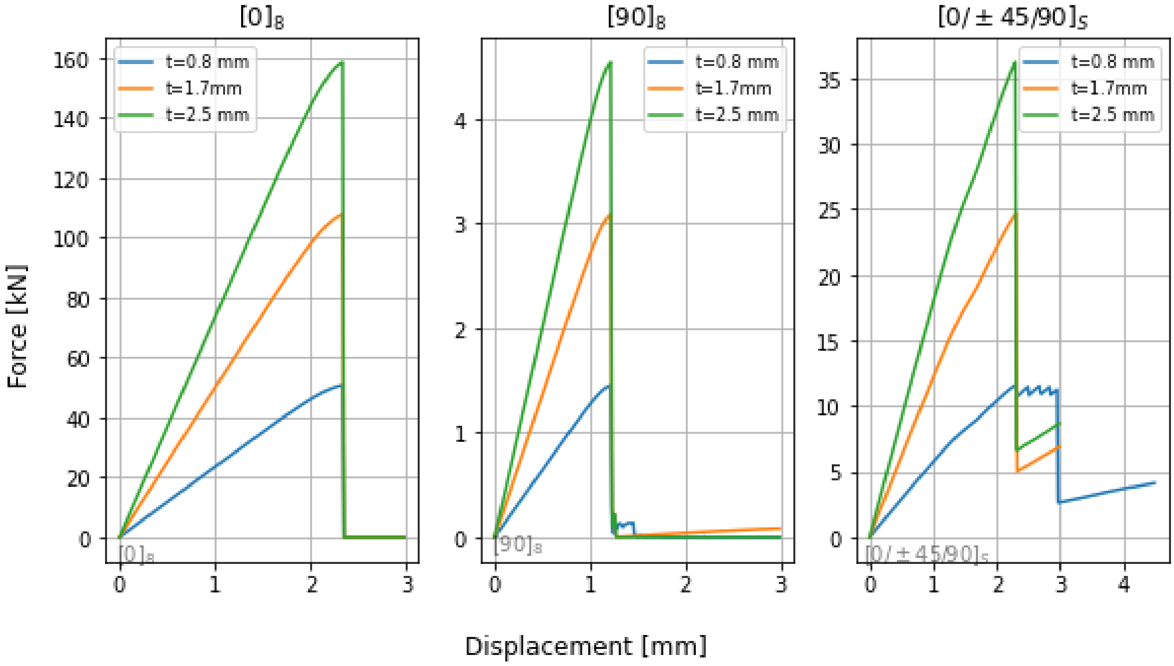


Figure 3.2: Results of the coupon simulations in LS-Dyna. Within each graph the layup of the coupon is the same, the only variable is the thickness of the coupon.

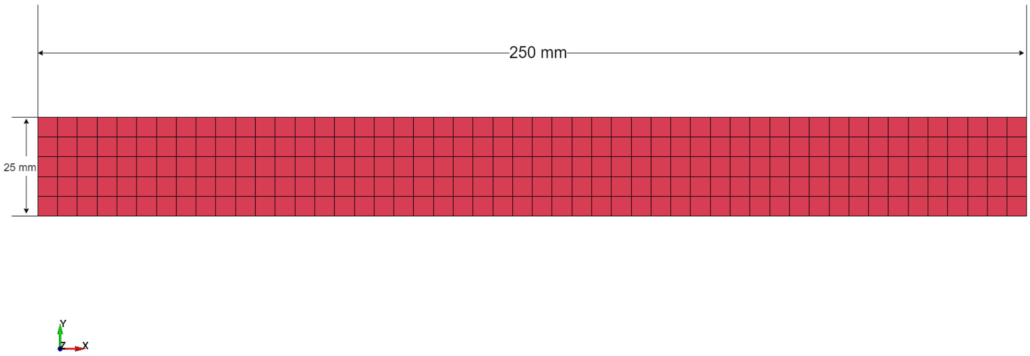


Figure 3.1: Dimensions for the coupon modelling. The pictured dimensions will be fixed throughout this chapter. Furthermore the coupon consists of shell elements of size 5 mm.

Table 3.1: Coupon Dataset build for the artificial neural networks. The results are read from LS-Dyna output files.

Layup	Sample Number	Thickness	Displacement at ultimate Load [mm], Ultimate Load [kN]
$[0]_8$	1 2 3	0.8 mm, 1.7 mm, 2.5 mm	[2.34, 50.8], [2.34, 108], [2.34, 159]
$[90]_8$	4 5 6	0.8 mm, 1.7 mm, 2.5 mm	[1.21, 1.45], [1.21, 3.09], [1.21, 4.54]
$[0/\pm45/90]_s$	7 8 9	0.8 mm, 1.7 mm, 2.5 mm	[2.3, 11.6], [2.3, 24.7], [2.3, 36.2]

3.2. Artificial Neural Network Setup and Training Strategy

As discussed in the literature study, a feedforward neural network (FNN) will be used for this research. From the literature study follows the elements that make up a network and training strategy. This section will go through the elements one by one.

First of all, the architecture (size) of the ANN needs to be determined. The architecture consists of the amount of layers and neurons within the layers. This rather small analysis contains 9 data samples. Increasing the number of neurons or layers increases the capacity of the ANN. However, it is crucial that the network is not overfitted and thereby learning the data training set by storing the training labels without forming a function to predict any unknown data samples. To prevent the ANN from overfitting, the network is limited to one hidden layer. For this analysis, two neuron activation functions were considered. These were the Sigmoid function and The Rectified Linear Unit(ReLu) neuron. The Sigmoid function is:

$$f(z) = \frac{1}{1+e^{-z}} \tag{3.1}$$

The ReLu function is:

$$f(z) = max(0, z) \tag{3.2}$$

Where z is equal to the input of the neuron, which can be calculated using equation 1.4. Both are non-linear functions, with well-defined derivatives, therefore they are expected to work well with gradient descent learning using backpropagation. According to [24], the Sigmoid function was most popular before the introduction of the ReLu neuron. Since this analysis is of small scale and not highly nonlinear it is expected that the chosen activation function is not influencing the results significantly (<1% error). This is because the results of the simulation are quite straightforward and could even be predicted with linear regression. It should therefore not be difficult for these neurons to find a low minimum of the cost function. ReLu neurons can suffer from the dying neuron problem, if their output converges to zero during training. This can happen if the input of the neuron is far below zero, and the weight changes are too small to make the neuron's output switch to a positive value. Therefore the Sigmoid neuron will also be employed for this analysis to see what activation function works with the least amount of neurons and thus the smallest chance of overfitting.

For the training algorithm, Stochastic Gradient Descent with backpropagation will be used in Tensorflow/Keras. This algorithm, named the optimizer in Tensorflow/Keras, has a learning rate of 0.001. The learning rate influences the weight changes of the network. This parameter is kept at this low value to ensure that it consistently converges to a minimum of the cost function. For the loss or cost function, the mean squared error (MSE) is selected. The MSE function was a commonly used cost function in the literature review, and therefore it is deemed a good starting point.

To train but also test the neural network performance, 7 out of 9 sample points in table 3.1 were fed to the neural network as training set, and two sample points were withheld from the network to see if it predicts the correct value. It is common practice to use a randomized selection of sample points from the dataset. However, in this analysis specifically, the ANN task will differ between interpolation and extrapolation depending on whether the test data samples consist of coupon thicknesses in between the training data samples. Due to this, the training/test data split is chosen manually. To investigate the extrapolation performance and the interpolation performance, two ANNs were trained with a corresponding training/test data split that forces the ANN to perform either task. Lastly, to teach the ANN that the ultimate load of a coupon is zero as the thickness decreases to zero 3 data samples are added for each layup. Each sample contains the layup and thickness zero as input and train labels [0,0]. This prevents the network from predicting negative ultimate loads when the thickness is close to zero.

3.3. Results/Conclusions

The best training results are achieved by setting up one Dense layer of 10 ReLu activation function neurons. More neurons work as well, but this only increases the capacity of the ANN unnecessarily. As stated before, the Sigmoid function is also tested. When comparing the test results there was no real distinction between the use of the Sigmoid neuron, so no results using this neuron are reported in this chapter. A minimum of the cost function is found by performing around 2200-2800 epochs, which is one training cycle in which all the training data is passed through the network once. The loss which is the mean squared error is usually below 0.001 on the training data. The data contains two output parameters, the displacement at the ultimate load and the ultimate load. The mean squared error is calculated for each output parameter, then the mean of these two errors is taken to find the final mean squared error.

Interpolation Artificial Neural Network

The interpolating ANN is trained without data samples 5 and 8 in table 3.1. The training process is visualized in Figure 3.3 below. In this particular training session, the network finds two breakthroughs to reduce the MSE on the training data.

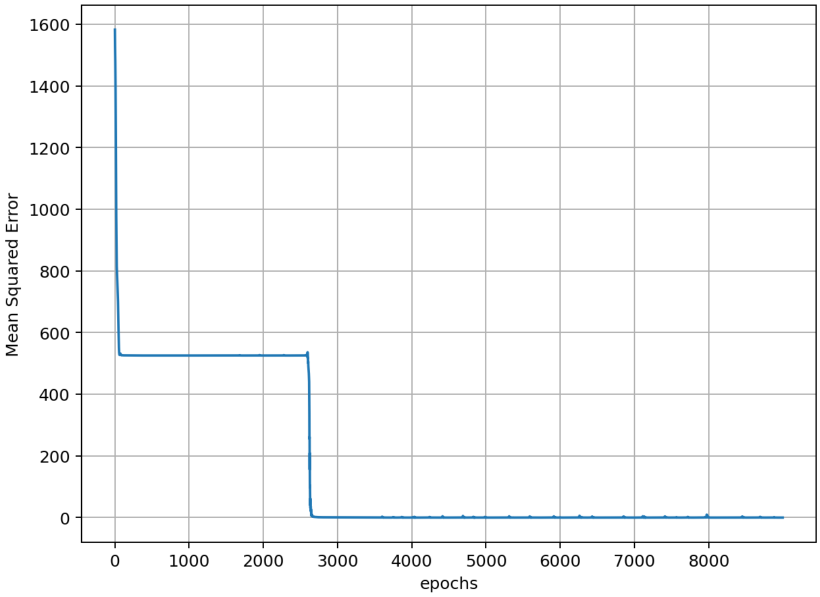


Figure 3.3: Training result of the ANN fed with data from table 3.1 excluding sample numbers 2 and 5.

The results of testing the two remaining data samples that are withheld from the ANN during training are found in Table 3.2. From the table, it can be seen that the network is spot on concerning the prediction of the ultimate load. However, There is a difference in the predicted displacement for the second prediction. Strangely, it did not manage to learn that the failure displacement is the same for the same layup.

Table 3.2: Predictions on the test data of the ANN.

Test Input	LS Dyna Result	ANN prediction
Layup: $[0]_8$, thickness= 1.7 mm	2.3 mm 108 kN	2.3 mm, 108 kN
Layup $[90]_8$, thickness=1.7 mm	1.21 mm 3.08 kN	1.29 mm, 3.09 kN

Extrapolation Artificial Neural Network

In this section 2 data samples with the maximum thickness of 2.5 mm will be witheld from the network. Specifically samples 6 and 9 in Table 3.1. The network is trained and the resulting predictions are seen in table 3.3. When comparing these values it is clear that the network is not able to extrapolate even though the MSE on the training data is below 0.001.

Table 3.3: Training result of the extrapolation ANN fed with data from table 3.1

Test Input	LS Dyna Result	ANN prediction
Layup: $[90]_8$, thickness= 2.5 mm	1.21 mm 4.54 kN	1.37 mm, 52.1 kN
Layup $[0/\pm 45/90]_S$, thickness=2.5 mm	2.3 mm 36.2 kN	3.6 mm, 19.9 kN

3.4. Chapter Conclusions

The most important conclusion drawn from the chapter is that the regression FNN does perform well, but it is important that when splitting a small dataset, the test data should lie within the design space rather than on the boundaries. This can be fixed by creating a different type of sampling method, and/or also a larger dataset. Furthermore, the required number of epochs during training was quite high for a small dataset. A common suggestion also found in [31] that could proof valuable is to try to normalize the input and output of the dataset. These two concepts will be applied further on in the thesis.

4

Stunning Fuselage Model

In this chapter, the STUNNING fuselage model is described and a couple of models that are reduced in size are introduced. The purpose of the chapter is to find out whether these reduced models can be used to deduce crashworthiness behavior of the full model, such that a neural network can be trained to predict crashworthiness behavior of the full model based on a model that is less computationally expensive.

4.1. Breakdown of the Stunning Fuselage subfloor section model

The STUNNING fuselage subfloor model, sometimes referred to as the original model in this chapter, is presented in Figures 4.1, and 4.2. The model represents one fuselage section of a regional commercial passenger aircraft cut off just above the cross-cargo beam, that carries the cargo floor of the aircraft. The cargo floor is not included in the model. In the bottom of Figure 4.2, the first part of the model is the crushing plate, which is a rigid plate with a thickness of 25 mm. The velocity of the crushing plate is in positive Z direction with a constant magnitude of 10 m/s, the simulation ends after 30 ms. The fuselage consists of parts modelled with shell elements. This simplifies/removes the details of the fuselage geometry significantly compared to the real design. Therefore, this model is not suited for detailed analysis of joints and other details of the fuselage, but rather the global behaviour of the structure on impact. All parts in Figure 4.1 are composite parts with material properties corresponding to Toray Cetex CT1225 Standard [53]. This material is a thermoplastic composite material, made from stacked unidirectional plies of carbon fibre cured in Polyaryletherketone (PAEK) resin. For the connection of the cross-cargo beam to the frame, metal plates are used to join these two components together.

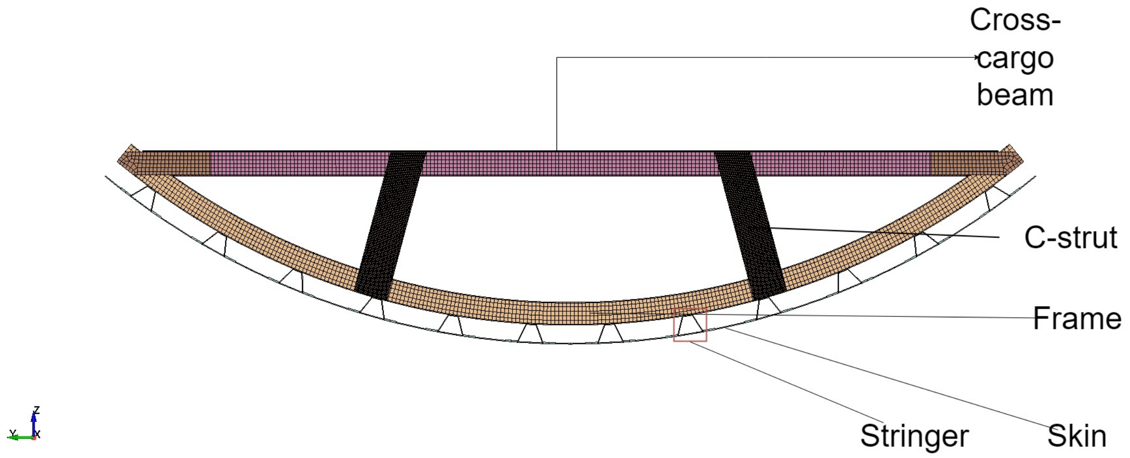


Figure 4.1: Front view of the STUNNING Fuselage subfloor section, indicated in the figure are the main elements that make up the assembly.

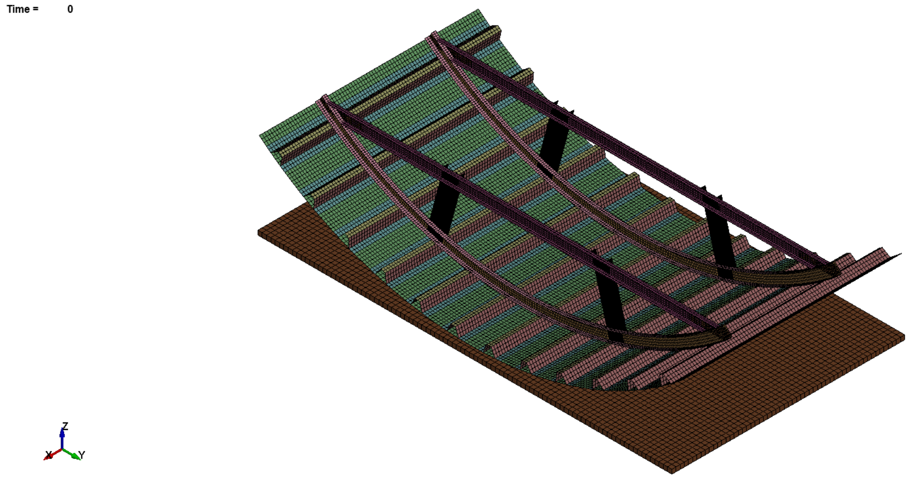


Figure 4.2: Isometric view of the STUNNING Fuselage subfloor section

4.1.1. Material (Damage) properties

The material model chosen for the composite parts of the structure is MAT058 in LS-Dyna. The specific properties used for all simulations in this Chapter are summarized in Table 4.1. The material model for the metal plates that connect the cross-cargo beam to the frame is MAT037 with the values of the material properties based on an aluminium AA6111-T4 alloy.

