Escola de Camins
Escola Tècnica Superior d'Enginyeria de Camins, Canals i Ports
UPC BARCELONATECH

TREBALL FINAL DE MÀSTER

# Coupling Uncertainty Quantification of the B-Pillar Stamping Process and the Side Crash using Machine Learning


Treball realitzat per:
**Andino Raymond Börst**

Dirigit per:
**Riccardo Rossi**
**Pedro Diez**


Màster en:
**Mètodes Numèrics en Enginyeria**

Barcelona, 29 de Junio de 2022


Departament d'Enginyeria Civil i Ambiental

**Abstract**

Crash tests in the automotive industry are essential to check a vehicle's safety equipment and are therefore extremely demanding. Numerous variables, that arise due to manufacturing tolerances, cause uncertainty in the results of these highly complex crash tests. With a crash test being so expensive to emulate and simulations being too computationally expensive to run Monte-Carlo simulations on them, other alternatives are needed to quantify this uncertainty in the crash tests. This is where the AQUA algorithm comes in that was initially developed by Dr. Marc Rocas [1]. This algorithm uses different machine learning techniques to quantify the uncertainty of a crash test with the FEM simulations of the given scenario and do a sensitivity analysis for the corresponding parameters. However, the initial version of the algorithm was missing some important features which are addressed in this thesis. In addition to the application on a side crash test of a dummy vehicle, the algorithm is also used on the stamping process of the B-pillar of the vehicle to obtain more details about the thickness mapping of this part. The outcome of the analysis for the stamping process is subsequently coupled with the side crash simulation of the dummy vehicle to include a more realistic representation of the thickness in the side crash model. The improvements made to the AQUA algorithm include a mixture of experts approach, the implementation of different non-linear dimensionality reduction methods and an optimized sampling method. This optimized sampling method uses an iterative cycle approach to enrich the existing data set with specific samples defined by areas of low certainty in the model. The results show that the improvements to the AQUA algorithm have a significant effect on the results. The introduction of clusters greatly improves the prediction capabilities of the algorithm and the new dimensionality reduction methods significantly enhances the amount of information being captured in the reduced space. Additionally, the optimized sampling method is more efficient, as better results can be obtained with fewer simulations and the accuracy of the classification model is improved as well. Consequently, the analysis of the stamping process shows that assuming a uniform thickness mapping for the B-pillar is unrealistic and yields inaccurate results, as many thinner and thicker areas are found throughout the part. Lastly, the coupling of the stamping process and the side crash simulation yielded much more detail in the results and uncovered outcomes that were not previously discovered due to the lack of details in the thickness mapping for the B-pillar.

# Contents

# List of Figures

# List of Tables

# Acronyms

**AQUA** Artificial Quantification for Uncertainty Anomalies.

**DR** Dimensionality Reduction.

**Euro NCAP** European New Car Assessment Programme.

**FEM** Finite Element Method.

**GBoost** Gradient Boosting.

**HAC** Hierarchical Agglomerative Clustering.

**IIHS** Insurance Institute for Highway Safety.

**kMC** k-Means Clustering.
**KNN** K Nearest Neighbours.
**kPCA** Kernel Principal Component Analysis.

**MC** Monte Carlo.
**MDS** Multidimensional Scaling.
**MoE** Mixture of Experts.

**OK** Ordinary Kriging.

**PC** Principal Component.
**PCA** Principal Component Analysis.

**QoI** Quantity of Interest.

**RF** Random Forest.
**RSE** Relative Squared Error.

**SM** Surrogate Model.
**SMACOF** Scaling by MAjorizing a COmplicated Function.

**t-SNE** t-Distributed Stochastic Neighbor Embedding.

**UI** User Interface.
**UMAP** Uniform Manifold Approximation and Projection.
**UQ** Uncertainty Quantification.
**USA** United States of America.

# Glossary

**AQUA**

**H** Set of input parameters.
**h** Input parameters.
**X** Set of output data.
**x** Output data.
**Z** Set of output data in reduced space.
**z** Output data in reduced space.
$\mu$ Mean.
$n_{data}$ Number of data points.
$n_{param}$ Number of parameters.
$n_x$ Dimension of original output data.
$n_z$ Dimension of reduced output data.
$\sigma$ Standard deviation.

**UMAP**

$a$ Parameter 1 defining the minimum distance and spread between the low dimensional points.
$b$ Parameter 2 defining the minimum distance and spread between the low dimensional points.
$C$ Cross entropy to be minimized.
$k$ Number of KNN.
$N$ Number of points in the data set.
$\rho$ Distance to the nearest neighbour.
$\sigma$ Smoothed normalisation factor.
$v$ Weights of the low dimensional space.
$w$ Weight.
$x$ Point.
$y$ Low dimensional point.

**Hierarchical Agglomerative Clustering**

$c$ Centroid of a cluster **X**.
$D$ Distance measure.
$\Delta$ Similarity measure.
**X** Set of points **x** forming a cluster.
**x** Point.

**Random Forest Classifier**

$G$ Gini index for a node in a decision tree.
$G_{rating}$ Gini rating for a node split in a decision tree.
$m$ Number of points used to train one decision tree in the RF.

$n_p$ Number of points in the whole data set of the RF.
$p$ Probability.
$t$ A node in a decision tree.

**Multidimensional Scaling**

$\mathbf{A}_{ij}$ $n \times n$ matrix with all zeros except entries $a_{ii} = a_{jj} = 1$ and $a_{ij} = a_{ji} = -1$.
$\mathbf{B}$ Matrix with entries defined by $b_{ij}$.
$b_{ij}$ Entries to matrix $\mathbf{B}$ defined as $w_{ij}d_{ij}/\hat{d}_{ij}(\mathbf{X})$.
$\hat{\mathbf{B}}$ Matrix with entries defined by $\hat{b}_{ij}$.
$\hat{b}_{ij}$ Entries to matrix $\hat{\mathbf{B}}$ defined as $w_{ij}d_{ij}/\hat{d}_{ij}(\mathbf{Y})$ if $\hat{d}_{ij}(\mathbf{Y}) > 0$ and 0 if $\hat{d}_{ij}(\mathbf{Y}) = 0$.
$d$ Euclidean distance between two points in the original full dimension.
$\hat{d}$ Euclidean distance between two points in the reduced dimension.
$\mathbf{e}$ Column of the identity matrix $\mathbf{I}$.
$g$ Majorizing function.
$\mathbf{I}$ Identity matrix.
$n$ Number of points in the space.
$\eta_d^2$ Total dispersion.
$\eta_{\hat{d}}^2$ Reconstructed dispersion.
$\rho$ Codispersion.
$\sigma$ Stress function.
$\sigma_{raw}$ Raw stress function.
$t$ Dimensions of the reduced space.
$V$ Matrix defined as $\sum_{i=1}^{n}\sum_{j=i+1}^{n} w_{ij}\mathbf{A}_{ij}$.
$w$ Non-negative weight.
$\mathbf{X}$ $n \times t$ matrix with the rows representing the coordinates of the points $\mathbf{x}$.
$\mathbf{x}$ Coordinates of a point in the reduced dimension.
$\mathbf{Y}$ Coordinate matrix $\mathbf{X}$ from the previous iteration.

**Iterative Cycle**

$\alpha$ Threshold Value.
$c_r$ Right classification certainty.
$c_w$ Wrong classification certainty.

# 1 Introduction and Motivation

The safety requirements on car manufacturers keep increasing at a steady pace to ensure the well-being of all passengers inside the vehicles. To comply with certain government regulations and deliver cars which are as safe as possible to the customers, crash tests are performed. These crash tests include several different scenarios and as can be seen in Figure 1.1, showing statistics of the Insurance Institute of Highway Safety (IIHS) in the United States of America (USA), these safety requirements have had a substantial impact on vehicle crash deaths throughout the years.

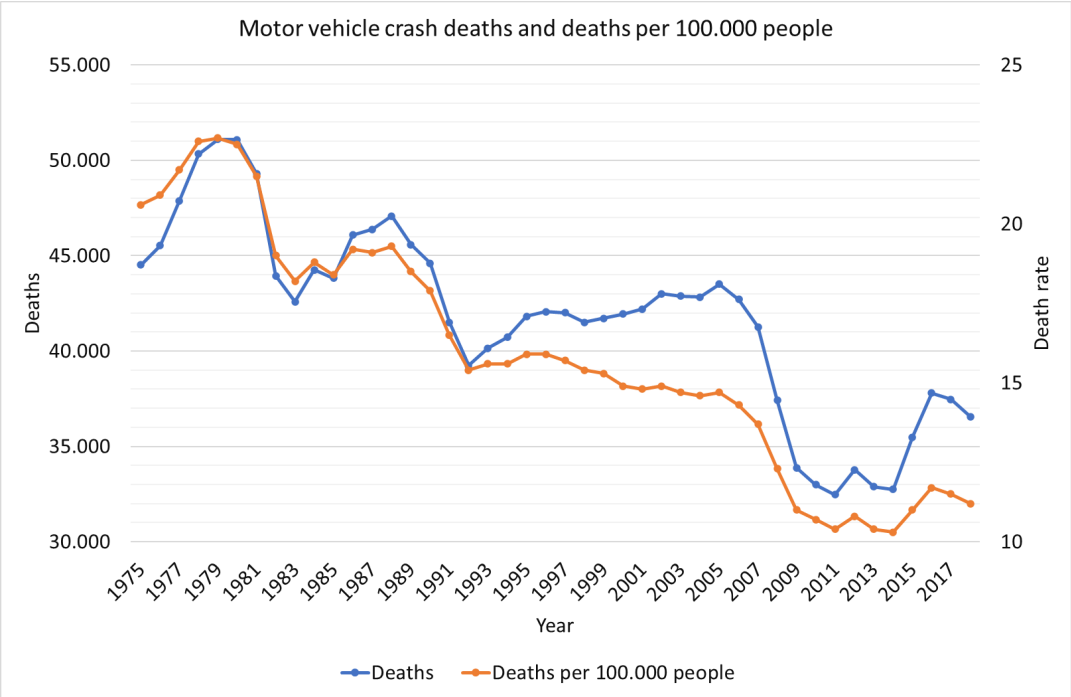



Figure 1.1: IIHS statistics on motor vehicle deaths and deaths per 100.000 people in the USA from 1975-2018. Adapted from [2].

Production and assembly processes in car manufacturing include several variables that can induce uncertainty in the exact outcome of the results. This includes for example the thickness of the B-pillar, which has an acceptable range of tolerance. This tolerance however, in combination with various other parameters that also have a tolerance, can amount to a substantial variance of the final results. Nonetheless, it is essential to have a range of all possible outcomes considering all of the possible parameter variations, as

any outcome needs to be accounted for.

Emulating a crash test is extremely expensive for car manufacturers, which is why the number of crashed vehicles during development is limited to approximately fifty crashes, of which twenty are side crashes, before production starts [7]. For this reason it is essential to be able to generate realistic results with simulations of these crashes. However, while much cheaper and more easily executable, a simulation is still very time consuming and costly, with a single optimized simulation of the side crash lasting more than 6 hours [7], meaning it is not possible to run a simulation for every possible combination of parameters. As an example, when considering seven parameters and only picking three values for each parameter instead of replicating the entire variable distribution, it would take 2187 simulations, lasting more than 13122 hours, to obtain a result for all possible combinations. Because of the fact that it is unfeasible to run so many simulations due to computational time and cost, it is necessary to develop a different method to quantify the uncertainty of a simulation and generate new results for different parameter combinations faster using Surrogate Models (SMs).

The side crash is one of the most important safety measures, as side crashes account for about a quarter of the vehicle deaths in the USA. According to the IIHS the difference in the survival rate of a driver in a vehicle with a good and a poor rating is 70 %. This means a driver in a vehicle with a poor IIHS side crash rating is 70 % more likely to die than a driver in a vehicle with a good rating. [10]
Moreover, the European New Car Assessment Programme (Euro NCAP) states that "side crashes account for the second highest frequency of death and serious injuries" [4]. These numbers clearly show how important it is for car manufacturers to achieve a good rating in this test, as it is one of the most decisive indicators of a vehicles safety.

One of the criteria to assess the results of a side crash is the intrusion of the B-pillar into the occupant compartment. While some intrusion is unavoidable, it should not be significant enough to endanger any passenger [10]. When analysing the side crash simulation at SEAT S.A. it was found that the thickness of the B-pillar plays a significant role in its deformation and intrusion into the passenger space [7]. For this reason it is desirable to have a realistic representation of the B-pillar thickness throughout the entire part, which is not uniform as is often assumed during simulations of the side crash. The thickness of the B-pillar in reality is not uniform, but instead shows a mapping of different thicknesses throughout. This is mainly due to the stamping process in which the B-pillar is produced. To include an adequate representation of the B-pillar thickness in the side crash simulation, it is therefore necessary to realistically simulate the B-pillar stamping process beforehand.

## 1.1 Problem Description

The objective of this thesis is to analyse methods already used at SEAT S.A. to quantify the uncertainty and generate SMs of FEM simulations, as well as finding and implementing new techniques to improve already existing methods. Moreover two simulation processes are to be coupled using these uncertainty quantification and SM techniques. The objective is to be able to have a more realistic representation of the B-pillar in the side crash simulation using the results of the stamping simulation. The procedure of this thesis can be seen in the flow chart in Figure 1.2. The first step is to examine the current implementation of the uncertainty quantification algorithm used by SEAT S.A. called Artificial Quantification for Uncertainty Anomalies (AQUA), which was developed by Dr. Marc Rocas [1]. This algorithm will be improved upon to include various new machine learning techniques to achieve better results. Additionally, an automated cycle algorithm will be implemented to define the new simulations to be generated, which are required to train the machine learning algorithms used in AQUA. This is done to maximize the amount of information conceived in as few simulations as possible. It is necessary to validate these changes to the algorithm and test the new automated cycle technique to identify the optimal machine learning techniques and their parameters. In parallel the FEM model of the stamping process is analysed, to be able to apply the AQUA algorithm on that simulation. The real-life manufacturing process of the B-pillar stamping is also analysed to identify which parameters of the process have a variation and how big their impact is on the final result. This information is necessary to perform the uncertainty quantification of the stamping process, as the distribution of the varying parameters needs to be known. After having validated and tested the modified algorithm and analysed the stamping process the algorithm can finally be applied to the simulation of the stamping process. This is done to quantify the uncertainty in the distribution of the thickness and also generate a SM of the stamping process, which will then be coupled to the side crash simulation. The following chapter focuses on coupling the results of the stamping process to the simulation of the side crash to be able to specify parameters of the stamping process and immediately generate results of the side crash with a more realistic mapping of the B-pillar thickness. Additionally, a chapter containing information on the practical implementation of the AQUA algorithm is included, which showcases the interface and functionality of the code. The final chapter concludes the work done in the thesis and gives an outlook into other potential developments in the future.

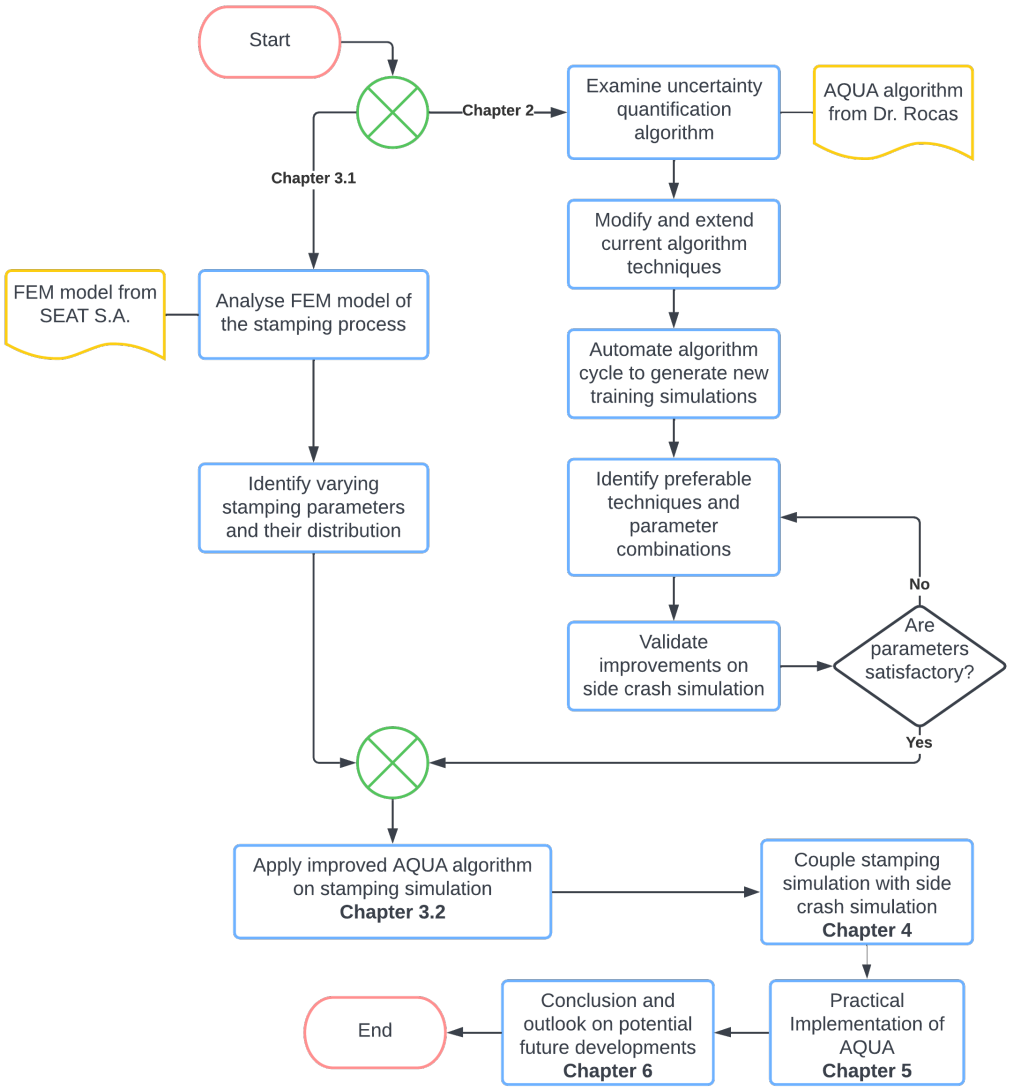



Figure 1.2: Flow chart of the procedure for this thesis.

## 1.2 State-of-the-Art in Uncertainty Quantification

Computer models, such as the simulations treated in this thesis, are deterministic in nature, yielding the same result when run over and over again. However this is not in line with physical experiments which include various factors that affect the outcome and result in different observations when being executed multiple times. Since it is not feasible to run so many physical tests it is desired to include this random error of physical experiments in the computational models. This process is called Uncertainty Quantification (UQ) [11]. The nature of uncertainty is itself distinguished in epistemic uncertainty and variability uncertainty, with the discussed use cases falling into the later. Epistemic

uncertainty is due to a lack of knowledge, while variability uncertainty is due to the implicit variability within a system [12].

Uncertainties in computer models can be categorized into different sources of uncertainty, namely parameter uncertainty, model inadequacy, residual variability, parametric variability, observation error and code uncertainty. The cases considered in this thesis fall into the category of parametric variability, where the input parameters are very well defined and their variability is known and can therefore be introduced into the computer model. The most common approach for tackling such a problem is the Monte Carlo MC simulation. However, this method cannot be implemented, if the cost to obtain a result of the computer model is extremely large, as many results are necessary for this approach to be effective. While a Modified MC analysis using Latin Hypercube sampling is more efficient than the standard MC analysis, it still requires many data points to perform adequately. [13]

Uncertainty propagation, which deals with the quantification of uncertainties in the output, compared to the inverse assessment of uncertainty, where given some experimental measures the inputs are calibrated, can be executed using various different methods [14]. As mentioned the MC simulation is the most common approach, but other methods are used in the industry as well. One alternative UQ method is the so-called Polynomial Chaos Expansion. This approach approximates the correlation between the inputs and the outputs with polynomials. Once an approximation is found it can even be used as a SM to calculate the results of new parameter combinations. However, the main problem of the Polynomial Chaos Expansion is that with an increased number of input parameters and output values the number of polynomials and their coefficients for the approximations increases exponentially. This can result in over-fitting and instabilities in the predictions. [14]

For this reason a combinatorial approach is used where a SM is trained using machine learning models to be able to generate enough samples for a Modified MC analysis. Even though a MC analysis is a very basic approach, convergence is guaranteed with an increasing number of samples, since the problem scales linearly and is applicable to any kind of problem type.

With UQ also comes the desire to analyse the sensitivity of the model. This is done to understand the correlation and the effects of the input on the output. Once the variations of the output are known it is desirable to understand which input parameters have the biggest impact on the results. Since variance based models, like the Sobol method, can take into account the entire input space and quantify multiple order interactions between parameters as well as non-linear responses it is the method of choice for this application. Even though it is usually compromised by a high computational cost, due to the use of the MC method with a SM this is made feasible.

All of this information gives a better understanding of the variability and the robust-

ness of the computational model and the real problem as well as a better understanding of the relation between the input and the output. This can be used in several applications, like for example identifying critical input parameters and ranges of values that should be avoided during the real-life process and the probability of an outcome behaving in a certain manner.

# 2 Artificial Quantification for Uncertainty Anomalies

The AQUA algorithm was developed by Dr. Marc Rocas and is used to quantify the uncertainty of a simulation, by measuring the influence of certain parameters and their sensitivity on the final outcome [1]. This is very important, as a vehicle model consists of many different components, which are all subject to production errors and tolerances that are not usually accounted for in the simulations. In addition to being able to quantify the uncertainty of a simulation the algorithm also defines a SM. This SM can be used to generate results for any new combination of parameters significantly faster and cheaper than running a full simulation, usually within seconds rather than hours. In fact it is possible to generate new results dynamically, allowing the user to see how the changes in a parameter affect the results immediately.

The first part of this chapter will focus on describing the fundamentals of the AQUA algorithm to give an introduction what the further work is based on. This part gives a short overview of the methods already implemented by Dr. Marc Rocas in [1]. It is worth noting that the entire code for the AQUA algorithm was created from scratch for this thesis, based on the information in [1]. To showcase the AQUA algorithm and its capabilities an example case is presented, which analyses the B-pillar in the side crash of a dummy car. Consequently the changes and improvements made to the algorithm will be explained and compared to the initial analysis of the side crash. A conclusion is given at the end, rounding out all of the improvements made to the algorithm and giving final recommendations on the application of the new version of AQUA.

## 2.1 The Algorithm

This section serves as an introduction to the AQUA algorithm and is fully based on the work done by Dr. Marc Rocas in [1].

The algorithm uses various machine learning techniques to achieve the desired results, in particular supervised and unsupervised learning techniques. The goal of the AQUA algorithm is to analyse an initially large dimensional data set related to a specific input array. This means the algorithm takes in two data sets, a matrix $\mathbf{H}$ describing the set of inputs $\mathbf{h}$ and a matrix $\mathbf{X}$ containing the set of output data $\mathbf{x}$ resulting from the input description. The inputs $\mathbf{h}$ can consist of any number of parameters, that influence the

results in the output $\mathbf{x}$. The process used to obtain $\mathbf{x}$ given $\mathbf{h}$ can be anything, although in this case it is a numerical simulation.

The input description $\mathbf{H}$, consisting of many different parameter combinations $\mathbf{h}$, and the output description $\mathbf{X}$, consisting of many different outputs $\mathbf{x}$, can therefore be described as matrices of the following form:

$$\begin{aligned} \mathbf{H} &= [\mathbf{h}_1, ..., \mathbf{h}_{n_{data}}] \in \mathbb{R}^{n_{param} \times n_{data}}, \\ \mathbf{X} &= [\mathbf{x}_1, ..., \mathbf{x}_{n_{data}}] \in \mathbb{R}^{n_x \times n_{data}}, \end{aligned} \tag{2.1}$$

with $n_{param}$ being the number of parameters for each data point, $n_{data}$ being the number of data points and $n_x$ being the size of the initial output data.

The set of different parameter combinations in $\mathbf{H}$ are to be analysed with their respective results $\mathbf{X}$, to obtain the uncertainty in the output due to each parameter. Additionally, a SM is created in the process. This SM is able to generate a result $\mathbf{x}$ for any new parameter combination $\mathbf{h}$ instantly, without having to go through the original process, in this case running a simulation. A flow chart displaying the whole process of the AQUA algorithm is shown in Figure 2.1.

Notably, the input description $\mathbf{H}$ is defined using a low-discrepancy sequence to ensure the entire input space is covered evenly, which would not be guaranteed when choosing the inputs randomly, and also results in faster convergence rates than the conventional MC method [15]. The low-discrepancy method used for the AQUA algorithm is the Halton sequence, which is a reproducible sequence. This means every time the algorithm is run the same sequence of samples will be generated, guaranteeing comparable results when running the algorithm with different methods and also allowing for a continuation of the series. When defining the input description it is important to use a uniform distribution of the input parameters, and not the real one. This is because the input description needs to contain information about as much of the potential input space as possible for the algorithm to discover and learn all possibilities, even if some combinations are very unlikely. The initial input description is extraordinarily important to the final outcome of the algorithm, as this is what is responsible for how well the machine learning algorithms are trained and how diverse the information is in the initial data. This is why one important improvement made to the algorithm, described in Section 2.6, deals with optimizing the initial input description $\mathbf{H}$.

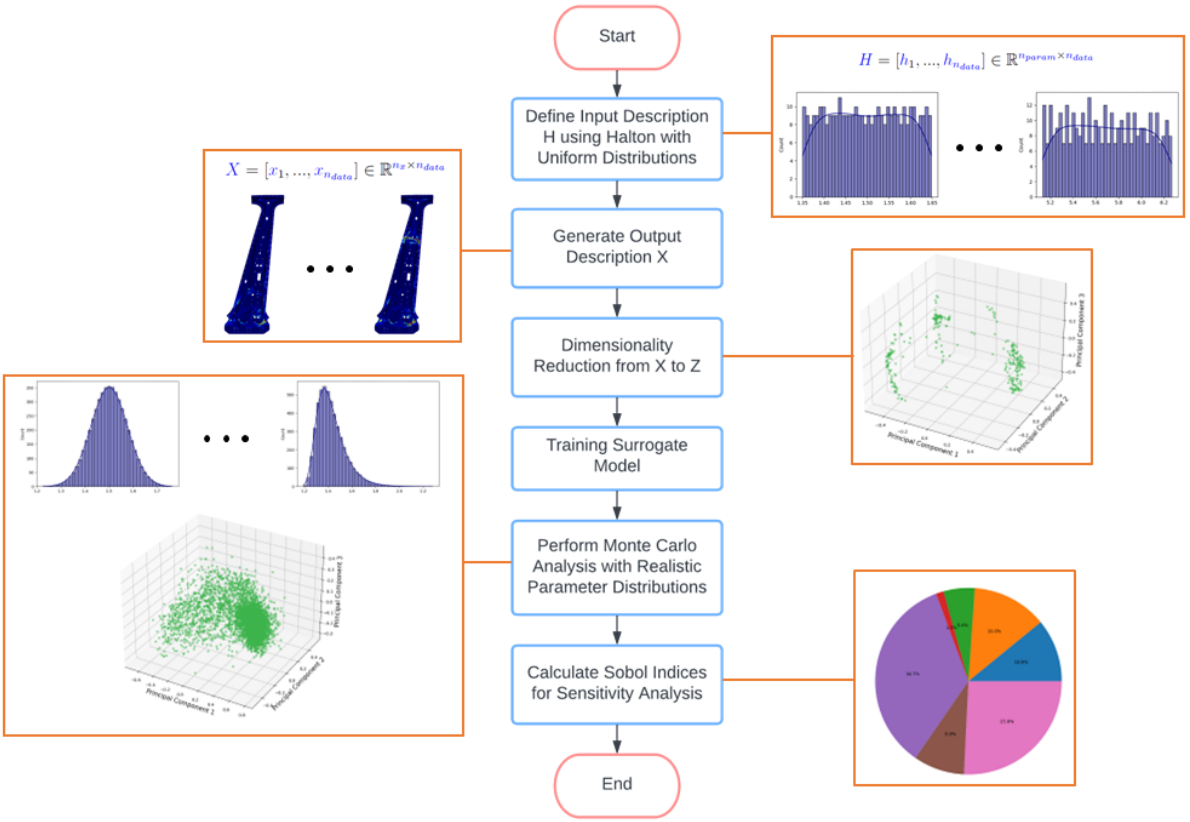



Figure 2.1: Flow chart of the AQUA algorithm.

The first step of the process is to apply a Dimensionality Reduction (DR) method on the given output data $\mathbf{X}$ and reduce it to $n_z$ dimensions. DR is an unsupervised machine learning technique, as it aims at recognizing patterns in the given data, with no additional information. The output data of the cases to be considered is of very high dimension and cannot be processed in the given form. Additionally, for this use case it is desirable to be able to visualize the data, which is why the original output dimension $n_x$ is reduced to two or three dimensions. Three dimensions are usually preferable, as this allows more information to be captured within the reduced space. Analysing data in a reduced space is very functional, as it simplifies the detection of clusters, outliers and general trends in the data. The new set of data in the reduced space representing the information of the output data is given as:

$$\mathbf{Z} = [\mathbf{z}_1, ..., \mathbf{z}_{n_{data}}] \in \mathbb{R}^{n_z \times n_{data}}. \tag{2.2}$$

The transformation of the data given in the full space $\mathbf{x}$ to the data in the reduced space $\mathbf{z}$ can be written as:

$$\mathbf{z} = \mathcal{G}(\mathbf{x}), \tag{2.3}$$

where $\mathcal{G}$ is the DR technique. The backward mapping, to obtain an original full dimensional result from data given in the reduced space, is stated as:

$$\mathbf{x} = \mathcal{G}^{-1}(\mathbf{z}). \tag{2.4}$$

Methods used to perform DR in the algorithm are Principal Component Analysis (PCA) and kernel Principal Component Analysis (kPCA), with other new methods being introduced in Section 2.5. PCA is a linear DR technique, which although being simple to implement, has difficulties in capturing information in highly non-linear data sets. Introducing a kernel to this method helps to combat this problem, although it does not solve it completely, as the amount of non-linear information that can be captured is highly dependent on the kernel that is used, and the fact that the kernel is able to represent the original data correctly. This means kPCA is not a universally applicable non-linear DR technique. For this reason new methods to capture very non-linear data sets are introduced to improve the representation of data in the reduced space, which has a very significant impact on the results of the following steps in the algorithm as well.

After transforming the output data into the low dimensional space the SM can be trained using this data. The SM is a supervised machine learning technique, also called regression model, using an input to predict a certain output. The three methods used in this step are Ordinary Kriging (OK), Gradient Boosting (GBoost) Regressor and Random Forest (RF) Regressor. Other regression models could be used, however it is important to point out that in the cases considered in this thesis the data is very non-linear, which is why the mentioned models were chosen. OK is an exception, as it is not entirely a non-linear regression method, but instead gives the best linear unbiased prediction, which is useful to estimate random effects [16, 17]. Throughout this thesis the GBoost Regressor was used.

Following the training of the SM a (MC) analysis can be performed, as now thousands of simulation results can be generated extremely fast and cost efficient. To be more precise a quasi-Monte Carlo method is used, as the samples are not defined randomly, but instead using the low-discrepancy Halton sequence, similarly to the definition of the input description $\mathbf{H}$. In this step it is important to have knowledge about the input parameter distributions, as this is essential for a realistic assessment of the problem. Unlike the input description, where a uniform distribution for each parameter was assumed, in this step the real parameter distributions are used to obtain a realistic representation of possible results.

In addition to reducing the dimensionality for analysis purposes, it is of great interest to also include an inverse transform of the DR, from $\mathbf{z}$ to $\mathbf{x}$, to obtain results in the original dimension. This is especially important when generating new results with the SM to be able to visualize the obtained information and analyse new parameter combinations which had not been simulated before. Interpreting the results of the SM directly is not suitable, as it yields information only in the reduced space. Since not every DR technique supports backward mapping to obtain an inverse transform a different method is used. Using all of the initial data the newly generated data point is interpolated.

This is done by multiplying the output data $\mathbf{x}$ of each of the initial data points by a specific weight for each of those points. To determine this weight the distance of the new data point to the initial data point is measured. The closer the points are together, the larger the weight for this point is, which ensures that similar data points have the highest influence in the outcome of the inverse transform. This approach is very similar to the K Nearest Neighbours (KNN) regression algorithm. A detailed explanation of the technique is found in [1].

The final step of the algorithm is to do a sensitivity analysis to quantify the effect and uncertainty each parameter induces into the results. To do this the Sobol method is used, which is a variance-based sensitivity analysis, on the samples generated during the quasi-MC method. The Sobol indices obtained with this analysis give an indication on how much impact each input parameter has on the final results. When using this method the first order, second order and total order interactions are analysed. First order interactions indicate how much the final outcome changes based on changes to only a single parameter, meaning how much variance a change in only one parameter causes in the result. The second order Sobol indices, similar to the first order indices, are an indication on how much variance in the final results is caused by a change or interaction of two parameters. In theory it would be possible to calculate higher order Sobol indices, however this would be too computationally expensive. The total order indices are the sum of the Sobol indices of all orders and indicate the total influence of a parameters variance on the results. It is important to note that the Sobol indices are calculated with the results in the reduced space, which means they are highly dependent on the reduced space generated by the DR method. [18]

The results gained from the quasi-MC analysis and the Sobol indices allows for a broad and realistic analysis of the problem at hand. The new enriched data set created in the quasi-MC analysis can identify new modes and behaviours in the process to be examined which were not previously discovered or considered. With such a rich amount of information it is possible to find the cause of certain occurrences in real life and relate them to the results of the simulations. It is possible to explore the entire space of achievable outcomes without having to run very long and costly simulations and also identify which parameters have the biggest impact on the results. This information can be used to optimize the production process to make sure that critical parameter combinations are avoided or flawed parts are identified immediately based on a few measurements of the input parameters.

Lastly, to know how many simulations are required to have a space that is rich enough to properly train the surrogate model, the Kullback-Leibler Divergence is calculated. To do this the full number of simulations is considered and compared to the probability distribution of the space with ten simulations less than currently implemented. The input space is increased with Halton points, until the Kullback-Leibler Divergence convergences.

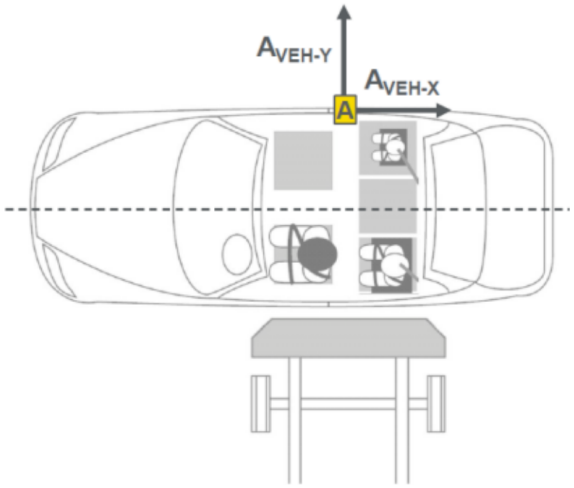
The next section showcases an example where the AQUA algorithm is applied to demonstrate its functionality and results. Subsequently all of the improvements made to the algorithm within the scope of this thesis are presented, including their application on the example model to visualize the improvements and see their direct effect. These improvements include extensions of already implemented ideas, as well as completely new approaches to have more possibilities in the analysis and quantification of uncertainty anomalies.

## 2.2 Example: Analysis of the B-Pillar in the Side Crash of a Dummy Car

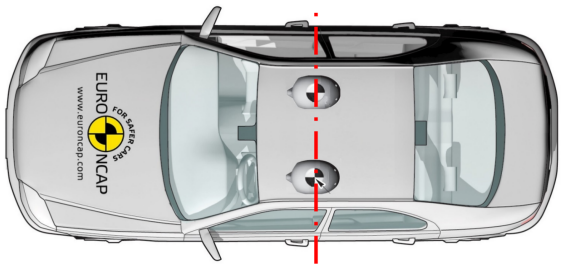
To showcase some results of the AQUA algorithm and display its functionality an example of a side crash simulation for a dummy car is presented. First, a general description of the problem statement is given. Subsequently the results of each step of the algorithm are shown and analysed. The problem to be analysed is a side crash test of a dummy car following the test protocol of the Euro NCAP [19]. As a side note, it is not possible to disclose any specific values regarding the parameter distributions and results due to confidentiality restrictions of SEAT S.A., hence all the confidential data will either be covered or substituted by variables.

### 2.2.1 Problem Description

The Euro NCAP side crash test is employed to test the protection of the passengers during a side impact on a vehicle. The intrusion into the passenger space is the crucial measurement for the ranking. During the test a deformable barrier mounted on a trolley is launched at the side of the vehicle at 60 km/h. The vehicle is stationary during the test and the impact of the barrier is targeted at the side in a right angle. The setup of this crash test and the impact line can be seen in Figure 2.2. [4]




(a) Setup




(b) Impact Line

Figure 2.2: Setup of the Euro NCAP side crash test (a) and the impact line (b) [3, 4].

A Euro NCAP side crash test of the 2020 Seat Leon during and after impact can be seen in Figure 2.3.

(a) During impact

(b) After impact

Figure 2.3: Side crash test of the 2020 Seat Leon during (a) and after (b) impact [5, 6].

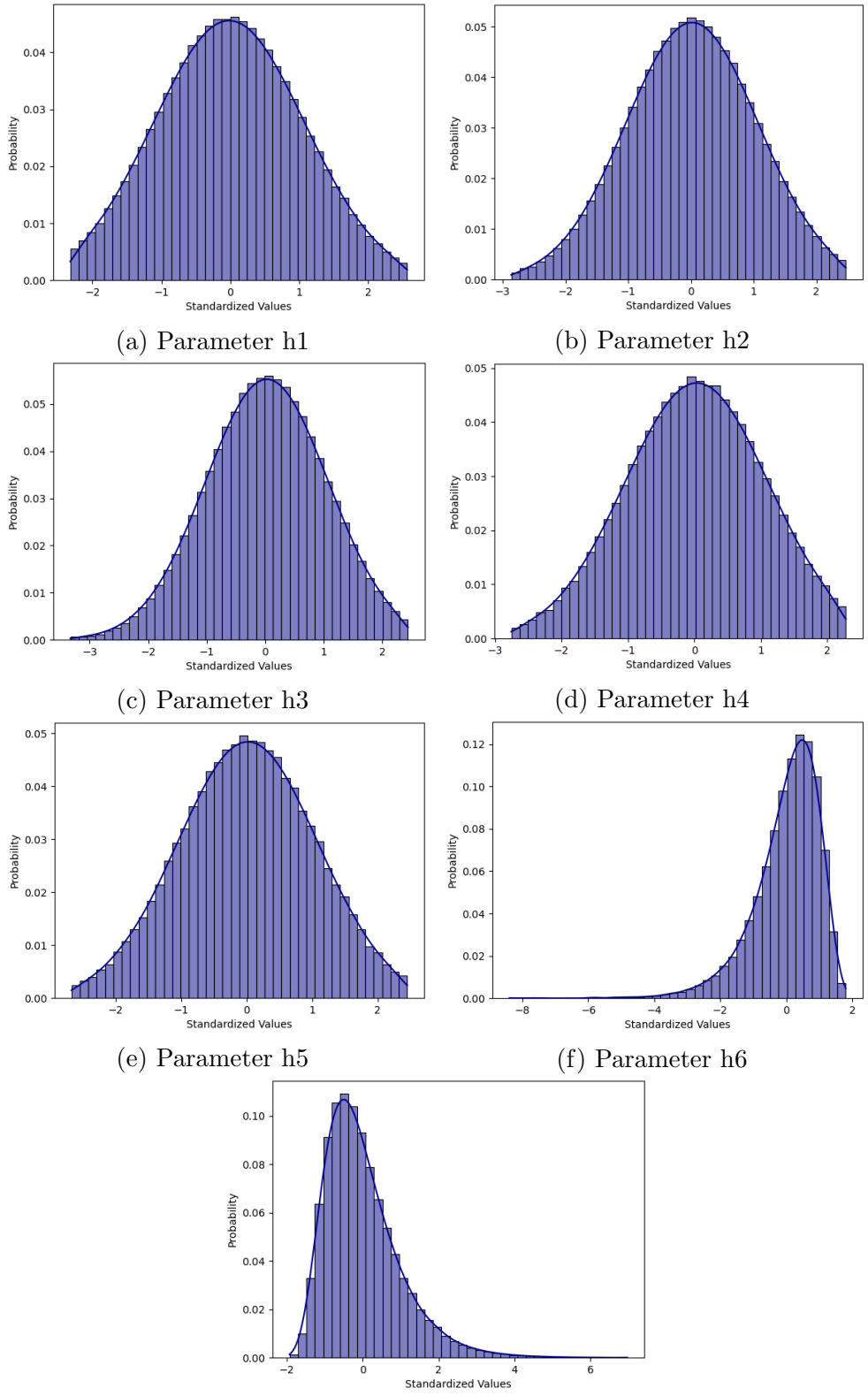
The simulation model of the side crash test contains numerous parameters that have a big impact on the final results. Especially the parameters defining the area around the B-pillar are extremely influential on the outcome of the simulation, as in real life the B-pillar and its surroundings are the most important factor for passing the crash test. As determined by the manufacturing department the most influential parameters that exhibit a certain amount of variation during the production process of the vehicle are shown in Table 2.1 [7]. Unfortunately their acceptable range cannot be disclosed, as this information is confidential and is therefore being standardized or substituted with variables. These are the parameters that will be used in the AQUA algorithm to

quantify the uncertainty of the side crash simulation.

| Parameter | Name | SI-unit |
|---|---|---|
| H1 | Thickness of B-pillar | mm |
| H2 | Thickness of retractor support | mm |
| H3 | Hardness of B-pillar | HV |
| H4 | Horizontal position of impact barrier | mm |
| H5 | Vertical position of impact barrier | mm |
| H6 | Material failure of retractor support | |
| H7 | Bearing factor of welding points | |

Table 2.1: Parameters of the side crash simulation to be varied in the uncertainty quantification analysis [7].

The hardness of the B-pillar, parameter h3, is given as the Vickers-Hardness of the material with the Vickers Pyramid Number (HV) as unit of measure. Parameter h6 modifies values of the material card used to model the retractor support in the simulation, which results in different failure behaviour of the material. A higher value of h6 means the material is more resistant to failure. Parameter h7, which is the bearing factor of the welding points, indicates the failure point and behaviour of the welding points. In this case the lower the value, the stronger the welding points. The exact distributions of the parameters listed in Table 2.1 are shown in Figure 2.4.




(a) Parameter h1

(b) Parameter h2

(c) Parameter h3

(d) Parameter h4

(e) Parameter h5

(f) Parameter h6

(g) Parameter h7

Figure 2.4: Probability distribution function with standardized values of the side crash parameters stated in Table 2.1 [7].

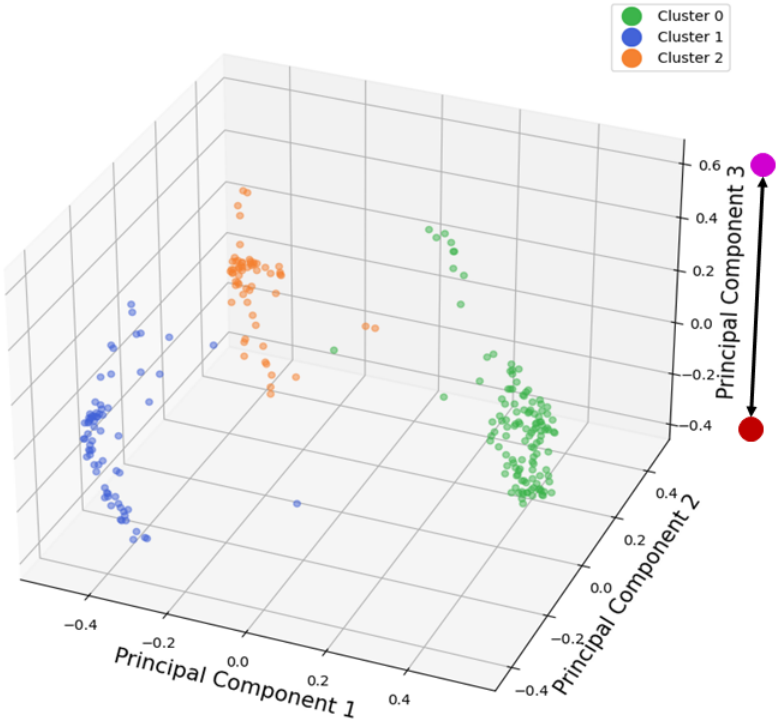
As can be seen, the first five parameters have a normal distribution, while parameter six has an inverted log-normal and parameter seven a log-normal distribution. The mean $\mu$ and standard deviation $\sigma$, as well as the scale and shift values of all the distributions are confidential and can therefore not be disclosed.

To quantify the results of the side crash simulation the plastic strain value of every shell element in the B-pillar at the last time step of the simulation is extracted and used as the output description $\mathbf{x}$. This Quantity of Interest (QoI) was chosen, because the deformation of the B-pillar is the main indicator for the results of the side crash. The dimension of the output description $n_x$ is therefore equivalent to the number of elements used to mesh the B-pillar.

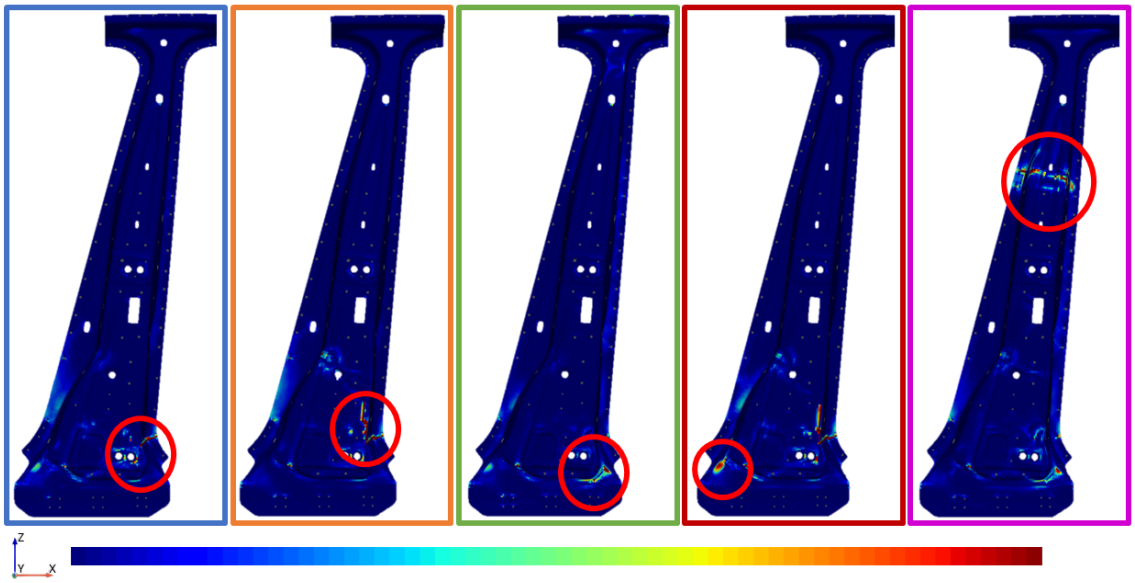
### 2.2.2 Results

The results presented in this section are based on 255 simulations run on parameter combinations obtained with the Halton sequence for a uniform parameter distribution of the values presented in Table 2.1.

The reduced space obtained after applying kPCA in the dimensionality reduction step using the 258 simulation results can be seen in Figure 2.5. Each point in the reduced space represents one simulation result. The location of each point is obtained from the DR of the plastic strain in the B-pillar for the respective simulation. Exact values of the scalar bar are confidential and are therefore not displayed, with blue representing a small plastic strain and red a large plastic strain.




(a) Reduced Space



(b) Plastic Strain on B-pillars

Figure 2.5: Reduced dimension (a) of the side crash model using kPCA and results of the plastic strain on the B-pillar (b). Each color frame around the B-pillar results represents the general behaviour of the results in the cluster with the corresponding color. The last two frames represent the general behaviour of the results depending on the value of PC 3 in combination with the respective cluster behaviour.

The reduced space shown in Figure 2.5a obtained with kPCA clearly shows three main clusters (colored in green, blue and orange) with specific characteristics in each of them. It is important to note that the clusters are only used for visual purposes at this point. Additionally it can be seen that a certain behaviour in the results can be observed along the third Principal Component (PC) axis. Figure 2.5b showcases some of the results found within each cluster with the red circles marking the defining characteristics of the respective clusters. The blue cluster exhibits a cut running from the lower right side toward the two holes in the lower part of the B-pillar. Looking at the results found in the orange cluster, a similar behaviour is identified, although the cut bends upward instead of running down towards the hole in the lower part of the B-pillar. Results in the green cluster do not have a cut at all, only an indentation on the right side of the B-pillar below the two holes. In addition to the three main clusters, another behaviour can be observed along the third PC axis, marked with red and pink. The result framed in red shows the plastic strain found within all of the clusters for points located in the lower spectrum of the third PC. The points with small values in the third PC have an indentation in the lower left part of the B-pillar. On the other hand points located on the upper spectrum of the third PC, marked and framed in pink, display an indentation at the top of the B-pillar. Notably only the simulations in the green cluster, which are the ones without any cuts at the bottom, that additionally have no indentation at the top are the results that successfully passed the Euro NCAP side crash test.

It is also visible that some simulations are located between clusters, although it is only very few. One example is shown in Figure 2.6.

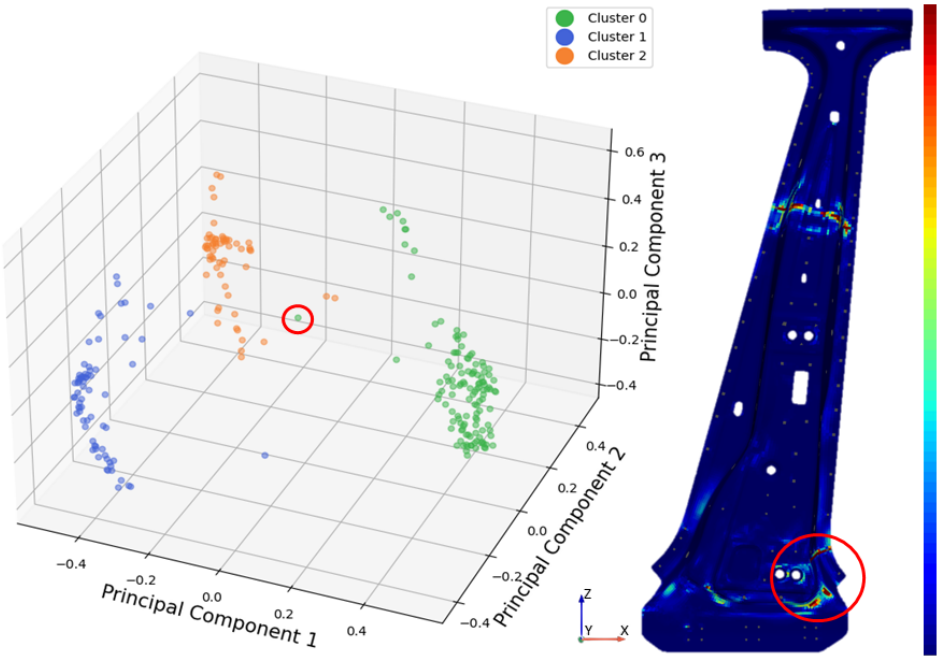



Figure 2.6: Outlier in reduced space of side crash simulation using kPCA.

Looking at the result of the outlier point that is located between the blue and the green cluster, it is visible that behaviour of both clusters is present. The question in this case is to decide which cluster this point should be assigned to. This topic will further be investigated in the following sections, more specifically Section 2.5.3.

The next step after the dimensionality reduction is the training of the SM and performing the MC analysis with realistic parameter distributions. Considering 5000 samples to be predicted, the MC analysis yields the reduced space shown in Figure 2.7.

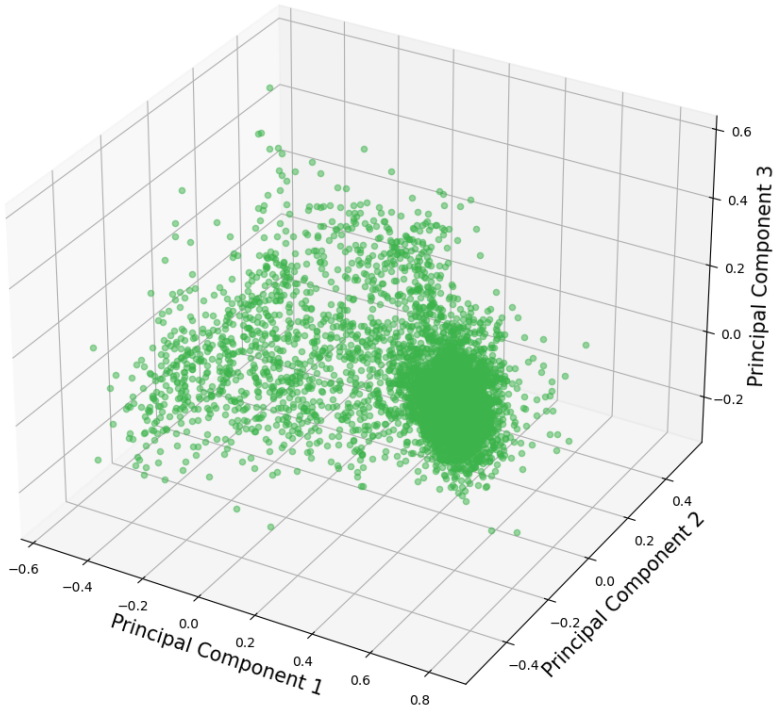
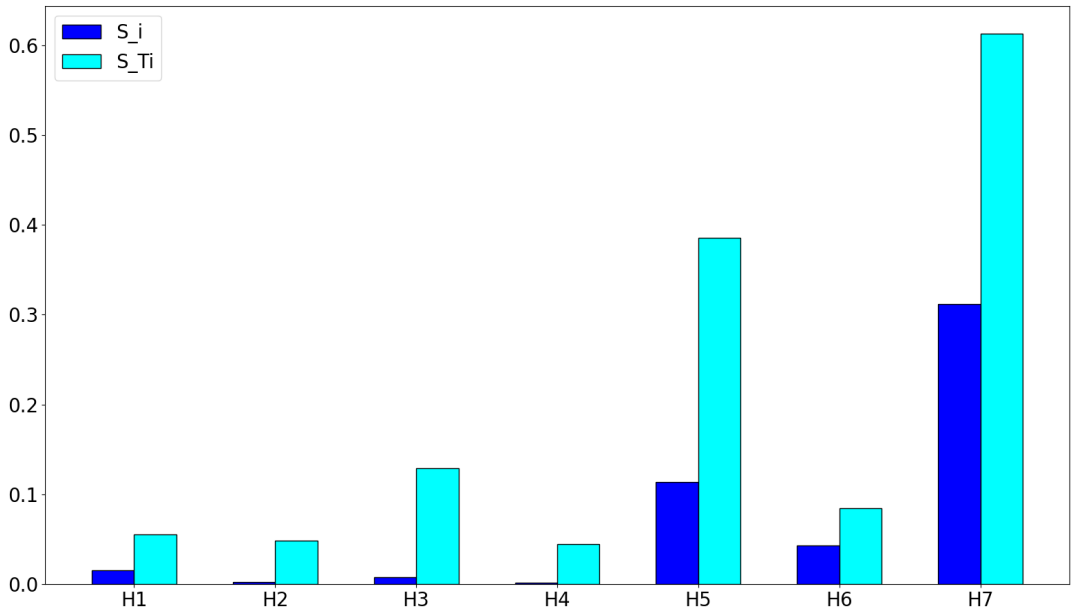



Figure 2.7: Reduced space with predicted samples of the MC analysis.

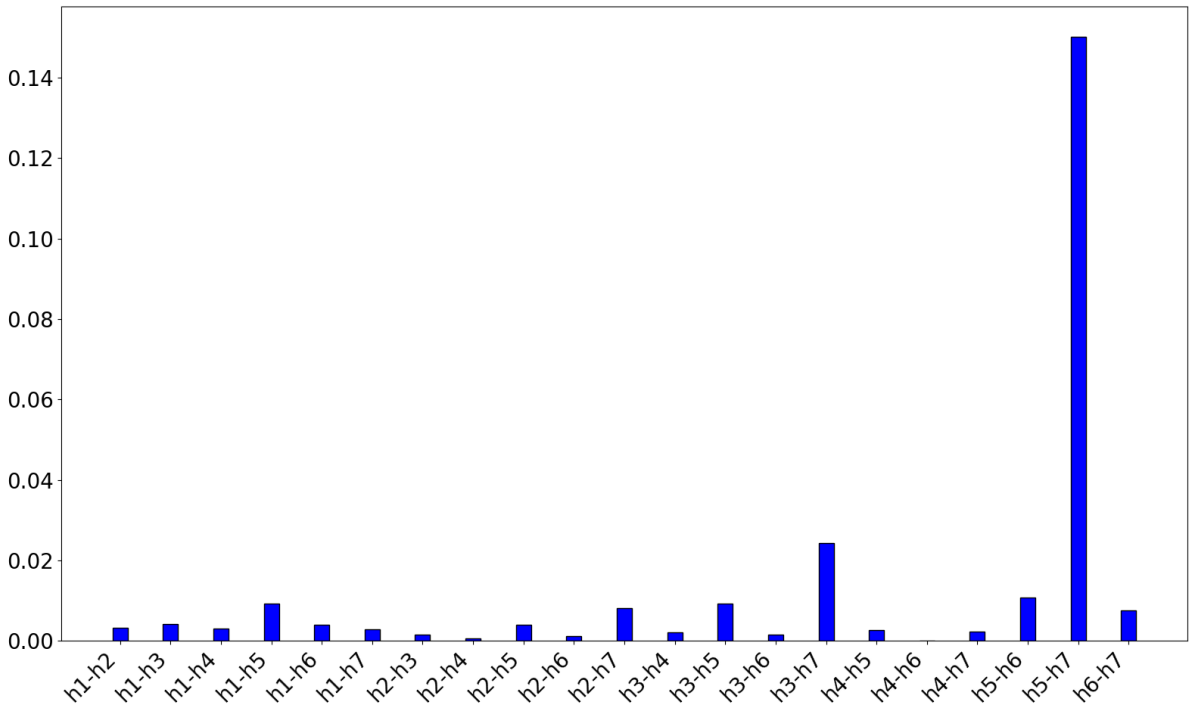
It can be seen that the predictions of the MC samples do not maintain the clustering structure found in Figure 2.5a. Instead the space between clusters disappears, as the SM predicts a multitude of points in the previously empty space, which is due to the interpolating nature of regression models. This is not desirable, as simulations in this space are not possible, or at least much less common than it appears with this MC analysis. The reason is that the simulation is not continuous in this space based on the information from the 258 initial simulations. Because the problem is very non-linear and not continuous, different approaches need to be explored, which will be explained in more detail in Section 2.3.

The last step of this analysis is to do a sensitivity analysis to identify how much influence each parameter has on the final outcome. This can be seen in the Sobol indices

presented in Figure 2.8.




(a) First (S_i) and total (S_Ti) order Sobol indices

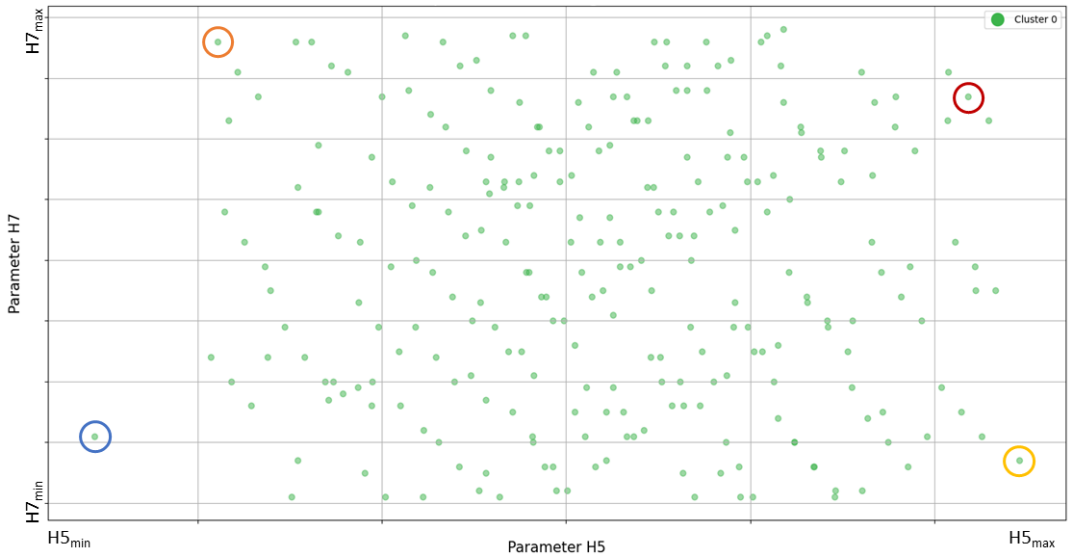



(b) Second order Sobol indices

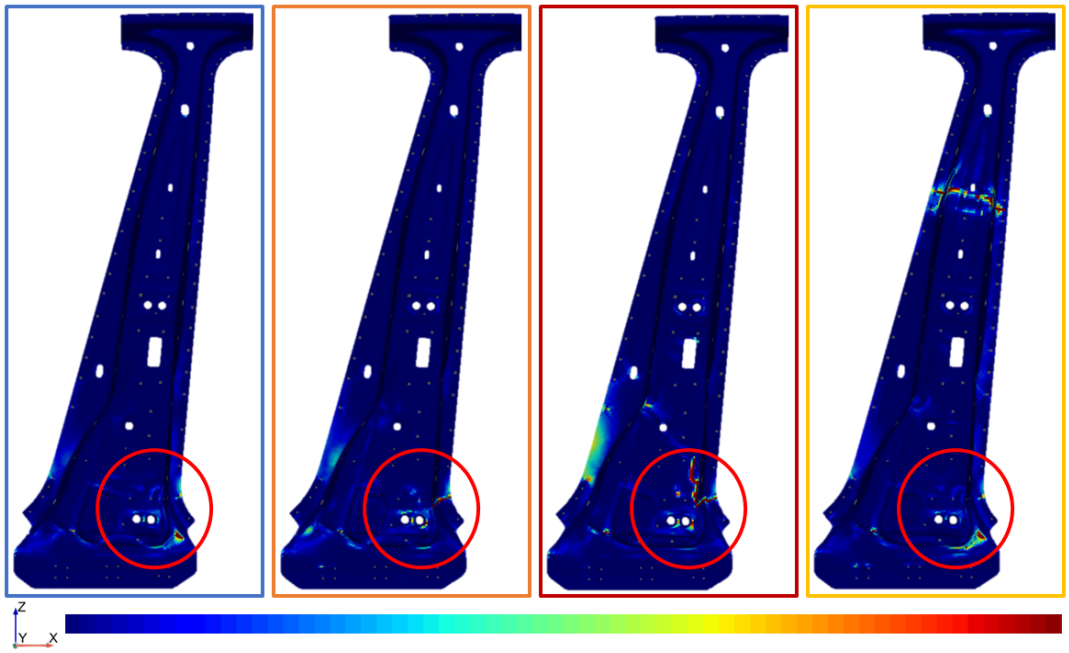
Figure 2.8: Sobol indices of first (S_i) and total (S_Ti) order (a) and of second order (b) for the side crash analysis.

Taking a look at the first and total order indices shown in Figure 2.8a a clear devia-

tion is noticeable between the first and total order indices. This indicates that there are many second and higher order interactions between the parameters, which is confirmed in Figure 2.8b. The parameter with the highest influence on the results is parameter h7, representing the bearing factor of the welding points. On the other hand parameters h2 and h4, which are the thickness of the retractor support and the horizontal position of the impact barrier, cause almost no variability in the results at all. Parameter h5, the vertical position of the impact barrier has the second highest influence on the variability of the results, followed by parameter h6, the material failure of the retractor support and parameter h1, the thickness of the B-pillar. With the thickness of the B-pillar having such a big influence on the results, but only considering a vague and not entirely realistic implementation of it assuming a constant thickness throughout, it would be desirable to implement a more realistic thickness distribution throughout the B-pillar. Analysing the second order indices the interaction between parameters h5 and h7 clearly stands out, which presumes a high correlation between the vertical position of the impact barrier and the bearing factor of the welding points. Reason being that if the impact barrier is higher, the lower part of the B-pillar, where the welding points are located, is more strongly affected by the lower edge of the barrier as the support structures at the bottom of the B-pillar are less effective. This in turn can cause major or minor deformations based on the bearing factor of the welding points. To better understand the correlation between parameter h5 and parameter h7 the data points were plotted over the values of these two parameters. The scatter plot and the corresponding results for the analysis are shown in Figure 2.9.




(a) Scatter plot over H5 and H7




(b) Plastic strain on B-pillars

Figure 2.9: Scatter plot over parameters H5 and H7 (a) and the plastic strain on the b-pillar for specific data points (b).