Table 4.1: Material properties used for the STUNNING fuselage section model

Property	**LS-Dyna abbreviation**	**Quantity**
Density	RO	1.59e-6 kg/mm^3
Young's modulus (Longitudinal direction)	EA	143 GPa
Young's modulus (Transverse direcction)	EB	8.8 GPa
Young's modulus (Normal direction)	EC	0 GPa
Poisson's ratio ()	PRBA	0.0185
Shear modulus ()	GAB	4.3 GPa
Shear modulus ()	GBC	3.4 GPa
Shear modulus ()	GCA	4.3 GPa
Factor to determine the minimum stress limit after stress maximum for:	SLIM__	see following 5 rows
fiber tension	SLIMT1	0.01
matrix tension	SLIMT2	0.1
fiber compression	SLIMC1	0.6
matrix compression	SLIMC2	1
shear	SLIMS	1
Time step for automatic element deletion	TSIZE	1e-7 ms
Maximum effective strain for element layer failure.	ERODS	-0.55 [-]
Softening reduction factor for strength in the crashfront	SOFT	1 for all parts expect C-strut 0.22 for C-strut
Strain at longitudinal compressive strength	E11C	0.008 [-]
Strain at longitudinal tensile strength	E11T	0.019 [-]
Strain at transverse compressive strength	E22C	0.028 [-]
Strain at transverse tensile strength	E22T	0.009 [-]
Strain at shear strength	GMS	0.05 [-]
Longitudinal compressive strength.	XC	1.089 GPa
Longitudinal tensile strength.	XT	2.755 GPa
Transverse compressive strength	YC	0.248 GPa
Transverse tensile strength	YT	0.078 GPa
Shear strength	SC	0.0483 GPa

4.1.2. Boundary conditions

The fuselage subfloor section is constrained by two types of boundary conditions. The first is a symmetry boundary condition visualized in Figure 4.3, these nodes are fixed in X direction and their rotation is constrained around every axis. This is indicated as a 100111 condition in LS-Dyna. Furthermore, along the parts where a full fuselage would usually continue, the nodes are fully fixed. This is indicated in Figure 4.4, and is indicated as a 111111 condition in LS-Dyna. Lastly, the rigid crushing plate is constrained in every rotation and the translation in X and Y direction.

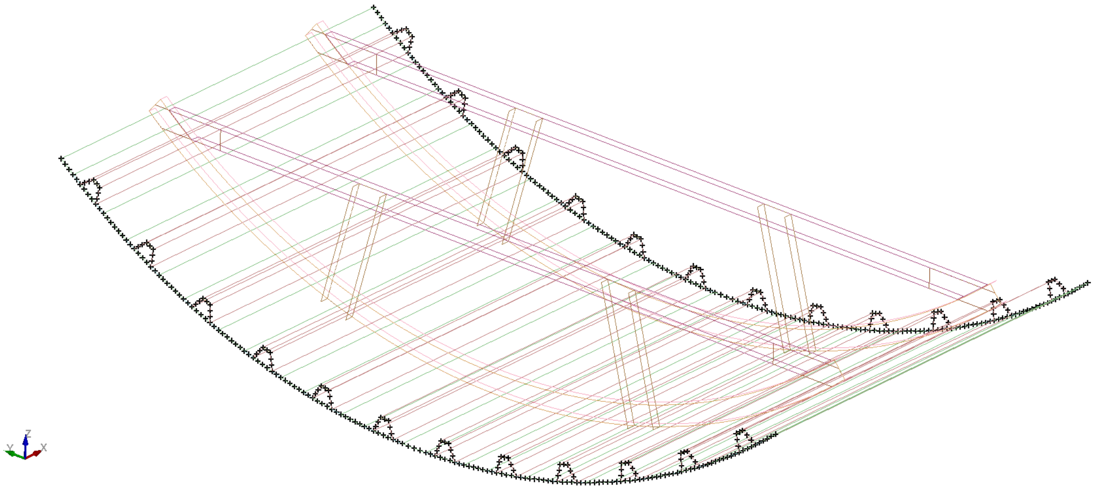


Figure 4.3: Symmetric boundary conditions on the outer nodes(in X direction) of the fuselage indicated in black.

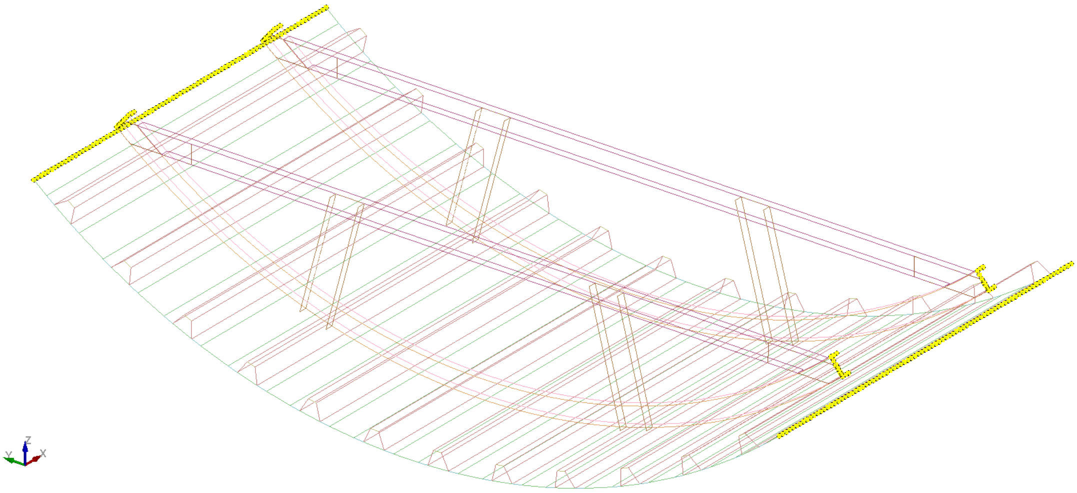


Figure 4.4: Fixed boundary condition of the fuselage indicated in yellow.

4.1.3. Mesh properties

The mesh of the cross-cargo beam consists of square elements of size 10 mm by 10 mm in the web, and the flanges of quad elements of size 12.6 mm by 10 mm. The frame web consists of elements of size 12 mm by 10 mm and similarly to the cross-cargo beam the flanges consist of elements of size 12.6 mm by 10 mm. The skin consists of elements of size 16 by 15 mm. The stringers consist of elements of size 15 mm by 16-18 mm. The C-struts, which are of greater interest, as their performance has a large influence on the crashworthiness kinematics have a much finer mesh than all previously described parts. The mesh size of these elements are of size 2.5 mm. The default ELFORM 2 in LS-Dyna is used for the shell elements (Belytschko-Tsay elements). These elements are used for every part except the crushing plate, which is a rigid solid.

4.1.4. Contact

Contact definitions can have a large influence on the results of the impact simulation. In LS-Dyna the contact is defined through a number of keyword cards. The first is an automatic general contact definition. This influences the contact of parts that are not in contact at the start of the impact, which is important as the impact of the plate will cause heavy distortion of the original geometry. Secondly, there are contact definitions for the parts that are joined together. These are defined through the TIED_SHELL_EDGE_TO_SURFACE card.

4.2. Results of the STUNNING Fuselage Model

The results of the simulation are visualized in Figure 4.5. The energy absorbed over the simulation time is presented in Figure 4.6. In the first 10 ms, the crushing plate flattens the skin and the frame, and the stress in the C-struts starts to rise causing them to twist and bend. After 20 ms the C-strut joint with the frame has failed. The stringers are being pushed into the C-struts which leads to the C-struts still carrying some load. after 20 ms the remaining structural integrity of all parts is lost but the structure in this compact form absorbs energy at a similar rate.

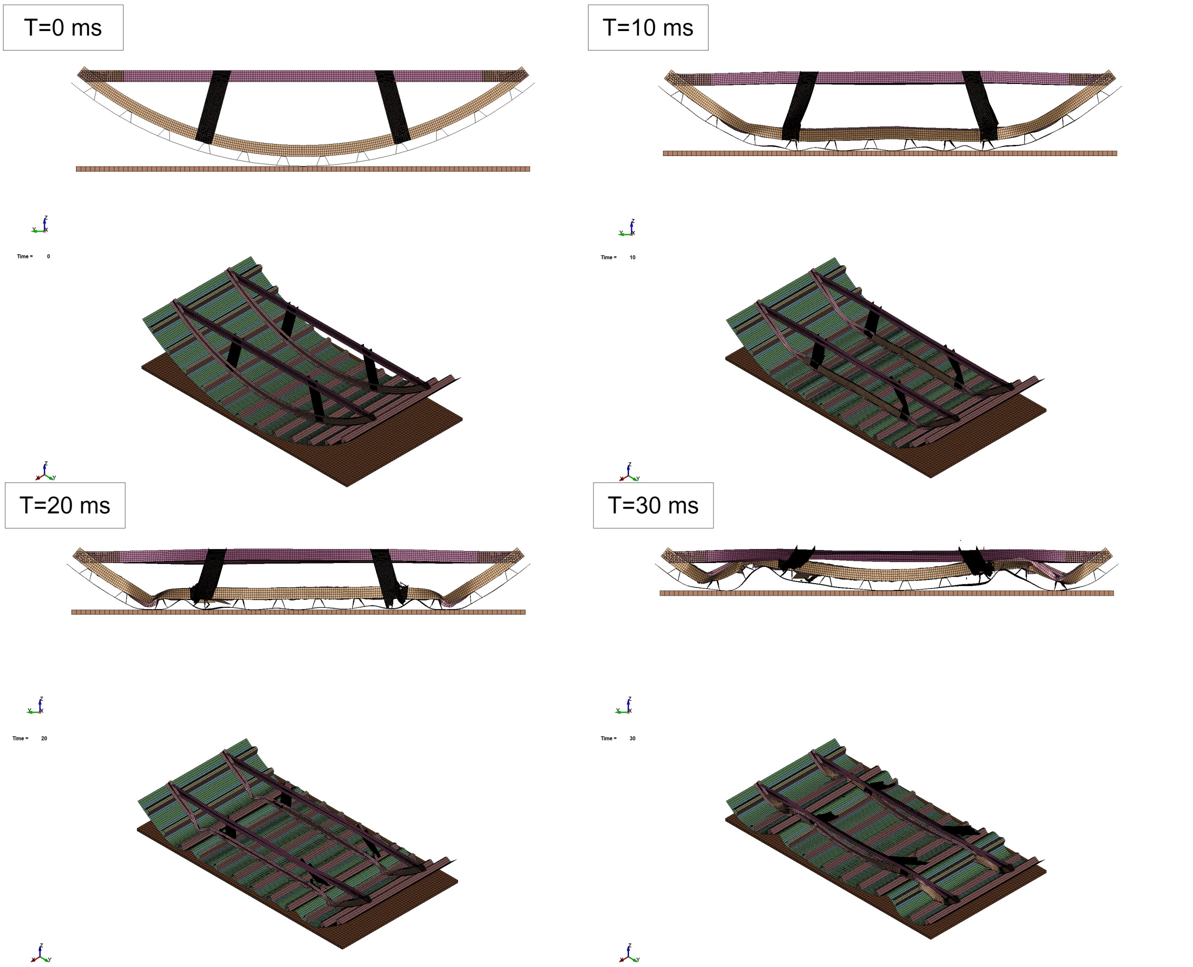


Figure 4.5: Crushing of the STUNNING fuselage section presented in a front view and isometric view in timeframes of 10 ms.

The crushing force is presented in Figure 4.7. This graph is obtained by using a low-pass digital filter (SAE filter) of 1000 Hz on the contact force data between the plate and the fuselage section. The mean crushing load is around 52 kN and the peak crushing load around 252 kN.

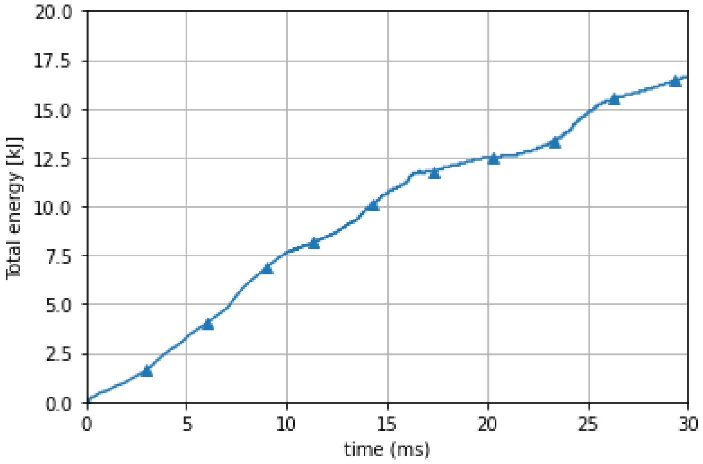


Figure 4.6: Total energy of the simulation over time for the original model.

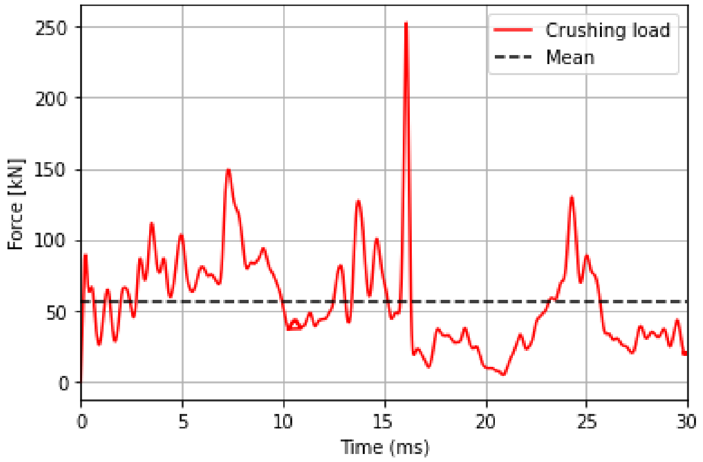


Figure 4.7: Simulation results of the load over time of the fuselage subfloor section.

The internal energy per part is seen in Figure 4.8. Note that the internal energy is different from the total energy, as it does not include hourglass energy, kinetic energy, and contact energy. The bottom of the omega stringers that are connected to the skin is modelled as one part, therefore these components can not be split. The internal energy of the stringer is the flanges and top of the stringer in the graph. From the graph, it becomes clear that the C-struts only function for the first 10 ms of the impact. Afterwards, the internal energy stays somewhat constant. The stringers accumulate about the same amount of energy, but their absorption is more gradual compared to the C-struts. for the skin, the absorption seems to rise in two separate instances, from 0 to about 10 ms and then from 15 to 25 ms. This is in accordance with what is seen in the failure mode in Figure 4.5. In the first 10 ms of the simulation, the skin between the C-struts flattens. Once this has happened the C-struts starts to be loaded until failure. As soon as the C-strut fails, just after 10 ms as the internal energy loses its slope in figure 4.8. After the C-struts failure, the skin starts to be deformed more heavily again, causing the slope of the internal energy of the skin to rise again and also the peak force seen around the same time in Figure 4.7. Lastly, the frame's internal energy seems to rise with a constant slope until around 12 ms. Afterward, the slope decreases slightly and then around 25 ms a bigger increase in slope is seen again. The large increase in slope near 25 ms is suspected to be the effect that the skin starts to compress and thus push inward into the frame as seen in the lower right corner in Figure 4.5.

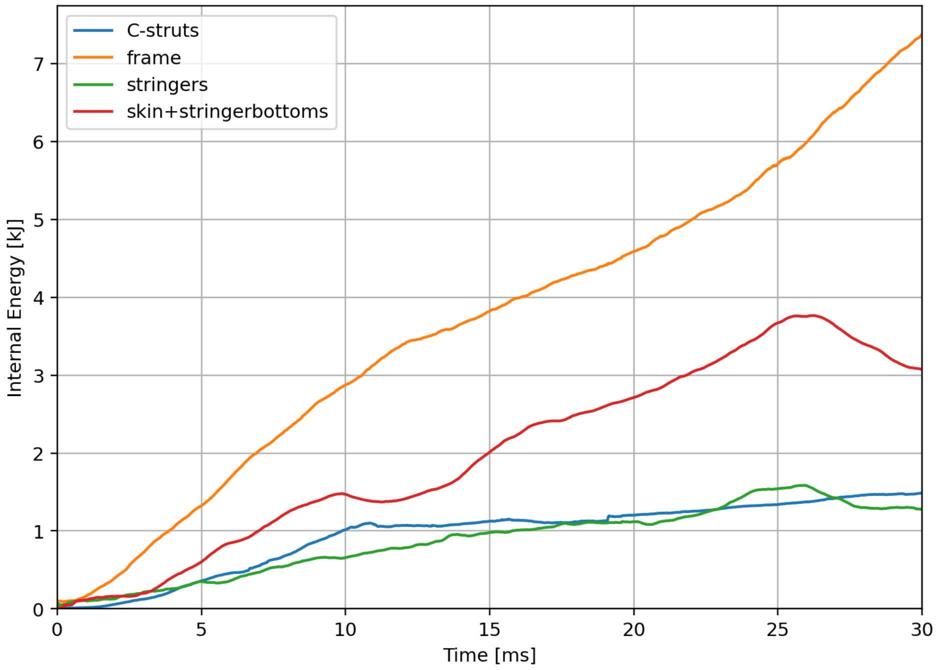


Figure 4.8: Internal energy for different parts during the impact simulation of the original STUNNING fuselage.