Figure 2.9 shows how parameter h5 and h7 interact with each other. If the bearing factor of the welding points is small a cut at the bottom part of the B-pillar is prevented. This can be seen in the blue and the golden framed points. When the bearing factor of the welding points (h7) is high, indicating a weak welding point, a cut is initiated at the bottom part of the B-pillar, seen in the orange and red framed points. When the vertical position of the impact barrier (h5) increases, so does the effect on the lower part

of the B-pillar, with an increase in deformation. As can be seen in the blue and orange framed points, the deformation and extend of the cut is kept to a minimum when the impact barrier is lower, compared to the red and golden framed points respectively as the support structures at the bottom of the B-pillar absorb less of the load. In other words, the deformation of the result framed in gold is much larger than the deformation of the result framed in blue, while still showcasing similar characteristics. Similarly the cut in the red framed result is much larger than the cut in the orange framed result. Overall if h7 is large, a cut is initiated and if h5 is large the respective deformation is amplified. This clearly shows the correlation between these two parameters, because coherent behaviour can be identified based on the combination of both values.

The main reason for such a high influence on the results of parameter h5, seen in the large first order Sobol index, is due to the correlation between a high parameter value of h5 and the indentation observed in the top part of the B-pillar. For the indentation to occur in the top part of the B-pillar a large h5 parameter value is obligatory. A higher position of the impact barrier is not synonymous with having an indentation at the top though, it is merely a requirement for this occurrence to appear. Similarly the high influence of parameter h7 on the results is due to the fact that the bearing factor of the welding points dictates the effect on the lower part of the B-pillar. When the bearing factor is small, indicating strong and good welding points, the deformation at the bottom of the B-pillar is kept to a minimum (green points in Figure 2.5), while larger values result in cuts and deeper indentations (blue and orange points in Figure 2.5).

Analysing parameter h1, the thickness of the B-pillar, more closely, yields interesting results as well. It is not only necessary to have a large value for h5 to obtain an indentation at the top part of the B-pillar, but also a small value of h1. This is visible in Figure 2.10, where all of the data points with a large third principle component also have a small value of h1.

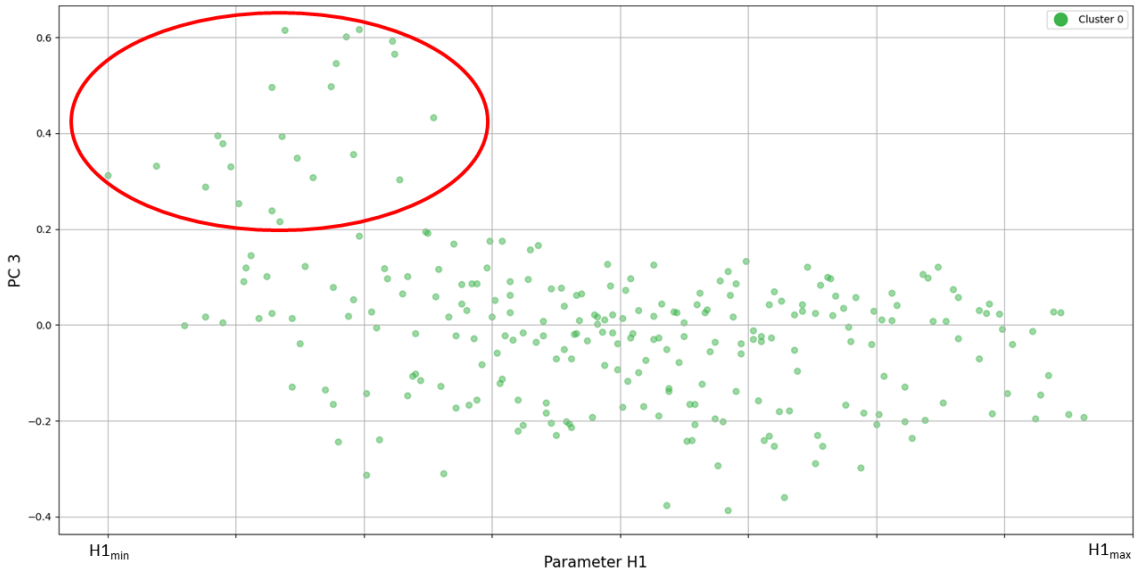



Figure 2.10: Scatter plot of data over parameter H1 and PC3.

Overall it can be said that the AQUA algorithm allows for a very in-depth analysis of the simulation, identifying uncertainties, parameter correlations and sensitivities. This leads to the identification of the parameters with major influence on the results. As was seen in the Sobol indices one of the more influential parameters is parameter h1, representing the thickness of the B-pillar. However this thickness is presumed to be constant throughout the entire part which is not conforming with reality. Since it is such an influential parameter it is desirable to implement a more realistic representation of the thickness throughout the B-pillar to capture the effects of this parameter with more detail and obtain more realistic results. This difficulty is further addressed in Chapter 4. On the other hand, it was also found that the thickness of the retractor support and the horizontal position of the impact barrier have almost no influence on the results and could be disregarded in future analysis.

## 2.3 Improvements to AQUA

The algorithm works very well in the stated form and a considerable amount of information and knowledge can be gained by its use. However, several features in the algorithm are left to be desired, as new problems and findings are uncovered during its application. This section gives an overview of the three main improvements that were made to the algorithm with a more detailed explanation in the following sections. In addition to these three main improvements to the algorithm itself a multitude of visualization and analytic tools were added to improve the usability and investigative use of the algorithm.

The first improvement to be made is implementing a so called Mixture of Experts (MoE) approach. The idea behind MoE is to train different experts for specific sections

of the data set. This means a model can have many different sections which have a specific type of behaviour that is unlike the other sections. In this case it is preferable to train a machine learning model for each of these sections, called experts, instead of training one global model. This approach is advantageous for capturing more details within each expert, and avoiding blurred transitions between sections. When analysing initial results presented in Section 2.2 it was found that often times the reduced space would exhibit clusters within the data. The space between the data is unreachable when running the real simulations due to physical reasons within the problem description. However, when only training one SM some of the predictions for new parameter combinations would appear in these empty spaces between clusters, because of the interpolating nature of regression models for unknown parameter combinations. This was clearly visible when showing the results in Figure 2.5a and Figure 2.7. If multiple SMs are trained for each of the appearing clusters, this prediction behaviour can be avoided. Implementing this approach therefore requires two extensions of the AQUA algorithm. The first one being clustering algorithms, which are able to identify the clusters in the data used to train the expert SMs. Secondly to be able to identify which parameter combination needs to be analysed by which SM responsible for a specific cluster, a classification algorithm needs to be implemented as well. The classification algorithm classifies the new parameter combinations to the according clusters based on the clustering of the initial data. The methods chosen for clustering the data are Hierarchical Agglomerative Clustering (HAC). For classification the Random Forest (RF) Classifier was selected.

The second refinement that needs to be done to the algorithm is implementing more DR techniques, especially for non-linear data sets. Reducing the dimension is the first step of the algorithm, which consequently means it has a big impact on the performance of the rest of the algorithm, as it sets the base for all other operations. The reason for implementing new DR techniques is the fact that the considered data is highly non-linear. The techniques that were used so far were only applicable for linear data sets or a very specific non-linear structure which needs to be identified. This is why a non-linear DR technique needs to be implemented, to deal with highly non-linear data and still capture as much information from the original dimension as possible. The DR techniques of choice for the given problem definition are the Multidimensional Scaling (MDS) and Uniform Manifold Approximation and Projection (UMAP) techniques that are both non-linear. UMAP, which is a fairly novel DR technique, falls into the category of favoring the preservation of local distances over global distances, unlike PCA and MDS, which aim at maintaining the pairwise distance structure for all of the data points. Having said that, UMAP still performs arguably better at keeping the global data structure than for example t-Distributed Stochastic Neighbor Embedding (t-SNE). This is especially useful for clustering, which is required for the MoE approach. [20]

The last improvement, but arguably the most important one, is a change to how the set of input parameters $\mathbf{H}$ is chosen. Previously only Halton points were selected to define the points for the set of input parameters. However, this method can lead to a very large set of data being required to contain enough information to cover the possible

output space. When using Halton points, areas which are already well described might be enriched even further, without improving the amount of information contained in the data substantially. Instead, it is desired to identify areas in the output space where very little information is known so far to strategically target these zones. To do this a automated cycle version of AQUA was developed which helps improve the detection of areas with little information in the data set and specifically run simulations aiming at enhancing the information density in these zones. Having information on the amount of undefined points also allows for the definition of another convergence condition, in addition to the Kullback-Leibler Divergence already used.

In addition to the three main modifications made to the AQUA algorithm, various other improvements regarding the usability, visualization and tools for analysis were implemented, which are used throughout the thesis to present the results. All of the solutions to the problems discovered in the initial version of AQUA explained in this chapter are further elaborated in the following sections. The methods and theories behind all of the improvements are explained in greater detail, with examples being presented alongside. A modified flow chart showing the three main changes to be made to the algorithm can be seen in Figure 2.11 with the additions marked in green.

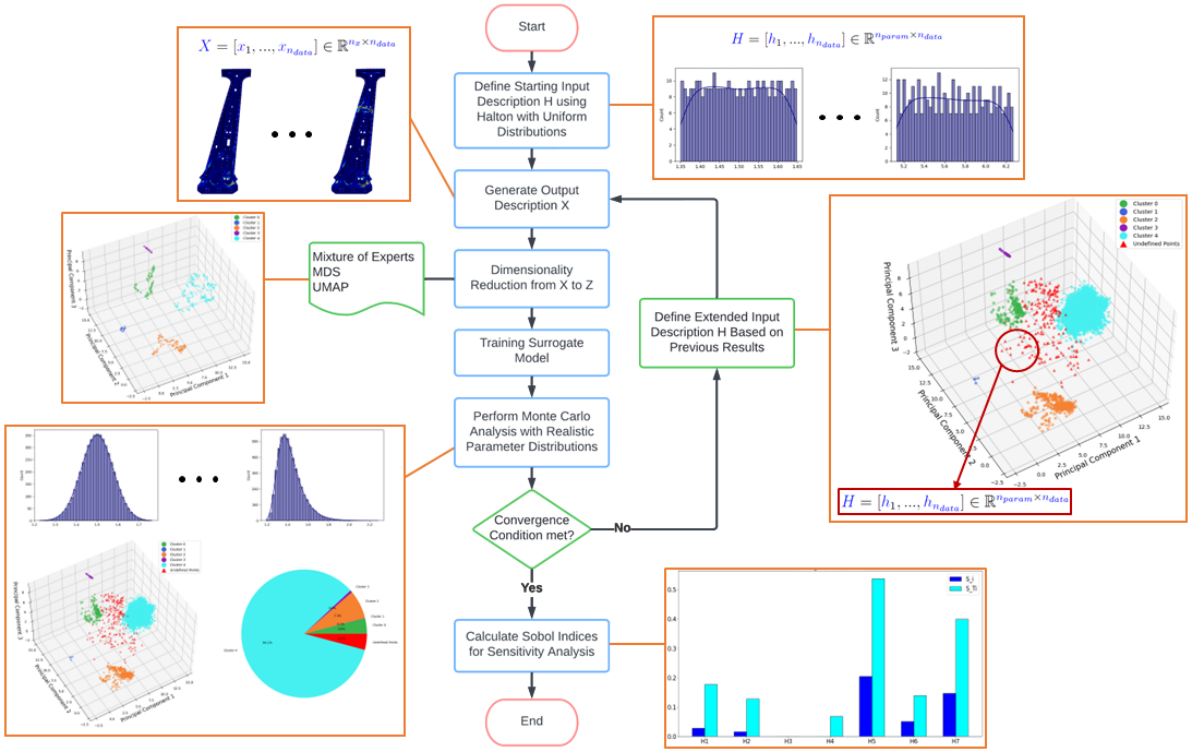



Figure 2.11: Modified flow chart of the AQUA algorithm with main changes to be made marked in green.

## 2.4 Mixture of Experts

The first main improvement to be implemented is the MoE approach. This approach is used to counteract the interpolating nature of regression models, which can cause some unrealistic results to be obtained. As could be seen in the side crash example a clear cluster formation could be identified. Solutions between these clusters are extremely unlikely or even impossible due to the fact that the simulation behaviour is not continuous in some cases. In the case of the side crash for example there is either no cut at the bottom right, or the cut is at least up to a certain depths, as once the cut is initiated it rips open until at least a certain point. This means that a result, where the cut only intrudes halfway to what is realistic is not desirable, but possible when only training one SM. With clustered data it is therefore also possible to do a mode identification, as each cluster belongs to a specific behaviour in the results, which can then be assigned a probability of occurring. By measuring the percentage of MC samples classified into each cluster, which correspond to a specific mode, the probability for each of these modes to occur in reality can be quantified.

In the MoE approach it is necessary to first cluster the data points in the reduced space to identify all of the potential modes in the results. After defining all the clusters, individual SMs for each cluster can be trained. To be able to know which SM needs to be used for which parameter combination or sample, a classification model has to be trained. The classification model then identifies which cluster the respective parameter combination belongs to and which SM to use for the regression. Classification algorithms are supervised machine learning techniques, as all of the data is labeled. Notably the classification algorithms return the certainty with which a parameter combination should be assigned to every cluster, where the largest certainty corresponds to the cluster in which the data point is classified in. This additional information is very useful and can be utilized for additional analysis tools, like the one explained in Section 2.6. As a side note, in the case that the clustering algorithm detects a cluster with fewer than ten data points, the number of clusters is decreased by one and the data is re-clustered, as a cluster with only ten data points contains too little data to appropriately train a SM.

The methods used for clustering and classification are explained in the following sections as well as a comparison of the MoE approach to the original example of the side crash shown in Section 2.2.

### 2.4.1 Clustering

For clustering mainly Hierarchical Agglomerative Clustering (HAC) and as an alternative k-Means Clustering (kMC) were selected as the methods of choice. HAC was chosen, as it is almost universally applicable to all kinds of data and has almost no requirements or constraints while being very versatile for all kinds of data structures [21]. Other techniques like for example Fuzzy Analysis Clustering (see for example [22]) or

Density Based Spatial Clustering (see for example [23]) were not implemented, as they have various drawbacks for the given data. This includes having previous knowledge of the data and space, which is not the case especially if the detection is supposed to be executed automatically, and other constraints like not having varying densities in the data or not being applicable to high dimensional data [24]. HAC has drawbacks of its own, which are further detailed later on, but is generally applicable to all kind of data. Additionally the clustering algorithm can be adjusted during the analysis, so it can be verified that the clustering was done correctly and it can be adjusted if needed. As an alternative the kMC method was implemented, as it is the most common technique with no requirements of previous knowledge or constraints on the data as well as having good adaptability for sparse data [21]. In the case that one method is not able to capture the clusters correctly, the other method can be selected instead. The idea is to not only cluster the set of output data in the reduced space $\mathbf{Z}$, but also be able to cluster the data in its original dimension $\mathbf{X}$. This is useful to verify that the dimensionality reduction method maintained the original problem structure.

The main reason for implementing HAC over any other technique is its simplicity and universal applicability, without needing any prior knowledge of the data. While Hierarchical clustering can also be done top-down, in Divisive Clustering, when determining the splits this requires an exponential number of subsets, which is not the case in the bottom-up HAC approach [24]. In the use case of this thesis there are no significant time requirements for the clustering method, as the data structures are not very large, so the time complexity of HAC, which is its main drawback, is not significant [21]. However if other models are to be analysed with the AQUA algorithm it might be necessary to implement other classification methods to deal with the given data structures. Since HAC is very sensitive to outliers and does not work well with missing data, although missing data is not a problem for the AQUA algorithm, it is helpful to have kMC as an alternative and also to be able to cluster the data in its original dimension [25].

### Hierarchical Agglomerative Clustering

HAC is a bottom-up approach where each data point starts out as its own single cluster and the clusters are successively merged together until either the desired number of clusters is reached or the cutting criteria is satisfied, not allowing for any more clusters to be merged. Using the cutting criteria allows the algorithm to determine the number of clusters without having to pre-define that number initially and can consist of different definitions. In the case of the AQUA algorithm this is not done, because it is assumed that no prior knowledge on the reduced space is available, as each DR method yields a different space. [21, 26]

The concept of HAC is to iteratively merge the two most similar clusters. The similarity of two clusters can be measured using linkage methods. The four linkage methods available are the single linkage, complete linkage, average linkage and ward linkage meth-

ods. The single, complete and average linkage methods are the most basic linkage types, where the similarity of two clusters is based on the distance between the two closest points in each cluster for single linkage, the distance between the two points that are furthest apart for the complete linkage and the average distance between all points in each cluster for the average linkage. The ward linkage method uses the centroids of the clusters to determine the similarity between two clusters. [8]

Denoting the distance between two points $\mathbf{x}$ of the set of points in a cluster $\mathbf{X}$ as $D(\mathbf{x}_i, \mathbf{x}_j)$ and the similarity measure between two clusters as $\Delta(\mathbf{X}_i, \mathbf{X}_j)$ the equations for all four linkage methods are as follows:

Single Linkage:

$$\Delta(\mathbf{X}_i, \mathbf{X}_j) = \min_{\mathbf{x}_i \in \mathbf{X}_i, \mathbf{x}_j \in \mathbf{X}_j} D(\mathbf{x}_i, \mathbf{x}_j), \tag{2.5}$$

Complete Linkage:

$$\Delta(\mathbf{X}_i, \mathbf{X}_j) = \max_{\mathbf{x}_i \in \mathbf{X}_i, \mathbf{x}_j \in \mathbf{X}_j} D(\mathbf{x}_i, \mathbf{x}_j), \tag{2.6}$$

Average Linkage:

$$\Delta(\mathbf{X}_i, \mathbf{X}_j) = \frac{1}{|\mathbf{X}_i||\mathbf{X}_j|} \sum_{\mathbf{x}_i \in \mathbf{X}_i} \sum_{\mathbf{x}_j \in \mathbf{X}_j} D(\mathbf{x}_i, \mathbf{x}_j), \tag{2.7}$$

Ward Linkage:

$$\Delta(\mathbf{X}_i, \mathbf{X}_j) = \frac{|\mathbf{X}_i||\mathbf{X}_j|}{|\mathbf{X}_i| + |\mathbf{X}_j|} \|c(\mathbf{X}_i) - c(\mathbf{X}_j)\|^2, \tag{2.8}$$

where $c(\mathbf{X}_i)$ stands for the centroid of $\mathbf{X}_i$ given as

$$c(\mathbf{X}_i) = \frac{1}{|\mathbf{X}_i|} \sum_{\mathbf{x} \in \mathbf{X}_i} \mathbf{x} \tag{2.9}$$

and $|\mathbf{X}_i|$ is the number of points in cluster $i$. [8]

The standard linkage method used in AQUA is the ward linkage method, as it was found after testing that for the type of scattered data obtained with the algorithm the ward linkage method works best, however this can be adjusted. Additionally the distance metric $D$ can be any distance metric with the implementation in AQUA permitting the use of the Euclidean, L1, L2, Manhattan or Cosine distance. Notably when selecting the ward linkage method the distance metric is automatically set to the Euclidean distance. To better visualize the linkage methods an illustration is shown in Figure 2.12.

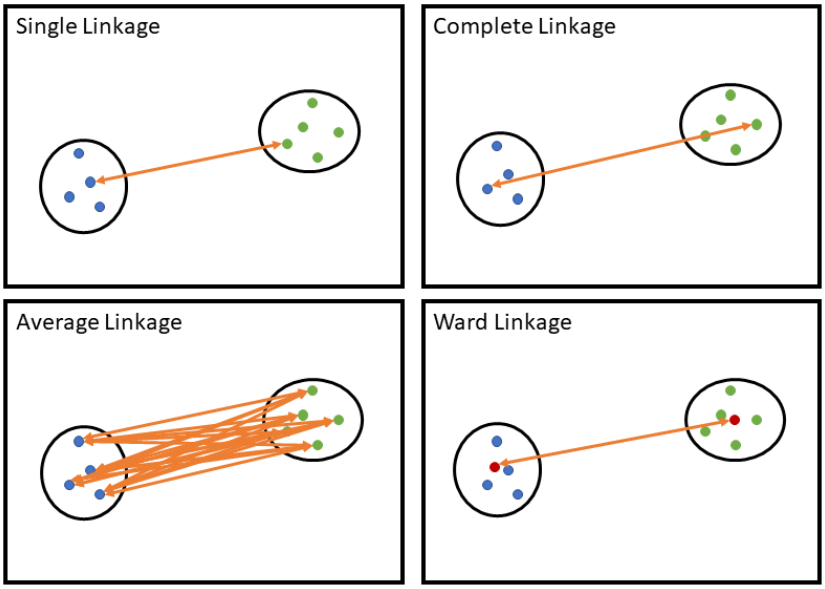



Figure 2.12: Linkage methods used in HAC. Adopted from [8]

Once all of the data points are clustered correctly the SMs for each of these clusters can be trained.

### 2.4.2 Classification

After having clustered the data and trained the SMs it is necessary to have the ability to classify new parameter combinations into these clusters, which enables the application of the correct SM. The next step, which is the MC analysis will generate thousands of samples that need to be predicted with the appropriate SMs. Unlike the clustering methods, which are an unsupervised machine learning technique, classification is a supervised machine learning technique like the SMs, as all of the data is labeled in this case. While the SM, which is a regression model, is also a supervised learning technique, it is very different from classification methods, as a classifier only needs to predict a predetermined category, while a regression model predicts continuous values. The goal of the classifier is to be able to identify which parameter combination belongs to which cluster.

The method used for classification is the Random Forest (RF) Classifier, with the Gradient Boosting (GBoost) method being a candidate for future implementations. The reason for choosing the RF Classifier over the GBoost Classifier is the fact that the GBoost Classifier needs parameter optimization depending on the problem being analysed. So to make the algorithm simpler to use and avoid complicated optimization problems for the classification model the RF Classifier is chosen, which requires little to no parameter tuning to perform well and in most cases performance is very similar to that of the GBoost Classifier, especially with noisy data. Additionally over-fitting is

less likely to occur with the RF classifier. Lastly the decision trees in the RF classifier can be visualized to get an idea of the decision path, which is useful to identify what parameter combinations result in certain outcomes. [27]

### Random Forest Classifier

Tree-structured classifiers are some of the most common classification techniques out there. However the RF classifier, while building on the basis of tree-structured classifiers, has some very significant benefits. The RF classifier is made up of a collection of tree-structured classifiers, hence the name Random Forest, with each of the tree-structured classifiers having an independent but identically distributed random vector. This feature is key, as this low or nonexistent correlation between the individual decision trees is the reason for its improved performance compared to an individual predictor. Each tree-structured classifier in the RF classifies the input with the majority of votes deciding the final classification. This procedure is also called bagging. The most significant and important improvement from this procedure is the classification accuracy. Importantly, compared to other classification algorithms, where only one model is trained, the RF approach is less prone to over-fitting, because of multiple trees voting on the same input space. Additional benefits include fairly good accuracy while being relatively robust to outliers and noise and being very simple and parallelizable. [28]

To ensure that the decision trees in the RF classifier are independent, only a part of the full training data, selected randomly, is used on training each decision tree. Subsequently the decision trees are split until the Gini index of the given split is 0, at which point this particular branch ends. The Gini index is defined as follows:

$$G(t) = 1 - \sum_{i=1}^{C} p(i|t)^2, \tag{2.10}$$

where $G(t)$ is the Gini index for node $t$, $C$ is the total number of classes, or in the case of this thesis clusters, and $p(i|t)$ is the probability of class $i$ in node $t$. This Gini index basically represents the probability of a randomly selected parameter combination from the training set to be classified incorrectly into this branch. When the Gini index is 0 it means the classification using this decision path is pure, as the probability of a wrong classification is 0, since all of the probabilities $p(i)$ are either 0 or 1. If the Gini index is not 0 for a certain node, it means a further split is necessary. To identify the ideal split, a value, denoted as Gini rating in this thesis, is calculated, which is given as:

$$G_{rating}(t_L, t_R) = p(t_L)G(t_L) + p(t_R)G(t_R), \tag{2.11}$$

with $G_{rating}(t_L, t_R)$ being the Gini rating for a split of node $t$ into the left node $t_L$ and the right node $t_R$. Then $p(t)$ is the probability of the split, given by the percentage of samples in that split, and $G(t)$ is the Gini index for that split. The best split is obtained

by minimizing the Gini rating. [29]

It is possible to use other criteria instead of the Gini index as well, like the cross-entropy or misclassification error, however these methods are not used in this thesis. An example for such a decision tree can be seen in Figure 2.13.

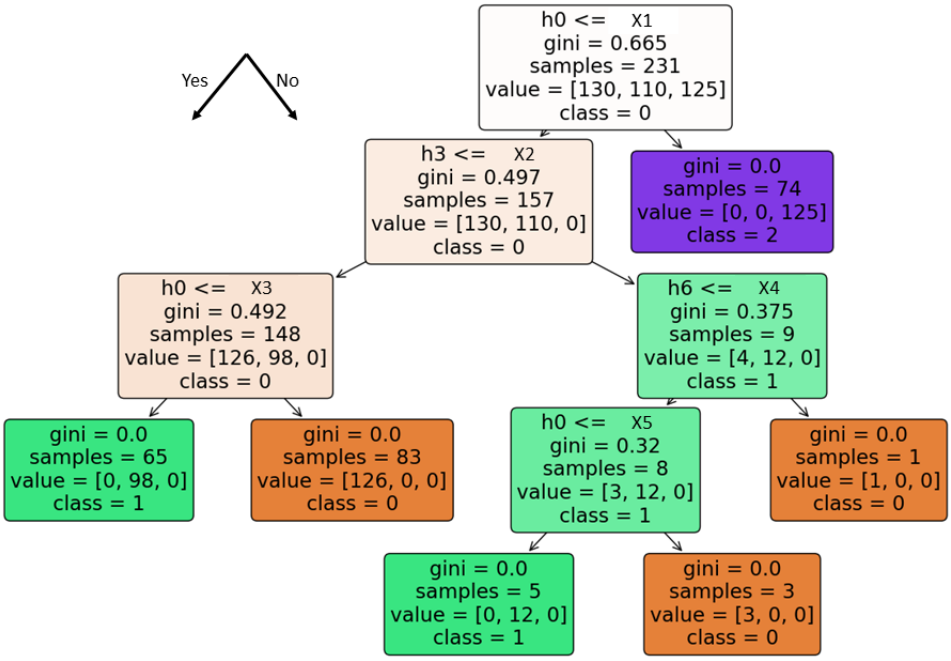



Figure 2.13: Example of a decision tree.

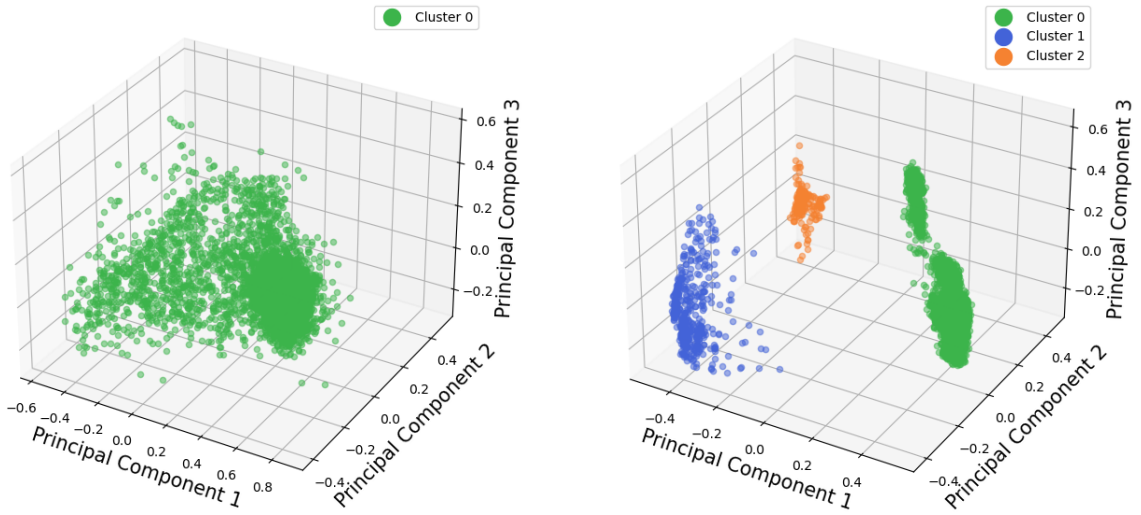
When generating a new classification tree for the RF classifier, $m \leq n_p$ input parameters are selected for the new tree. $m$ is the number of randomly selected input parameters to be used for training the specific classification tree out of the total $n_p$ input parameters. Usually $m$ is chosen to be $\sqrt{n_p}$. The idea is to reduce the correlation between the trees by reducing $m$ and therefore reduce the variance of the whole classification model. It is worth noting that in reality $m$ can depend on the problem and an experienced user can change this value depending on the problem to be analysed. However, since the AQUA algorithm is to be kept as user friendly as possible the default value of $\sqrt{n_p}$ is used, as this still yields very good results in most cases and reduces the amount of tuning that needs to be done by the user. [27]

Additionally to decrease the correlation between the decision trees even further, only 2/3 of all the available samples are used for training each decision tree in the implementation used in this thesis. The samples are selected randomly just like the input parameters. Lastly the number of classification trees to be trained for the RF classifier is set to 100. It was found that increasing the number of trees did not improve the classification accuracy with the number of input parameters and clusters found in the

problem definitions of this thesis. However if the number of input parameters were to increase substantially for a new analysis, this parameter would potentially have to be adjusted. [30]

### 2.4.3 Application on Side Crash of Dummy Car

Cluster detection was first mentioned in the initial analysis of the side crash in Section 2.2. Even though for that analysis the clusters were only used for visual purposes, the MoE approach takes this information to the next level, by considering the clusters and training different SMs for each of these clusters. Figure 2.14 shows the predictions of the MC samples using two different approaches. The first approach only uses one SM to predict all of the MC samples, while the other method uses the MoE approach, training a SM for each cluster that was detected in Figure 2.5.




(a) One SM (b) MoE approach

Figure 2.14: Reduced space of the MC samples for the side crash example using one SM for the entire data (a) and using the MoE approach (b).

Comparing both scatter plots in Figure 2.14 it is clearly visible, that the clustered structure of the original data is very well maintained when using the MoE approach. The empty space between the clusters is not filled with unrealistic data points blurring the line between different cluster behaviours. This is exactly what the MoE approach was supposed to avoid. To proof that this method is better than using only one SM the error of the SMs predictions is calculated. To do this the Relative Squared Error (RSE) is used. Since the values to be considered are vectors the norm of the differences is used. The equation for the SM error is therefore written as:

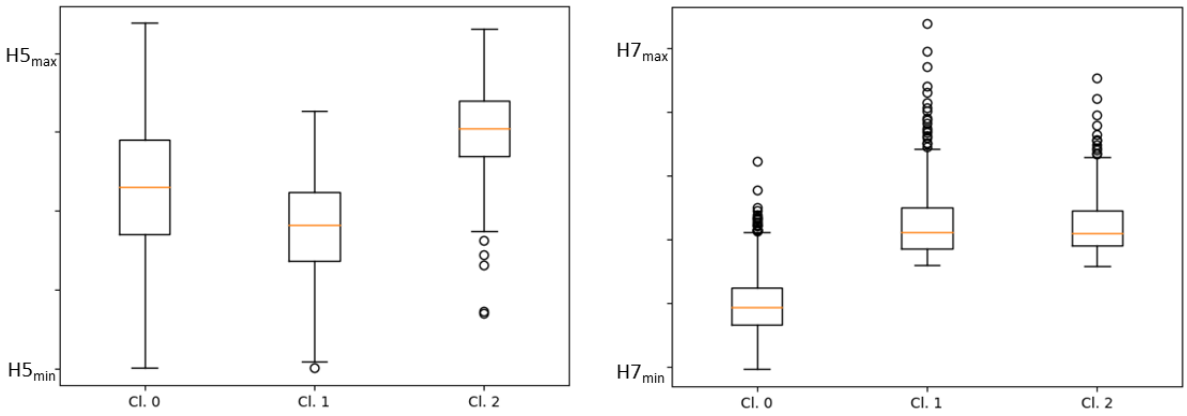
$$RSE = \frac{\sum_{i=1}^{n} \|y_i - \hat{y}_i\|^2}{\sum_{i=1}^{n} \|y_i - \bar{y}\|^2}, \tag{2.12}$$

with $y_i$ being the target value of the sample to be predicted, $\hat{y}_i$ being the predicted value from the SM, $\bar{y}$ being the mean of $y$ given as $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$ and $n$ the sample size. The result of the RSE is an indication as to how well the SM performs compared to a simple model using the average as the prediction. Values closer to 0 are better, while values over 1 are unacceptable, as then it would be more accurate to just take the average as the prediction always. The reason for dividing by the difference of the target value and the average value instead of just dividing by the target value is that the magnitude of the target value is highly dependent on the reduced space and target values closer to zero would then automatically have a higher error, even though the distance between the prediction and the target value is not greater than for target values that have very large coordinate values. This process requires the input and reduced output description to be split into a train and test set. To avoid any bias with the clustered data the train-test split is stratified with a ratio of 80% of the data for training and 20% for testing. Additionally this train-test split is done five times with different random splits to increase the data size.

The results for the data set presented in Section 2.2 of the side crash result in a SM error of 0.05 for the MoE approach and a SM error of 0.31 for the model with only one SM. This is a very significant improvement as the MoE approach has an error six times smaller. Notably it should be taken into account that the MoE approach also induces a classification error into the results, as some points might be classified into the wrong cluster. To calculate the classification error the same train-test split procedure is used as for the SM error but instead obtaining the percentage of rightly classified points. When using only one SM this classification error naturally does not exist, but for the MoE approach of the side crash example it results in an error of 17%. It is worth noting that these errors are very conservative and are better in the final models used, as the final models are trained using all of the data and not only 80% of it. Lastly, and most importantly, the classification error can be improved by increasing the number of data points and therefore the training set. As of right now, with the given data set, the total error of the MoE approach is an error of 17% in classification and an error of 0.05 for the SM compared to the SM error of 0.31 obtained with the original approach. However, the classification model can be improved significantly by increasing the amount of data points or selecting the data points more intelligently, which is addressed in Section 2.6 and the SM error can also be improved slightly by choosing different DR methods as explained in Section 2.5. On the other hand, the SM error when using only one SM cannot be improved so easily. This is because the amount of data points required to identify the clusters and avoid the blurring of clusters needs to be extremely large. In fact it would need to be substantially larger than the amount needed in the MoE approach, making it unfeasible.

Another benefit of using the MoE approach is that, in addition to the previously men-

tioned advantages, it allows for mode identification. Since the data is now divided into clusters with certain characteristics in the results, it is possible to assign a probability of occurrence to each cluster after the MC analysis. Subsequently the range for all the parameters in each cluster can be found, giving the ability to identify critical parameter combinations. Using the results of the MC analysis depicted in Figure 2.14b the different mode probabilities can be obtained. Due to confidentiality concerns the exact values cannot be disclosed. However, cluster 0 shows by far the highest probability, followed by cluster 1 and cluster 2. Assuming a hypothetical probability $p_0$ for cluster one, this means there is a $p_0$ probability that a randomly selected parameter combination with the given distributions results in a plastic deformation of the B-pillar with similar characteristics to the ones found in cluster 0. This is extremely useful, as it gives the probability for certain behaviour to occur in real life. To find any specific parameter values resulting in specific behaviour the parameter ranges in each cluster a plotted. Figure 2.15 shows the ranges for parameters h5 and h7, which were already analysed in Section 2.2.




(a) Parameter H5 (b) Parameter H7

Figure 2.15: Range of values in each cluster for parameter H5 (a) and parameter H7 (b).

As was already found based on looking at the results in Section 2.2 parameter h7, the bearing factor of the welding points, dictates whether the result will have behaviour similar to cluster 0 or cluster 1 and 2. All of the points in cluster 0, which is the most desirable cluster for the Euro NCAP ranking, have a low bearing factor of the welding points. In a real application this could for example be used to put an emphasis on the welding process and the quality of the welding points in production to significantly improve the results in the side crash. Parameter h5, the vertical position of the impact point, does not show characteristics as clearly as parameter h7, although this is mainly due to the fact that the effect of varying this parameter causes the results to shift within each cluster, and not shift clusters. As was explained in the initial analysis a higher value of parameter H5 is required for the indentation to appear at the top of the B-pillar, but this can happen in any cluster. Interestingly though, it can be seen

that for results to appear in cluster 2 slightly higher values of h5 are required. This is the orange cluster with the cut facing upwards. To understand the reason for this the complete FEM model would have to be analysed to find any part interactions causing this behaviour. Unfortunately due to confidentiality reasons the FEM model is not available for this analysis. However it is interesting to point out that the Audi AG has made a similar discovery to this, as they introduced a device that is able to lift the Audi A8 up by 80 mm in the event of a side crash, essentially lowering the point of impact [31].

### 2.4.4 Conclusion

In conclusion the introduction of the MoE approach caused the predictions of the MC samples to maintain the clustered structure and significantly improved the SM error. Even though it also induced a classification error into the algorithm the accuracy is still substantially better than the initial approach and most importantly the classification error can be improved significantly with different sampling approaches addressed in Section 2.6. Lastly the new approach allows for mode detection and the identification of the probability for specific occurrences in the results. Lastly the clustered data gives the option to find parameter combinations and values that result in certain outcomes, giving the option to identify desirable and undesirable parameter combinations.

## 2.5 Dimensionality Reduction

The DR step is an extremely important step in the AQUA algorithm, even more so with the MoE approach introduced in the previous section, as it is the first major step in the algorithm and therefore has influence on all of the following steps. The data to be analysed can be very non-linear and show complex behaviour, which is why the DR techniques used need to be able to capture this information. The PCA used previously is good for linear data and while the kernel Principle Component Analysis (kPCA) can capture non-linear structures in the data it is necessary to have some kind of previous knowledge on what kernel would best portray the given data structure. So the main requirement for the new DR techniques is that they are able to capture non-linear behaviour and maintain the data structure of the original dimension as well as possible. With the introduction of the MoE approach it is also desirable to have a technique that favours the preservation of local distances over global distances to facilitate cluster recognition and improve the SMs. The techniques chosen for this task are MDS and UMAP.

In general DR techniques can be grouped into methods that aim at preserving the distance structure for all the data points, namely matrix factorization methods, and the methods that give preference to preserving the local distances over the global distances, namely neighbour graph methods [20]. To have more flexibility and for the code to be

more adaptable one method from each of these categories was selected. MDS is a method that tries to maintain the global distance structure among all data points, which is a very good method to identify outliers and have a general idea of what the data structure looks like. However, spread out data and outliers are not desirable for either clustering, as could be seen in Section 2.4, the regression models used for the SMs or the backward mappings. This is why another method from the latter category was chosen to complement MDS, namely UMAP. Other methods that fall into the same category as UMAP, like Isomap and t-Distributed Stochastic Neighbor Embedding (t-SNE) were tried as well, but UMAP was found to yield better results overall.

### 2.5.1 Multidimensional Scaling

MDS is a dimensionality reduction technique aimed at maintaining the global distances between all the points from the original space in the reduced space. This means in general the method tries to approximate

$$d_{ij} \overset{!}{=} \|\mathbf{x}_i - \mathbf{x}_j\|_2 = \hat{d}_{ij}, \tag{2.13}$$

as closely as possible for every point in the set. $d$ is the euclidean distance between two points in the original full dimension and $\mathbf{x}$ are the coordinates of a point in the reduced dimension used to calculate $\hat{d}$, representing the euclidean distance between the same two points in the reduced space. Because it is not always possible to exactly maintain all distances from the original space in the reduced space, the best possible approximation is sought after. Theoretically this method is not only applicable with distance measures, but also dissimilarity measures or other types of correlation measures. When the similarity measure is not metric the non-metric version of MDS needs to be used. However in this case the metric MDS is used, since the similarity measure is the distance between the points. [32]

To find the optimal position of all the points to approximate the distances in the original space as close as possible, a loss function is minimized. The most common loss function is the normed sum-of-squares of the errors in the generated space, which is defined as stress $\sigma$. This stress function is given as:

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n}\sum_{j=i+1}^{n}(\hat{d}_{ij} - d_{ij})^2}{\sum_{i=1}^{n}\sum_{j=i+1}^{n} d_{ij}^2}}, \tag{2.14}$$

with the numerator of the fraction being called the raw stress function. [33]

The difficulty comes in finding the optimal configuration of point coordinates $\mathbf{x}$ that minimizes this stress function $\sigma$. This is in part due to the fact that the function to be minimized has many variables, $n \times t$ variables to be exact, where $n$ is the number of points in the space and $t$ is the dimension of the reduced space. So for the side crash

example analysed in Section 2.2 where 255 points were in the data set and the reduced space is of dimension 3, it is necessary to optimize $255 \times 3 = 765$ variables that are all affecting each other. Therefore $\sigma$ is a function of $\mathbf{X}$, with $\mathbf{X}$ being a $n \times t$ matrix where each row represents the coordinates $\mathbf{x}$ of a point, because $\hat{d}$ is dependent on $\mathbf{X}$ according to its definition in Equation (2.13).

The idea is to start with a base configuration that is randomly selected and iteratively improve the configuration until the stress value converges. This does not however guarantee that the global minimum is found, as local minima will cause this method to converge as well. Consequently this method is run multiple times, in this case four times, with different randomly selected starting points to increase the chances of finding the global minimum. Out of these four runs the space with the smallest stress value is chosen. Initially methods like gradient descent were used, but the method used in this thesis is known as the SMACOF (Scaling by MAjorizing a COmplicated Function) method. [33, 30]

The following procedures a based on [34, 9, 35]. Instead of trying to minimize a very complex function, like $\sigma(\mathbf{X})$, a simpler function $g(\mathbf{X}, \mathbf{Y})$ is minimized instead. $\mathbf{Y}$ is the coordinate matrix $\mathbf{X}$ but from the previous iteration and $g(\mathbf{X}, \mathbf{Y})$ is the majorizing function. The requirements for $g$ are:

1. $\sigma(\mathbf{Y}) = g(\mathbf{Y}, \mathbf{Y})$,
2. $\sigma(\mathbf{X}) \leq g(\mathbf{X}, \mathbf{Y})$,
3. $g(\mathbf{X}, \mathbf{Y})$ must be simple (usually linear or quadratic).

The reason for choosing a simple majorizing function instead of using the original complex function is that it is simple and has guaranteed descent. Additionally it is easy to compute and the step size procedure is much simpler compared to other methods. First it is necessary introduce an alternative notation for the euclidean distance:

$$\begin{aligned} \hat{d}_{ij}^2(\mathbf{X}) &= \sum_{s=1}^{t}(x_{is} - x_{js})^2 = \sum_{s=1}^{t}[\mathbf{x}_s^T(\mathbf{e}_i - \mathbf{e}_j)]^2 = \sum_{s=1}^{t}\mathbf{x}_s^T(\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^T\mathbf{x}_s \\ &= \operatorname{tr}\mathbf{X}^T\mathbf{A}_{ij}\mathbf{X}, \end{aligned} \tag{2.15}$$

where $\mathbf{e}_i$ is the $i$-th column of the identity matrix $\mathbf{I}$, $\mathbf{x}_s$ is the $s$-th column of $\mathbf{X}$ and $\mathbf{A}_{ij}$ is a $n \times n$ matrix of zeros except the entries $a_{ii} = a_{jj} = 1$ and $a_{ij} = a_{ji} = -1$.

Now decomposing the raw stress function, which is the numerator of $\sigma$ defined in Equation (2.14), with the addition of a non-negative weight $w$ to indicate the importance

of the residual combination $ij$ that is usually set to 1, as:

$$\begin{aligned}\sigma_{raw}{}^2(\mathbf{X}) &= \sum_{i=1}^{n}\sum_{j=i+1}^{n} w_{ij}(d_{ij} - \hat{d}_{ij}(\mathbf{X}))^2 \\ &= \sum_{i=1}^{n}\sum_{j=i+1}^{n} w_{ij}d_{ij}^2 + \sum_{i=1}^{n}\sum_{j=i+1}^{n} w_{ij}\hat{d}_{ij}^2(\mathbf{X}) - 2\sum_{i=1}^{n}\sum_{j=i+1}^{n} w_{ij}d_{ij}\hat{d}_{ij}(\mathbf{X}) \\ &= \eta_d^2 + \eta_{\hat{d}}^2(\mathbf{X}) - 2\rho(\mathbf{X}),\end{aligned} \tag{2.16}$$

with $\eta_d^2$ is the total dispersion, $\eta_{\hat{d}}^2(\mathbf{X})$ is the reconstructed dispersion and $\rho(\mathbf{X})$ is the codispersion.

Combining Equation (2.15) and Equation (2.16) results in the following definitions:

$$\eta_{\hat{d}}^2(\mathbf{X}) = \sum_{i=1}^{n}\sum_{j=i+1}^{n} w_{ij}\hat{d}_{ij}^2(\mathbf{X}) = \operatorname{tr}\mathbf{X}^T\left(\sum_{i=1}^{n}\sum_{j=i+1}^{n} w_{ij}\mathbf{A}_{ij}\right)\mathbf{X} = \operatorname{tr}\mathbf{X}^T V\mathbf{X}, \tag{2.17a}$$

$$\rho(\mathbf{X}) = \sum_{i=1}^{n}\sum_{j=i+1}^{n} w_{ij}d_{ij} = \operatorname{tr}\mathbf{X}^T\left(\sum_{i=1}^{n}\sum_{j=i+1}^{n} b_{ij}\mathbf{A}_{ij}\right)\mathbf{X} = \operatorname{tr}\mathbf{X}^T\mathbf{B}(\mathbf{X})\mathbf{X}, \tag{2.17b}$$

where $b_{ij} = w_{ij}d_{ij}/\hat{d}_{ij}(\mathbf{X})$.

The minimization problem is now simplified to the two functions $\eta_{\hat{d}}^2(\mathbf{X})$, which is a straight-forward quadratic function, and $\rho(\mathbf{X})$, which is more difficult to solve. However assuming that $\hat{d}(\mathbf{Y}) > 0$ and applying the Cauchy-Schwarz inequality ([36, 37]) shows that a linear majorization of $-\hat{d}_{ij}(\mathbf{X})$ is given by $-\operatorname{tr}\mathbf{X}^T(\hat{d}(\mathbf{X})\mathbf{A}_{ij})\mathbf{Y}$, which is a linear function and easy to solve. This results in a new definition of $\rho(\mathbf{X})$ given by:

$$-\rho(\mathbf{X}) = -\sum_{i=1}^{n}\sum_{j=i+1}^{n} w_{ij}d_{ij} \leq -\operatorname{tr}\mathbf{X}^T\left(\sum_{i=1}^{n}\sum_{j=i+1}^{n} \hat{b}_{ij}\mathbf{A}_{ij}\right)\mathbf{X} = -\operatorname{tr}\mathbf{X}^T\hat{\mathbf{B}}(\mathbf{Y})\mathbf{Y}, \tag{2.18}$$

with $\hat{b}_{ij}$ given as:

$$\hat{b}_{ij} = \begin{cases} w_{ij}d_{ij}/\hat{d}_{ij}(\mathbf{Y}) & \text{if } \hat{d}_{ij}(\mathbf{Y}) > 0, \\ 0 & \text{if } \hat{d}_{ij}(\mathbf{Y}) = 0. \end{cases} \tag{2.19}$$

Considering all of the above transformation this results in the majorizing inequality stated as

$$\begin{aligned}\sigma_{raw}{}^2(\mathbf{X}) &= \eta_d^2 + \eta_{\hat{d}}^2(\mathbf{X}) - 2\rho(\mathbf{X}) \\ &\leq \eta_d^2 + \operatorname{tr}\mathbf{X}^T V\mathbf{X} - 2\operatorname{tr}\mathbf{X}^T\hat{\mathbf{B}}(\mathbf{Y})\mathbf{Y} = g(\mathbf{X},\mathbf{Y}).\end{aligned} \tag{2.20}$$

Now to find the new updated $\mathbf{X}$ for the next iteration the derivative of $g(\mathbf{X}, \mathbf{Y})$ is equated to zero like so:

$$\begin{aligned} \frac{\partial g(\mathbf{X}, \mathbf{Y})}{\partial \mathbf{X}} = 2V\mathbf{X} - 2\hat{\mathbf{B}}(\mathbf{Y})\mathbf{Y} \stackrel{!}{=} 0 \\ \Rightarrow \mathbf{X} \stackrel{!}{=} V^{-}\hat{\mathbf{B}}(\mathbf{Y})\mathbf{Y}, \end{aligned} \tag{2.21}$$

with $V^{-}$ being the Moore-Penrose inverse of $V$. Additionally if $w_{ij} = 1$, which is usually the case, the equation simplifies to $\mathbf{X} \stackrel{!}{=} n^{-1}\hat{\mathbf{B}}(\mathbf{Y})\mathbf{Y}$.

To visualize this process an example is presented in Figure 2.16.

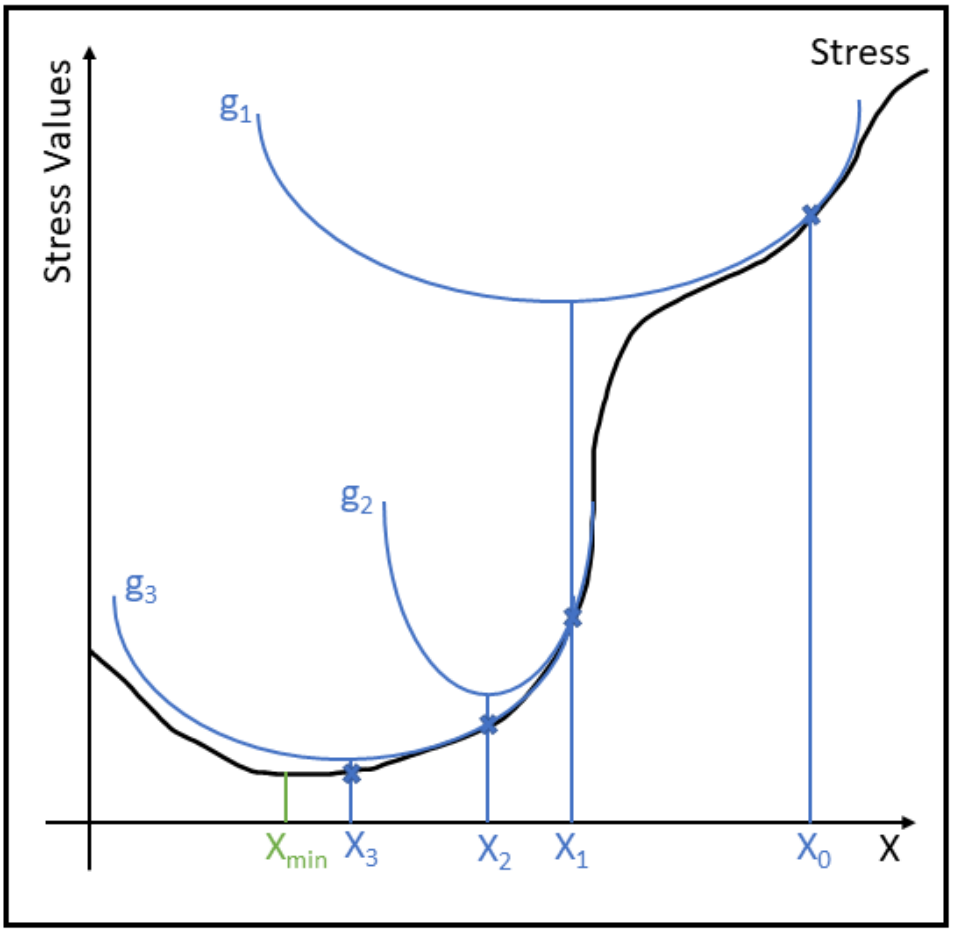



Figure 2.16: Convergence example of SMACOF minimization procedure. Based on [9]

### 2.5.2 Uniform Manifold Approximation and Projection

Similar to MDS the UMAP method tries to reduce the dimension of a large dimensional data set maintaining the similarities or distances between the points in the original space

and the reduced space. Unlike MDS however, this method emphasises local distances over global distances, meaning clustered points (i.e. similar points) are kept close to each other in the reduced space, while points from other clusters are actively separated further apart. This is done by using the KNN, making the number of KNN the most important hyperparameter of this method. These KNN are then used to arrange the data points according to their similarities and dissimilarities with other points using fuzzy simplicial sets. However, this is not exactly representative of the original space like in the case of MDS, which tries to maintain the distances between all points as well as possible. Even though it does not try to exactly reproduce the distances from the original space, prioritizing the preservation of similar points emphasises clusters in the data which are better visualized. [20]

This section will not go into too much mathematical details, as this is not the purpose of this thesis. Instead a focus is set on explaining the main thoughts behind the method and how it is implemented. For more information regarding the mathematical details of UMAP it is recommended to look at [20]. All of the following work in this section is based on information from the original paper for UMAP [20].

UMAP can be split into three main functions, namely defining the local fuzzy simplicial sets for every point in the data set, defining the initial location of points in the reduced space through spectral embedding and finally optimizing the embedding in the reduced space.

As input UMAP takes the distance between all of the points. To obtain the fuzzy simplicial sets it is not enough to just take the distances directly. As a side note, a simplicial set is a model that captures a topological space using simplices and their relations. For more information on simplicial sets it is recommended to look at [38, 39, 40]. One requisite for building a simplicial set out of a topological space, that in theory captures all of the important information, is that the data is uniformly distributed. If the data is not uniformly distributed the simplicial set contains gaps and clumps, which means the full topological space is not represented correctly. This means a metric needs to be defined that makes the data in the topological space seem like its uniformly distributed. Instead of using the distances, weights $w$ are used instead, that can be defined depending on the structure of the topological space. In other words every section in the topological space, depending on how dense the data points are in that section, will map to the euclidean space on a different scale. The simplicial set is set up in a way that the whole data set appears uniformly distributed, implicating that very densely packed points will have similar magnitude of weights as points that are in sparse sections of the space. To have a better notion of how dense an area around a point in the topological space is and ensure that the topological space is locally connected, a fuzzy notion of the distances is introduced. This fuzzy notion will define the weights $w$ to this point based on the distances between multiple nearest points around it. How many nearest points are considered is based on the hyperparameter $k$ that is defined by the user, which determines how many KNN are used to determine the weights of the fuzzy simplicial set.

The first step to generate the local fuzzy simplicial sets for each point is to determine what value all the weights $w$ to that point should sum up to. This value is determined by taking the binary logarithm of the number of KNN, namely $k$. For example when $k = 8$ is selected, all the weights of the $k$ KNN of each point should sum up to $\log_2 8 = 3$. Having this condition ensures the perception of a uniformly distributed topological space. Notably the weights, or similarity scores, are not necessarily symmetrical. For example when defining the fuzzy simplicial set for point 1 and point 2, the weight from point 1 to point 2 might be different than the weight from point 2 to point 1. This is because the two points have different KNN resulting in different magnitudes of the weights. The equation used to obtain the weights relative to each point, ensuring that the sum of the scores equals the binary logarithm of the number of KNN, is given by:

$$w(x_i, x_j) = \exp\left(\frac{-\max(0, d(x_i, x_j) - \rho_i)}{\sigma_i}\right) \quad \text{for } 1 \leq i \leq N \text{ and } 1 \leq j \leq k, \tag{2.22}$$

where $d(x_i, x_j)$ is the distance between point $x_i$ and point $x_j$ and $\rho_i$ is the distance to the closest nearest neighbour of point $x_i$. $\sigma_i$ is the smoothed normalisation factor that ensures that:

$$\sum_{j=1}^{k} w(x_i, x_j) \stackrel{!}{=} \log_2 k. \tag{2.23}$$

The value of $\sigma$ is different for every point, as it depends on the KNN around this point, and needs to be determined such that Equation (2.23) is satisfied.

Since the weights between two points can be asymmetrical they need to be made symmetrical. However unlike t-SNE, where the average between the two weights is calculated, UMAP uses the definition of the probabilistic fuzzy union to combine the weights defined as follows:

$$w_{ij} = w_i + w_j - w_i w_j. \tag{2.24}$$

After having defined all of the symmetrical weights between the points it is finally possible to generate a low dimensional representation of the original topological space. To do this a low dimensional topological space is generated and optimized such as to depict the simplicial sets as closely as possible. The first graph for the low dimensional representation is initialized using spectral embedding (for more information on spectral embedding refer to [41, 42, 43]). Subsequently to optimize this representation the distance between the two representations is measured using the cross-entropy which is dependent on the following term $C$ to be minimized:

$$C = -\sum_{i=1}^{N}\sum_{j=1}^{k}\left[w_{ij}\log(v_{ij}) + (1 - w_{ij})\log(1 - v_{ij})\right] \tag{2.25}$$

where $v$ are the weights of the low dimensional embedding defined as:

$$v_{ij} = \frac{1}{1 + a\|y_i - y_j\|_2^{2b}}, \tag{2.26}$$

with $y$ being the low dimensional points and $a$ and $b$ being parameters resulting from an attractive and repulsive gradient expression that is influenced by other user defined hyperparameters. These two parameters $a$ and $b$ decide how tight the points in the low dimensional space can be packed together. Finally the optimization of the embedding is carried out by stochastic gradient descent (for more information on stochastic gradient descent refer to [44, 45, 46, 47]).

The main hyperparameters for the UMAP algorithm are the $k$ number of KNN, the minimum distance and the spread between the points in the low dimensional space. These parameters are very important for the performance of the algorithm and need to be set according to the problem set. In the case of the problems analysed in the thesis $k$ was set to 8, the minimum distance was set to 0.5 and the spread was set to 1. It is important to note that the size of the clusters and the distance between the clusters in the reduced space of UMAP is not representative of reality. These distances are exaggerated to emphasize clusters and highlight similarities and dissimilarities between points. This is very useful for visual analysis, but not necessarily when doing a sensitivity analysis for example. Compared to t-SNE the main advantage is more control over the results with the use of hyperparameters, but especially a better global representation of the data. While the distances between clusters and their sizes are not necessarily equivalent to the original space, the location relative to each other is much more accurate than when using t-SNE, where clusters are placed in totally random locations in the reduced space.

### 2.5.3 Modifications for better Clustering

As was mentioned in Section 2.3 it is sometimes possible that the UMAP algorithm includes intermediate points in unexpected clusters (i.e. the points were included in a different fuzzy simplicial set). This phenomenon is elaborated on in the next section (see Section 2.5.4 showing the application of the two presented methods on the dummy side crash example. As was explained in the previous section UMAP uses KNN to obtain a local simplicial set, which is the reason this happens. To avoid this, a modification to the UMAP algorithm is presented, which ensures that all points are included in the correct predefined clusters and additionally the clusters are very clearly separated.

Based on the fact that UMAP takes a similarity matrix as input, in this case containing the euclidean distances between all the points, this matrix can be modified to obtain the desired outcome. To ensure that all the points are included in the correct clusters and clearer cluster boundaries are defined, the similarity matrix is modified in way that all of the points who do not belong to the same clusters a further separated apart. This means if two points belong to the same cluster the distance between these

two points stays untouched in the modified similarity matrix. However if two points are not in the same cluster, the distance between these two points is increased in the modified similarity matrix. To make sure that all the points that are not in the same cluster are separated far enough, the distance is increased by adding the maximum distance between two points in the whole data set to the base distance. This not only ensures that the points which are not in the same cluster are separated further, but also that the structure within each cluster is maintained, as the distances between points in the same clusters stays the same.

The workflow of this method is as follows:

1. Cluster the data points based on different possible criteria,
2. Get similarity matrix by calculating euclidean distance between all the points in the data set,
3. Identify maximum distance between two points in the data set,
4. Add maximum distance in similarity matrix to any point combinations that are not in the same cluster.

To visualize this concept a plot is shown in Figure 2.17.

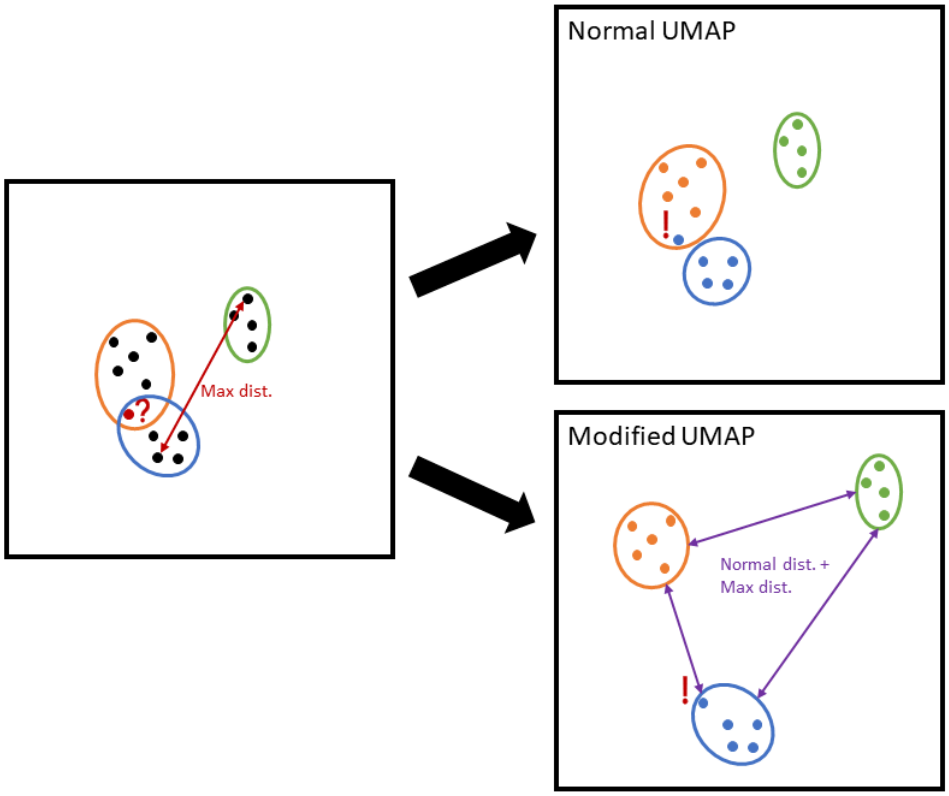


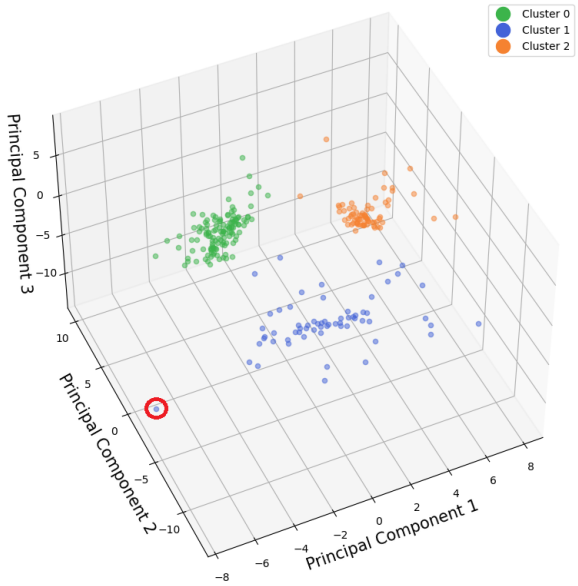

Figure 2.17: Modification of point distances for the modified version of UMAP.

To determine the clusters before doing the DR with the modified version of UMAP it is possible to use the data points in the original dimension. This means the points are clustered based on the data in the full dimension, before applying any DR technique. Another option is to use another DR technique to determine the clusters and then use these cluster classifications for the modified version of UMAP. Based on the data sets analysed in this thesis it is recommended to define the clusters in the original dimension.

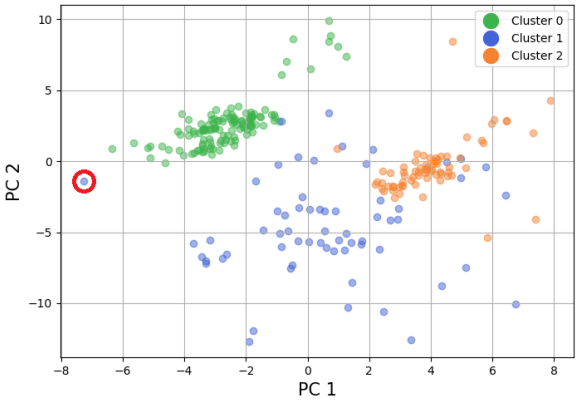
Notably, the distance between the clusters with this method is not representative of the real distance or similarity between the clusters in the original space anymore. However, the distance within the clusters still is representative of the similarity between the points.

### 2.5.4 Application on Side Crash of Dummy Car

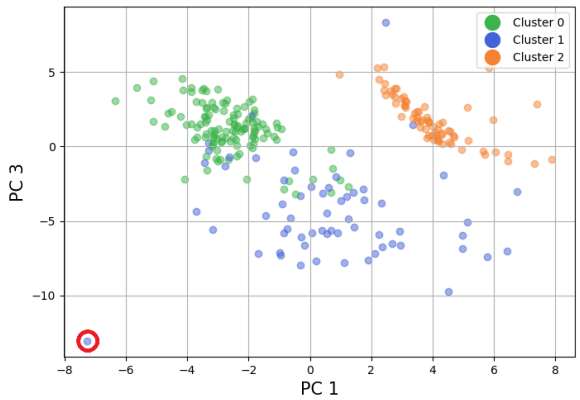
As a reminder of what the reduced space using kPCA looked like with the given clusters and results, it can be seen in the analysis of the initial results in Section 2.2 specifically in Figure 2.5. For the first comparison the MDS method is used with the reduced space being shown in Figure 2.18 and the plastic strain of the B-pillar in the results shown in Figure 2.19. The cluster order is different than for kPCA, but still three main clusters are detected. Because it is quite difficult to demonstrate a three dimensional plot in a screenshot the different axes are plotted in two dimensions as well to have a better understanding of the space.