4.3. Model Sectioning

In this section, some models are presented that attempt to simulate similar crashworthiness behaviour as the STUNNING fuselage section. The goal here is to check if these reduced models would be candidates to train a neural network with. Since neural network training requires a large dataset containing many simulations, a reduction in model size would heavily speed up the process and also reduce computing cost.

4.3.1. Half Model

Two different models will be investigated, the first being a model which is cut in X direction to obtain half the size of the original fuselage section. This model and its crush simulation are presented in Figure 4.9. The model is identical to the original model, so the mesh, material properties, and contact definitions are all left unchanged. A slight annotation is performed for the boundary condition seen in Figure 4.3. The original boundary condition at the side that is removed when halving the model is now applied to the new edge of the skin so that the boundary conditions on the outer edges of the skin are similar.

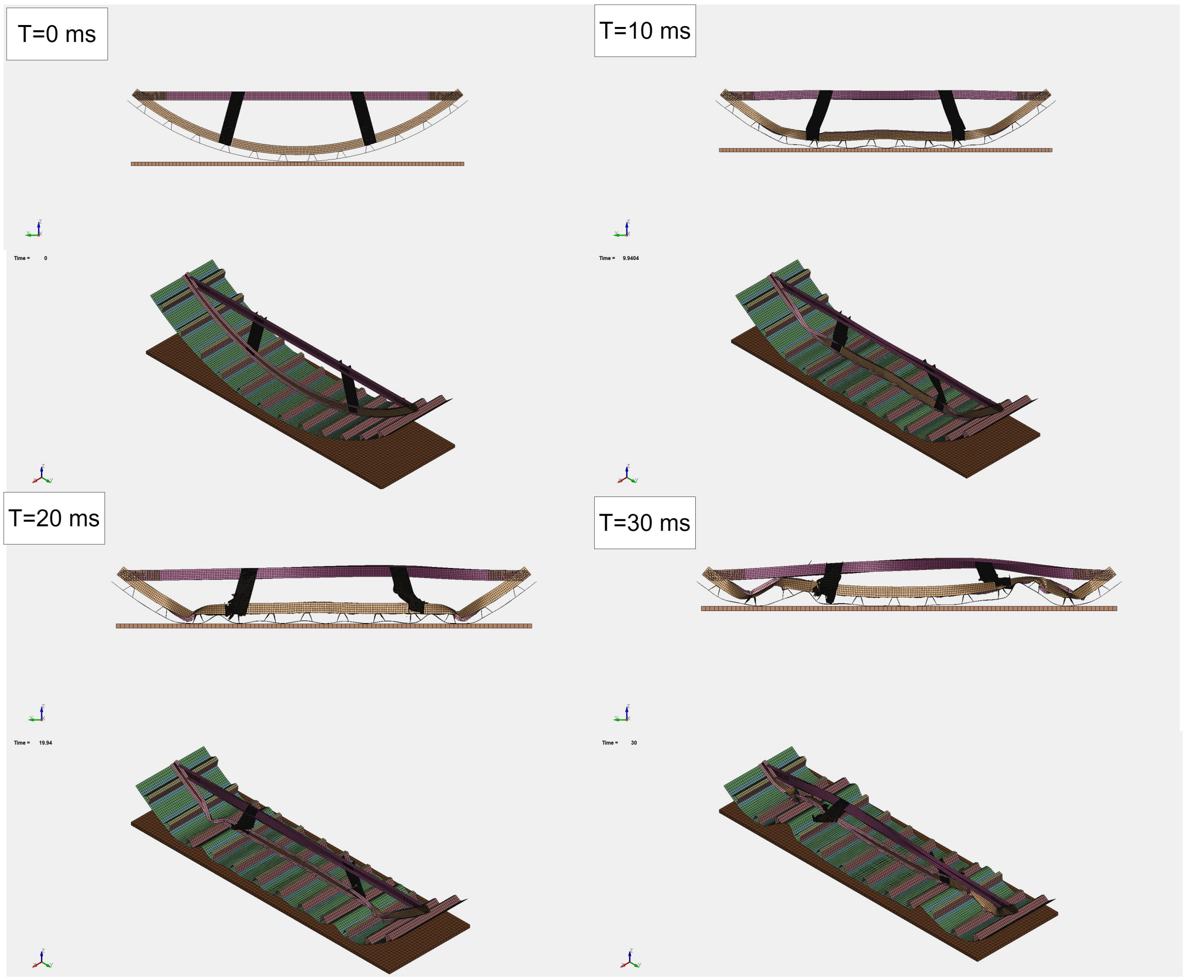


Figure 4.9: Crushing of the half fuselage section model.

4.3.2. Frame-C-strut-beam model

The second model that was derived was the Frame-C-strut-beam model presented in Figure 4.10. This model was based on the idea that the skin and stringers do not contribute largely to the crashworthiness performance, and can also not be easily adapted for crashworthiness in particular, because these parts carry other functions in normal operation of the aircraft. So therefore these parts are removed from the model, and the remaining parts need to be constrained so that the crush behavior is consequent with the original model. The first run that was performed was without any added boundary conditions, only the fixed boundary conditions on the outer edges of the frame are present. The result is seen in Figure 4.10. From the figure, it is clear that the center of the frame is somewhat crushed in the first 10 ms. However, afterward the structure starts to rotate, causing the crushing to stop.

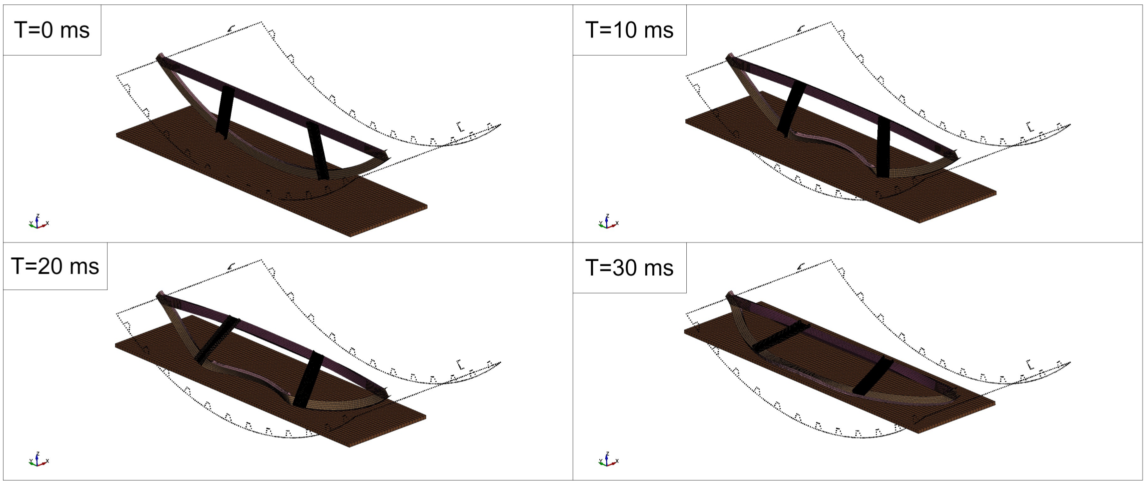


Figure 4.10: Frame-C-strut-beam model without any constraints apart from a fixed constraint at the outer edges of the frame.

To overcome this problem the frame was constrained, according to the placement of the stringers of the fuselage section. The nodes in Figure 4.11 were constrained in X direction. The resulting simulation is presented in Figure 4.12.

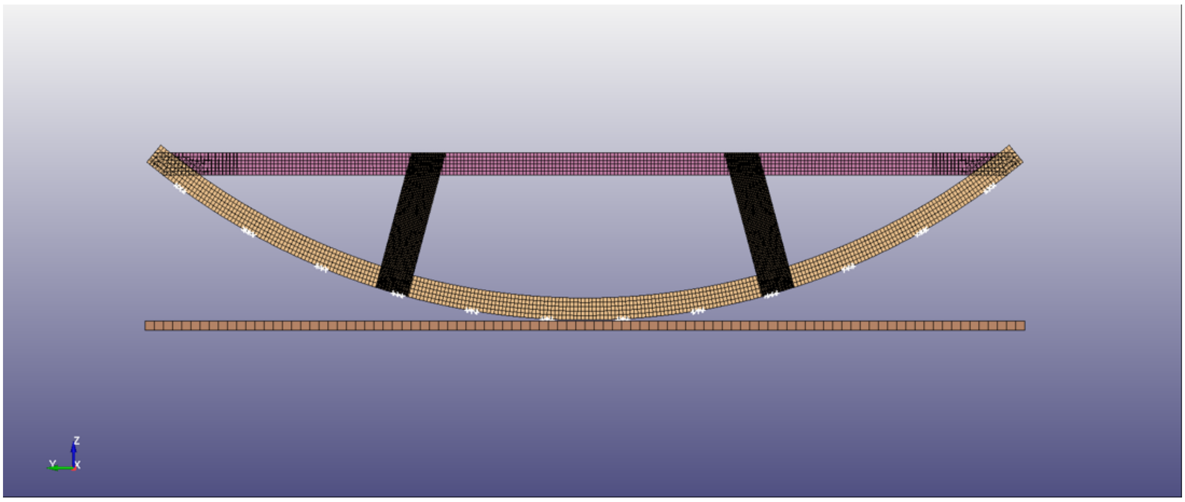


Figure 4.11: Nodes constrained in X direction highlighted in white along the bottom of the frame.

From Figure 4.12, it is very clear that the deformation is not at all similar to the crushing behavior seen in the original model. The frame folds inwards without the flattening behavior that can be seen in the original model (top right of Figure 4.5). Also, the cross-cargo beam is experiencing a completely different failure mode in this model. Therefore it is decided that the use of this model is not pursued in this thesis.

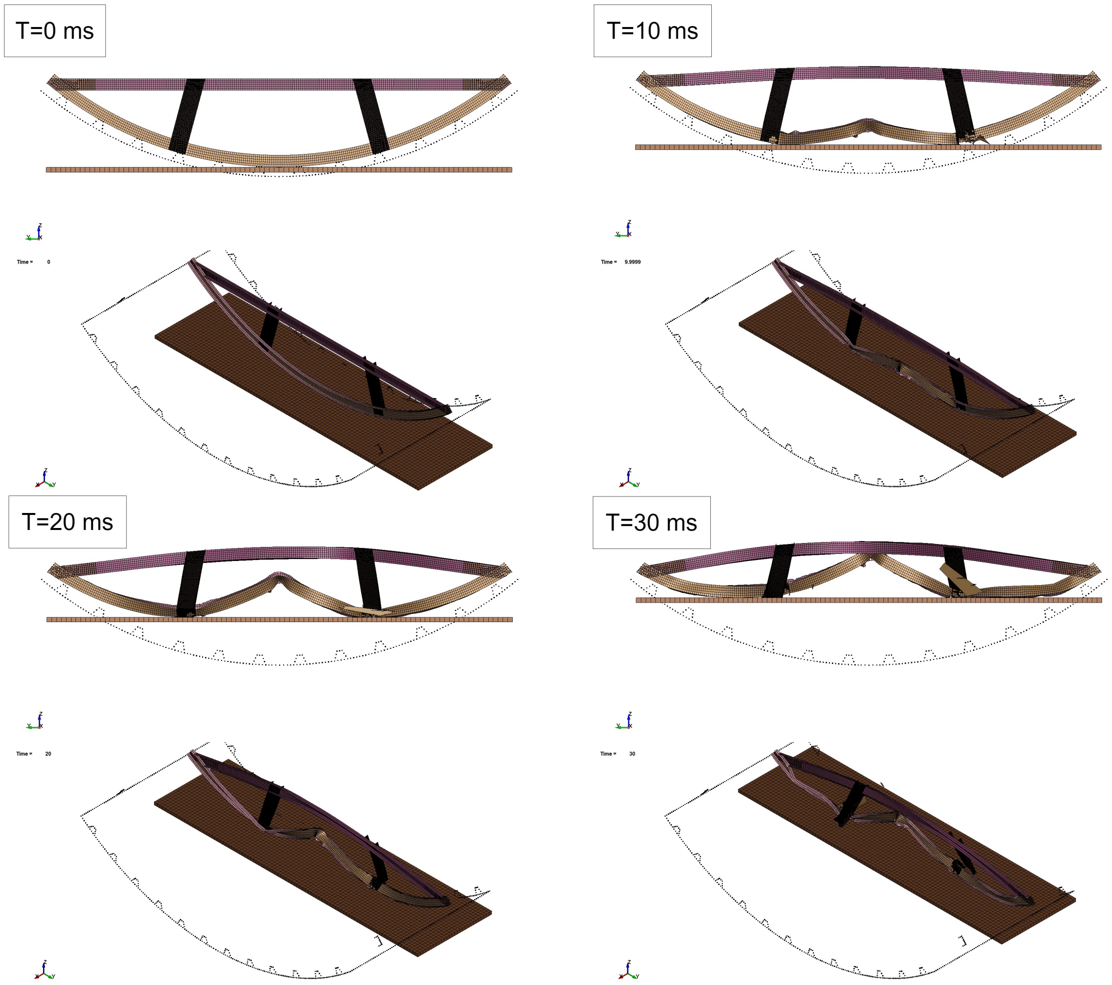


Figure 4.12: Crushing of the Frame-C-strut-beam model with X-constrained nodes along the bottom of the frame.

4.3.3. Comparison Half to Original STUNNING Model

Failure of half the STUNNING fuselage model and the original model is depicted in Figures 4.5, 4.9 respectively. During the first 10 ms of the impact, the two models show very similar behaviour; as the crushing plate makes contact with the skin, the skin starts to flatten and the stringers orientation becomes distorted as a result. Furthermore, within a few milliseconds, the circular frame starts to flatten in both models. after the first 10 ms, the C-struts in both models are heavily deformed but have not yet broken in two. This happens around 13.3 ms for the half model and 14.6 ms for the original full model. From failure of the C-strut, both models show a similar trend where the skin moves up into the cross cargo beam, and the way in which the skin compresses near the end is somewhat symmetric in both models.

In Figure 4.13 a comparison is given of the total energy of both models. It shows extremely similar behavior in energy absorption over time. The total energy of the original full STUNNING fuselage model is halved as it is expected that the half model would absorb half the amount of energy. This similarity carries over somewhat to the crushing forces in Figure 4.14. The crushing force is obtained with a SAE filter of 1000 Hz for both simulations. Also, note that the force is halved for the original fuselage in order to compare these two models. The large peak force around 16 ms is missing in the half model, however. This is not necessarily an issue if the peak force is not a considered parameter for the neural network to predict. The mean crushing force is quite close around 28 kN for the original model and 32 kN for the half model.

Both simulations in Figures 4.13 and 4.14 were performed with the same number of CPU cores (4). The Total CPU Time was around 32 hours for the half model, and for the full model 58 hours. This is a significant decrease in computing time if a neural network training set would be built from the half model instead of the full original model.

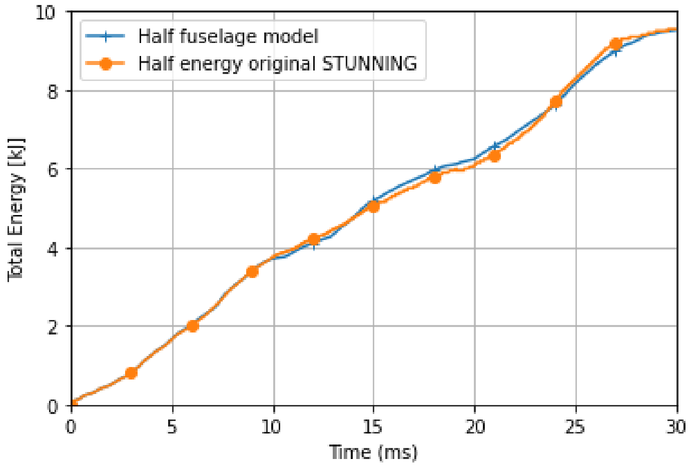


Figure 4.13: Total energy of the half model and the original STUNNING model. The total energy of the original model is scaled by a factor 0.5, whereas the total energy of the half fuselage model is unchanged.

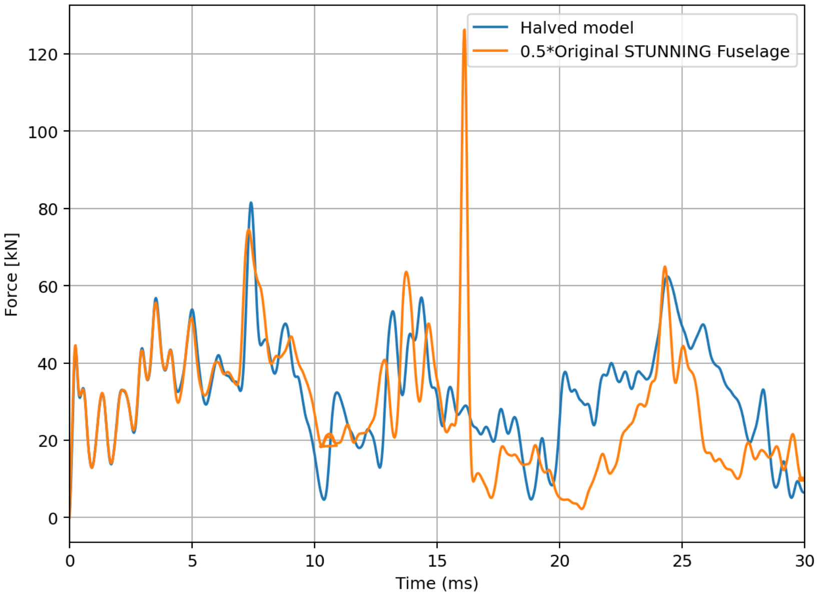


Figure 4.14: Crushing force of the half model and the original STUNNING model. The Crushing force of the original model is scaled by a factor 0.5, whereas the total energy of the half fuselage model is unchanged.