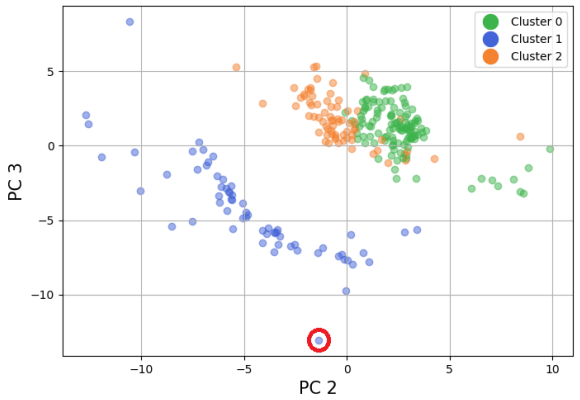

(a) 3D




(b) PC 1 - PC 2




(c) PC 1 - PC 3




(d) PC 2 - PC 3

Figure 2.18: Reduced space of the side crash model using MDS in three dimensions (a) and in two dimensions (b, c and d).

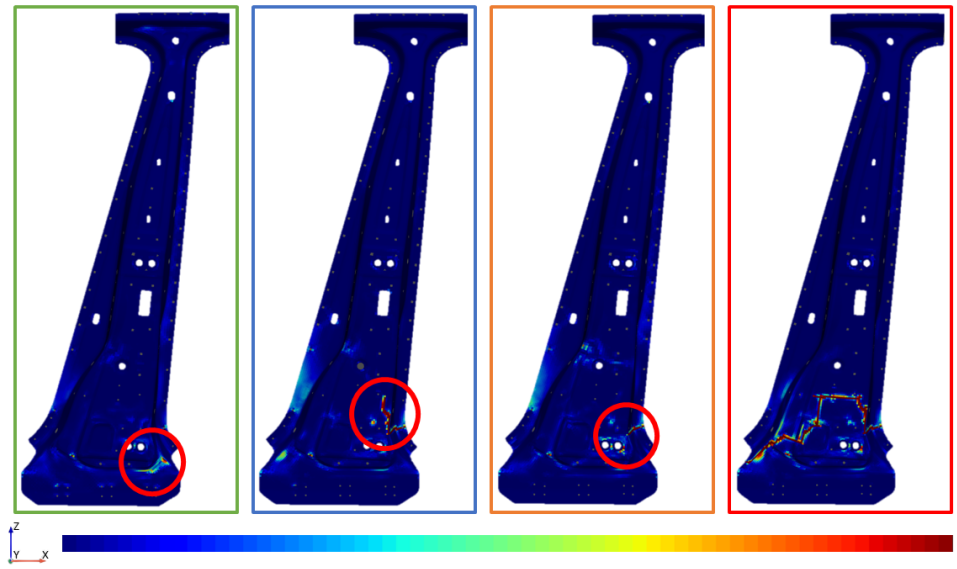
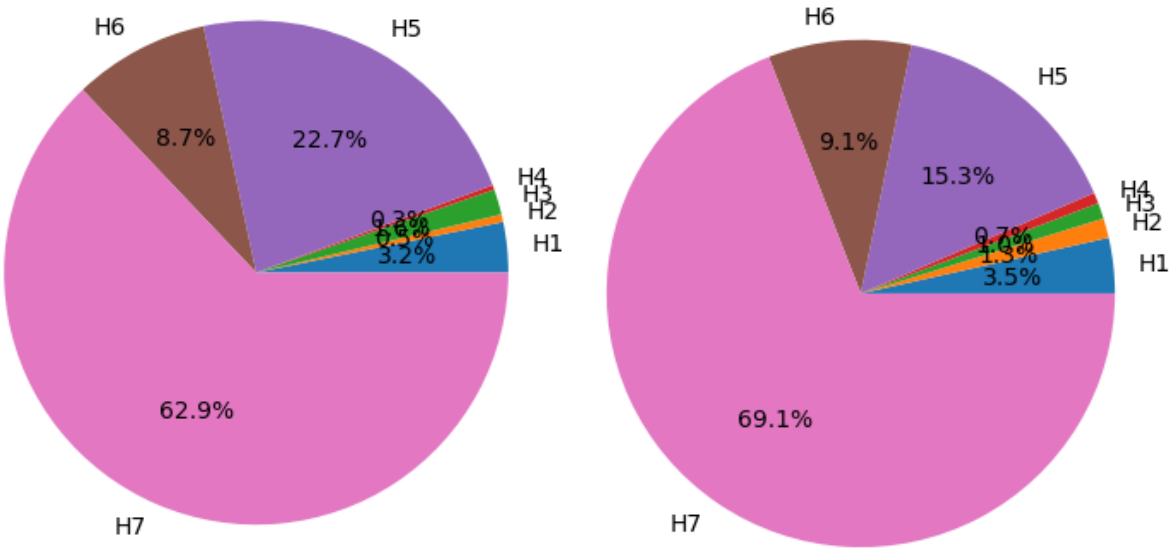


Figure 2.19: Plastic strain in the B-pillar for the reduced space generated using MDS. Each color frame around the B-pillar results represents the general behaviour of the results found in the clusters with the corresponding color.

Figure 2.19 shows the corresponding B-pillar plastic strain for the clusters detected in the reduced space. The characteristic behaviour for each cluster is the same as the one found in the initial analysis using kPCA as DR technique, however when using the MDS method the spacing of the original space is maintained better. This is especially visible when looking at the blue cluster, which was a lot more compressed using kPCA, but in reality the points are spread much further apart and the points within this cluster are more different than the points within the other clusters. In general within each cluster the same behaviour can be identified in specific regions of the cluster as shown in Figure 2.5 where an indentation at the top can be found for data points located on the upper range of the third PC. With MDS this behaviour cannot be assigned to a specific axis or PC, but it is still found in specific sections of the clusters. More interestingly though, and the reason why MDS is an improvement over kPCA, is the fact that the global distances from the original higher dimensional space are maintained in the low dimensional space. This means the found clusters and the spacing of the points truly reflect the similarity and dissimilarity of the points. With kPCA it was not possible to distinguish the points belonging to the orange cluster (the blue cluster in the MDS space) as the points are all very close together, which is not representative of the true distribution as can be seen in the MDS space where all of the points in the blue cluster are fairly far apart. This shows that the blue cluster has a considerable amount of uncertainty as the results in this cluster can vary more significantly than initially anticipated and are very unstable. This is especially important for the sensitivity analysis with the Sobol indices, as with kPCA the variance in this cluster is not captured correctly. The first order Sobol indices for both the kPCA and MDS methods can be seen in Figure 2.20.




(a) kPCA (b) MDS

Figure 2.20: First order Sobol indices of the side crash model using kPCA (a) and MDS (b).

In general the percentages of the first order Sobol indices are fairly similar for both DR methods except parameters h5 and h7. As was found in the analysis these two parameters have a very big influence on which cluster the parameter combination is classified in and where in the cluster it is placed. For the kPCA method the position in the third PC indicated whether an indentation was present at the top of the B-pillar or not and parameter h5 had a very big influence on that. Given the fact that with MDS this tendency is not present the first order Sobol index of parameter h5 is smaller, as the variance to get to the points with the indentation at the top is smaller as well, in other words the clusters do not have such an elongated shape as in the space with kPCA resulting in a smaller variance caused by parameter h5. On the other hand the first order Sobol index of parameter h7 increased. This can be traced back to the fact that the clusters are further apart in the space obtained using MDS compared to using kPCA, as parameter h7 is highly influential on which cluster the parameter combination is classified in. This shows that even though the general sensitivity of the parameters was captured correctly with kPCA, the DR is highly influential on the exact results of the Sobol indices, as the reduced space dictates the variance in the data points. For this reason it is essential to use a technique, like MDS, which tries to maintain the global structure of the original space as closely as possible, as this will result in the best and most realistic approximation of the variance in the sensitivity analysis.

In addition to that, the detection of outliers is significantly facilitated with the MDS method as the points are not artificially clustered together. For example, the point marked in red in Figure 2.18 and 2.19, which in the reduced space of MDS is clearly far away from the other points in the cluster, shows some very extreme behaviour with the

B-pillar completely ripping apart at the bottom. On the other hand in the kPCA space this simulation is packed in the middle of the cluster not being differentiated from the other results that clearly do not show such extreme behaviour. To illustrate this the same outlier simulation point marked in Figure 2.18 for the MDS space is also marked for the kPCA space in Figure 2.21.

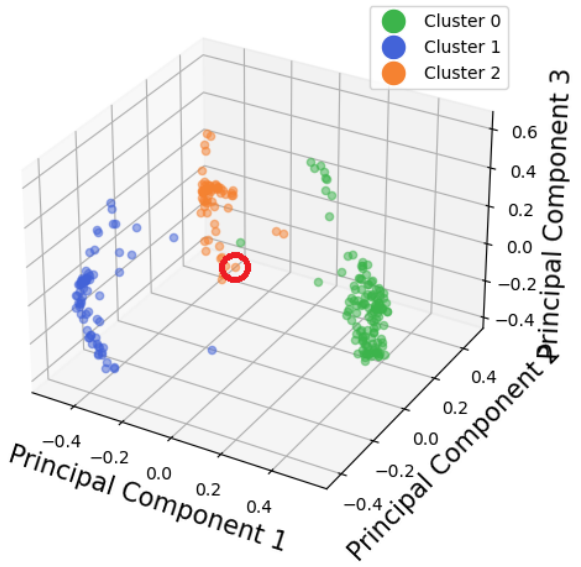



Figure 2.21: Reduced space of side crash model with kPCA and outlier marking.

Unfortunately there are some drawbacks to this method as well. Because the points in the blue cluster of the MDS space are more spread apart the SM error increases as the predictions are slightly less accurate in that cluster. However, it is questionable if this better accuracy is actually representative of better approximating the results, or if it is only due to the fact that all of the points are closer together and the error is therefore smaller, since the predictions will automatically be closer to the real points. Also the problem of how to assign intermediate points between clusters, which was mentioned in the analysis with kPCA already, still remains with this method. Sometimes it is not exactly clear where the cluster boundaries are and this can cause difficulties for the clustering and classification algorithms. Overall MDS is extremely useful for analysis purposes, as it gives a true representation of the distances and therefore similarities between the points and is more realistic for the sensitivity analysis when calculating the Sobol indices. Additionally it is very useful for outlier detection.

Now to tackle the problem of MDS that in some cases the cluster boundaries might not be very clearly defined, UMAP was introduced, which like MDS is supposed to maintain the local structure of the original space in each cluster, while forming clearer boundaries between the clusters. The reduced space obtained with UMAP using 8 KNN, as this was determined as the best value for this problem, can be seen in Figure 2.22.

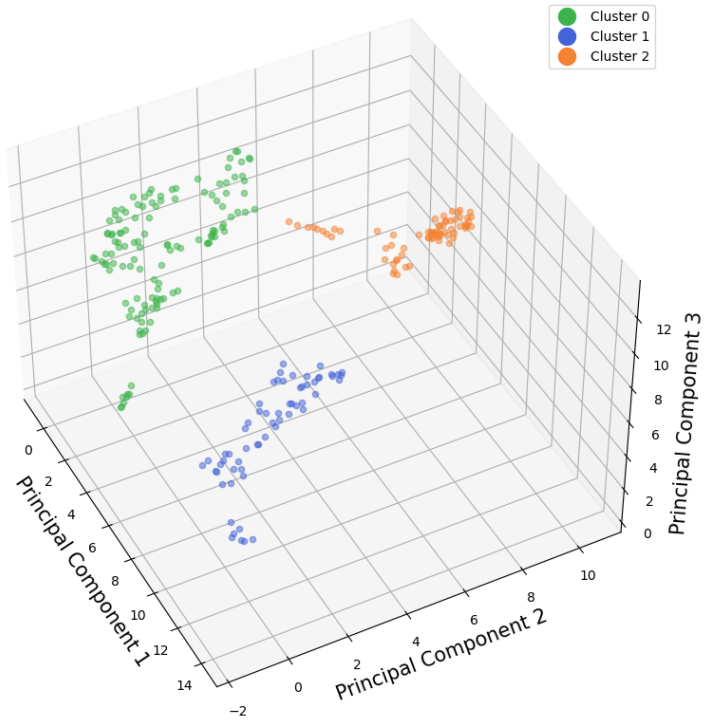
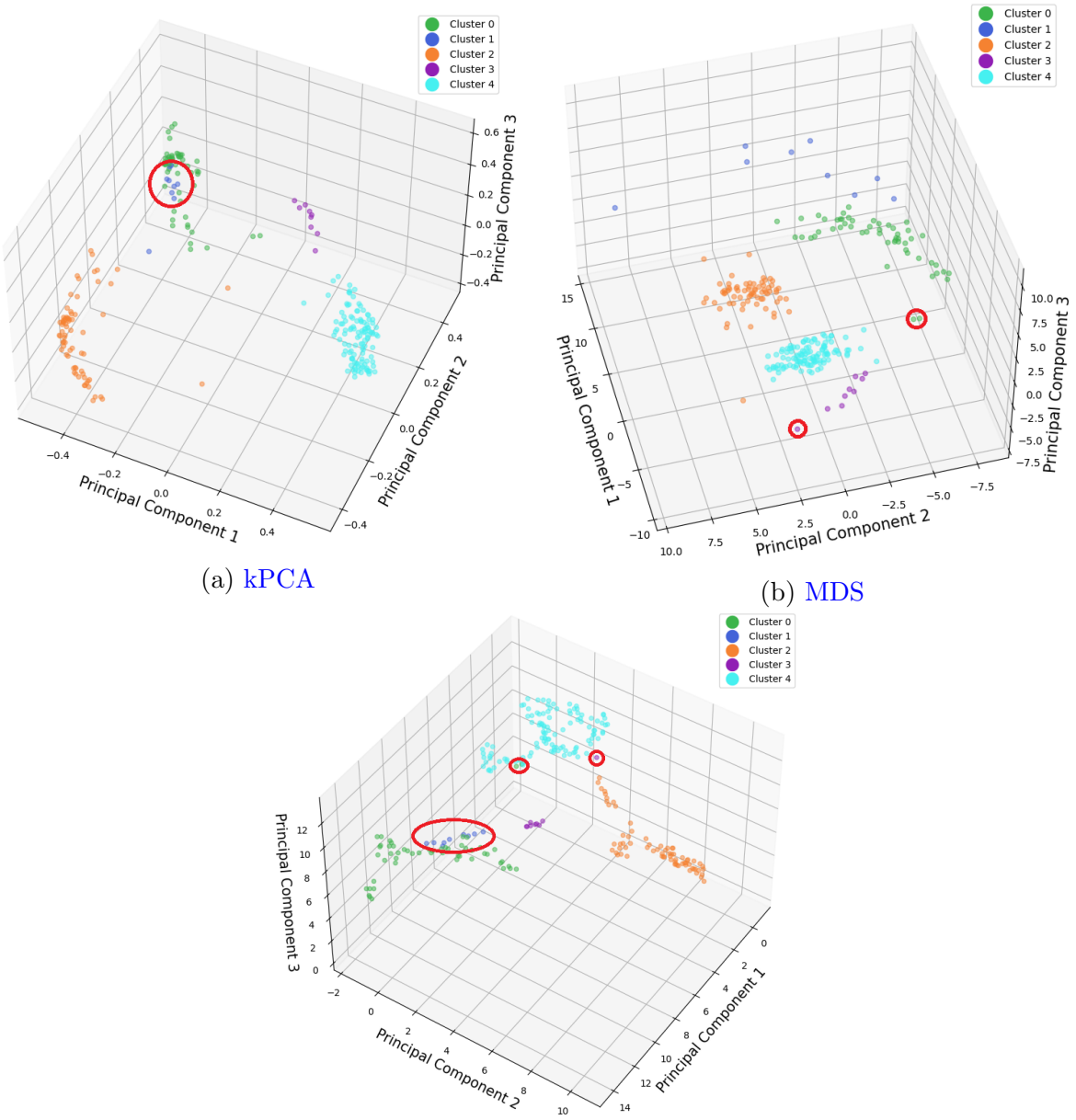



Figure 2.22: Reduced space of side crash model with UMAP.

The clusters are split into the same characteristic clusters as with MDS as shown in Figure 2.19. It can be seen that the UMAP method generates much clearer boundaries between the clusters, and the data points are better contained especially for the blue cluster. While the overarching structure of the points is still recognizable. Unfortunately it was found that the UMAP algorithm struggles with intermediate points and quite often will place some of these points in the wrong cluster. To show this more clearly the data points were clustered in the original dimension, meaning the full large dimensional data was used instead of the low dimensional data from the reduced space, and the reduced space was then plotted with the identified clusters from the full dimensional space. These plots, made for kPCA, MDS and UMAP, can be seen in Figure 2.23. To better visualize this point and try to differentiate the result behaviour even more it was chosen to cluster the data into five clusters instead of three. The clusters were generated in the full dimensional space for all three cases, so the clusters and data points in each cluster are exactly the same.




(a) kPCA

(b) MDS

(c) UMAP

Figure 2.23: Reduced space for the side crash model with clustering in the full dimensional space using kPCA (a), MDS (b) and UMAP (c).

The first thing to notice is that the blue cluster is very close to the green cluster in the space generated by kPCA. While the same is true for UMAP, the spacing between the blue and the green cluster is slightly larger, as in kPCA the blue cluster (encircled in red) is right in the middle of the green cluster. Comparing this to the space generated by MDS it is very different, as the blue cluster here is representative of the outliers that are spread far apart. The closest cluster is the green cluster here too, but the points are significantly further away than with the two other methods. This confirms the point about

how MDS is great for outlier detection and gives a good idea of what the original space looks like. However, comparing UMAP to MDS it is clear that the cluster boundaries are much better defined in UMAP and the points are more evenly spaced in each cluster. Additionally, a large number of details are captured within the clusters of the UMAP space. This is beneficial for the clustering algorithm, even though it is not relevant in these plots as the clusters were defined in the full space, and the regression model as well. Now it seems as though the space generated by kPCA has similar characteristics as the space of UMAP in regards to spacing and boundary definition. Apart from the fact that the spacing in kPCA does not exactly reflect the spacing in the original dimension whereas with UMAP it does, the main benefit is that no knowledge of the data structure is required with UMAP. The kPCA space is generated using a cosine kernel, which for the side crash example seems to be working quite well. However this is not necessarily the case for every model, which will be seen in Chapter 3, and sometimes no kernel might approximate the data structure properly, which makes UMAP superior to kPCA.

The two other red circles in the UMAP space are points that were pushed into the wrong cluster by the UMAP algorithm. UMAP is very prone to pushing intermediate points into wrong clusters, and this is clearly visible here. Two green points and one purple point are pushed into the turquoise cluster. And looking at the two circles in the MDS space, which represent these same points, it is immediately noticeable that these points are very close to the turquoise cluster, almost being in the middle between their corresponding cluster and the turquoise cluster. This is why UMAP failed to correctly place these points in the right clusters. To avoid this and ensure that all the points are placed in the correct cluster, while still being able to take advantage of the benefits of UMAP, the modified version of UMAP was developed. The resulting space using the modified UMAP version with the same clusters from the full space can be seen in Figure 2.24.

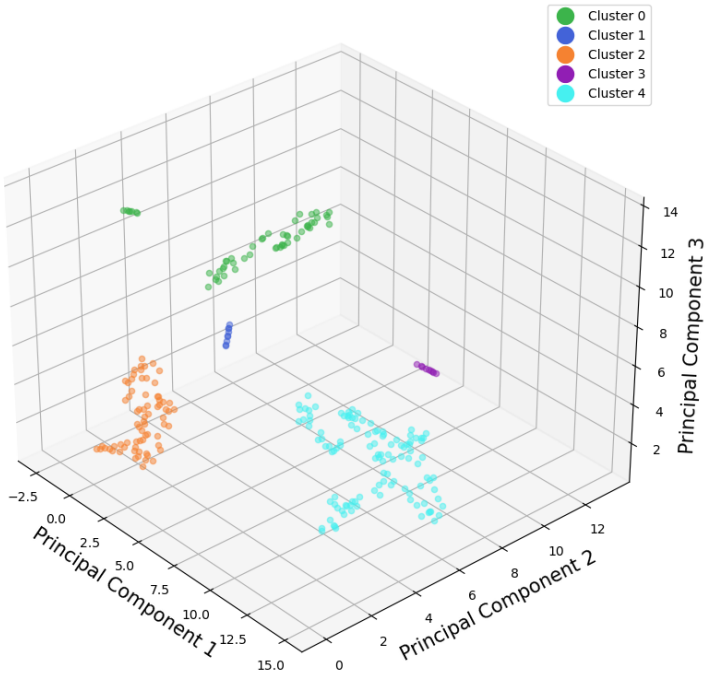
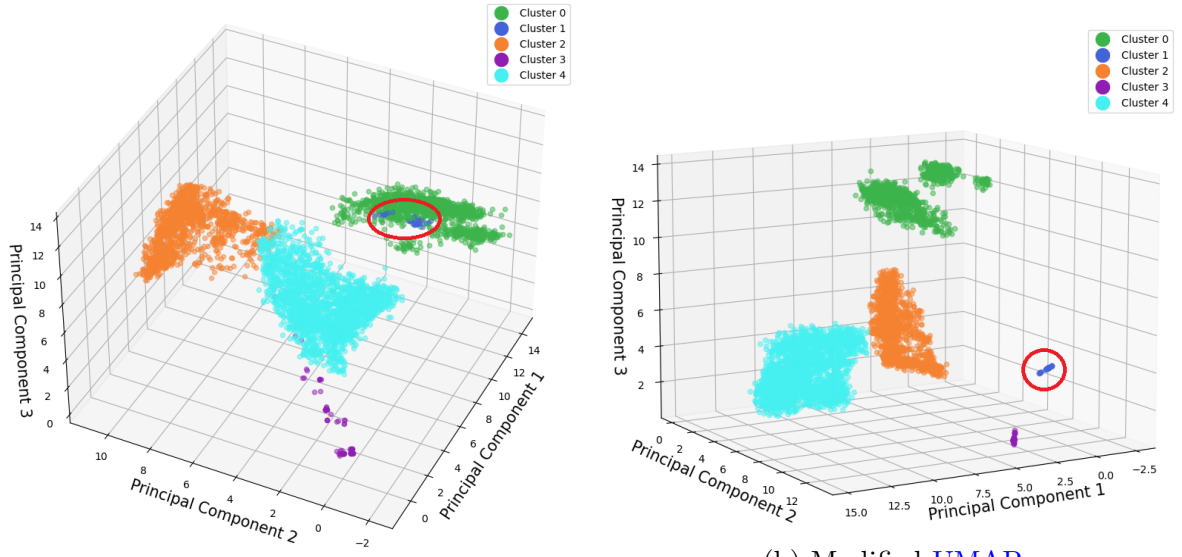



Figure 2.24: Reduced space of side crash model with modified UMAP.

With the modified UMAP algorithm all of the cluster boundaries are very clearly defined, with a considerable amount of detail being captured within each cluster, and no points being pushed into a wrong cluster. Such clear cluster boundaries are very important, as this ensures that the algorithm does not overlap any clusters which is detrimental for the clear distinction of points predicted by the SM and especially the backward mapping. This is both true for MDS and the regular version of UMAP, as could be seen with the blue and green clusters being so close to each other in both cases in Figure 2.23. To illustrate this the predicted MC samples are shown in Figure 2.25 for both the regular UMAP and the modified UMAP, as this phenomenon is more clearly visible with the regular UMAP, although it can occur with MDS as well. Even though this is very important for the clustering and backward mapping, it is worth noting that this does wrongly influence the results of the sensitivity analysis with the Sobol indices. It is extremely important to only do the sensitivity analysis with the MDS method, as it is the only method that truly reflects the variance of the points over the whole space.

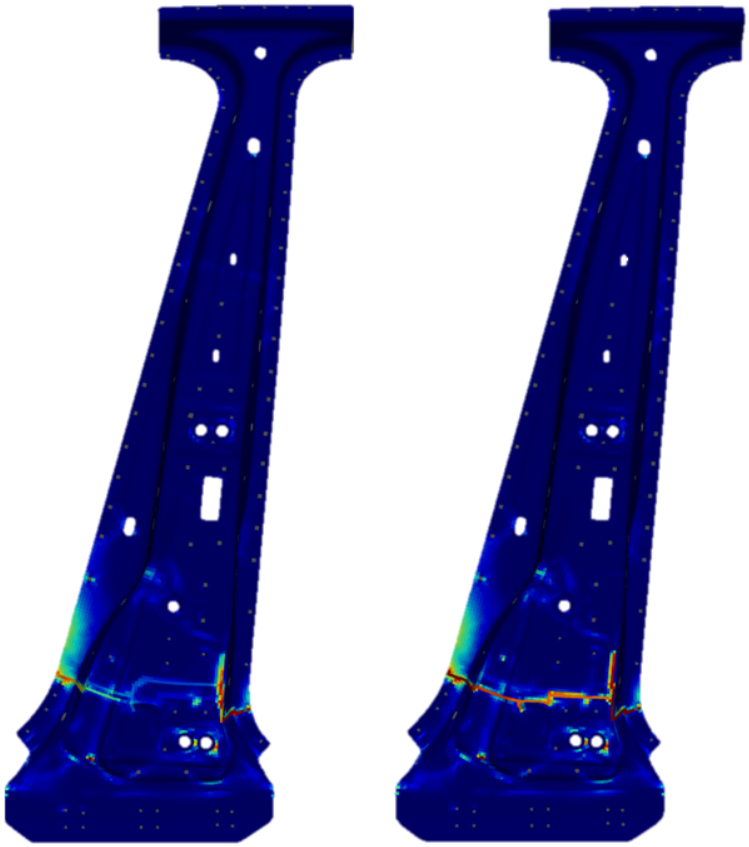



(a) Regular UMAP

(b) Modified UMAP

Figure 2.25: Reduced space of the predicted MC samples for the side crash model with clustering in the full dimensional space using regular UMAP (a) and the modified UMAP (b).

As can be seen the clusters in the modified UMAP version are not overlapping at all compared to the regular UMAP version. This ensures that the different result behaviours are better distinguished, especially in the backward mapping. For the backward mapping having the clusters so far apart enforces the interpolation with data points of the same clusters, whereas overlapping cluster points will be interpolated with points from different clusters resulting in blurred outcomes that are a mix of both clusters. To visualize this the backward mappings of a prediction in the blue cluster for the regular and modified UMAP versions are shown in Figure 2.26. The modified version of UMAP separates the blue cluster better, resulting in a much clearer backward mapping, while the regular version of UMAP overlaps the blue and the green cluster, resulting in a blurred backward mapping.



(a) Regular UMAP (b) Modified UMAP

Figure 2.26: Backward mapping of sample predictions for the side crash model in the blue cluster of Figure 2.25 using regular UMAP (a) and the modified UMAP (b).

As a last note it is important to mention that the distance between clusters for both the UMAP and the modified UMAP methods is not representative of the real distance between the clusters in the original full dimensional space. This means cluster similarities cannot be observed with these two methods. This is visible with the blue and green clusters for example, which are usually very close to each other, but the modified UMAP version placed the blue cluster further away from the green cluster.

### 2.5.5 Conclusion

In conclusion it can be said that MDS is a very good method to be used when the raw data needs to be analysed, in other words to get an exact representation of the spacing between all of the points from the full space in the reduced space and being able to easily identify outliers. MDS yields the closest representation as to how the points would be scattered in the original full space. However, the boundaries of the clusters with MDS are not always very clearly defined. This can be complicated when analysing

the predictions of the MC analysis, as the clusters can mix and overlap, and is especially problematic for the backward mapping. For this case the UMAP method offers a great solution, as it emphasizes the local distances in the clusters over the global distance of all points. This results in the natural formation of clusters within the reduced space with very clear cluster boundaries, facilitating the analysis of the clustered data and especially improving the clarity of the SM predictions and the backward mapping. Unfortunately this does not always work flawlessly as some intermediate points frequently find their way into wrong clusters. This lead to a modified version of UMAP where the clusters are defined beforehand based on different possible criteria, and the distances between the points in each cluster are modified in a way to guarantee the formation of the wanted clusters, without having any points slip into the wrong clusters. The recommended way to define the clusters is by finding the clusters in the original full dimensional space.

Overall it can be said that the best method to analyse the raw data, identify outliers and do the sensitivity analysis is MDS. However when analysing the MC results and especially for the backward mapping, meaning the generation of results in the original dimension, it is advisable to use the modified UMAP version with clustering done in the original full dimensional space. As a last note the sensitivity analysis should not be done with any other method than MDS, even not PCA or kPCA, as MDS is the only method that truly reflects the variance between all points, while the other methods generate an alternate reduced space that is not entirely representative of the original fully dimensional space. While having an altered reduced space does bring its own benefits as shown, it is not suitable for the sensitivity analysis.

## 2.6 Optimized Sampling

The last main improvement to be made to the AQUA algorithm is a self-optimizing training set. Instead of running new simulations with parameter combinations defined by the Halton sequence, the goal is to identify the output space that is least explored and specifically target this space with new simulations.

### 2.6.1 Iterative Cycle Algorithm

The concept of the iterative cycle is to start out with a standard input description **H** of 60 data points that is generated with the Halton sequence as done previously. Instead of enriching the training set by extending the input description **H** with more Halton points, the new parameter combinations are selected based on the points with the lowest classification certainty in the MoE approach. As a reminder to the MoE approach explained in Section 2.4, when doing the MC analysis with multiple clusters, first each sample needs to be classified into a cluster for the right SM to be used for the prediction of the sample. The classification methods give a classification certainty for each cluster

to each predicted sample point. To avoid points with a very low certainty to be classified into wrong clusters and distorting the results with potentially wrong classifications, a new cluster containing these "undefined points" is added. A sample will be classified as an undefined point when its classification certainty is smaller than the threshold value ($\alpha$) defined for the current training set. To obtain more accurate certainties it is worth pointing out that the classification model needs to be calibrated first.

To determine $\alpha$ it is first necessary to identify the right classification certainties and wrong classification certainties. The right classification certainties are the certainties with which a classification was done correctly, while the wrong classification certainties are the certainties with which a classification was done incorrectly. For these values to be determined, the data used to train the classification method needs to be split into a train and a test split. The classifier is then trained with the training set and tested on the test set. The certainties with which the test set was classified are the values assigned to the set of wrong classification certainties and right classification certainties. To obtain the threshold value $\alpha$ the cumulative distribution of both the right and wrong classification certainties are analysed. Instead of using the average, which can skew the results significantly and might not be able to correctly identify undefined points, $\alpha$ is determined with confidence scores of both certainties. In other words it is determined with what maximum certainty 95% of the wrongly classified points are assigned, and with what minimum certainty 20% of rightly classified points are assigned. This way it is possible to know with what certainty only 5% of wrong classifications should occur and with what certainty at least 20% of right classifications should occur. The threshold value $\alpha$ is then chosen as the maximum of both of these values to ensure the best results possible. The equation to determine $\alpha$ can therefore be written as:

$$\alpha = \max(c_w(95\%), c_r(20\%)), \tag{2.27}$$

where $c_r$ is the right classification certainty for a given cumulative probability and $c_w$ is the wrong classification certainty for a given cumulative probability.

This approach does not only consider the outright performance of the classification algorithm, but also its robustness, as it considers the confidence with which right and wrong classifications are made and specifically adjusts the threshold value to account for this. An illustration of this process is shown in Figure 2.27.

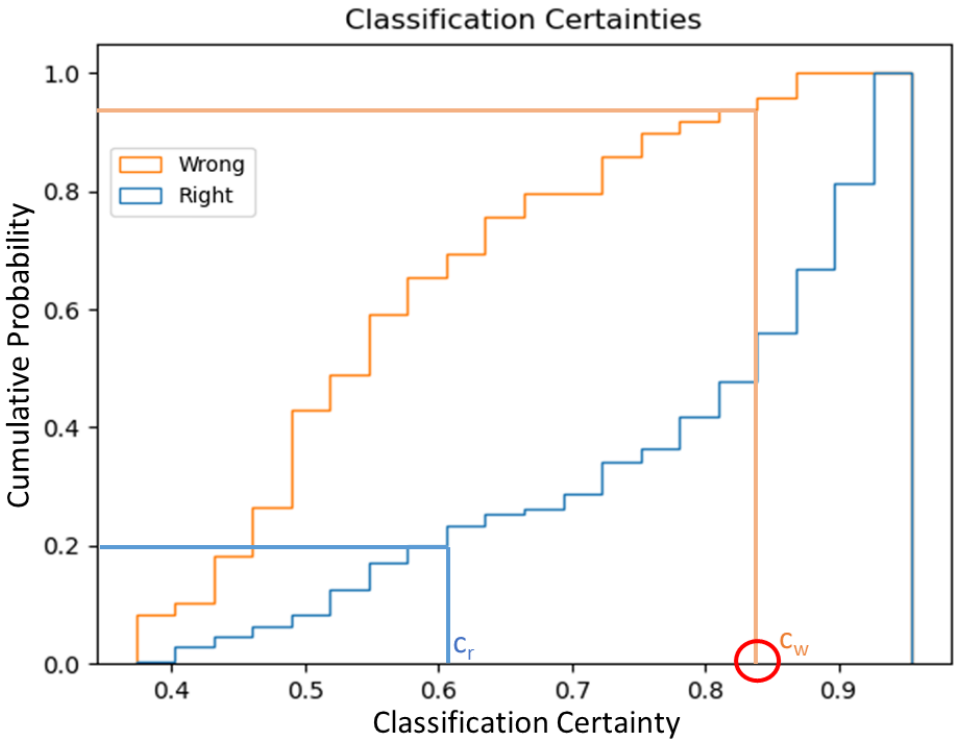



Figure 2.27: Cumulative distribution function of the classification certainties to determine the threshold value $\alpha$.

In the given example shown in Figure 2.27 $\alpha$ would be set to a value of 83% as this certainty, which is the maximum certainty for 95% of wrongly classified points, is larger than the minimum certainty for 20% of rightly classified points, with a certainty of 61%.

Importantly, to avoid any bias when determining the right and wrong classification certainties, the data is split into five different test and train splits with the final set of right and wrong classification certainties being the culmination of all the five splits. Additionally the split is done stratifying the data, to ensure an equal amount of data from each cluster is included in the train and test split respectively. Lastly, it is worth noting that this method is very conservative, as the final classification model is trained using all of the data and the points in the test set are undefined points of previous iterations, which means the final model used in the algorithm will perform much better than determined with the train-test-split. A flow chart of the process is shown in Figure 2.28. Notably the final classification model is still trained using all of the data. The train-test-split is only done to identify the approximate accuracy of the classification model and determine the classification certainties.

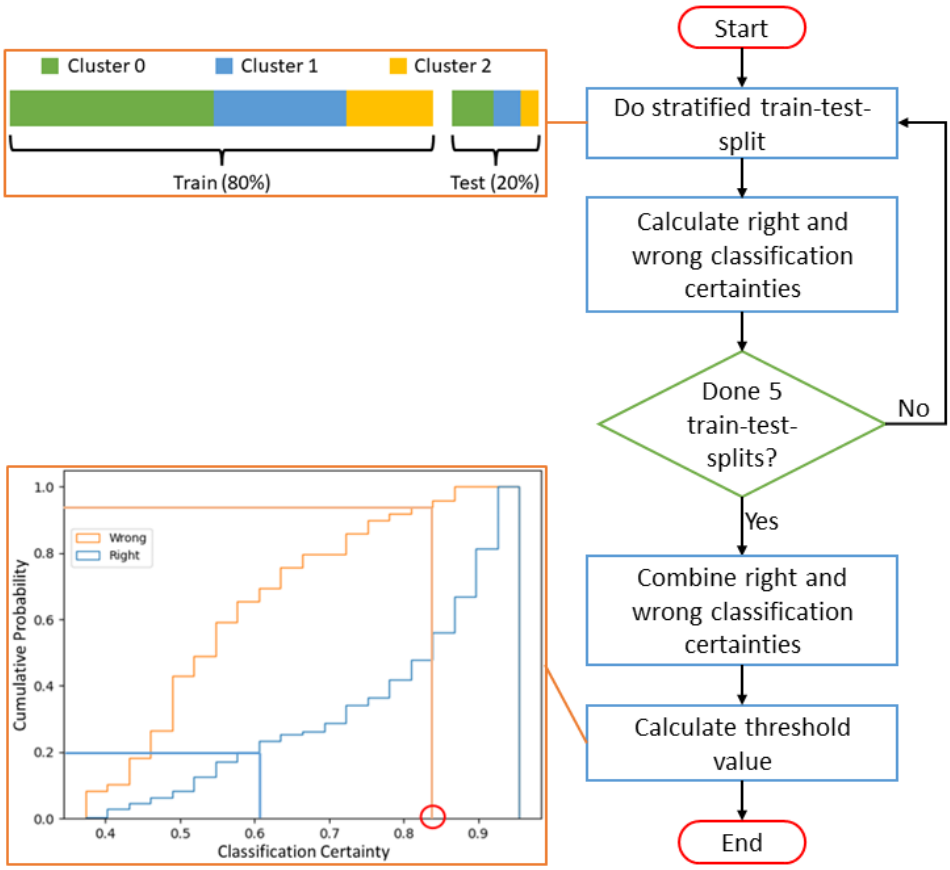



Figure 2.28: Flow chart of the process to obtain the threshold value $\alpha$.

After determining $\alpha$, all of the samples which are not classified with a certainty above that threshold value $\alpha$ will be grouped as undefined points. This means one additional cluster is added to the initial number of clusters. This is done, so that points that would normally be classified with a very low certainty, and therefore likely be classified incorrectly, do not distort the outcome of the final results. Additionally it gives an indication as to how well the classification model is trained and if the data set is rich enough to capture most of the important behaviours found in the respective simulation model. If the percentage of undefined points is very high, it means that the set of input parameters **H** does not contain enough parameter combinations to demonstrate all the possible results for the machine learning algorithms to be trained appropriately. On the contrary, if the percentage of undefined points is very low it means that the input description is very well defined and a substantial amount of the possible results is captured. It is important to keep in mind that this is only an indication, as it is still possible that most of the behaviours are captured, and the number of undefined points is very low, but there is not enough data to train the machine learning algorithm. This is particularly likely when the parameter combinations do not show a specific pattern for each cluster, as this makes training the machine learning algorithms more difficult.

Since the certainty of every sample can be measured like this, it is possible to identify the samples with the lowest certainty, which in turn should be the parameter combinations that the machine learning algorithms have the least information about. To specifically target the areas of least knowledge, to most efficiently improve the training data and in turn the machine learning algorithms, it is desirable to therefore select the most uncertain points to be simulated and added to the training set for the next cycle. This way the AQUA algorithm identifies the parameter combinations with the highest uncertainty, and therefore the least knowledge at the current cycle, to then expand the input and output descriptions with these specific parameter combinations for the next cycle. This process is then repeated, until the training data for the machine learning algorithms is well enough defined and the percentage of undefined points is small enough, which is the convergence condition shown in the flow chart in Figure 2.11. For the simulation models chosen here the percentage of undefined points necessary for the convergence condition to be met was set to 2 %. However if more than 20 % of samples are classified as undefined points the algorithm will expand the input and output description using Halton points, as it was decided that at this point too little knowledge is contained in the data set that a broader expansion of the data set is necessary.

To determine the most uncertain parameter combinations all of the samples that were classified as uncertain points are ordered by their classification certainty in ascending order. The eight points with the smallest classification certainty are then selected to be simulated and added to the input description $\mathbf{H}$ and output description $\mathbf{X}$ for the next cycle. Like this the points, which the machine learning model deems as having the least amount of information for, are added to the training model for the next cycle and parameter combinations which had little previous information are targeted specifically. As future work it could be interesting to calculate a threshold value for each cluster based on the classification certainties in each cluster, to even further improve the accuracy of the classification model, although this is even more difficult to implement with very few data points.

As a side note, if no clusters can be detected because the output of the simulation is continuous, then the convergence condition is based on the error of the SM introduced in Section 2.4.3 and the new data points are chosen using Halton points. Notably the SM error should be calculated with the MDS method as it is the most representative for what the error would be in the original dimension and the threshold value for the error is set to 0.01 in this case. In addition to that, when running this cycle algorithm it is necessary to automatically identify the number of clusters in the data. For this, the Silhouette method is used, where a Silhouette score is calculated for multiple number of clusters. In other words the data will be clustered into two to ten clusters, and for each number of clusters the Silhouette score is determined. The Silhouette score is a number between $-1$ and 1, with a larger value corresponding to a better cohesion between points in the same cluster [48]. This means the number of clusters with the highest Silhouette score is the number of clusters, which best represents the number of clusters found in the data. The advantage of this method is that it can be applied to any clustering technique

and compared to the elbow method, which is another technique to identify the ideal number of clusters, it can more easily be interpreted by the algorithm itself, as it is just necessary to find the maximum Silhouette score. The Silhouette score is the average of the Silhouette coefficients, which are calculated for every point in the data set. The Silhouette coefficient for a point $i$ is defined as follows:

$$s(i) = \frac{b(i) - a(i)}{max(a(i), b(i)} \quad \text{with} \quad -1 \leq s(i) \leq 1, \tag{2.28}$$

where $a(i)$ is the average dissimilarity of $i$ to all other points in the same cluster and $b(i)$ is the average dissimilarity of $i$ to all other points in the closest other cluster [48]. This of course means that the minimum number of clusters is two. It is assumed that there will not be more than ten clusters, which is why the analysis is only done for a number of clusters between two and ten. Notably if there are no clusters, this needs to be set by the user manually, which is why an initial analysis of the results needs to be done after the first 60 Halton points to determine if clusters are detected and if this automatic cluster detection or a predetermined number of clusters should be used.

Regarding the convergence condition, while it is true that the percentage of undefined points is a good indicator as to how much information is captured in the data, as well as being a good indicator of the performance of the classifier, it is not necessarily a good indicator for sufficient information being available to train the machine learning algorithms. This means that in theory enough information about the possible outcomes is contained in a relatively small data set, however it might not be enough data to train a machine learning algorithm. As machine learning algorithms usually rely on ample amounts of data to identify patterns, it can be difficult to train a machine learning algorithm with very few data points. For this reason it is still advisable to look at the score of the regression model as well. It is worth noting that the more important factor is the classification score, since if the right cluster has been selected, the regression model then works quite well, because the cluster selection is the more impactful decision. If the wrong cluster is selected, then the prediction of the regression model is guaranteed to be wrong, while if the right cluster was selected, the main behaviour of the results has already been predetermined by the cluster behaviour. For this reason it might be necessary to add more simulations to the data set, if the current data set is deemed to be too small.

In conclusion, the convergence condition is met when either clusters are detected and the percentage of undefined points is below the limit value, in this case set to 2%, or no clusters are detected and the SM error is below a limit value, in this case set to 0.01. The reason for not always including the SM error condition as convergence condition, even when considering the undefined points, is because if there are clusters the most important decision for the prediction is the correct classification into the clusters. While it is still important that the SM error is kept as small as possible, once the correct cluster has been found the most important information about the outcome of that parameter combination is already known. Within the cluster itself the SM error is not as

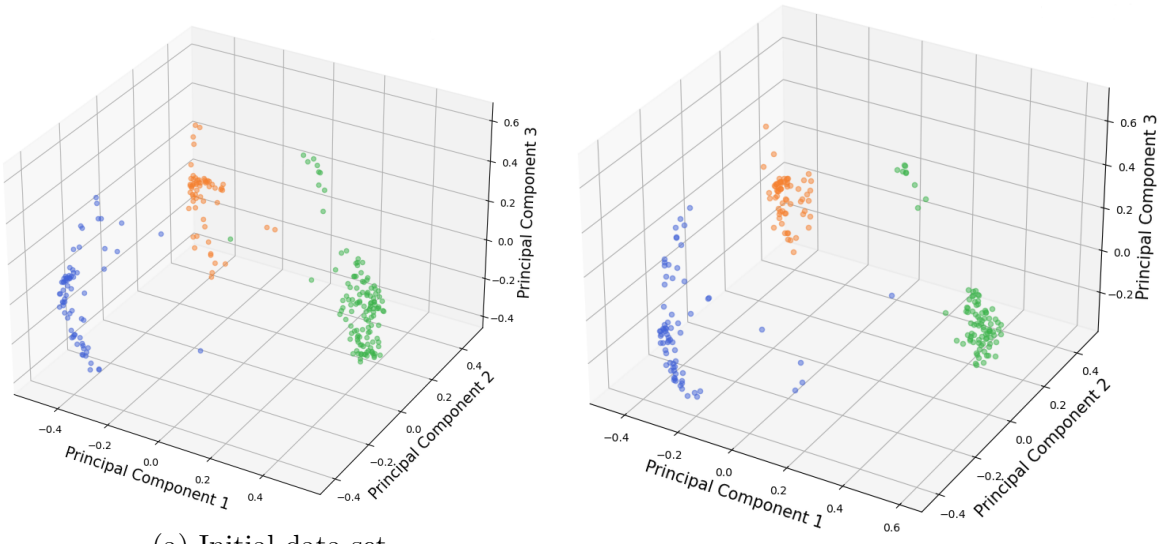
significant anymore, especially compared to a model only having one cluster, where the correct prediction of the SM is the only information given on the results.

When clusters are detected the recommended method to identify the clusters is to find the clusters in the original full dimensional space, as it is more accurate. When clustering in the original full dimensional space the DR technique is not significant for the cycle method until the final analysis step, as only the classification in the clusters is important, and not the exact location of the points in the reduced space. However, if the clusters are determined in the reduced space, the DR technique is extremely significant as discussed in the previous sections.

Like in the preceding sections the application of this method is demonstrated on the example of the side crash of the dummy car discussed in Section 2.2.

### 2.6.2 Application on Side Crash of Dummy Car

After initializing the iterative cycle and running the first 60 simulations with parameter combinations generated by the Halton sequence, 13.06% of the MC samples were classified as undefined points. The percentage of undefined points was calculated using the clusters identified in the original full dimensional space. In fact as expected from the initial analysis of the side crash in Section 2.2 three clusters were found. With the given percentage of undefined points the next parameter combinations to be simulated are defined by the MC samples with the least amount of certainty. As per the recommendations of the previous sections the clusters are obtained with the original full dimensional data. Convergence was finally met after 228 simulations with 1.72% of undefined points. At this point still only three clusters are found. The reduced space, using kPCA as DR technique, obtained with both methods is depicted in Figure 2.29.

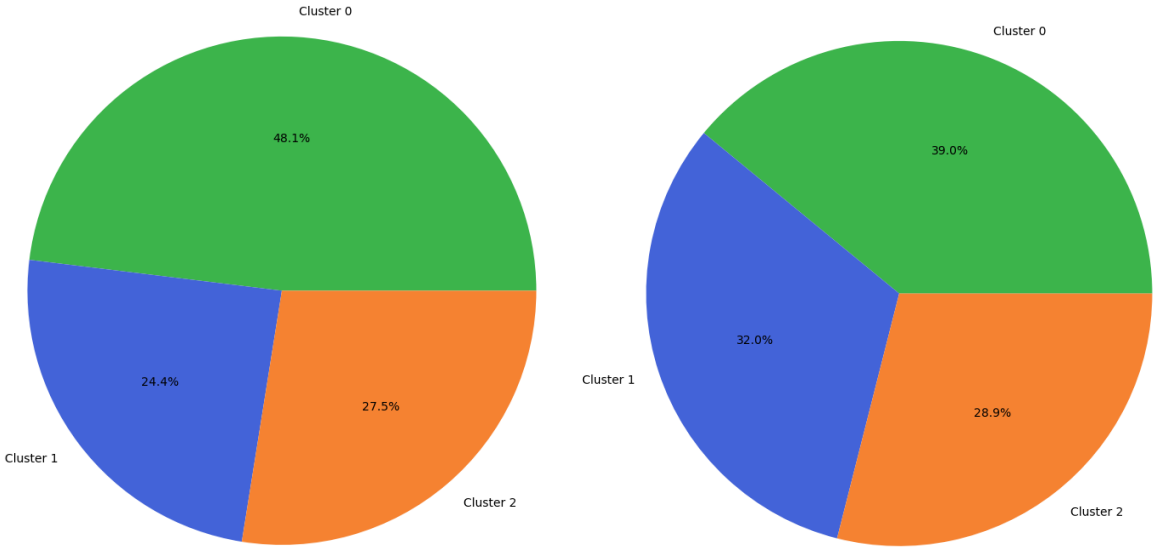



(a) Initial data set

(b) Iterative cycle data set

Figure 2.29: Reduced space using kPCA for the initial data set (a) and the iterative cycle data set (b) of the side crash simulation.

The number of simulations to reach convergence is already lower than the 255 simulations used in the initial analysis of the side crash. However, to really show the effect of the iterative cycle the percentage of undefined points for the initial simulation run needs to be compared. With 255 simulations the initial simulation run still has 7.98% of undefined points. This is because the Halton points do not lead to an equally spread out data set over the three clusters and it does not necessarily contain information about areas with low certainty like is the case with the iterative cycle set. To see this the number of points in each of the three clusters is shown in Figure 2.30.



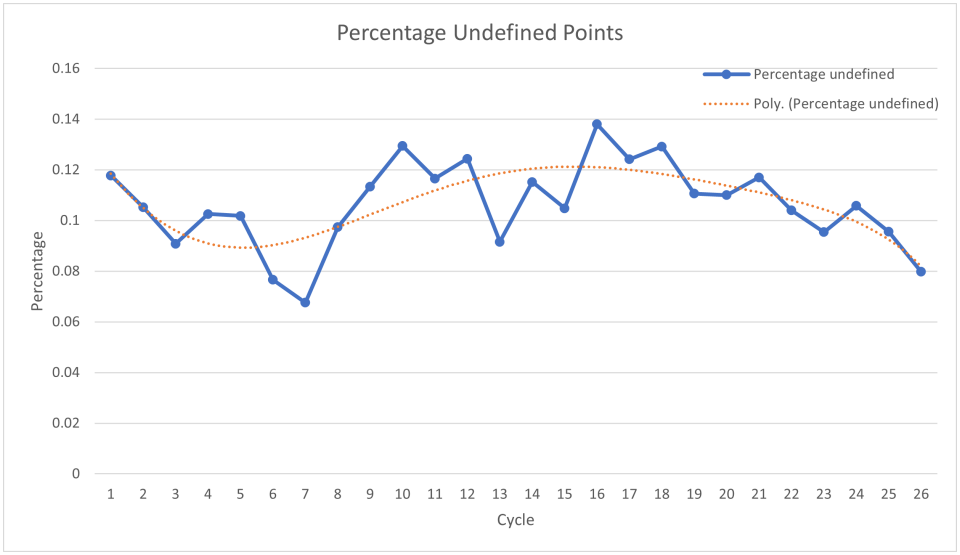

(a) Initial data set

(b) Iterative cycle data set

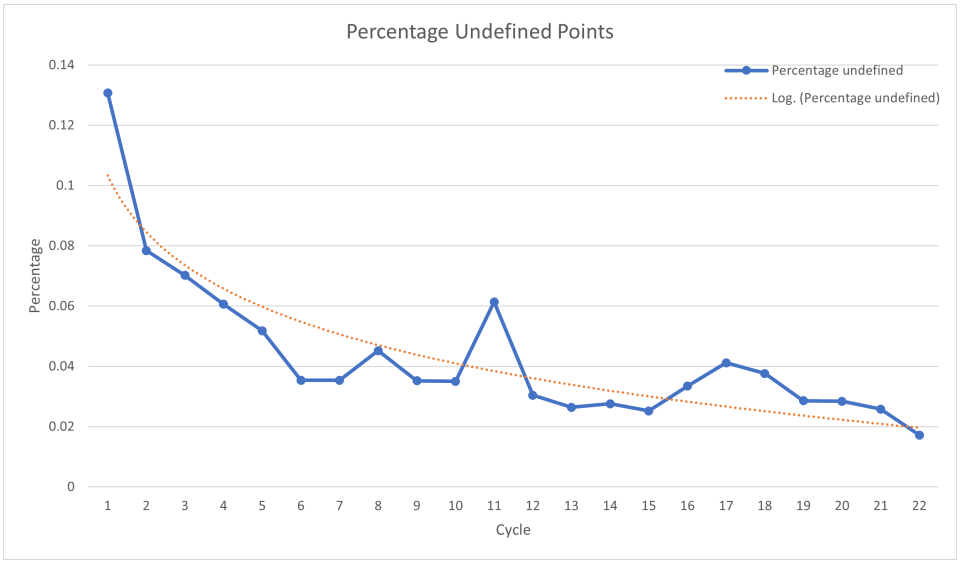
Figure 2.30: Cluster occurrences for the initial data set (a) and the iterative cycle data set (b) of the side crash simulation.

As can be seen the data set generated with the iterative cycle has a much more even distribution of points across the three clusters. This is extremely important for the classification method, as it ensures that enough data points are supplied by each cluster for the classification method to be trained properly. The initial data set clearly has a bias toward cluster 0 as this is the most common behaviour of the results. It is important to note that these distributions do not coincide with the final mode probabilities, as the parameter combinations for the simulations are generated with uniform distributions and not the real distributions found in the side crash. The real parameter distributions are used in the MC analysis and therefore result in completely different number of cluster occurrences. The more evenly distributed data points in the iterative cycle data set show that the idea of choosing the MC samples with the lowest certainty does work very well, as the clusters with smaller probabilities of occurring are actively targeted and enriched with this method. This method is a lot more efficient, as it is not necessarily to add new data points to the already well defined clusters, which is what happens in most cases when using the Halton sequence to define the new parameter combinations.

The convergence of the percentage of undefined points for both the initial data set and the iterative cycle data set are shown in Figure 2.31 to demonstrate the efficiency of the iterative cycle method.




(a) Initial data set



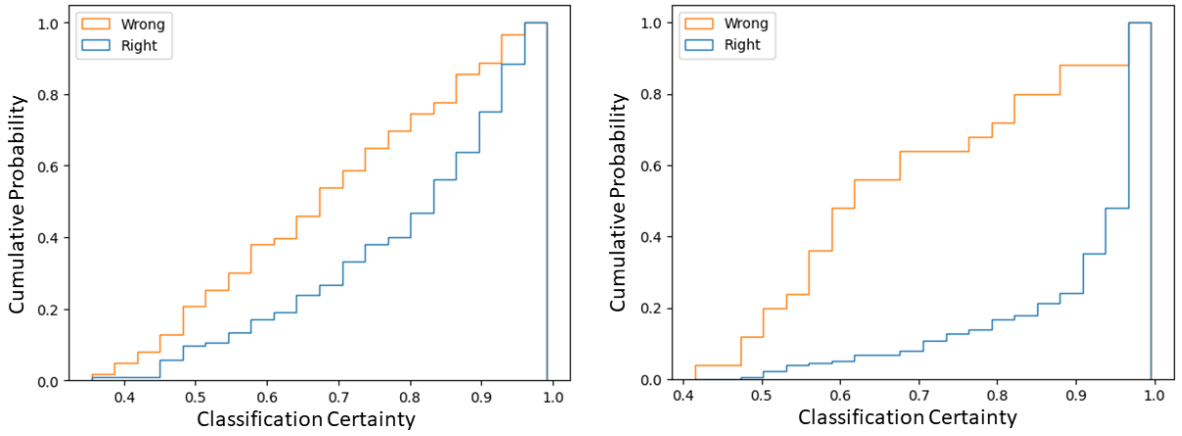

(b) Iterative cycle data set

Figure 2.31: Convergence plot of the percentage of undefined points for the initial data set (a) and the iterative cycle data set (b) of the side crash simulation.

The initial data set was approximated with a polynomial function of order six and the iterative cycle data set with a logarithmic function. It is clearly visible that the iterative cycle method converges quickly in the beginning and approaches the 2% value more slowly in the end, like a logarithmic function. The spike in the percentage of undefined points in cycle 21 for the iterative cycle method is because the algorithm detected four instead of three clusters, which increased the uncertainty in the model. The initial data set on the other hand does not show any convergence criteria so far. It would probably require a lot more simulations for the percentage of undefined points to stabilize and reach the 2% threshold value. The reason why the percentage of undefined points can go

up in the initial data set even though more data points are added is due to the fact that the classification algorithm becomes overconfident, because numerous points from the main cluster are included in the data set that are well defined. This is why the threshold value $\alpha$ is higher and more points are classified as undefined when running the MC analysis. So even though the initial data set contains more points than the iterative cycle data set, some clusters in the initial data set contain fewer points than the iterative cycle data set, which is the main cause of uncertainty in the classification of the initial data set.

It is difficult to compare how well one classification model is doing compared to the other when training with the initial data set and the iterative cycle data set, as there is no independent data set that could be used as a test set. To cross-check the two data sets the two classification models were used to predict the other data sets labels. Meaning one classification model was trained with the initial data set and tested on the iterative cycle data set and vice versa. The results show that when using the classification model that was trained with the initial data set and tested on the iterative cycle data set only 63% of points were classified correctly. On the other hand when using the classification model that was trained with the iterative cycle data set and tested on the initial data set 87% of points were classified correctly. This is not entirely representative of the models performances, but it definitely shows that the initial data set does not have enough data points in areas of low certainty to classify these points correctly. The iterative cycle data set on the other hand contains more points in areas of low certainty and enough points in areas of high certainty to get an extremely good classification score, considering the small amount of data points available. The classification certainties of the wrongly classified points and rightly classified points for both classification models are plotted in Figure 2.32, to show that the classification model trained with the initial data set is overconfident in areas of low certainty.




(a) Initial data set

(b) Iterative cycle data set

Figure 2.32: Classification certainties for the models of the initial data set (a) and the iterative cycle data set (b) of the side crash simulation.

Figure 2.32 shows how the model for the iterative cycle data set does the correct classifications with a very high certainty and the wrong classifications with a low certainty. In this case most of the wrong classifications should be filtered out by the iterative cycle method, as the difference in classification certainty between right and wrong classifications is very distinct. On the other hand, the model trained with the initial data set shows that a multitude of correct classifications have a low certainty and numerous wrong classifications have a very high certainty. This is exactly what the iterative cycle method tries to avoid, as with such classification certainties it is very difficult to filter out wrong classifications in the MC analysis. The reason for saying that the model of the initial data set is overconfident is because it has a very high certainty when making wrong classifications.

Using the threshold value $\alpha$ defined by the AQUA run with the iterative cycle data set to be $\alpha = 0.64$ and applying it on Figure 2.32b, it would mean that 60% of wrongly classified points would be filtered out and set as undefined points. Considering that the model had a classification score of 87% this means only $(100\% - 87\%) \times (100\% - 60\%) = 5.2\%$ of all points would be included in the final analysis with a wrong classification. It also needs to be taken into account that in this case approximately 7% of rightly classified points would be filtered out leaving $87\% \times (100\% - 7\%) = 81\%$ of all points included in the analysis with a right classification. This means the algorithm would have an accuracy of $\frac{81\%}{81\%+5.2\%} = 93.9\%$ in the analysed data set. Considering the little number of sample points in the training model this is an extremely good achievement and an improvement of almost 7% compared to the initial classification score. Additionally it should be noted that the data set that this was tested on is not representative of the samples used in the actual MC analysis for the side crash, as uniform distributions were used here. This means that the accuracy for the MC analysis with the iterative cycle data set is most likely even better, as more samples will be included belonging to cluster 0, which is very well defined in the model. A visual aid to these calculations is shown in Figure 2.33.

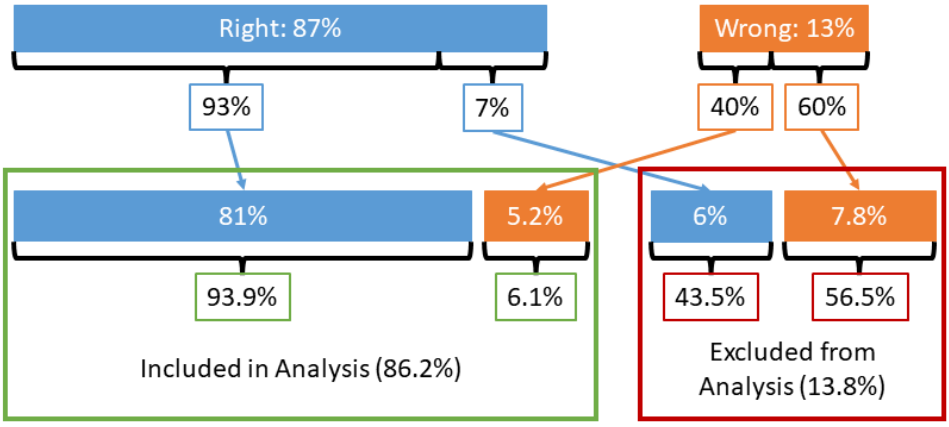



Figure 2.33: Accuracy of the classification model trained with the iterative cycle data set in the final analysis.

Going through the same process with the classification model trained with the initial data set a threshold value $\alpha$ of 0.85 is determined. This means that 80% of wrongly classified points are filtered out, but also that 55% of rightly classified points are filtered out. With a classification score of 63% this results in 7.4% of points with a wrong classification to be considered in the final analysis and 34.7% of points with a right classification. The final accuracy is then calculated to be $\frac{34.7\%}{34.7\%+7.4\%} = 82.4\%$. This is significantly better than the 63% accuracy that would have been captured if all the points were considered. However, this is still significantly less than the analysis of the iterative cycle model and only 42.1% of samples are left in the analysis. The iterative cycle model on the other hand still had 86.2% of samples left in the analysis, which is also an important factor to consider.

Overall this shows how the iterative cycle method significantly improves the convergence rate of the classification model and improves the efficiency, as better accuracy is achieved with less data points. Additionally it was shown how the method of excluding the undefined points from the analysis remarkably improves the accuracy for the final analysis.

### 2.6.3 Conclusion

Finally, it can be said that while improvements can still be made to the optimized sampling idea with different techniques of determining the threshold value $\alpha$ and doing a more specific classification analysis for each cluster, the newly introduced method works very well and significantly improves results. Using the iterative cycle as an optimized sampling method substantially reduces the number of simulations needed for the classification model to be trained properly. The new method results in a much more evenly distributed data set across clusters. The convergence condition is also more robust, as the previous condition used in [1], namely based on the Kullback-Leibler divergence, could be misleading if no parameter combinations are generated by the Halton sequence that result in a divergence of the data set. In addition to that, the new convergence condition directly measures the performance of the classification model. It not only verifies that all the different result behaviours are captured in the data set, like the previous condition was meant to do, but also that enough data points are contained in the data set to properly train the machine learning algorithms. Lastly, the method of filtering out points and classifying them as undefined points significantly improves the accuracy of the classification model in the MC analysis, as wrongly classified points are filtered out with the threshold value $\alpha$. While this method excludes correct classifications from the analysis as well, the improvement in accuracy outweighs the loss of correctly classified points. The improvement in accuracy is very important, as wrongly classified points in the MC analysis remarkably influence the results of the sensitivity analysis and uncertainty quantification. By excluding a small percentage of undefined points, even if a few correctly classified points are included, the performance of the AQUA analysis is considerably boosted.

## 2.7 Final Conclusion

In conclusion a few very important improvements were made to the AQUA algorithm that improve the overall analysis capabilities, efficiency and accuracy. The three main improvements include:

1. a MoE approach to avoid blurring the clustered data,
2. new DR techniques to better capture non-linear information and encourage cluster formation and
3. an optimized sampling technique to reduce the number of simulations needed and improve classification accuracy of the classification model used in the MoE approach.

First, the MoE approach introduced a way to detect and maintain clustered data within the AQUA analysis. When analysing the reduced space of the problem, different SMs are trained for each cluster instead of just training one, leading to much more accurate predictions and also enabling mode detection. It is possible to determine a probability for specific behaviour to occur, to see how likely it is that certain incidents take place when doing the real life process. Finally, it is possible to identify parameter combinations that lead to specific outcomes, as the parameter ranges can be analysed for the clusters, to find patterns in the parameters. With this information it is possible to improve quality control and manufacturing processes as critical parameter combinations can be identified and bad combinations avoided or good combinations promoted.

Next, new DR techniques were introduced with the goal of better capturing non-linear information in the data and encouraging cluster building for a better visual analysis of the MC samples and an improved backward mapping. The implemented methods are MDS, UMAP and a modified version of UMAP. All of these methods can capture non-linear relationships in data, however MDS maintains distances between all points, while UMAP favors local distances over global distances with the use of KNN. It was determined that MDS is the best method to analyse the data in its rawest form, as the distance between all the points in the original space is maintained as well as possible in the reduced space. Outliers can easily be identified and relationships between clusters are made clear. Additionally, this is the method of choice for the sensitivity analysis, as it is the only method where the distances between all points is representative of the original full dimensional space and therefore the variance between the points is also the best representation for use in the sensitivity analysis. When no clusters are detected MDS is also the recommended method, as the distributions of the points is the most realistic for the SM. However when clusters are detected, using UMAP enhances the cluster boundaries further, creating a better distinction between clusters. Especially for the backward mapping, meaning the generation of a result in the original full dimension

based on the coordinates of a data point in the reduced dimension, this is very beneficial. This is because clearly separated clusters ensure that the points in the same cluster are by far the most prominent influence in the generation of a backward mapping, and other cluster points do not blur this result. Additionally, this aids in the visual analysis of the MC results, as the clusters do not mix and overlap. Because UMAP is prone to placing points in wrong clusters some times the modified version of UMAP was developed, which ensures that all the points are placed in the desired clusters and the clusters are very clearly defined and spread apart. Since the two UMAP versions show a somewhat altered representation of the real distances between points, they should not be used for the sensitivity analysis. The general recommendation is to use MDS for the sensitivity analysis and to identify outliers and to use the modified version of UMAP for backward mapping and the visual analysis of the MC results.