5 Design Space and Performance Parameters

In this chapter a sensitivity analysis is performed on some different design variables that make up the design space dimensions. The C-strut of the fuselage section is identified as an interesting part to parameterize, because it was seen in Chapter 4 that it plays an important role in how the failure progresses as the STUNNING model is crushed. Two design variables for the C-strut are introduced in section 5.1. A small sensitivity analysis is then done, to check whether the effect of these design variables on the global crashworthiness behavior is significant. This is important because a surrogate model will be built in chapter 6 that predicts the crashworthiness behavior given the design variables as input. If these design variables have no significant effect on the crashworthiness behavior, there is no point in building a surrogate model for them. In section 5.3, a sensitivity analysis is done based on changes in the number of CPUs used to parallelize the LS-Dyna simulation of the full STUNNING fuselage section model performed in Chapter 4. Lastly, The conclusions from these analyses are presented in section 5.4.

5.1. Design variations of the C-strut

Different models of the C-strut are generated as LS-Dyna input files. The C-strut can be adjusted in two main ways. The first is the angle of the C-strut, the second is the position of the C-strut. The angle of the C-strut is defined in Figure 5.1, where an angle of $\alpha = 15$° can be seen, which is the original value for the STUNNING fuselage. The C-struts are mirrored in the Y-axis, and the C-strut orientation does not change when comparing the C-struts in X direction (see Figure for axis system). Two extremities of this angle are simulated: 0 and 30° (Figure 5.2 and 5.3). Due to the fact that a large number of input files need to be created to obtain a neural network training data set, automatic generation of input files is a necessity. Therefore, a script has been written in Python that generates mesh and element formulations of the C-strut. The script requires the input of the original XYZ data from the C-struts, the requested angle, and the requested translation distance along the Y-axis. The script extracts the location of the top nodes, the nodes with the highest Z-value. These nodes are then translated along the Y-axis according to the requested translation. Then the nodes are projected onto the bottom of the frame at the requested angle. The top and bottom nodes are then defined, and horizontal meshing lines are drawn so that elements of the right size are created.

With this, a new LS-Dyna file is created to obtain a new geometry of the C-strut. A difficulty arises in generating a consistent mesh size and geometry, meaning that slight variations can be observed when comparing two different points in the design space, especially along the bottom of the C-strut where the C-strut meets the frame. Generally an element size of $2.5mm$ is aimed for which was the approximate size of the elements on the original model.

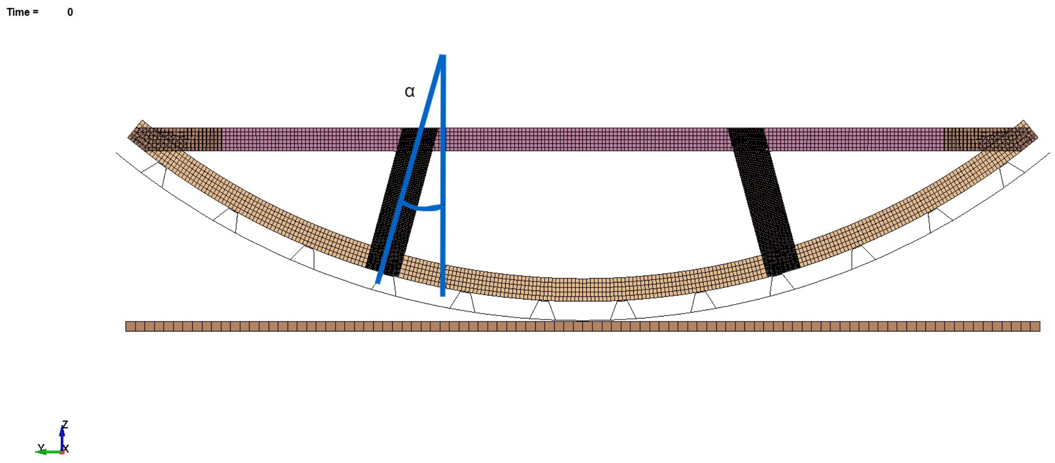


Figure 5.1: Definition of the angle α of the C-strut.

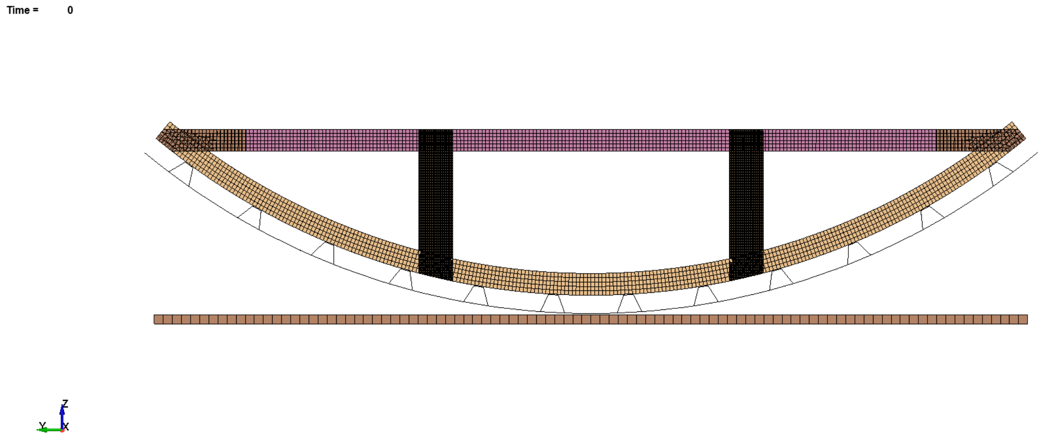


Figure 5.2: The C-struts placed at an angle $\alpha = 0°$

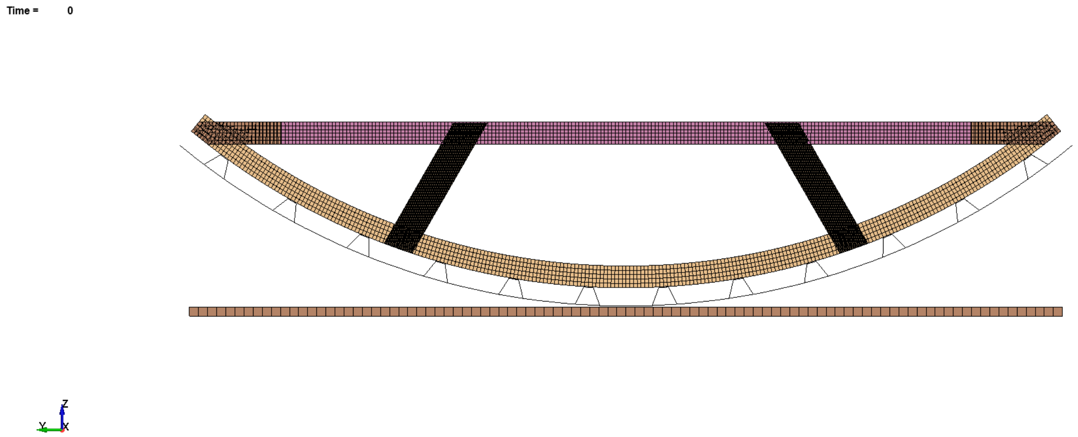


Figure 5.3: The C-struts placed at an angle $\alpha = 30°$

The second varation, moving the C-strut, means translating the C-struts in Y direction. For example in Figure 5.4 below the C-struts are translated 100mm outward in Y-direction from their original position.

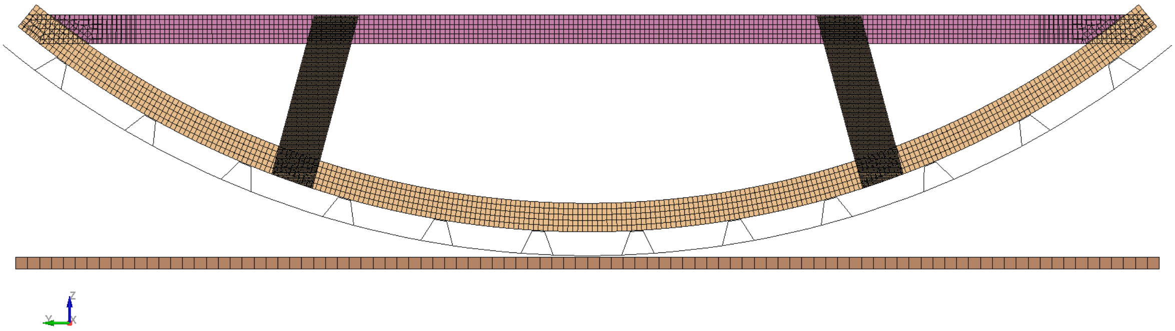


Figure 5.4: The C-struts placed at the original angle $\alpha = 15°$, but both moved outward with 100mm in Y-direction.

5.2. Design Space Sensitivity analysis

The simulations of the variations presented in Figures 5.2, 5.3, and 5.4 are performed to see how sensitive the different design space dimensions are to the crashworthiness indicators discussed previously. The resultant total energy of each energy over time is presented in Figure 5.5. From the figure it is quite clear that each adaptation causes a different total energy path except for the model that moves the C-struts outward and the C-strut angled at 30 deg. From inspection of the failure mode of these two simulations, it becomes clear that if the C-strut joint on the frame is moved too far outward, the cross cargo beam will fail and the C-struts will not see a complete failure after 30 ms of crushing. Therefore an additional proposed design variable apart from the ones mentioned in section 5.1 would be the layup of the cross cargo beam. That would be to prevent this cross-cargo beam failure mode from occurring on multiple points in the design space and create results that are too similar. Vice versa, a possibility would be to keep the cross-cargo beam parameters constant and then investigate if the neural network can accurately predict at what angles/position of the C-strut this alternate failure mode occurs. When looking at the crushing force of the plate against time in figure 5.6, the difference between the test with the outward moved C-strut and the test where the C-strut is at its original position but angled at 30 degree is more easily seen.

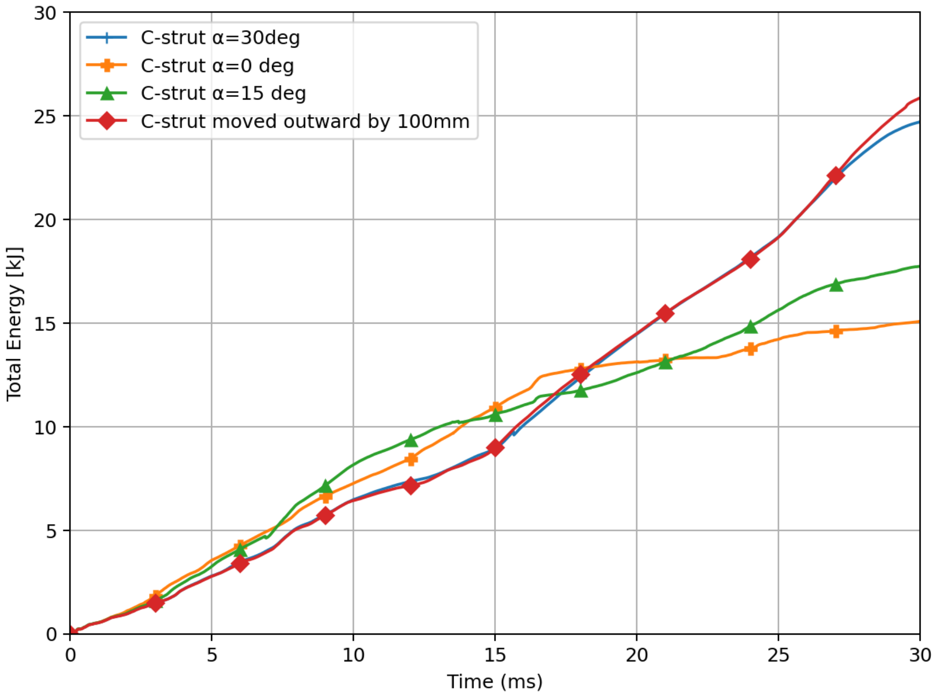


Figure 5.5: Total energy of the simulation for different design variations of the C-struts plotted over time.

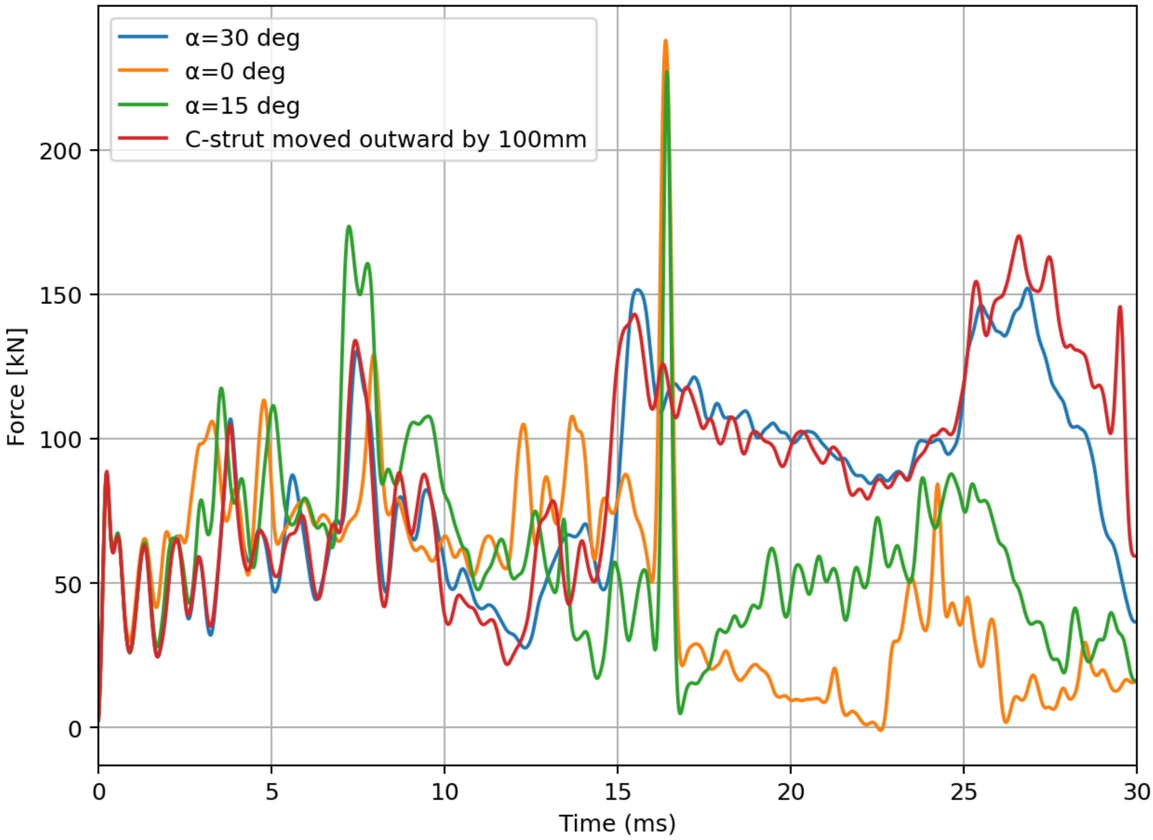


Figure 5.6: Different Contact force plots over time of design variations.

Table 5.1: Data read from Figure 5.6.

Design Varation	Mean Force	Peak Force
$\alpha = 30°$	83 kN	152 kN
$\alpha = 0°$	51 kN	238 kN
$\alpha = 15°$	61 kN	227 kN
Moved C-strut outward by 100 mm ($a = 15°$)	86.5 kN	170 kN

5.3. Parallelization Sensitivity Analysis

To build a large dataset a great amount of computing power is required to do the necessary simulations of the model. To speed up simulations multiple CPUs are used for one simulation in a strategy known as parallelization. There are two common ways to share the work of one simulation, time domain parallelization, and space domain parallelization. The LS-Dyna solver used for this research uses the space domain parallelization method. In figure 5.7 different wall times and CPU times can be seen for different amounts of CPUs used per simulation. The wall time is the real-time measured between when the job was started and when it was finished. The CPU time is then the sum of the durations that each CPU was being used by the solver. From this data, the formula WallTime*NCPU=CPUtime seems to fit, which either means that no CPUs were ever idle or idle CPU time is still added to the total CPU time. These simulation were performed on the HPC12 cluster of the TU Delft. The simulation performed for this is always the same input file, namely the STUNNING original fuselage model. The cluster consists of different nodes representing CPUs, and not all CPUs are of the same model. Specific allocation of nodes to a simulation cannot be controlled for this research. When looking at the top graph in Figure 5.7, the ideal relation between number of CPUs and wall time would be a linear relation. Linearity in the reduction of the wall time is still observed when going from 1 to 4 CPUs but it decays less using 8 and 16 CPUs.

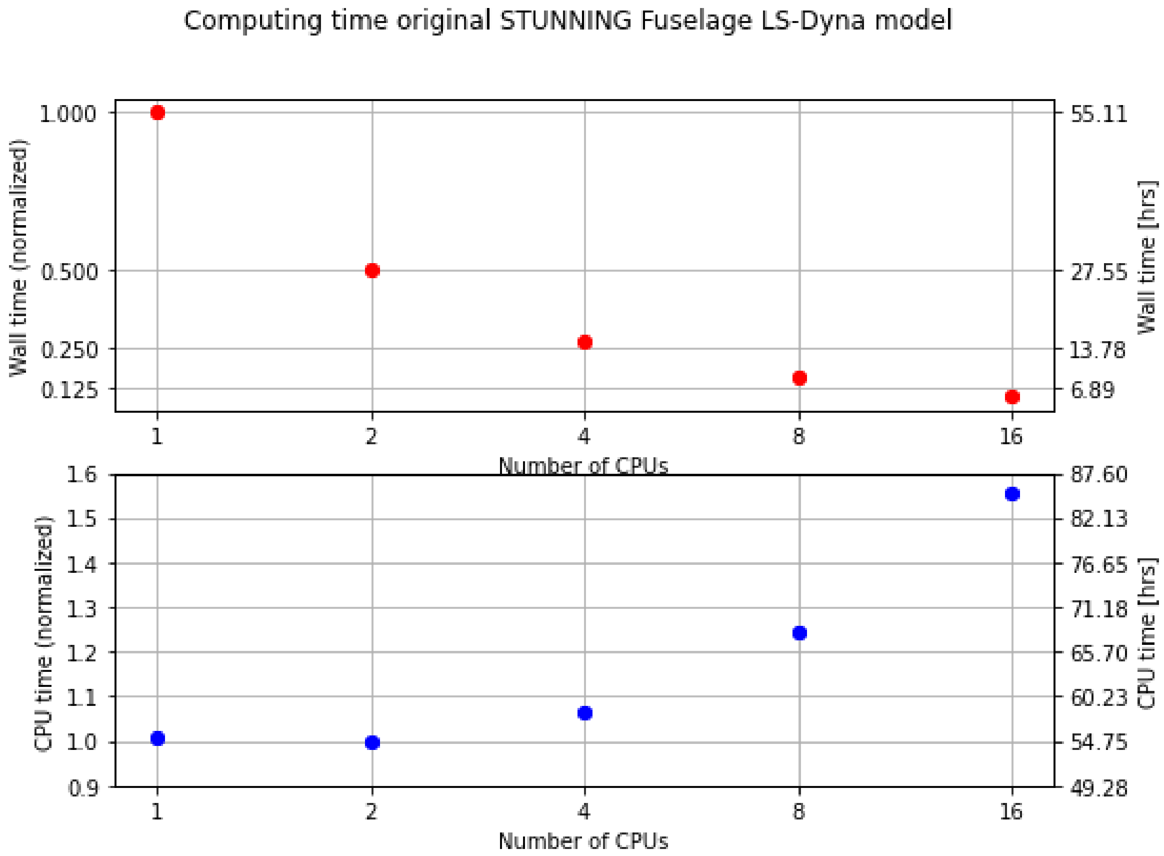


Figure 5.7: Computing times of the original STUNNING fuselage model using different numbers of CPUs for paralellization of the model. The wall time indicating the time passed between starting and finishing the simulations(without interruptions/pauses) and CPU time indicating the sum of the amount of time that each CPU was active during the simulation.

When comparing the results of the different Parallelizations, there is a significant difference after two-thirds of the simulation has been performed (at real time= 20 ms). This difference can be observed in Figure 5.8. In the figure it is clear that the difference between the use of 8 CPUs and 2 CPUs is the largest, giving a total energy of around 18 kJ vs 15 kJ for 2 CPUs. When using no parallelization (1 CPU) the total energy is between these two extremities. Also when using 4 CPUs the total energy is the closest to this. In the interest of creating a consistent training dataset to feed to a neural network, the decision is taken to run all simulations with a standard number of CPUs. Alternatively/Additionally the simulation can also be reduced to only simulate the first 20 ms of the impact on the STUNNING fuselage. This would be the go-to strategy if the number of computer resources available varies and therefore the number of CPUs per simulation too. The disadvantage of this would be that a third of the structure's failure is no longer simulated, which is not preferred generally, so this will not be done.

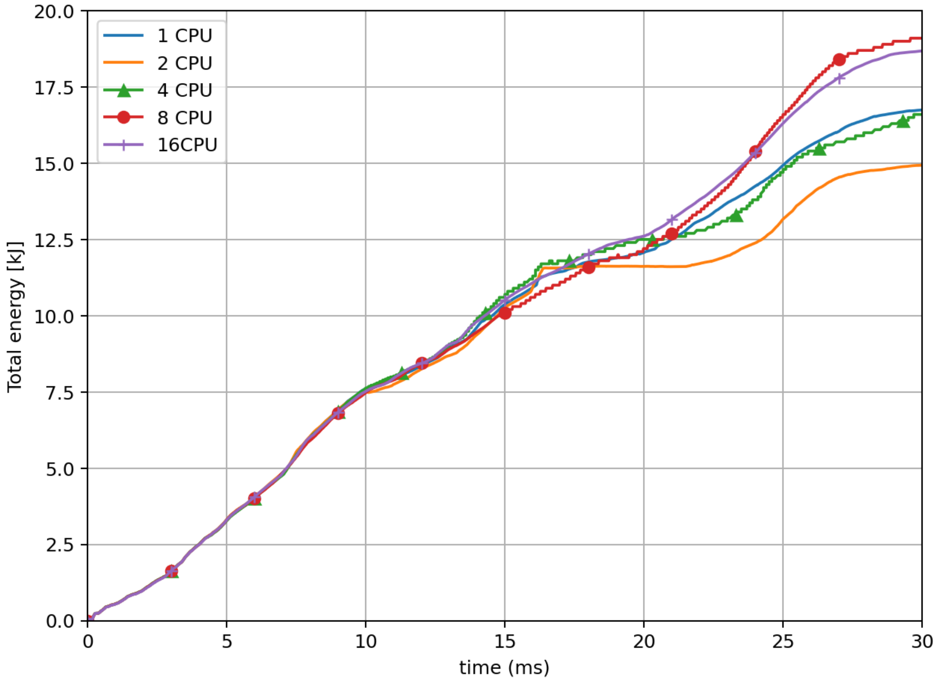


Figure 5.8: Total energy of the STUNNING fuselage simulation as function of real time for different numbers of CPU used per simulation.

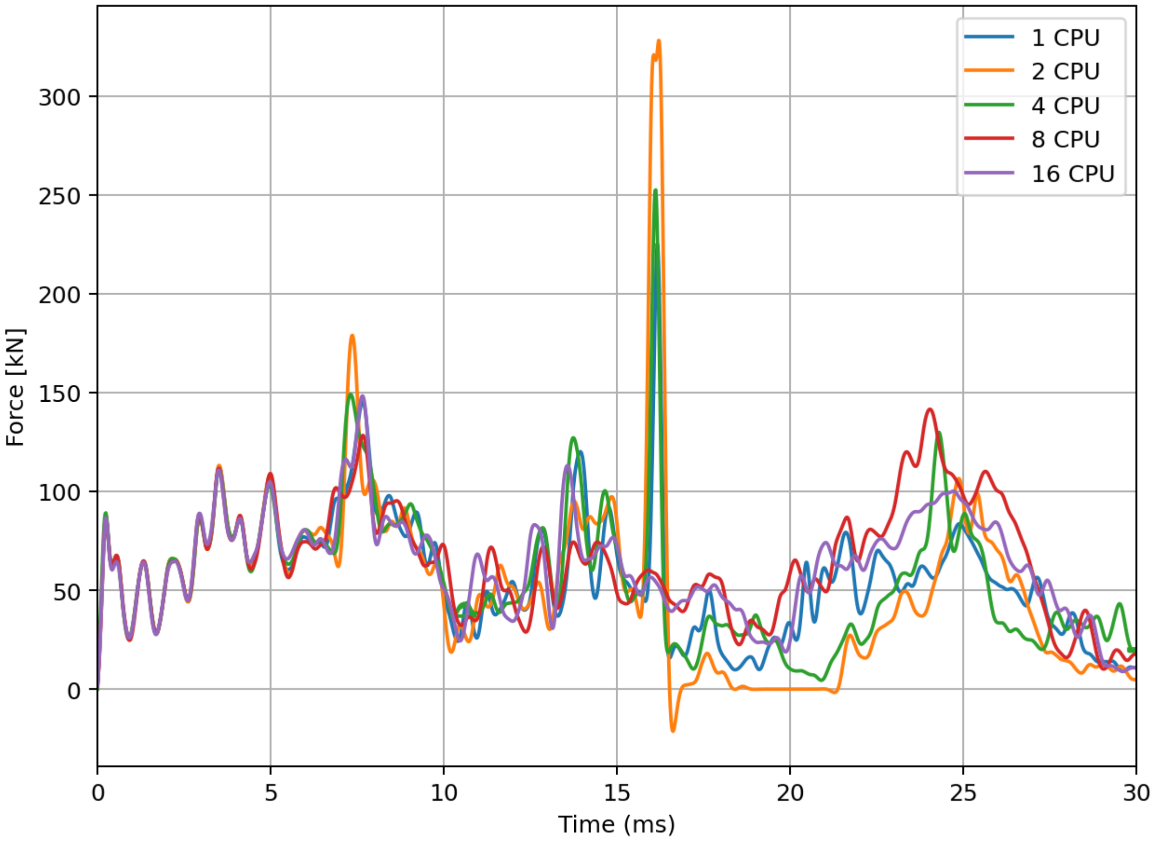


Figure 5.9: Crushing force of the different parallelization simulations. Filtered with a SAE 1000 Hz filter.

5.4. Final Design Space and Crashworthiness Indicators

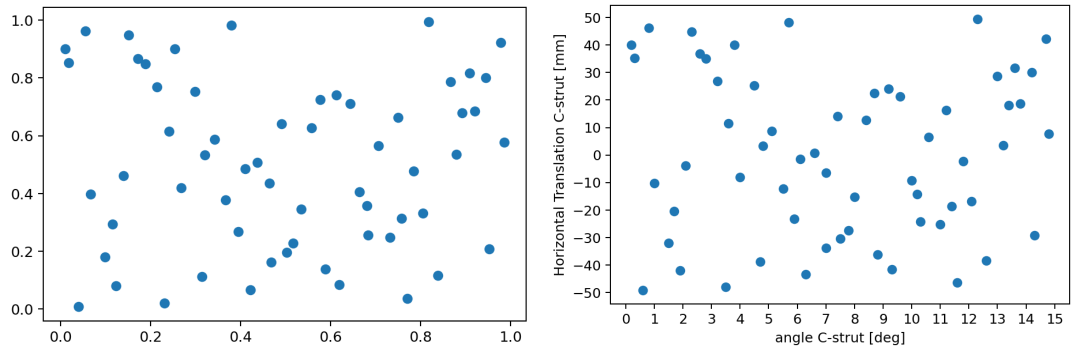
The final design space that will be used for the training of the Artificial Neural Network (ANN) surrogate model will consist of 2 dimensions. The angle of the C-strut and the translation in Y direction, are presented in Figure 5.1 and 5.4 respectively. The third variable considered in this chapter, the layup or thickness of the cross cargo beam was left out. The reasoning behind this is twofold. First of all, the complexity that a third dimension in the design space would bring would require an exponential increase in the number of data samples that need to be gathered for the surrogate model. This does not necessarily have to be a problem if it turns out that the surrogate model will do well within a design space of two dimensions, a third dimension can always be added. Secondly, It was also decided that the failure mode that was caused by placing the C-strut too far outward or with a large angle was not of interest in the research. In an attempt to exclude it as much as possible, the angle of the C-strut will be limited from 0 to 15 deg, and the translation of the C-strut will be limited to -50 to 50 mm. A translation of 50 mm means that both C-struts move 50mm outward, increasing the gap between them by 100 mm and vice versa for -50 mm. An important note to make is that the effect on the crashworthiness performance of these two design variables is not intuitive. This makes inspection and verification of the results very difficult. On the upside, however, it creates an opportunity to show the true value of an ANN surrogate model as it will try to learn from any data and subsequently can help identify the relation between the design space dimensions and the crashworthiness performance. The crashworthiness indicators that are of interest for the thesis are the total energy absorbed and the mean crushing force. The total energy and the mean crushing force both give an indication of how well the structure can dissipate the kinetic energy and thus give a measurement of how capable the subfloor structure would be in the event of a crash. Furthermore, these two parameters are proportional to each other. The specific energy absorbed in frame of this analysis would be the total energy divided by the mass of the structure. The changes in mass for the design variables ar Since the mean crushing force is more tedious to obtain, given that the force graphs always need to be filtered, it was therefore decided to take the total energy as the parameter that the Artificial Neural Network Surrogate model, presented in the next chapter(chapter 6), will be trained on. Another crashworthiness indicator that would ideally be predicted by the neural network is the peak force, which is an indication of the maximum acceleration that the occupant in the event of a crash would experience. The peak force however is difficult to model accurately with the current model in LS-Dyna, for example, the number of CPUs can affect the peak force significantly. This can be seen in section 5.3 in Figure 5.9. In this Figure, peak forces can be observed from over 300kN to about 140 kN for the same input file but using a different number of CPUs in parallel. This enormous deviation of a factor of 2 can not lead to a consistent data set when running many deviations of the C-strut.

6
Artificial Neural Network Surrogate Modelling for the STUNNING Fuselage

In Chapter 5, a basis has been established for the design space, the simulation parameters, and the model that will be parametrized for the surrogate model that will be built as the final analysis for this thesis. The surrogate model will be in the form of an Artificial Neural Network (ANN). Section 6.1 will present the methodology of building a training dataset, from which the ANN will learn to predict the final total energy after 30 ms of simulated impact. Section 6.2 will go through the setup of the ANN and explain how it is formed. Then the networks are trained and the results are analyzed in section 6.3. Lastly, section 6.4 will present the conclusions drawn from the analysis.

6.1. Sampling of Design space and Building Training dataset

To build a Training dataset the design space needs to be sampled. For this, in the literature study, Chapter 1, a common method for this was presented. This is the Optimal Hypercube Latin Sampling (OHLS) method. There are several different strategies when discretizing the design space. The two most viable for this research are the centered OHLS and maximin OHLS. centered OHLS centers the sampling points within the sampling intervals, and maximin maximizes the minimum distance between points. This has a consequence however that the distance between two points along one axis of the design space varies. For ANN training, that consequence is not necessary or beneficial in any way. On the other hand, due to the high non-linear dynamics of the simulation spreading out the sample points within the design space can potentially offer a better learning progression of the ANN. Therefore the maximin OHLS is used for the sampling. The Literature in chapter 1 used between 48 to around a 100 simulations to build a training dataset. Since no existing method or rule was found within the literature on determining this number, the number of sampling points is set at 60 for this research. Additional data can always be added, in form of a more dense sampling set, or an extension of the design space. The ANN can in that case be retrained. The sampling is visualized in Figure 6.1, the sampling method returns a normalized set. This set can then be transformed into a non-normalized set representing the values that are passed on to the next step.



(a) Normalized version of the design space,

(b) The design space with the design variables on each axis.

Figure 6.1: The design space created with random seed number 226. To go from the Normalized to the actual design space, the X-axis is multiplied by 15 to obtain the angle of the C-strut (defined in Figure 5.1), and the Y axis is multiplied by 100 and then translated -50 to obtain the horizontal translation of each C-strut.

The next step is the generation of the input files. 60 versions of the STUNNING Fuselage need to be created according to the sampling in Figure 6.1. In Section 4.3.3 it was concluded that running the half STUNNING model results in very similar crush behaviour and that the crashworthiness indicators such as total energy and mean crushing force are very similar albeit halved of the original model. Because of this it was decided that, to save on required computing power, the half model is used for the building of the training dataset. This means that the assumption is made from this point that the half STUNNING model total absorbed energy will always be half of the original STUNNING model regardless of the design variation. It is important to highlight that the sample points in Figure 6.1b do all comply to the symmetry in the vertical center line of the fuselage skin. A script written in Python generates the mesh and shell element definitions of all samples. Subsequently, the input models are uploaded to the High Performance Cluster (HPC) 12 of the TU Delft. The cluster contains groups of different processors. They are of different make and model and therefore have different numbers of cores (one core is referred to as one CPU). Each processor has access to local RAM and a hard disk, together these components form a node. The simulations are distributed over the available nodes. This is a relevant detail of this analysis because the LS-DYNA manual [51] mentions that consistency problems are inherent in numerical simulations of impact problems where structural bifurcations under compressive loads are common. Due to the way the software orders the summations, small round-off errors can result in different outcomes for the same input file. More importantly, is that the manual also mentions that the results can even differ between different computers. Discrepancies caused by running simulations on different nodes of the cluster are not avoided for this research. In section 5.3 it was found that running the simulations with four CPUs creates the most consistent result compared to the model that ran on one CPU. Furthermore, the cluster consists of nodes containing 16 or 32 CPUs and is consistently being used for other research. Therefore selecting four CPUs as standard rather than a higher amount for every simulation is deemed the best choice, due to the fact that the large amount of simulations can be spread across nodes that are vacant more easily.