The last main improvement is the optimized sampling method that uses an iterative cycle to specifically target areas of low certainty in the data set and enrich these areas in the next cycle. As a side note, it is worth mentioning that it is recommended to define the clusters in the original full dimensional space, as the clustering is more accurate this way. It was shown that by using the iterative cycle method the number of simulations, needed to properly train the machine learning algorithms and represent most of the possible outcomes, is significantly reduced with the iterative cycle method. Additionally, filtering the MC points and classifying them as undefined points remarkably improves the accuracy of the classification models by excluding wrong classifications from the analysis. This way the analysed data is not polluted with wrong classifications and the results of the AQUA analysis will be more accurate.

# 3 B-Pillar Stamping Process

Due to the results of the AQUA analysis for the side crash of the dummy car pointing out the significance of the B-pillar thickness, it is desirable to have a better understanding of this measure. The original analysis approximated this value as a uniform distribution throughout the whole B-pillar, which is not realistic. To get a more comprehensive insight into the shape and thickness of the part it is necessary to analyse the production process of the part. This will result in a more realistic representation of the B-pillar and its thickness mapping. The first part of this chapter focuses on understanding the stamping process of the B-pillar and the parameters of interest that will be used for the AQUA analysis. Subsequently, the outcome of the AQUA algorithm for the stamping process are presented and analysed finishing this chapter with a conclusion about the analysis of the stamping process and its possible applications.

## 3.1 Simulation of the Stamping Process

The stamping process for the production of the B-pillar is divided into eight main steps, which are the following:

1. Transport
2. Gravity and settling
3. Cooling on tools
4. Closing
5. Stamping
6. Quenching
7. Spring-back
8. Cooling on air

The first step is the transportation of the steel sheet that will be formed, also called blank, to the stamping machine. It is important to include this step, because the sheet has a very high initial temperature and is already starting to cool down at this stage and the temperature of the steel is an essential parameter during this process. The second step of the simulation is placing the sheet onto the lower die and letting the gravitational

forces settle the sheet into its final resting position before the actual stamping. Subsequently, one step is dedicated to the time the sheet spends cooling off on the tooling, while the stamping machine is initializing the stamping itself. The next step consists of the top die closing down onto the sheet. Notably the two dies are pre-heated to a certain temperature, so that the temperature shock between the matrices and the steel sheet are minimized. After the dies engage with the sheet the actual stamping of the sheet occurs with a predefined press pressure. Once the targeted press pressure is reached and the desired shape of the steel sheet is obtained it needs to be cooled in the quenching step, which is done for a predetermined time. Quenching is extremely important during the stamping process, as this step is essentially responsible for the final material properties of the B-pillar. While quenching the sheet, the tool stays closed with water running through the top and bottom dies to cool down the matrices and then in turn the part itself. The penultimate step simulates the opening of the dies and the spring-back of the sheet into its final shape. Lastly, the cooling on air is included in the simulation as well, as the steel sheet is still quite hot at this stage. The cooling still has an influence on the material properties and the shape of the B-pillar, as the steel sheet is settling into its final shape. To better visualize this process an illustration is shown in Figure 3.1. Due confidentiality concerns it is again not possible to disclose any specific values for this process. [7]

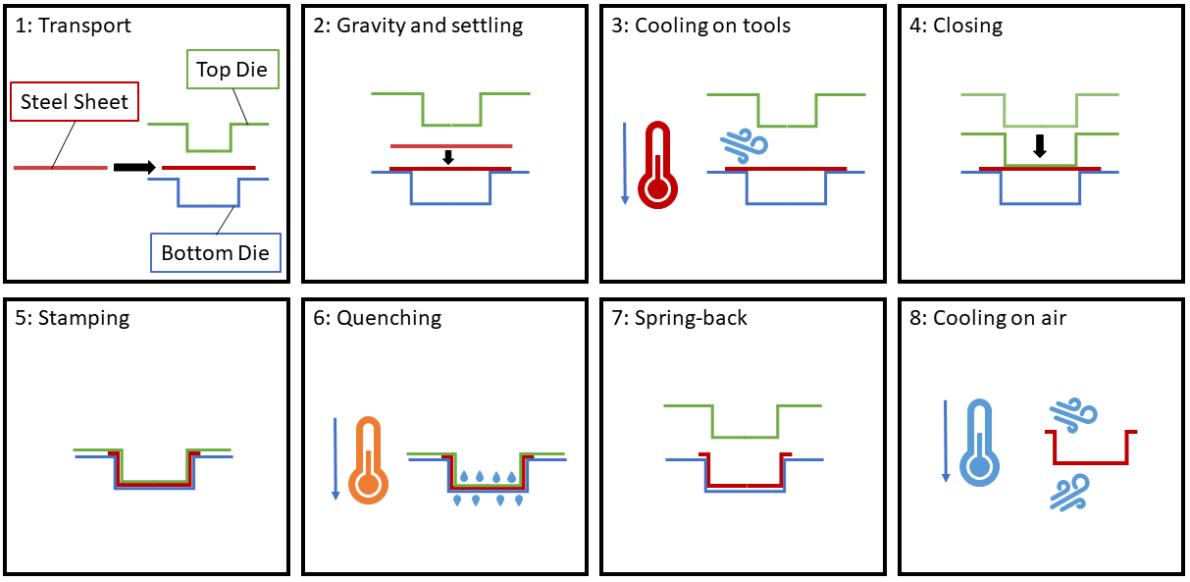



Figure 3.1: Simulation process for the stamping of the B-pillar [7].

The variables of interest for the stamping process, as determined by the manufacturing department at SEAT S.A., are listed in Table 3.1 [7].

| Parameter | Name | SI-unit |
|---|---|---|
| H1 | Initial sheet thickness | mm |
| H2 | Initial sheet temperature | °C |
| H3 | Transfer time | s |
| H4 | Time cooling on tools | s |
| H5 | Temperature of lower stamping die | °C |
| H6 | Temperature of upper stamping die | °C |
| H7 | Quenching time | s |

Table 3.1: Parameters of the stamping simulation to be varied in the uncertainty quantification analysis [7].

Unfortunately it was not possible to obtain any information about the true distributions of the stamping parameters found in production. Only the mean of each parameter is known, which is why a normal distribution with an 8% standard deviation and a truncation at a 10% deviation below and above the mean is assumed for all parameters. The distribution of all the parameters stated in Table 3.1 used for the AQUA analysis is shown in Figure 3.2. Due to confidentiality reasons the values are standardized, like in the previous examples.

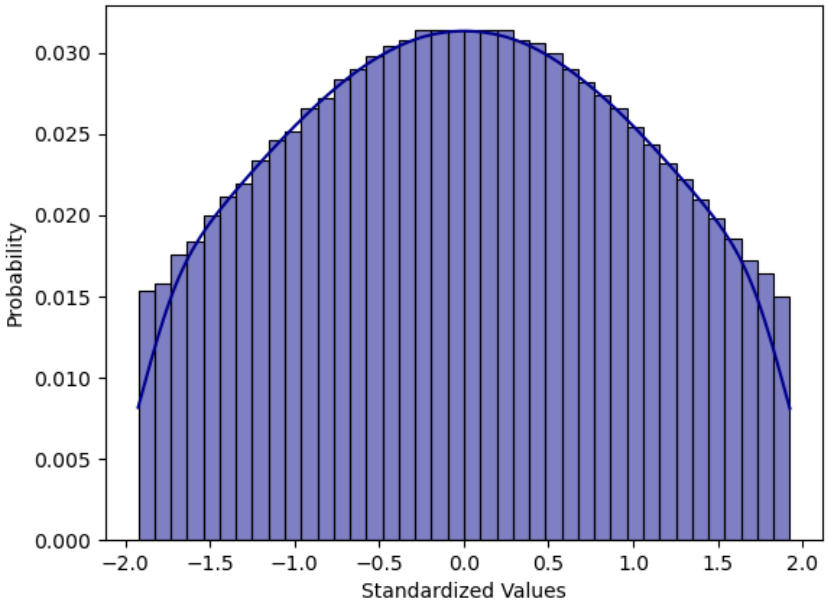



Figure 3.2: Probability distribution function with standardized values used for the stamping parameters stated in Table 3.1.

It is worth mentioning that the simulation model of the stamping process usually includes re-meshing the part during the simulation at various points in time to save computational time. It is not necessary to have an extremely refined mesh during the

whole process, as some parts of the process do not require as much detail in the results as others. For example the transportation step does not require such a refined mesh, as the whole part just moves together and is not deformed in any way only cooling down in a uniform way. The stamping step however requires a lot more refinement in areas of high deformation to capture all of the deformation, strains and stresses in the material with sufficient detail. Since for the AQUA analysis it is essential that all of the simulation results have the same output format it is not possible to allow re-meshing during the simulation. For that reason in this case the re-meshing is deactivated and the finest mesh necessary during the whole process is used for the entire process. This ensures that the final mesh of the part is always the same, no matter how many times the simulation is run.

## 3.2 AQUA Results

In the AQUA analysis of the stamping process the QoI is the thickness. Unlike the side crash example this means that the thickness, and not the plastic strain, of each shell element of the B-pillar is exported from the simulation results. The output description $\mathbf{x}$ of a simulation therefore is a vector with the same length as the number of shell elements in the B-pillar, containing the thickness for each of it.

After launching the iterative cycle version of AQUA for the stamping process the first step is to analyse the initial 60 simulations based on the parameter combinations generated with the Halton sequence. This is important to get a general idea of the behaviour of the simulation due to parameter variations and to see if any cluster building is occurring or if the results are continuous. The initial reduced space with the first 60 simulations can be seen in Figure 3.3. As per the recommendation made in Section 2.6 the first analysis is done with MDS, as it most closely represents the original full dimensional space.

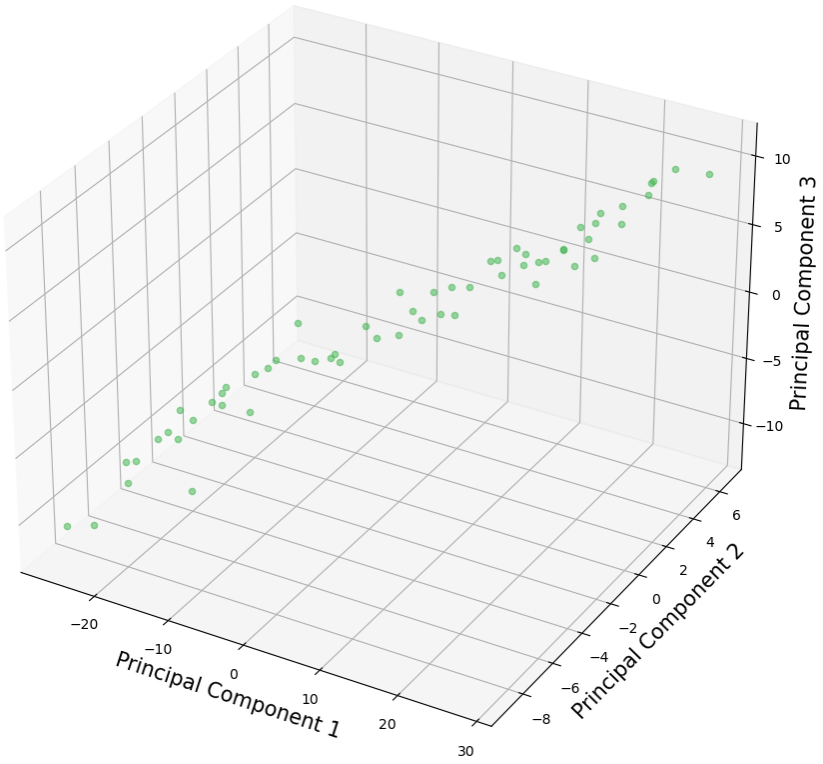



Figure 3.3: Reduced space generated with MDS of the stamping process for the first 60 simulations based on parameter combinations obtained with the Halton sequence.

It is clear that there is no cluster building in this model, unlike the side crash example, and the data is continuous. This gives reason to only using one cluster for this analysis and not apply the MoE approach. The convergence condition for this model is therefore based on the error of the SM and not the percentage of undefined points. Since the SM error is used as the defining value for convergence the DR selected is MDS, as it best represents the error to be found in the original dimension as stated in Section 2.6. Additionally, the new simulations will be generated by extending the Halton sequence. Using these conditions, convergence was reached after 148 simulations, where the SM error was smaller than 0.01 with a value of 0.0092. The final reduced space for the analysis with all the 148 data points can be seen in Figure 3.4.

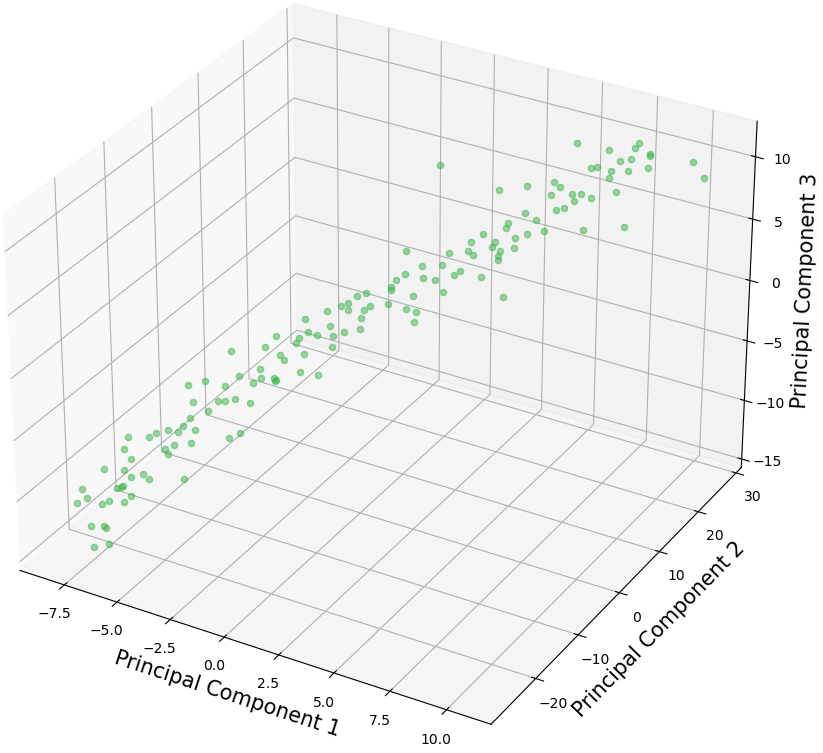



Figure 3.4: Final reduced space generated with MDS of the 148 simulations for the stamping process.

The final space looks evenly distributed, which is desirable for the predictions of the SM to work well. There are no clear outliers to be found, which leads to the assumption that this simulation model is very robust and the results are continuous without any mayor differences in the results behaviour, like seen in the side crash. In fact the reduced space is almost linear with the results increasing in thickness gradually along the line of data points in the reduced space. At one end of the line the thinnest B-pillar is found, while at the other end the thickest B-pillar is found. Some of the results are shown in Figure 3.5. The exact values of the scalar bar cannot be shown due to confidentiality, however the red end of the spectrum represents thicker values and the blue end represents thinner values.

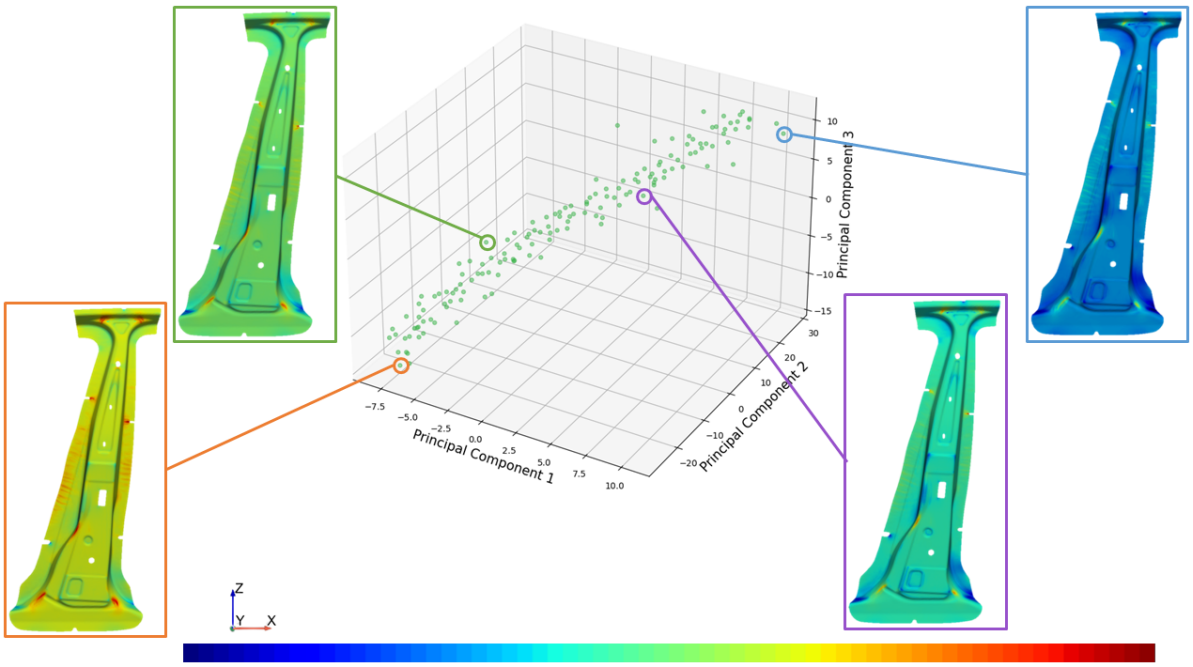



Figure 3.5: Thickness mappings of the B-pillar in the reduced space of the stamping process.

Figure 3.5 clearly shows a linear progression of the results along the recognizable line in the reduced space. The most reasonable presumption would be to correlate the main cause of variance in this model on the initial sheet thickness. To confirm this a look at the Sobol indices, presented in Figure 3.6, can be taken.

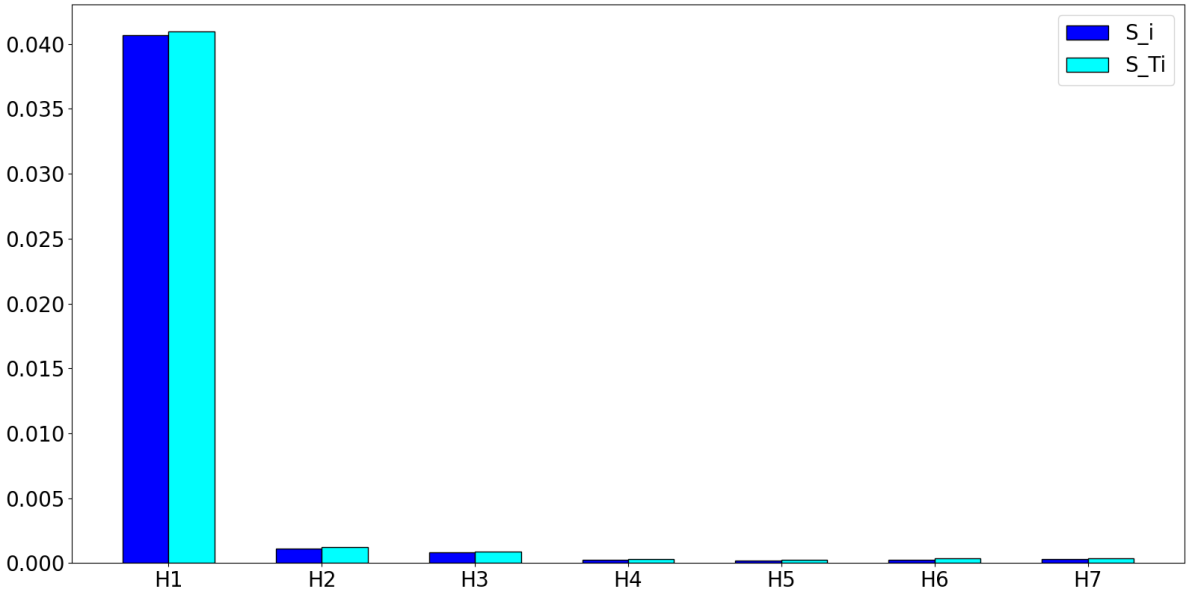



Figure 3.6: Sobol indices for the stamping process.

It is clearly visible that the initial sheet thickness is the most important parameter for the stamping process. This was expected, since a thin initial sheet would most likely not result in a thicker final part afterwards. All the other parameters have quite a small influence compared to the initial sheet thickness, as these parameters likely have a higher influence on other attributes of the B-pillar, like the hardness and crystalline structure of the material. It would be interesting to analyse these aspects of the B-pillar as well, however this is not done in this thesis. If four parameters had to be chosen with the highest influence in this model it would be the first four parameters.

Just to see what the reduced space would look like with UMAP as DR technique, the results are shown in Figure 3.7.

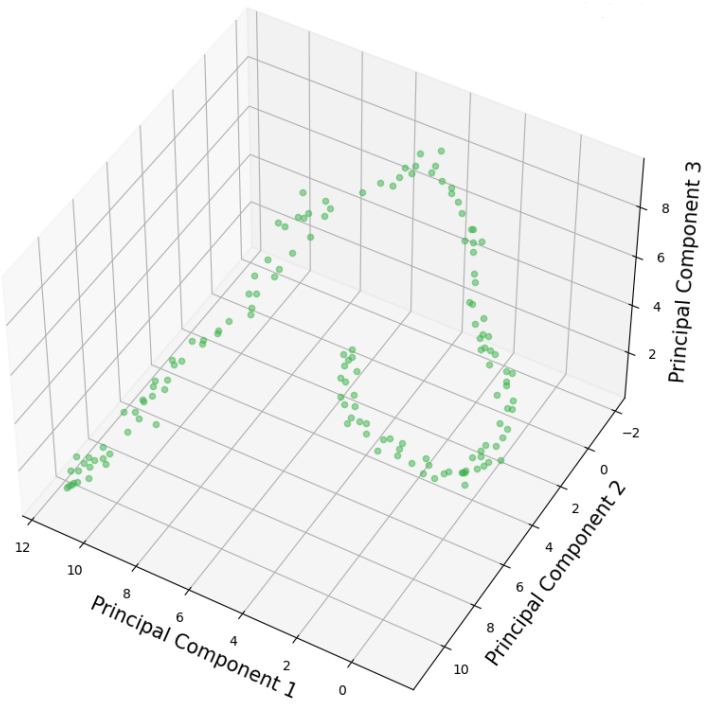



Figure 3.7: Reduced space of the stamping process using UMAP.

The reduced space generated with UMAP shows a very distinct spiral that, much like with MDS, has the results increase in thickness from one end to the other. Although it is not a straight line, the behaviour is similar to that of MDS, since there is only one cluster. The main difference is that the data points in the UMAP space are packed more closely together and not spread as far apart along the line as in the MDS space. Additionally, the line of points is not straight like with MDS but spiraling through the space. The differences are not very significant and both methods could be used for the analysis of the stamping process. Mainly the Sobol indices are affected slightly by the spiraling nature of the UMAP space, but since all the other parameters except h1 do not have a very big influence this is not noticeable. In the end, for further analysis and

applications the MDS method was selected, as it exhibits slightly more variation along the line, potentially including a bit more detail in the backward mapping.

Lastly, this data set is a good example to show, why kPCA is not always easily applicable on any type of data. Using a cosine kernel, like for the side crash example, yields very bad results for the stamping data set. The reduced space with some results showing the thickness mappings of two neighbouring points is depicted in Figure 3.8.

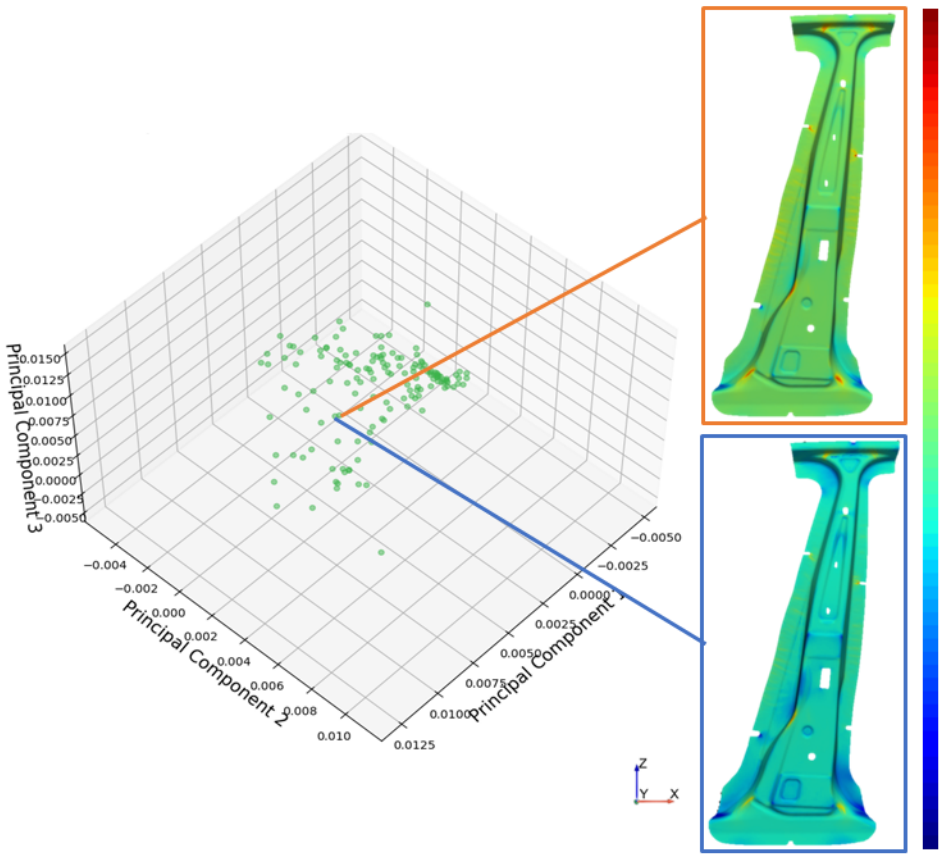



Figure 3.8: Reduced space of the stamping process using kPCA and two thickness mappings of neighbouring points.

Figure 3.8 shows that two adjacent points in the reduced space generated with kPCA have completely different thickness mappings. This is not desirable in the reduced space, especially considering that the other techniques were able to capture the continuity of the data. This could be resolved by changing the kernel function, however it is not guaranteed that a kernel function exists that can capture the given data structure correctly. Additionally, it requires more parameter tuning that can be avoided with the other methods.

## 3.3 Conclusion and Further Applications

In conclusion, the AQUA analysis of the stamping process shows a clear continuation of the results going from thin to thick. This is quite a contrast compared to the side crash analysis where the results were discontinuous, showing very different behaviour with no continuous evolution from one result behaviour to the other. As expected, the most important parameter for the thickness of the B-pillar in the stamping process is the initial sheet thickness, as it sets the base for the entire process. However, the discoveries made by running the simulation of the stamping process and doing an AQUA analysis with it are still insightful. It is obvious that the initial assumption of the side crash analysis is not good, as the thickness mappings found in this analysis clearly show thinner and thicker areas throughout the B-pillar, especially around the edge of the part and in areas of high deformation like corners.

The next step would be to implement the thickness mappings found in this analysis in the side crash simulations to include a realistic representation of the B-pillar instead of assuming a uniform distribution of the thickness. This topic is addressed in Chapter 4. In addition to analysing the thickness of the B-pillar it would also be interesting to analyse the hardness of the B-pillar. As was found in the AQUA analysis of the side crash in Section 2.2 the hardness of the B-pillar is also very influential (see Figure 2.8). This parameter is also approximated with a uniform distribution throughout the whole B-pillar, which is not very realistic as is the case with the thickness. Theoretically the same analysis process could be applied for the hardness of the B-pillar, exporting the hardness in each shell element in addition to the thickness. However, due to time restrictions this was not tackled in this thesis and is left for future work.

# 4 Coupling the Stamping Process and the Side Crash

As was noticed during the first analysis of the side crash in Section 2.2 the thickness of the B-pillar has quite a significant influence on the results. However, the thickness was approximated by a uniform thickness throughout the whole B-pillar in that initial analysis, which is not realistic. To get a more realistic representation of the thickness mapping in the B-pillar the stamping process was simulated and analysed with AQUA. Since a better understanding of the B-pillar thickness and a SM that is able to generate new results instantly is available, the goal is to directly couple the SM of the stamping process with the AQUA analysis of the side crash to include more realistic implementations of the B-pillar thickness in the analysis. This chapter will first explain the procedure in more detail as to how the coupling of the stamping process and the side crash is done. Subsequently the results obtained by the coupled model are presented and compared to the results of the regular side crash analysis with the uniform thickness of the B-pillar and a conclusion is given at the end.

## 4.1 Procedure

The general idea of coupling the stamping process of the B-pillar with the side crash is to include a more realistic representation of the B-pillar in the side crash simulation. Instead of having a uniform thickness, a realistic thickness mapping is used. The first step is to generate a SM for the stamping process that is able to generate the thickness mappings from the stamping process for new parameter combinations. This SM is then used to generate the according thickness mappings instantly for the side crash model. Instead of having the seven input parameters of which one was the uniform thickness of the B-pillar like in Section 2.2, the uniform thickness parameter (parameter h1 in Section 2.2) is replaced by parameters describing the stamping process. Instead of having only parameters related to the side crash model, the coupled model will have input parameters related to the stamping process and the side crash.

After analysing the results of the AQUA algorithm for the stamping process in Chapter 3 and the side crash in Section 2.2 with the respective initial parameter combinations, the most important parameters were selected for the coupled model. It is desirable to filter out the parameters that are not influential, as a higher number of parameters results in an exponentially higher number of possible combinations that need to be analysed.

This is known as the curse of dimensionality and would result in a lot more simulations being necessary for the problem to be described appropriately [49, 50]. Additionally, this can cause the machine learning algorithms to be over-fitted, as it becomes more difficult to find correlations between the parameters. Based on the previous results of the two AQUA analysis, four parameters from each simulation model were chosen for the coupled model shown in Table 4.1. The parameters taken from the initial stamping process analysis are the first four and the parameters taken from the side crash are parameters h3, h5, h6 and h7.

| Model | Parameter | Name | SI-unit |
|---|---|---|---|
| Stamping | H1 | Initial sheet thickness | mm |
| | H2 | Initial sheet temperature | °C |
| | H3 | Transfer time | s |
| | H4 | time cooling on tools | s |
| Side Crash | H5 | Hardness of B-pillar | HV |
| | H6 | Vertical position of impact point | mm |
| | H7 | Material failure of retractor support | |
| | H8 | Bearing factor of welding points | |

Table 4.1: Parameters of the coupled side crash simulation to be varied in the uncertainty quantification analysis.

The distributions of all the parameters are the same as used for the initial analysis shown in Figure 3.2 for the stamping process and in Figure 2.4 for the side crash.

As can be seen in the parameter selection, some parameters of the initial analysis of the stamping process are not considered in the coupled model. This means the whole AQUA process including the generation of a new input description needs to be redone with the new parameters. The same procedure as in Chapter 3 is repeated with only the four parameters selected here. Once the iterative cycle converged the input and output description of the stamping process can be passed to the coupled version of the AQUA algorithm. When initiating the iterative cycle (see Section 2.6) for the coupled version of AQUA the inputs are now the input and output description of the stamping process, as well as the parameter definitions shown in Table 4.1.

The iterative cycle of the coupled version of AQUA will run the AQUA algorithm twice. The first run of AQUA takes the input ($\mathbf{H}$) and output ($\mathbf{X}$) description of the stamping process to obtain the SM used to generate new thickness mappings of the B-pillar instantly. The second run of AQUA, which is the iterative cycle version in the case of the coupling, takes the parameter information and generates new parameter combinations to be calculated as described in Section 2.6. Once the parameter combinations are defined, the SM of the stamping process, obtained from the first run of AQUA, generates the corresponding thickness mappings based on the first four parameters. Subsequently

the side crash simulations are launched with the respective thickness mapping generated by the SM of the first AQUA run and the modified values based on the last four parameters. The rest of the process is identical to the normal iterative version of AQUA. Once the convergence condition of the second AQUA run is met, the iterative cycle is finished and the data set is ready for analysis.

Notably once the input and output description for the coupled side crash is generated, it is not necessary to pass the input and output description of the stamping process as input anymore. This is only required when generating the new simulations for the coupled side crash. When the input and output description of the coupled side crash is ready, to analyse the results the regular version of AQUA can be used. At this point only one run is necessary, as it is not necessary to generate new thickness mappings anymore.