The results of the simulations are visualized in Figure 6.2 and 6.3. The total energies of all simulations are plotted over time in figure 6.3, while in figure 6.2 the final total energies are plotted against the design variables. From figure 6.3 it is easy to determine whether any simulation results show unrealistic behavior. A couple of simulations were resubmitted to the cluster for example, due to the total energy graph showing values of over $1e7$ J. This is completely disproportionate compared to the majority of the results, which are in the area of 8-10 kJ near the end of the simulation(at 30 ms). After rerunning, the simulations showed results of the same order of magnitude as the others. Figure 6.2 shows a surface, which is obtained by doing a Delaunay triangulation in Python. This triangulation method is not based on the physics of the problem, and therefore is not accurate in between the data points. The triangulation simply tries to fit a smooth surface to the data points. This is advantageous, because it is difficult to read from a simple scatter plot which areas of the domain show for example a low final total energy.

As a last step, the ANN is trained on a normalized dataset. The normalized dataset for the input

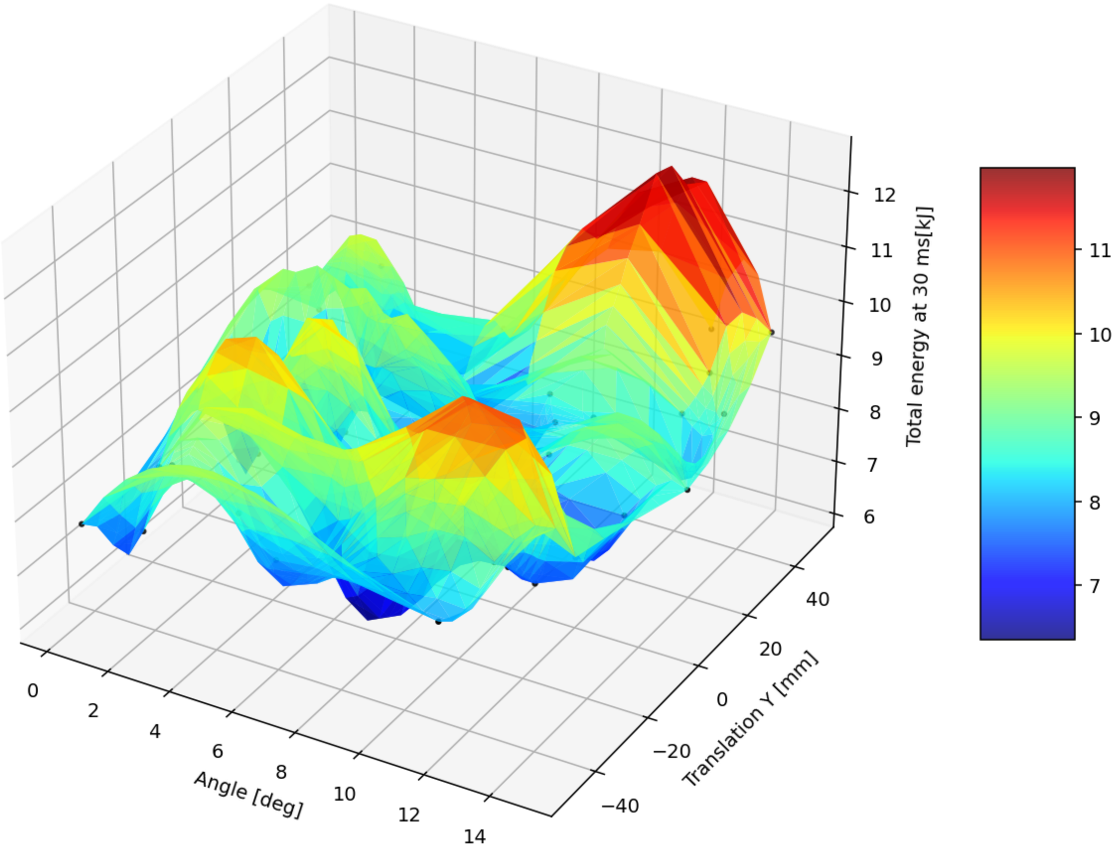


Figure 6.2: Surface fit to 60 samples using Delaunay triangulation on the design space dimensions

was already defined in Figure 6.1a. To normalize the training data a MinMax scaler is used:

$$y_{\text{norm}} = \frac{y_{\text{true}} - y_{\text{min}}}{y_{\text{max}} - y_{\text{min}}} \tag{6.1}$$

Where y_{norm} is a normalized train label, y_{true} is the unscaled train label, and $y_{\text{min}}/y_{\text{max}}$ are the minimum and maximum train labels found in the dataset respectively.

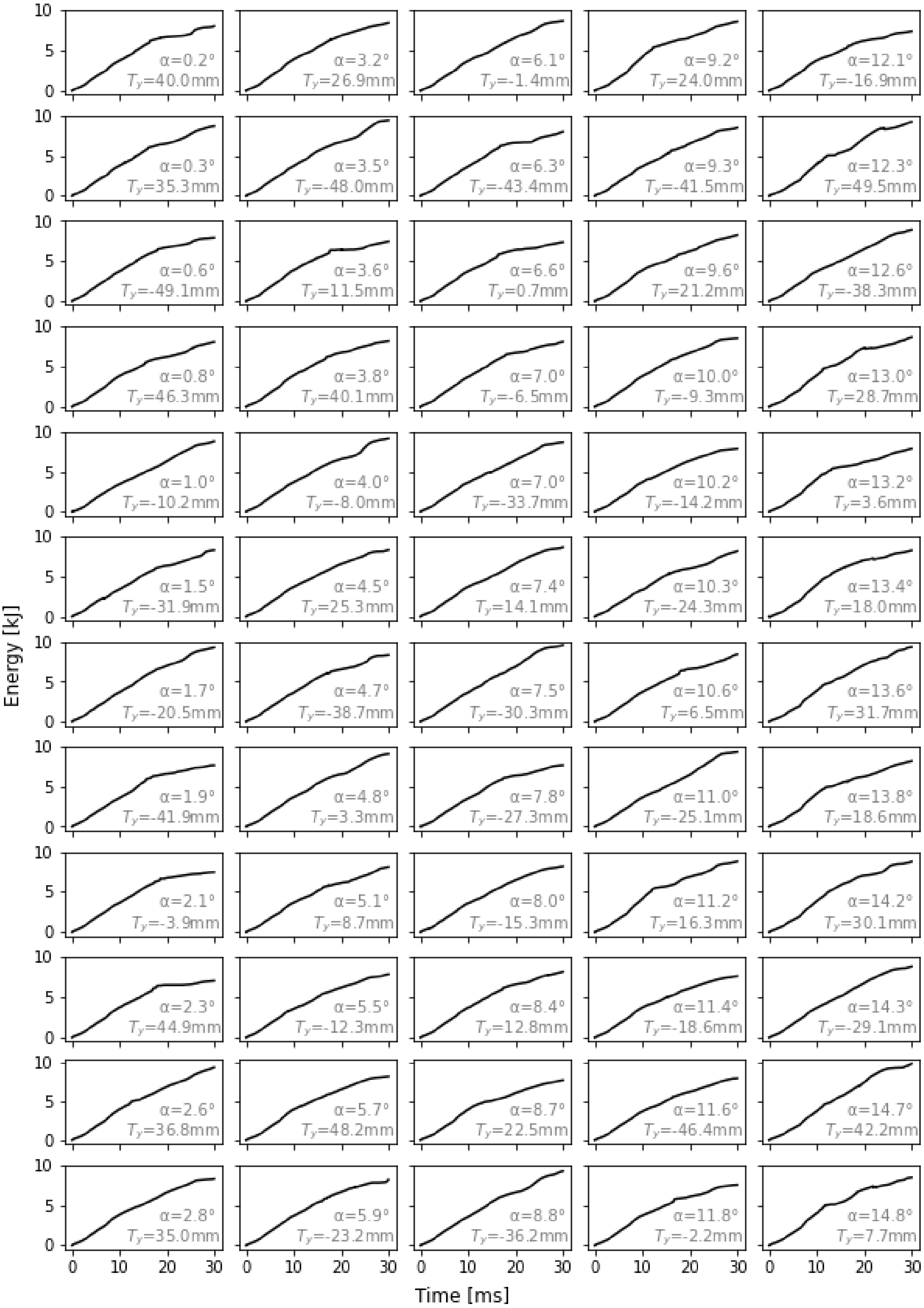


Figure 6.3: Total energy of the model over simulation time for the 60 samples found in 6.1b. Each graph is watermarked with the value of the angle in ° and the translation of each of the two C-struts in Y direction (T_y).

6.2. Artificial Neural Network Setup

As discussed in the literature study Chapter 1, this research makes use of Feedforward Neural Networks (FNNs). The FNN consists of a number of neuron layers that passes information between each layer. In FNNs the information is only fed in one direction (the forward direction). For the building of the FNN, the Tensorflow/Keras Python packages are utilized. The FNN performs a regression task similar to the work in chapter 3.

A number of hyperparameters and design choices need to be determined that make up the learning process, the architecture of the network, and the evaluation of the network. To do so, a couple of initial networks were trained and tested. Each aspect is discussed going forward.

6.2.1. Learning Process

The network is trained using Stochastic Gradient Descent with backpropagation. The network makes use of the Adaptive Movement Estimation algorithm (Adam) optimizer to do so. The Adam optimizer, described in [54], uses two decay rates to slow down the learning as the number of learning iterations(epoch) increases.

The cost functions considered were the Mean Absolute Error (MAE) and the Mean Squared Error (MSE). Throughout a couple of initial tests, sensitivity analysis was performed on both cost functions. From these tests, no observable difference was found between the final loss of the FNN. It was therefore decided to choose the MSE as the cost function, as this is a common cost function in literature and it was already used before in the analysis of chapter 3 where it performed effectively. From here, the final hyperparameters were tuned by hand to obtain a learning process reliably converges to a similar loss value on every run.

Parameter	Symbol	Quantity
learning rate	lr	0.0001
The Exponential Decay Rate for the 1st moment estimates	β_1	0.9
The exponential Decay Rate for the 2nd moment estimates.	β_2	0.999

6.2.2. Feedforward Neural Network Architecture

For the architecture of the network, there is no intention to set all parameters in stone, but rather to define a range over which the architecture can be changed. This way, an optimal network architecture can be found, and variance between the results can offer information about what is happening in the network. Diagnosing this can sometimes be difficult due to the nature of the neural network, which is most easily compared to a black box. First of all, from the literature research in 1, three promising candidates came forward for the neuron activation function. These are the Rectified Linear Unit(ReLu), the Sigmoid activation function, and the Gaussian Error Linear Unit activation function (GeLu). Other activation functions exist, but they are not applicable to this research, as they perform different tasks such as classification. The Sigmoid and ReLu functions have already been used in chapter 3. The GeLu function is added because in chapter 3 the difference between using ReLu and Sigmoid neurons did not change the training speed or the error of the network after training.The definition of the GeLu function is:

$$f(z) = 0.5z\left(1 + \tanh\left[\sqrt{2/\pi}\left(z + 0.044715z^3\right)\right]\right) \tag{6.2}$$

Where z is equal to the input of the neuron, which is defined in equation 1.4. The GeLu function [55] has characteristics from both ReLu and Sigmoid which can be seen in figure 6.4. First of all, the GeLu function has an activation value for inputs below zero, similar to the Sigmoid function. The ReLu has no activation value for inputs below zero. Secondly, the GeLu function converges to the same behavior as the ReLu function for inputs well above zero. The expectation therefore, is that due to their differences, the different neurons types will allow the network to converge to different minima of the cost function, so that a more elaborate comparison can be made.

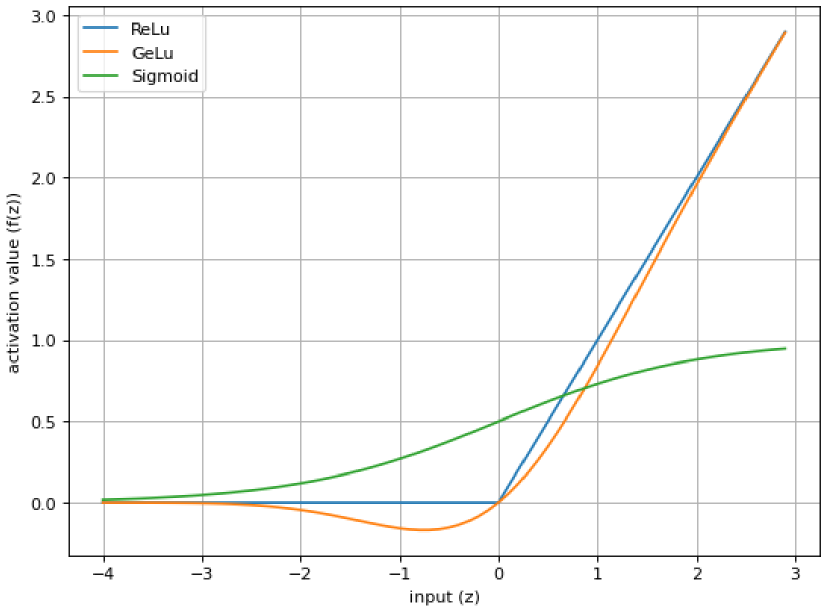


Figure 6.4: The ReLu, GeLu and Sigmoid neurons activation functions compared.

From initial testing, it becomes clear that both networks with both a single and double hidden layer perform similarly. Two assumptions are made throughout the analysis, which is that the neuron activation function is constant along the network (within the layer and with respect to other layers). Secondly, the use of two hidden layers is not increasing the effectiveness of the networks, and therefore will not be considered. The number of neurons in the network set the number of trainable parameters, and thus the capacity of the network. The FNNs have a range of 10 to 80 neurons in their hidden layer. The step size within this range is set at 10 neurons, creating a total of 8 steps. The reasoning behind setting the maximum number of neurons at 80 is simply that there are no more than 60 samples in the dataset. Therefore, increasing the number of neurons far above the dataset size will only allow the network to overfit because every neuron weight can be set to one dataset sample. Additionally, in the case of ReLu neurons, the ’dying neuron’ problem can occur when a neurons output is converging to zero as the training sets off. This means that these neurons do not contribute anything to the performance of the network and thus the name dying neuron. From the maximum number of neurons, the stepsize of 10 was chosen because it was the smallest step size in which a considerable change in performance can be seen within the first few training tests.

6.2.3. Evaluation of the Network

The evaluation of the network, after training, is an important part of the analysis. Ideally, the network must produce the same accuracy when its asked to predict a train label of an unknown sample. In practice, however, it is common that the error on data not seen by the FNN during training is higher. To evaluate this difference a part of the dataset produced in section 6.1 is withheld from the network. This section of the data is called the test data. An important pitfall to look out for is when the network overfits, causing the error on training data to be low while the error on the test data is significantly higher. This is because the network stores the train labels of the test data without building any significant algorithm. The test data is determined by a variable called the validation split. The validation split determines the ratio between the number of samples in the test dataset to the training dataset. It is important to note that this split consists of the same samples for each network trained if the split between two FNNs is constant. This is important to prevent noisy results for evaluation. This also means that if the validation split is doubled for example, then the test dataset will contain the same samples plus an equal amount of new samples.

6.3. Results of Training

With the setup described in previous section, 48 FNN were trained using the 60 samples described in section 6.1. Each sample contains the normalized angle and normalized translated position of the C-strut as input, and the normalized final total energy at 30 ms as train label. The resulting MSE on the training data and the MSE on the test data are reported each time after the network is trained. However, due to the fact that the MSE is normalized, the values are meaningless on their own. Therefore, the MSE will be scaled back into Joule, and then, the MSE is transformed into the Root Mean Square Error(RMSE). The tables containing the MSE are put in Appendix A and the training MSE and validation MSE specifically can be found in the final two columns of Table A.1 respectively. Table 6.1 contains the RMSE in Joule. As can be seen in the table, the number of neurons was increased with steps of 10, starting with 10 and ending with 80 neurons. Furthermore two validation split values were used. 0.10 making up a test dataset of 6 samples and 0.20 making up a Test dataset of 12 samples.

Table 6.1: Results of the training of the networks with the Root Mean Square Error in Joule for easier interpretation of the results. These networks were trained/tested with a total dataset length of 60 samples.

Val. split	number of neurons	neuron activation func	training RMSE[J]	val. RMSE[J]
0.10	10	ReLu	549.169	806.862
0.10	20	ReLu	577.581	960.231
0.10	30	ReLu	574.747	951.689
0.10	40	ReLu	482.786	636.869
0.10	50	ReLu	467.079	607.079
0.10	60	ReLu	474.486	728.782
0.10	70	ReLu	450.927	554.699
0.10	80	ReLu	454.066	616.292
0.10	10	Sigmoid	581.810	729.830
0.10	20	Sigmoid	587.825	822.133
0.10	30	Sigmoid	585.552	794.571
0.10	40	Sigmoid	586.832	805.433
0.10	50	Sigmoid	588.250	823.175
0.10	60	Sigmoid	588.204	823.304
0.10	70	Sigmoid	588.390	820.595
0.10	80	Sigmoid	587.708	816.676
0.10	10	GeLu	551.182	571.711
0.10	20	GeLu	578.741	735.485
0.10	30	GeLu	579.487	733.994
0.10	40	GeLu	579.455	734.942
0.10	50	GeLu	579.498	737.296
0.10	60	GeLu	579.547	738.606
0.10	70	GeLu	579.544	735.850
0.10	80	GeLu	579.492	733.345

Table 6.1 continued

Val. split	number of neurons	neuron activation func	training RMSE[J]	val. RMSE[J]
0.20	10	ReLu	554.161	1055.700
0.20	20	ReLu	554.417	914.733
0.20	30	ReLu	511.727	946.220
0.20	40	ReLu	496.705	828.882
0.20	50	ReLu	499.368	933.994
0.20	60	ReLu	486.632	745.076
0.20	70	ReLu	499.554	718.395
0.20	80	ReLu	471.303	1027.771
0.20	10	Sigmoid	580.843	807.430
0.20	20	Sigmoid	580.675	809.849
0.20	30	Sigmoid	580.719	812.840
0.20	40	Sigmoid	580.731	815.300
0.20	50	Sigmoid	580.689	811.834
0.20	60	Sigmoid	580.728	814.769
0.20	70	Sigmoid	580.685	811.934
0.20	80	Sigmoid	580.686	811.977
0.20	10	GeLu	575.715	912.405
0.20	20	GeLu	578.229	947.015
0.20	30	GeLu	573.546	864.346
0.20	40	GeLu	576.522	939.924
0.20	50	GeLu	575.219	914.000
0.20	60	GeLu	577.065	967.185
0.20	70	GeLu	577.017	967.031
0.20	80	GeLu	575.937	941.864

From this point on, it was decided to lengthen the training dataset with another 60 samples. The complete dataset, now containing 120 samples is seen in Figure 6.5. They were generated through the OHLS method similar to the previous 60 samples. Another surface is fitted to the data using a Delauney triangulation, which is seen in figure 6.6. For this round of training, The validation split is kept constant at 0.10, equivalent 12 samples of the dataset. The result of training the networks with a dataset of 120 samples is seen in Table 6.2. As explained before, the table containing the MSE that is used as cost function during training is also put in appendix A (Table A.2).

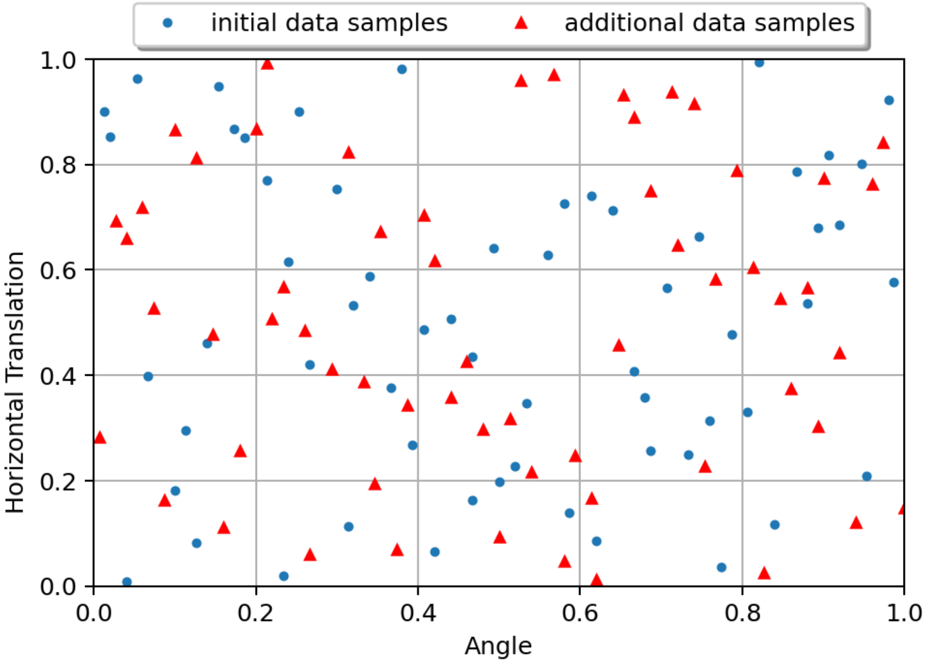


Figure 6.5: Expanded normalized dataset containing 120 samples in total. The horizontal axis represents the angle of the C-struts (normalized), and the vertical axis represents the normalized horizontal translation of the C-struts.

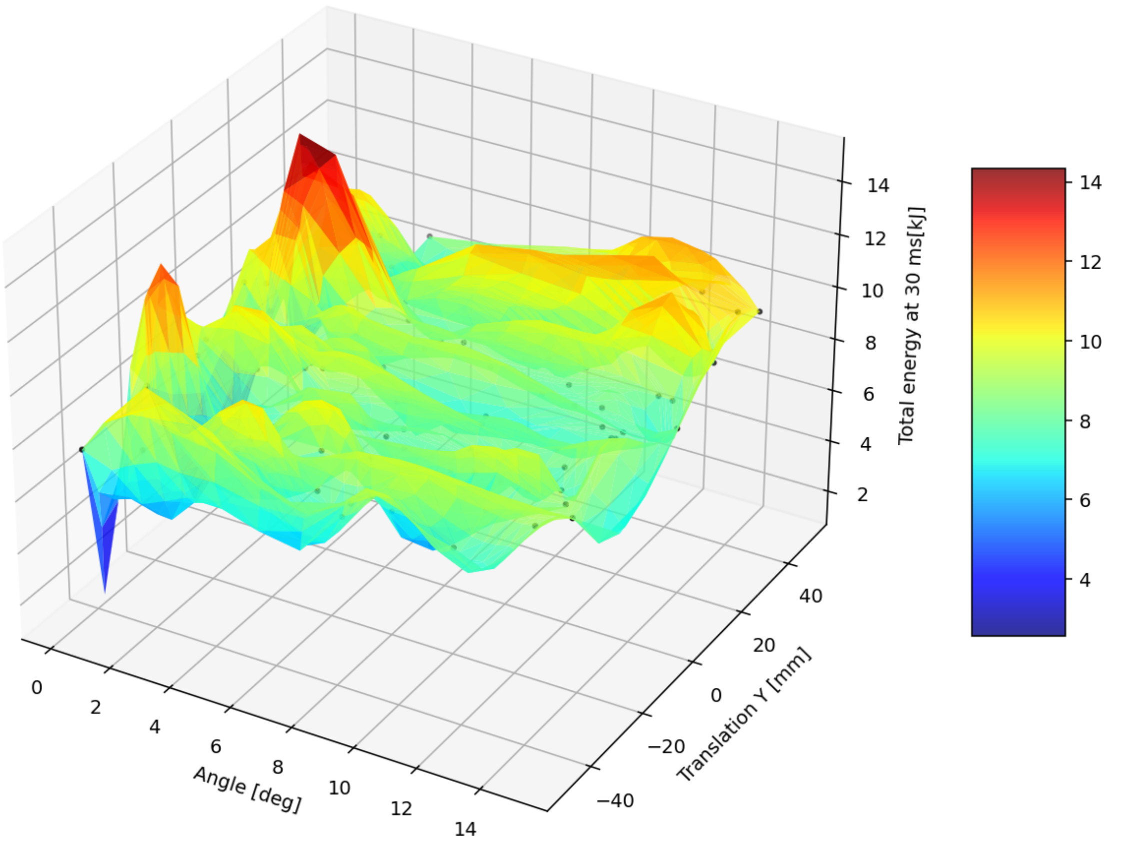


Figure 6.6: Surface fit to the training dataset of a 120 samples.

Best candidates analysis

Comparing the results, the lowest training error FNNs from the 60 sample dataset had a training Root mean Square Error (RMSE) of 451 J and a validation RMSE of 555 J. This network has 70 ReLu neurons and a validation split of 0.10. The lowest training error on the FNNs trained on 120 samples was a RMSE 577 J, with a validation RMSE of 770 J. This network has 50 ReLu neurons. This is a large error margin given that the total energy at the end of the simulation ranges between around 7-12 kJ.

Possible reasons why the error is so high could be that either the data received by the network is too noisy and thus it is not able to learn any further, or a breakthrough in training error can only be achieved with much more data samples. The latter defeats the purpose of the research, as increasing the sampling size exponentially means that it becomes less beneficial to build a surrogate model in this way. The former however, the noise of the data, could be improved upon. It was already clear from the CPU sensitivity analysis in Chapter 5, that the results of the simulations are inconsistent between CPUs and different computers/nodes. One suggestion to decrease the noise, which is out of scope for this research, would be to run all simulations on the same system. Another would be to add dropout to the FNN. A dropout layer deletes data streams during training within the network. This forces the network to find alternative ways to interpret and process the given data. Dropout is only generally good to bring down the validation loss, so it is not a solution to bring down the overall error of the network but rather to bring the training loss and validation loss closer together.

Effect of the validation split

The validation split determines the ratio between the number of samples trained and tested on. The validation split is only changed on the smaller dataset, containing 60 samples in table 6.1. When

Table 6.2: Results of training the networks with a dataset of 120 samples. A constant validation split of 0.1 (equivalent to 12 validation samples) is used and the remaining 108 samples are used for training. The Root Mean Square error in Joule is reported for easier interpretation of the results (the network was trained on the normalized mean squared error).

Val. split	number of neurons	neuron activation func	training RMSE[J]	val. RMSE[J]
0.10	10	ReLu	613.594	899.700
0.10	20	ReLu	602.061	822.015
0.10	30	ReLu	595.365	778.336
0.10	40	ReLu	600.711	812.474
0.10	50	ReLu	577.063	769.589
0.10	60	ReLu	590.204	883.292
0.10	70	ReLu	591.518	803.425
0.10	80	ReLu	580.283	788.517
0.10	10	Sigmoid	643.868	912.573
0.10	20	Sigmoid	642.440	905.955
0.10	30	Sigmoid	642.164	902.498
0.10	40	Sigmoid	642.297	909.346
0.10	50	Sigmoid	642.250	904.437
0.10	60	Sigmoid	641.693	910.068
0.10	70	Sigmoid	642.429	901.542
0.10	80	Sigmoid	642.028	905.850
0.10	10	GeLu	607.943	811.022
0.10	20	GeLu	622.514	814.865
0.10	30	GeLu	622.407	815.649
0.10	40	GeLu	622.235	815.934
0.10	50	GeLu	621.988	815.416
0.10	60	GeLu	621.929	814.590
0.10	70	GeLu	622.217	816.077
0.10	80	GeLu	622.138	813.904

comparing, for example, GeLu networks validation RMSE, it can be seen that the validation error rises with around 25%. This is not so apparant in the networks with the other activation functions, so a possibility would be that it is caused by the GeLu network not being able to fit. This can be caused by a number of things, the most obvious being is that the GeLu network is trained less effectively on less data. Although this is plausible, that would be a strange conclusion given that the training errors are very much the same between the 0.1 and 0.2 validation split GeLu networks. A possible iteration of the analysis could be performed, where the dropout strategy provided previously can possibly bring better performance to the validation loss of the GeLu neural networks with a validation split of 0.2.

Effect of the architecture size

From the results of the 60 sample set, again presented in table 6.1, it can be seen that for both Sigmoid and GeLu networks, the training and validation error stays somewhat constant. This could indicate the increase of capacity does not lead to a different algorithm but it rather decreases the weights of the network, with increasing neurons so that the same net result is obtained. The ReLu networks, however, do what was expected in advance of the analysis. Namely, the training error reduces as the capacity increases. This means that the network arranges a different calculation method with every increase in capacity. It can also be seen that the validation loss decreases when going from 10 neurons to higher numbers and starts to increase again in the final step going from 70 to 80 neurons. This seems consistent with the expectation that over fitting occurs from 80 neurons or higher.

Moving on to the network that was trained and tested on the 120 sample datasets, in tables A.2,6.2. It seems again that the Sigmoid and GeLu networks are not affected by changes in architecture size. For the ReLu networks however, no overfitting starts to occur as the architecture grows to 80 neurons. This is most likely due to the fact that the training dataset is larger and thus overfitting is not yet occuring at this size.

6.4. Chapter Conclusions

The Artificial Neural Networks developed for this chapter were all of one kind; the feedforward neural network. Within this family of networks three neurons types and eight different architectures have been tested and one hyperparameter,the validation split, in particular has been studied a bit more extensively. Two datasets were developed of different sizes (60 and 120) containing design variations of the C-strut of the STUNNING Fuselage model. To create these sets, the Optimal Latin Hypercube Sampling(OHLS) was used. The input files were generated with the strategy described in Chapter 5. Once the networks were trained, the most important conclusion that was drawn is that the validation error of the total energy, the error on predictions of design samples that the FNN has never seen is quite significant at 555 J root mean square error. Given that the total energy for the data typically ranges between 8 and 12 kJ this is an error that is not satisfactory for detailed design. For application in initial exploration of design spaces, the methodology presented in this chapter could still prove useful. Especially because, the relation between the design variables and the resulting total energy absorption during crushing was unknown beforehand.

Another important conclusion is that the doubling of data samples did not equate to a lower training error or validation error of the networks. This could indicate that the dataset is quite noisy, and that the surface that the ANN is trying to fit to the data cannot be found with this type of network or possibly any other neural network.

7 Conclusions

This thesis, divided into 6 Chapters, aims to build a methodology for the building of artificial neural network-based surrogate models that can predict or assess the global behavior of a large fuselage structure subjected to a crash. More specifically, in the thesis a design space consisting of dimensions that have no easily predictable effect on the crashworthiness performance was considered. Fulfilling this purpose would offer a tool to engineers that can broadly explore design spaces. This way, crashworthiness performance can already be developed in the early design stages of an aircraft. The fuselage subfloor structure used in this research is made from thermoplastic fibre-reinforced composite laminates. Although this material generally offers great prospects, it also complicates the design crashworthiness certification due to the brittle failure behavior of the material. As a consequence, the regulatory bodies of the aeronautical industry have published Special Condition reports for aircraft certification with significant composite material usage in the primary structure. In recent years, the use of artificial intelligence has shown promising results in development of crashworthiness of sub-assembly structures in various industries.

Material Modelling and First Surrogate Model
At the starting point of the thesis, three material models applicable to composite laminates modelling, MAT054, MAT058, and MAT261 were investigated. These material models were applied to a coupon that was subjected to tension testing according to the ASTM standards. Each material model is based on different failure criteria, and therefore showed differences in the load curves. It was noted that the material models also have different pre-failure behavior, causing slight differences in stiffness. Due to the increased complexity of MAT261, there was a significant step in needed computation time for MAT261. This material model uses a more physics-based approach, and is taking into account some extra failure modes such as fibre kinking compared to the other two models. The resulting ultimate load and stiffness do not differ too much, however. Due to the increased complexity of MAT261, the numerical efficiency is a lot lower, causing a computation time that is nearly three times higher compared to material models MAT054 and M058 in coupon tension models.

With the use of the coupon model, the first artificial neural network surrogate model was built. For the coupons a design space was built, with three distinct layups and three different coupon thicknesses creating a dataset of 9 simulations. The resulting ultimate load and failure displacement were extracted and formed the train labels for the surrogate model. Subsequently, a small network with Rectified Linear Units was built and trained, giving accurate results in testing when interpolating within the design space.

STUNNING Fuselage Modelling
The Fuselage Demonstrator subfloor section, often named the STUNNING fuselage (section) in this report, was introduced. The fuselage section is modelled with a slightly simplified version compared to its real-life counterpart, omitting small parts and details that do not contribute to the crashworthiness behavior. Three new models were derived from here, which were reduced in size by making use of the symmetry and by removing parts of the structure that had their stiffness dominantly faced in the perpendicular direction from the crushing force. The reason that these models were investigated came from the interest to create a less computationally expensive training dataset for the surrogate model.

The two models that removed the parts entirely showed very different behavior to the original model, and were abandoned as a consequence. The model that reduced the length of the fuselage section in longitudinal direction by making use of symmetry showed very promising similarity in crushing behavior, and had a similar mean crushing force and total energy absorption.

Sensitivity Analyses
The Design space for the artificial neural network surrogate models was investigated through a sensitivity analysis of the design space dimensions on the results of the total energy and crushing force of the structure. Both design space dimensions were focused on changing the geometry of the C-strut. Furthermore, a sensitivity analysis was done on the influence of the number of CPUs per simulation on the results of the original STUNNING fuselage section and the computing time. From the sensitivity analysis, it was seen that the total energy and crushing forces differ measurably when plotting them. Additionally, a failure mode of the fuselage section was discovered which is suspected to be caused by placing the bottom of the C-struts too far outward. In an attempt to exclude this failure mode, the design space was limited a bit more than in the simulations performed for the sensitivity study. In the CPU sensitivity study, it was seen that the simulation results are highly inconsistent when comparing them between different CPU parallelization strategies. To limit the effect of this variance the decision was made to fix the number of CPUs used per simulation. Also, the number of CPUs that were chosen took into consideration the available resources in the High Performance Cluster used for this thesis. Finally, the train target was determined, which was the total energy of the simulation. With that, all the elements needed for the artificial neural network surrogate model were in place.

STUNNING Artificial Neural Network Surrogate Models
Finally, in chapter 6 of the thesis, an artificial neural network was built that could predict the total energy of a 30 ms crushing simulation of the STUNNING fuselage subfloor section model. The design space was sampled twice using optimal Latin hypercube sampling, creating two batches of data containing 60 samples per batch. Three types of feedforward neural networks were developed, each differentiating itself by a different neuron function used. These were the rectified linear units, the sigmoid networks, and the Gaussian error networks. The effects of using more or less training data, the architecture of the network, and the effect of using smaller and bigger validation datasets were investigated. The most important observation made through this analysis is that with the doubling of training data, no significant change was seen for the accuracy of the model. Finally, a root mean square error was achieved on the largest training dataset of 770 J which corresponds to a disrepancy of around 10% in the worst case. The applicability of this methodology therefore limits itself to early design explorations of design spaces of crashworthy structures, especially in situations where the relation between design variables and the resulting performance are unknown.

References

[1] Transport Aircraft Crashworthiness and Ditching Working Group. *Transport Aircraft Crashworthiness and Ditching Working Group Report to FAA*. Release/Revision B. Federal Aviation Administration, 2018.

[2] Federal Aviation Administration. *14 CFR Part 25 Subpart C 'structures' - Airworthiness Standards: Transport Category Airplanes*. Federal Aviation Administration.

[3] Po Xue, L. Ding, F. Qiao, and T. Yu. "Crashworthiness Study of a Civil Aircraft Fuselage Section". In: *Latin American Journal of Solids and Structures* 11 (Jan. 2014), pp. 1615–1627. DOI: `10.1590/S1679-78252014000900007`.

[4] A.M. Eiband. *Human tolerance to rapidly applied accelerations: a summary of the literature*. Memorandum 19980228043. NASA, 1959.

[5] C. Lawrence, E.L. Fasanella, A. Tabiei, J.W. Brinkley, and D.M. Shemwell. *The use of a vehicle acceleration exposure limit model and a finite element crash test dummy model to evaluate the risk of injuries during Orion crew module landings*. Technical Memorandum 20080018587. NASA, 2008.

[6] M. Waimer, T. Feser, P. Schatrow, and D. Schueler. "Crash concepts for CFRP transport aircraft – comparison of the traditional bend frame concept versus the developments in a tension absorbers concept". In: *International Journal of Crashworthiness* 23.2 (2018), pp. 193–218. DOI: `10.1080/13588265.2017.1341279`.

[7] M. Guida, F. Marulo, and S. Abrate. "Advances in crash dynamics for aircraft safety". In: *Progress in Aerospace Sciences* 98 (2018), pp. 106–123. ISSN: 0376-0421. DOI: `https://doi.org/10.1016/j.paerosci.2018.03.008`.

[8] H. Mou, J. Xie, and Z. Feng. "Research status and future development of crashworthiness of civil aircraft fuselage structures: An overview". In: *Progress in Aerospace Sciences* 119 (2020). ISSN: 0376-0421.

[9] Z. Yu, X. Zhou, X. Zhou, Y. Zhang, and Q. Zhu. "Crashworthy subfloor structure of civil aircraft via inclined inward-folding composite tubes". In: *Composites Part B: Engineering* 189 (2020), p. 107887. ISSN: 1359-8368. DOI: `https://doi.org/10.1016/j.compositesb.2020.107887`.

[10] G. Zhu, G. Sun, H. Yu, S. Li, and Q. Li. "Energy absorption of metal, composite and metal/composite hybrid structures under oblique crushing loading". In: *International Journal of Mechanical Sciences* 135 (2018), pp. 458–483. ISSN: 0020-7403. DOI: `https://doi.org/10.1016/j.ijmecsci.2017.11.017`.

[11] W. An, S. Wang, and X. Han. "Crashworthiness Optimization Design of Regional Airliner's Fuselage Section Through Topometry Optimization". In: *AIAA Journal* 59.11 (2021), pp. 4754–4763. DOI: `10.2514/1.J060049`.

[12] C. Jiang, K. Wang, D. Cui, L. Ren, and Y. Hu. "Study on the Relationship between Intrusion Velocity and Occupant Injury of Side Impact". In: *2017 9th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA)*. 2017, pp. 324–326. DOI: `10.1109/ICMTMA.2017.0084`.

[13] Federal Aviation Administration. *Transport Airplane Cabin Interiors Crashworthiness Handbook*. AC 25-17A. Aircraft Certification Service, 2009.

[14] Federal Aviation Administration. *Special Conditions: Airbus A350-900 Airplane; Crashworthiness, Emergency Landing Conditions*. 25–537–SC. 2014.

[15] Federal Aviation Administration. *Special Conditions: Boeing Model 787-8 Airplane; Crashworthiness*. 25–362–SC. 2007.

[16] Federal Aviation Administration. *Advisory circular, Composite Aircraft Structure*. AC 20–107B. AIR-100, 2010.

[17] Z. Feng, J. Xie, S. Song, H. Mou, X. Su, and X. Zhang. “The failure mechanism and energy-absorbing characteristics of composite thin-walled C-channels subject to low-speed axial compression.” In: *Journal of Composite Materials* 53(16) (2019). ISSN: 1530-793X.

[18] M. Waimer. “Development of a Kinematics Model for the Assessment of Global Crash Scenarios of a Composite Transport Aircraft Fuselage”. PhD thesis. Universität Stuttgart, Jan. 2013, p. 276.

[19] H. Mou, J. Xie, Y. Liu, K. Cheng, and Z. Feng. “Impact test and numerical simulation of typical sub-cargo fuselage section of civil aircraft”. In: *Aerospace Science and Technology* 107 (2020), p. 106305. ISSN: 1270-9638. DOI: https://doi.org/10.1016/j.ast.2020.106305.

[20] S Hashemi and AC Walton. “A systematic approach to aircraft crashworthiness and impact surface material models”. In: *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering* 214 (May 2000), pp. 265–280. DOI: 10.1243/0954410001532051.

[21] L. Weiß and C. Hühne. “Experimental investigation on the deformation of CFRP cylinders using piezo-actuators”. In: *Composite Structures* 132 (Nov. 2015), pp. 944–951. DOI: 10.1016/j.compstruct.2015.07.002.

[22] K.E. Jackson and J.B. Putnam. *Simulation of a Full-Scale Crash Test of a Fokker F28 Fellowship Aircraft*. Technical Memorandum 20200002017. NASA, 2020.

[23] R.Anyoha. *The History of Artificial Intelligence*. 2017. URL: https://sitn.hms.harvard.edu/flash/2017/history-artificial-intelligence/ (visited on Nov. 26, 2022).

[24] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[25] J. Akst. *A Primer: Artificial Intelligence Versus Neural Networks*. 2019. URL: https://www.the-scientist.com/magazine-issue/artificial-intelligence-versus-neural-networks-65802 (visited on Nov. 27, 2022).

[26] T. Mitchell. *Machine Learning*. ISBN 0070428077. McGraw Hill, 1997.

[27] A. Khan, D.-K. Ko, S. C. Lim, and H. S. Kim. “Structural vibration-based classification and prediction of delamination in smart composite laminates using deep learning neural network”. In: *Composites part B* 161 (2019). ISSN: 1359-8368.

[28] X. Glorot, A. Bordes, and Y. Bengio. “Deep Sparse Rectifier Neural Networks”. In: *AISTATS*. 2011.

[29] Y. Upadhyay. *Introduction to FeedForward Neural Networks*. 2019. URL: https://towardsdatascience.com/feed-forward-neural-networks-c503faa46620 (visited on Nov. 11, 2021).

[30] J. Weng Z. Jin S. Zhang. “Immune particle swarm optimization of SUV rollover crashworthiness and weight”. In: *Structural and Multidisciplinary Optimization 64* (2021), pp. 1161–1174. DOI: https://doi.org/10.1007/s00158-021-02906-2.

[31] M. Nielsen. *Neural Networks and Deep Learning*. Determination Press, 2015.

[32] D.W. Abueidda, S. Koric, and N.A. Sobh. “Topology optimization of 2D structures with nonlinearities using deep learning”. In: *Computers & Structures* 237 (2020), p. 106283. ISSN: 0045-7949. DOI: https://doi.org/10.1016/j.compstruc.2020.106283.

[33] A. Chandrasekhar and K. Suresh. “TOuNN: Topology Optimization using Neural Networks”. In: Structural and Multidisciplinary Optimization, 2020.

[34] I. P. Rosinha, K.V. Gernaey, J.M. Woodley, and U. Krühne. “Topology optimization for biocatalytic microreactor configurations”. In: *12th International Symposium on Process Systems Engineering and 25th European Symposium on Computer Aided Process Engineering*. Ed. by Krist V. Gernaey, Jakob K. Huusom, and Rafiqul Gani. Vol. 37. Computer Aided Chemical Engineering. Elsevier, 2015, pp. 1463–1468. DOI: https://doi.org/10.1016/B978-0-444-63577-8.50089-9. URL: https://www.sciencedirect.com/science/article/pii/B9780444635778500899.

[35] A.I.J. Forrester, A. Sóbester, and A.J. Keane. *Engineering Design via Surrogate Modelling: A Practical Guide*. https://onlinelibrary.wiley.com/doi/book/10.1002/9780470770801. John Wiley & Sons, Ltd, 2008.

[36] C. Li and I. Y. Kim. “Lightweight Optimal Design of a Rear Bumper System Based on Surrogate Models”. In: 2015.

[37] Abaqus documentation. *Optimal Latin Hypercube Technique*. 2017. URL: `https://abaqus-docs.mit.edu/2017/English/IhrComponentMap/ihr-c-Reference-OptimalLatin.htm` (visited on Nov. 1, 2022).

[38] F.A.C. Viana, G. Venter, and V. Balabanov. “An algorithm for fast optimal Latin hypercube design of experiments”. In: *International Journal for Numerical Methods in Engineering* 82.2 (2010), pp. 135–156. DOI: `https://doi.org/10.1002/nme.2750`.

[39] Z. Jiang and M. Gu. “Optimization of a fender structure for the crashworthiness design”. In: *Materials & Design* 31.3 (2010), pp. 1085–1095. ISSN: 0261-3069. DOI: `https://doi.org/10.1016/j.matdes.2009.09.047`.

[40] L. Lanzi, C. Bisagni, and S. Ricci. “Crashworthiness optimization of helicopter subfloor based on decomposition and global approximation”. In: *Structural and Multidisciplinary Optimization* 27 (2004), pp. 401–410. DOI: `https://doi.org/10.1007/s00158-004-0394-z`.

[41] O. Laban, S. Gowid, E. Mahdi, and F. Musharavati. “Experimental investigation and artificial intelligence-based modeling of the residual impact damage effect on the crashworthiness of braided Carbon/Kevlar tubes”. In: *Composite Structures* 243 (2020), p. 112247. ISSN: 0263-8223. DOI: `10.1016/j.compstruct.2020.112247`.

[42] W. Wang, H. Wang, J. Zhou, H. Fan, and X. Liu. “Machine learning prediction of mechanical properties of braided-textile reinforced tubular structures”. In: *Materials & Design* 212 (2021), p. 110181. ISSN: 0264-1275. DOI: `10.1016/j.matdes.2021.110181`.

[43] Livermore Software Technology Corporation. *LS-DYNA Material Selector*. 2019. URL: `https://www.lstc.com/dynamat/` (visited on Apr. 8, 2022).

[44] S. T. Pinho, L. Iannucci, and P. Robinson. “Physically based failure models and criteria for laminated fibre-reinforced composites with emphasis on fibre kinking. Part II: FE implementation”. In: *Composites Part A: Applied Science and Manufacturing* 37 (5 May 2006), pp. 766–777. ISSN: 1359835X. DOI: `10.1016/J.COMPOSITESA.2005.06.008`.

[45] S. T. Pinho, L. Iannucci, and P. Robinson. “Physically-based failure models and criteria for laminated fibre-reinforced composites with emphasis on fibre kinking: Part I: Development”. In: *Composites Part A: Applied Science and Manufacturing* 37 (1 Jan. 2006), pp. 63–73. ISSN: 1359-835X. DOI: `10.1016/J.COMPOSITESA.2005.04.016`.

[46] M. Osborne, P. Feraboli, and U. Shumlak. “Single-Element Characterization of the LS-DYNA MAT54 Material Model”. 2012.

[47] A. Matzenmiller, J. Lubliner, and R. L. Taylor. “A constitutive model for anisotropic damage in fiber-composites”. In: *Mechanics of Materials* 20 (2 Apr. 1995), pp. 125–152. ISSN: 0167-6636. DOI: `10.1016/0167-6636(94)00053-0`.

[48] A. S. Kaddour, M. J. Hinton, P. A. Smith, and S. Li. “Mechanical properties and details of composite laminates for the test cases used in the third world-wide failure exercise:” in: *Sage Journal* 47 (20-21 Sept. 2013), pp. 2427–2442. ISSN: 00219983. DOI: `10.1177/0021998313499477`.

[49] Oak Ridge National Laboratory. *Crashworthiness of Carbon Fiber Composites*. URL: `https://energy.ornl.gov/CFCrush/rate_tests/rate_tests.cgi`.

[50] A. Cherniaev, J. Montesano, and C. Butcher. “5 th International LS-DYNA ® Users Conference Modeling the Axial Crush Response of CFRP Tubes using MAT054, MAT058 and MAT262 in LS-DYNA ®”. In: ().

[51] Livermore Software Technology Corporation. *LS-DYNA® KEYWORD USER’S MANUAL*. Version R13-Vol I. 2021. URL: `https://www.dynasupport.com/manuals`.

[52] P. Feraboli. “Crushing Behavior of a Composite Corrugated Specimen Representative of an Aircraft Subfloor: Experiment and Simulation DOT/FAA/AR-11/21 September 2013”. In: (Sept. 2013). DOI: `10.13140/RG.2.1.3860.9128`.

[53] *Toray Cetex® TC1225 - Toray Advanced Composites*. URL: `https://www.toraytac.com/product-explorer/products/gXuK/Toray-Cetex-TC1225`.

[54] D.P. Kingma and J.L. Ba. “ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION”. In: (2015). DOI: `https://doi.org/10.48550/arXiv.1412.6980`.

[55] D. Hendrycks and K. Gimpel. “Bridging Nonlinearities and Stochastic Regularizers with Gaussian Error Linear Units”. In: *CoRR* abs/1606.08415 (2016). arXiv: `1606.08415`. URL: `http://arxiv.org/abs/1606.08415`.

A
Training data and results

Table A.1: Results of training FNNs with 60 samples that make up the Training/Test data together.

Val. split	number of neurons	neuron activation func	training MSE	val. MSE
0.10	10	ReLu	0.039	0.084
0.10	20	ReLu	0.043	0.119
0.10	30	ReLu	0.043	0.117
0.10	40	ReLu	0.030	0.052
0.10	50	ReLu	0.028	0.047
0.10	60	ReLu	0.029	0.068
0.10	70	ReLu	0.026	0.040
0.10	80	ReLu	0.027	0.049
0.10	10	Sigmoid	0.044	0.069
0.10	20	Sigmoid	0.044	0.087
0.10	30	Sigmoid	0.044	0.081
0.10	40	Sigmoid	0.044	0.084
0.10	50	Sigmoid	0.045	0.087
0.10	60	Sigmoid	0.045	0.087
0.10	70	Sigmoid	0.045	0.087
0.10	80	Sigmoid	0.044	0.086
0.10	10	RBF	0.039	0.042
0.10	20	RBF	0.043	0.070
0.10	30	RBF	0.043	0.069
0.10	40	RBF	0.043	0.070
0.10	50	RBF	0.043	0.070
0.10	60	RBF	0.043	0.070
0.10	70	RBF	0.043	0.070
0.10	80	RBF	0.043	0.069

Val. split	number of neurons	neuron activation func	training MSE	val. MSE
0.20	10	ReLu	0.040	0.144
0.20	20	ReLu	0.040	0.108
0.20	30	ReLu	0.034	0.115
0.20	40	ReLu	0.032	0.088
0.20	50	ReLu	0.032	0.112
0.20	60	ReLu	0.030	0.071
0.20	70	ReLu	0.032	0.066
0.20	80	ReLu	0.029	0.136
0.20	10	Sigmoid	0.043	0.084
0.20	20	Sigmoid	0.043	0.084
0.20	30	Sigmoid	0.043	0.085
0.20	40	Sigmoid	0.043	0.086
0.20	50	Sigmoid	0.043	0.085
0.20	60	Sigmoid	0.043	0.085
0.20	70	Sigmoid	0.043	0.085
0.20	80	Sigmoid	0.043	0.085
0.20	10	RBF	0.043	0.107
0.20	20	RBF	0.043	0.115
0.20	30	RBF	0.042	0.096
0.20	40	RBF	0.043	0.114
0.20	50	RBF	0.043	0.108
0.20	60	RBF	0.043	0.120
0.20	70	RBF	0.043	0.120
0.20	80	RBF	0.043	0.114

Table A.2: One layer FNN trained on 120 samples with a constant validation split.

Val. split	number of neurons	neuron activation func	training MSE	val. MSE
0.10	10	ReLu	0.030	0.064
0.10	20	ReLu	0.029	0.053
0.10	30	ReLu	0.028	0.048
0.10	40	ReLu	0.028	0.052
0.10	50	ReLu	0.026	0.047
0.10	60	ReLu	0.027	0.062
0.10	70	ReLu	0.028	0.051
0.10	80	ReLu	0.027	0.049
0.10	10	Sigmoid	0.033	0.066
0.10	20	Sigmoid	0.033	0.065
0.10	30	Sigmoid	0.033	0.064
0.10	40	Sigmoid	0.033	0.065
0.10	50	Sigmoid	0.033	0.065
0.10	60	Sigmoid	0.033	0.065
0.10	70	Sigmoid	0.033	0.064
0.10	80	Sigmoid	0.033	0.065
0.10	10	RBF	0.029	0.052
0.10	20	RBF	0.031	0.052
0.10	30	RBF	0.031	0.053
0.10	40	RBF	0.031	0.053
0.10	50	RBF	0.031	0.052
0.10	60	RBF	0.031	0.052
0.10	70	RBF	0.031	0.053
0.10	80	RBF	0.031	0.052