To better illustrate this process a depiction of the process is shown in Figure 4.1.

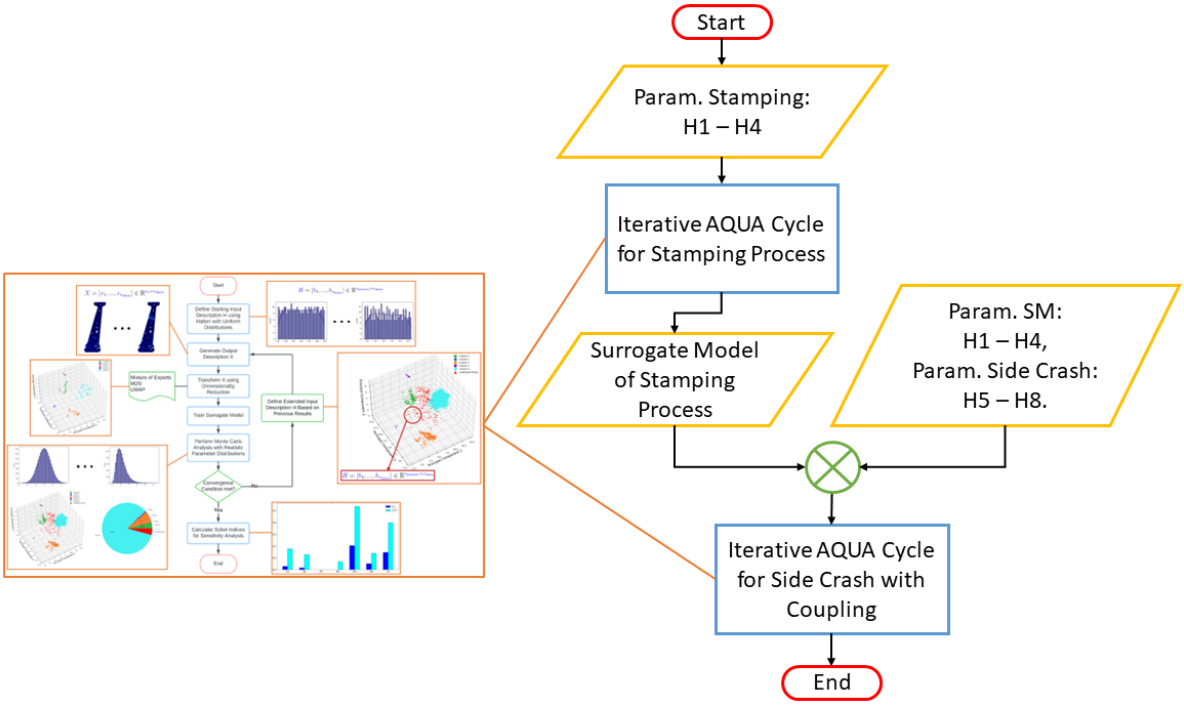
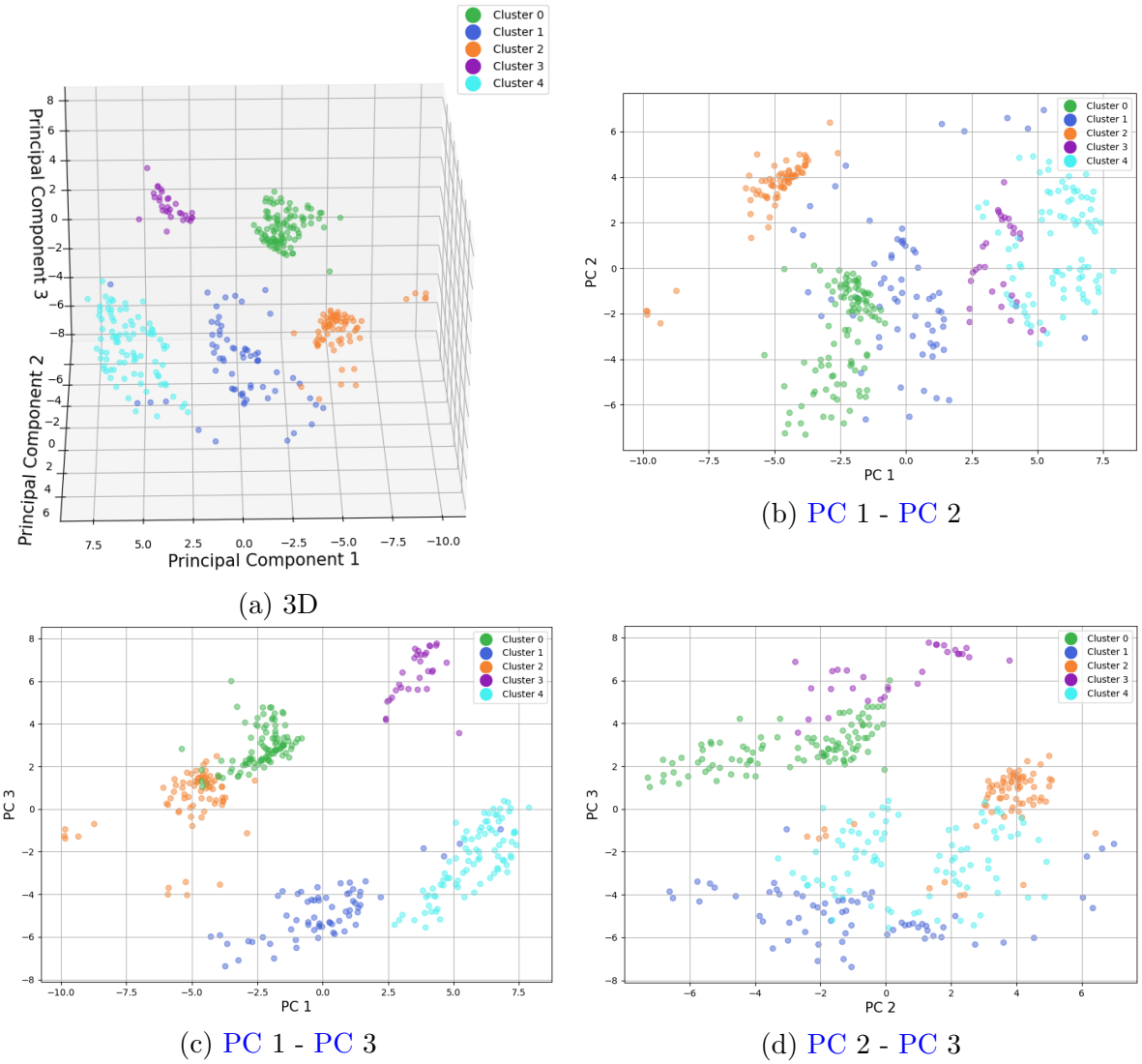



Figure 4.1: Process of the AQUA algorithm for the coupled version of the side crash analysis.

## 4.2 AQUA Results

Convergence for the coupled version of the dummy side crash simulation was reached after 356 simulations. This is significantly more than the regular version of the dummy

side crash, which converged after only 228 simulations. As the number of parameters increases, and especially the number of relevant parameters increases, the complexity of the model increases as well, leading to a much higher number of simulations needed to capture all the information. Additionally, the coupling of the thickness mapping introduces a lot more variance into the model. Unlike the regular side crash model, the AQUA algorithm identifies five clusters instead of three in the data of the coupled side crash model. Notably, the clustering was done with the full dimensional data and not the data in the reduced dimension. The reduced space of the coupled side crash model using MDS as DR technique can be seen in Figure 4.2.




(a) 3D

(b) PC 1 - PC 2

(c) PC 1 - PC 3

(d) PC 2 - PC 3

Figure 4.2: Reduced space for the coupled side crash model with clustering in the full dimensional space using MDS in 3D and 2D.

To better visualize the reduced space and make the differentiation between clusters

easier in the plot, the reduced space of the data set using the modified version of UMAP is presented in Figure 4.3. This is done to aid with the analysis of the plot. The results shown in this section were generated based on the reduced space obtained with MDS.

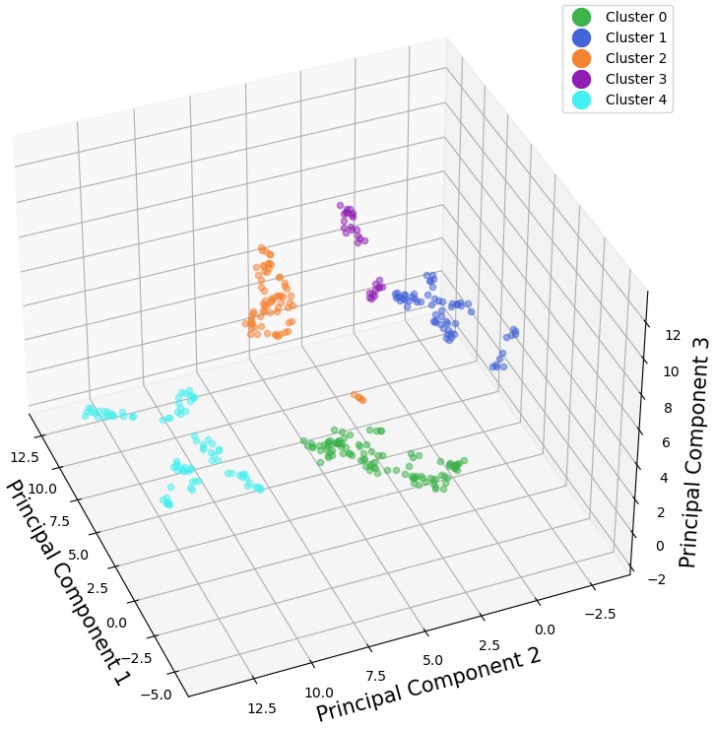



Figure 4.3: Reduced space for the coupled side crash model with clustering in the full dimensional space using the modified version of UMAP.

The corresponding characteristics of the plastic strain in the results for each of the clusters can be seen in Figure 4.4.

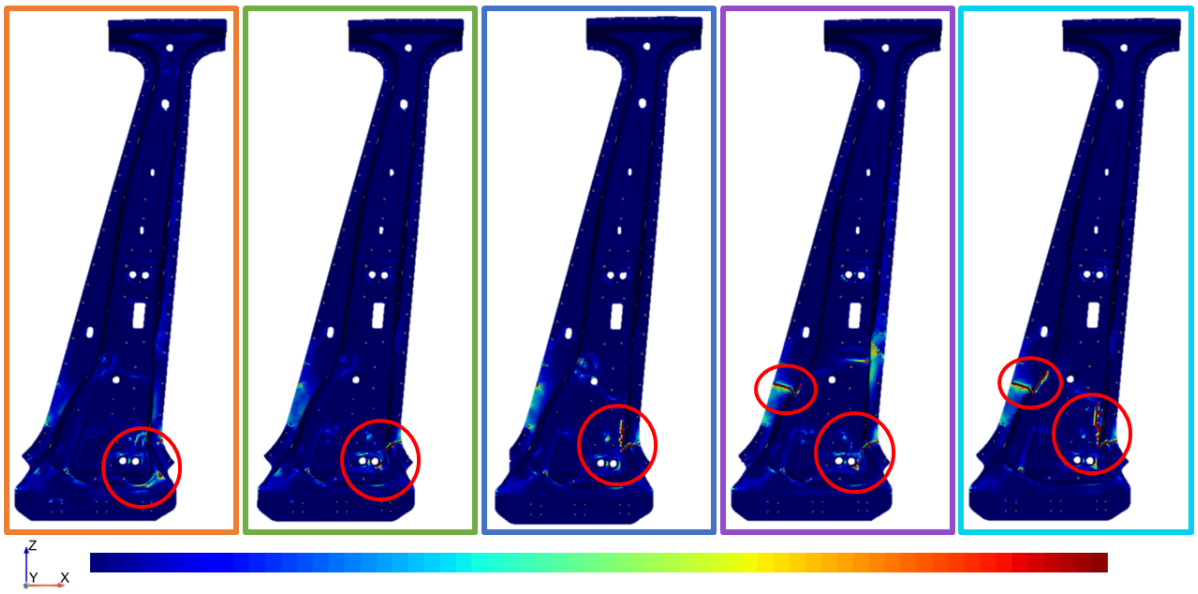



Figure 4.4: The plastic strain characteristics of the results in each cluster from Figure 4.2 framed in the according color.

As can be seen in Figure 4.4 the first three clusters, namely the orange, green and blue cluster, show the same characteristics in the results as the three main clusters from the regular dummy side crash analysis (see Section 2.2). The two additional clusters have a cut on the left side of the B-pillar, in addition to the behaviour of the green and the blue clusters. For examples the purple cluster has the same behaviour as the green cluster, in addition to the cut on the left. The cyan cluster has the same characteristics as the blue cluster, in addition to the cut on the left side. The cut from on the left side almost never occurred in the analysis of the regular dummy side crash model. The inclusion of these two clusters, with the cut on the left of the B-pillar being a lot more common in the coupled model than the regular model, clearly shows how impactful the thickness mapping is compared to the assumption of a uniform thickness.

When initially analysing the regular side crash model, the cut from the left was completely unnoticed, meaning such a result was not being anticipated by the AQUA analysis even though such a result is possible. The sensitivity analysis of the regular side crash analysis did not account for this type of outcome, resulting in a potentially overly positive analysis. In other words, the results from the AQUA analysis for the regular side crash model might seem better than reality, underestimating the likelihood of bad outcomes. This can be seen by comparing the cluster occurrences in the MC samples. Unfortunately it is not possible to disclose the exact percentages for each cluster due to confidentiality, but it can be said that the cluster probability for the orange cluster, which is the only outcome passing the side crash test, has significantly dropped in the coupled analysis. The purple cluster is the least likely, followed by the blue cluster, which is in line with the results of the initial analysis. However the probability for the green cluster is much higher than in the initial analysis, especially when including the

probability of the cyan cluster as well, which has similar characteristics to the green cluster. Overall it can be said that the probability for a downward facing cut from the right side was significantly underestimated in the initial analysis. Additionally the cut from the left side found in the results of the purple and cyan clusters was not identified in the initial analysis of the side crash model.

As a first step it is important to understand which parameters have a big influence on the results to narrow down what parameters might be causing such a difference in the results between the regular and the coupled side crash model. For that reason the first and total order Sobol indices are analysed, which are shown in Figure 4.5.

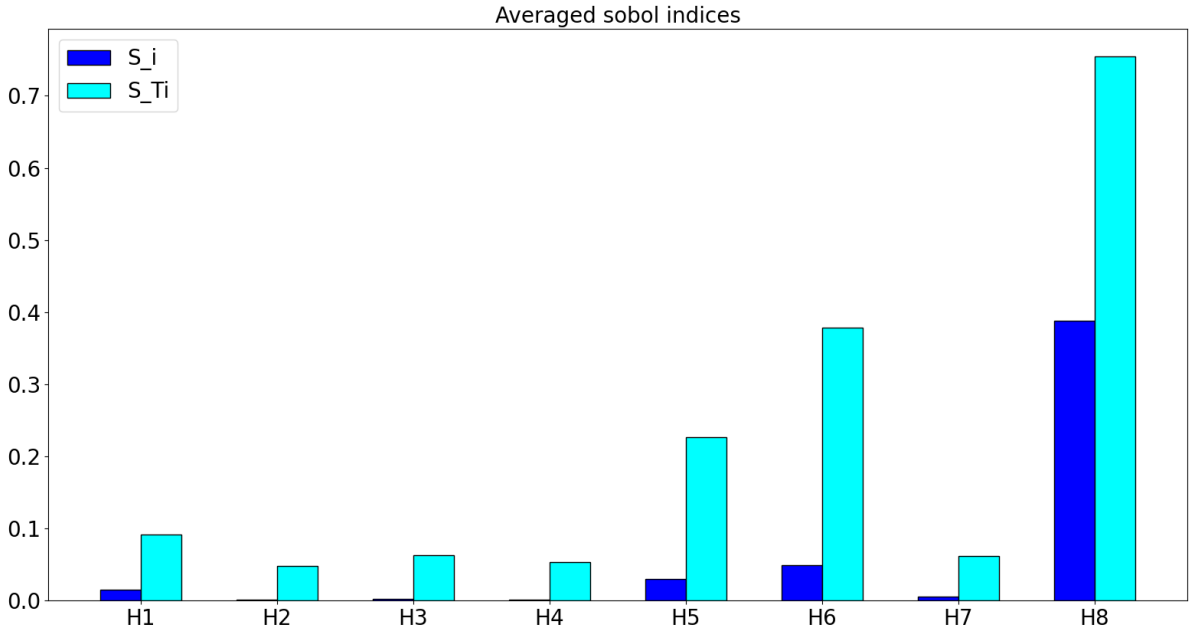
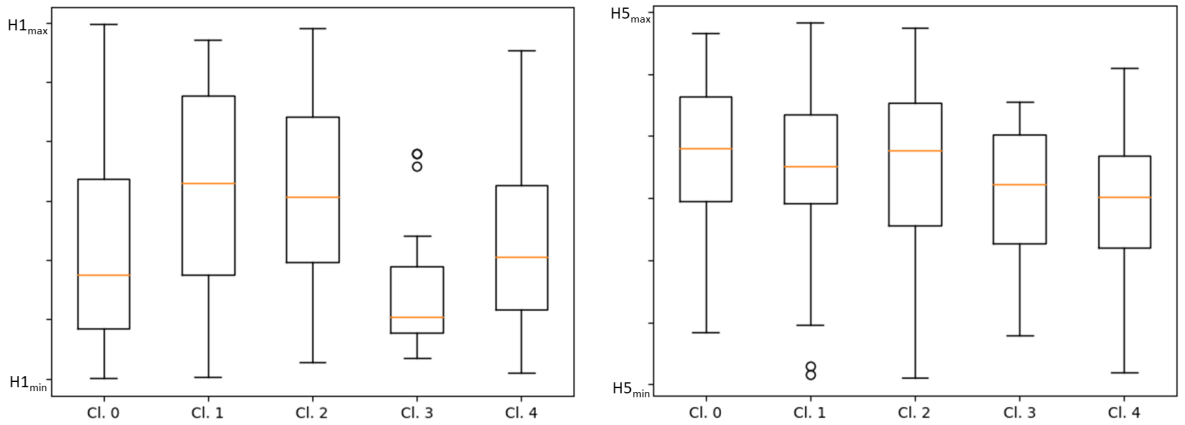



Figure 4.5: First and total order Sobol indices for the coupled dummy side crash model.

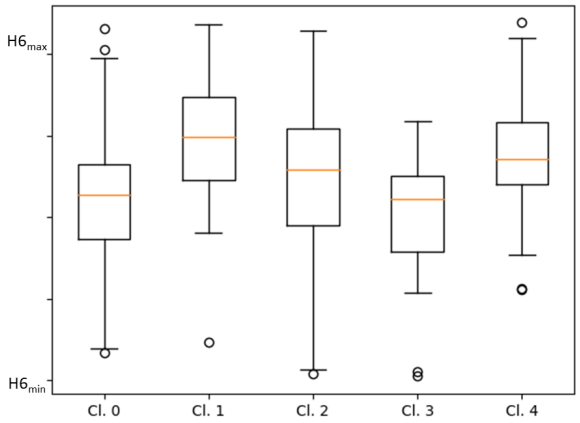
As expected the parameters h5 to h8 have a similar level of influence as in the initial side crash analysis. The quality of the welding points is again the most influential parameter followed by the vertical position of the impact barrier and the hardness of the B-pillar. The parameters related to the stamping process are generally less influential than the side crash parameters. While especially parameters h2 to h4 have a very low first order Sobol index, the total order Sobol index is quite significant, indicating various higher order interactions. The main parameters to analyse further are therefore parameters h1, h5, h6 and h8. To have an even better understanding of the effects of each parameter on the results the parameter ranges for these four parameters are plotted for each cluster. These plots are shown in Figure 4.6.



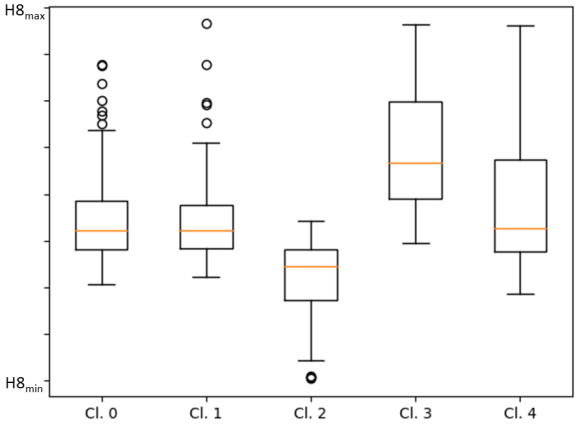

(a) H1




(b) H5




(c) H6



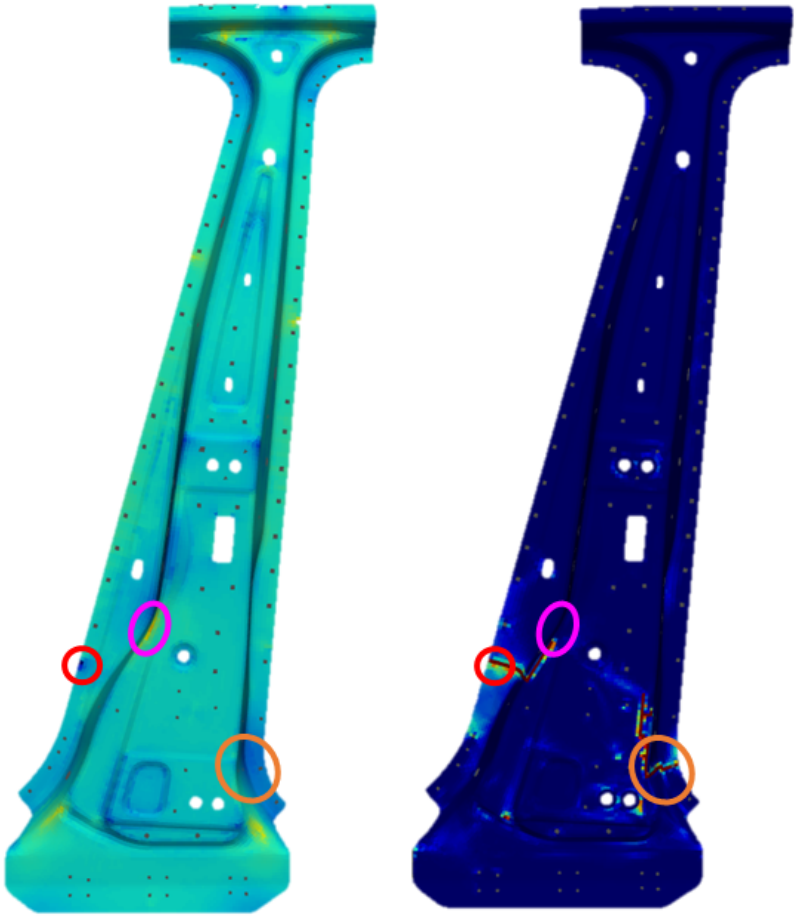

(d) H8

Figure 4.6: Parameter ranges for each cluster of the coupled side crash model.

Based on the range of values for parameter h5 in each cluster, it seems that this parameter causes the results to vary within the cluster and not so much to jump between clusters. This is due to the fact that the parameter value h5 takes on almost any value in all of the clusters, not indicating any specific condition in a cluster. Parameter h8 on the other hand clearly has an inclination for the results to end up in cluster two (the orange cluster) when its values are low. This observation was already made in the regular side crash analysis, as a low value of h8 prevents any cuts from occurring. Additionally it can be observed that the two clusters with the cut from the left (clusters 3 and 4) have exceptionally high values of h8. This indicates that a very bad quality of welding points can more likely result in this cut appearing from the left. Looking at the ranges for parameter h6 it can be made out that a high impact point of the barrier is required for the cut on the right side to face upward. This is based on the ranges for cluster 1 and cluster 4 which both do not have any small values for h6 and are the clusters with the cut on the right side going up. Lastly, parameter h1 is analysed, which is most likely to be the factor for the cut on the left, since it is the one parameter of the selected four affecting the stamping process. It is clear that cluster 3, which is one of the clusters

with the cut on the left, is only obtained with a low value of h1. Interestingly, cluster 4 which also has this cut does contain points which have larger values of h1. It shows that not only a single parameter is responsible for the cut on the left to appear, but it is a combination of different parameters. In fact the only certain assumption that can be made based on the value of only one parameter is for low values of h8 resulting in cluster 2.

To understand why it is more likely that cuts occur in the coupled version of the dummy side crash, and especially why a cut from the left appears that was not common in the initial version of the dummy side crash, the thickness mappings need to be analysed. Figure 4.7 shows a thickness mapping that resulted in a point of the cyan cluster and the plastic strain of a point in the cyan cluster.



(a) Thickness (b) Plastic Strain

Figure 4.7: Thickness mapping (a) and plastic strain (b) of a simulation from the cyan cluster.

The first point to compare is the area encircled in red. This area is the point of ini-

tiation for the cut from the left side in the B-pillar with the plastic strain. This point exactly corresponds to an area of lower thickness compared to the rest of the B-pillar as seen in the thickness mapping with a darker color. The second point of interest is the pink circle, which is the point where this cut stops. Based on the thickness mapping, this can be explained by an area of higher thickness, preventing the cut from expanding even further past this point. Lastly is the area encircled in orange indicating another zone of the B-pillar that is thinner compared to the rest of the part. This zone is the area in which the cut from the right side of the B-pillar starts. Such a zone, with a lower thickness, explains why it is more likely for these cuts to appear in the coupled model than the regular model.

It is true that the behaviour found in any of the clusters cannot be attributed to a single parameter. The only exception being cluster 2 which is highly correlated to parameter h8. Rather it is a combination of multiple parameters leading to this significant difference in the results between the coupled and the regular model of the side crash. However, the implementation of the thickness mapping does cause the analysis to be a lot more detailed and realistic, capturing new behaviours and resulting in different outcomes. Notably the initial version of the dummy side crash underestimated the probability of undesirable outcomes quite significantly.

## 4.3 Conclusion

In conclusion the coupling of the stamping process with the dummy side crash simulation yielded satisfactory results. A lot more detail was captured using the thickness mapping obtained with the stamping simulation compared to the assumption of a uniform thickness mapping. Two additional clusters were discovered with behaviour that was not previously found in the data, due to different areas of the B-pillar being thicker or thinner relative to the base thickness. This has very significant implications on the results of the sensitivity analysis, as the initial analysis of the dummy side crash model crucially underestimated the probability of undesirable outcomes. This can be very dangerous, since without coupling the stamping process to the side crash model, the two additional clusters would not have been discovered and it would encourage the engineers to be overconfident in their design. Having this additional amount of detail in the analysis ensures that the results are more in line with what to expect in reality.

Seeing the impact of coupling the thickness mapping to the dummy side crash simulation begs the question if other assumptions, which neglect a certain amount of detail in the model, cause the results of the analysis to be better than they should be. In general it is more desirable to have more conservative results to make sure that the worst cases are avoided and not the other way around. It is clearly not possible to have such a realistic representation of every part in the simulation of the dummy side crash, since otherwise the computational cost would be unacceptable. However, for important aspects of the

model that have a big influence on the results it is desirable to have a more realistic representation and to avoid bold assumptions that can skew the results. In the case of the dummy side crash model the next parameter to look into would be the hardness of the B-pillar, as this parameter also has quite a significant influence on the results and is still presumed to be uniform throughout the whole part. Especially considering that the stamping process of the B-pillar is already coupled to the side crash simulation with the thickness, the obvious next step would be to also include the hardness in this coupling.

Overall the coupling proved its worth by capturing a lot more detail in the possible outcomes and resulting in a much more realistic analysis of the side crash, which were the main objectives for the coupling.

# 5 Practical Implementation of AQUA

To showcase how the AQUA algorithm is used and implemented, the User Interface (UI) developed to launch the different versions of the algorithms is shown and explained in this chapter. The three different versions of the AQUA algorithm are the base algorithm, which is used to run an analysis for an already existing data set. The second version is the iterative algorithm that is used to generate the data set by running the iterative cycle explained in Section 2.6. Lastly the coupled algorithm is used to launch the iterative cycle to generate the data set for the coupled side crash analysis using the SM of the stamping process to include a more realistic thickness mapping of the B-pillar. Notably, even the data set of the coupled side crash model can be analysed with the base version of the algorithm once the data set has been generated. The coupled algorithm is only used to generate the data set.

## 5.1 The Base Algorithm

To use the base algorithm it is assumed that a data set already exists containing the input description **H** and the output description **X**. These two matrices are saved in csv-files which are then imported into the program when it is launched. The minimum requirement for the code to be launched is to have a folder containing two files called "H.csv" and "X.csv". For the results to be visualized it is necessary to have access to the mesh data, in the form of the connectivity matrix and the nodal coordinates, of the FEM part that is being analysed. In the case of the examples in this thesis it would be the mesh data of the B-pillar in the stamping process or the side crash simulation. This file is a text file and is called "plotting.txt". In general this means a folder containing three files needs to be set up containing the files called "H.csv", "X.csv" and "plotting.txt".

When launching the base AQUA algorithm a window pops up asking the user to select which machine learning methods to use during the analysis and which folder the files are located in. Importantly, a text field is included in that window that contains the information about the parameters and their distributions, as these are necessary for the MC simulation. The window is shown in Figure 5.1. The parameter values were substituted by variables in the following figure due to confidentiality concerns.

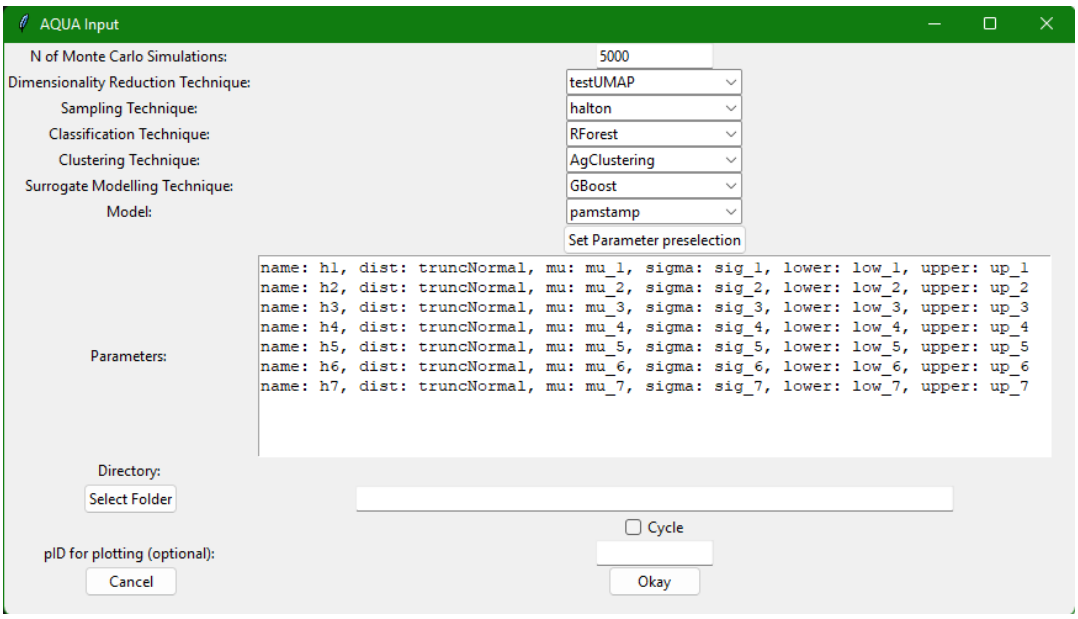



Figure 5.1: Main window to launch the base version of the AQUA algorithm.

The inputs that can be modified are the number of MC simulations, the DR technique, the sampling technique, the classification technique, the clustering technique and the SM technique. All of the available options for each input are displayed as a drop-down menu and only these options can be selected, so the user cannot modify the available options. If a new method is to be added the code itself needs to be modified to include the desired method in the drop-down menu. The available models include the stamping process of the B-pillar denoted as "pamstamp" as seen in Figure 5.1, the dummy side crash simulation and the coupled dummy side crash simulation. Notably when selecting the coupled dummy side crash simulation as model in this version of the algorithm, it will only do the analysis of an already existing data set. The coupled algorithm to launch the simulations and generate the data set of the coupled dummy side crash is explained in Section 5.3. To simplify the parameter definition when pressing the button "Set Parameter preselection" the text box containing the parameter information is changed to the default parameter distributions for the given model. Below the text box for the parameter definition is the input for the directory of the folder in which the files for the analysis are contained. To select a folder by browsing through the file-app of the computer the button "Select Folder" can be used. This allows the user to browse through the file-app and select the desired folder. The text box for the directory is then automatically filled with the selected folder directory so that manual adjustments can still be made if necessary. The checkbox denoted as "Cycle" is used to launch the iterative algorithm and will be explained in Section 5.2. The last input option denoted as "pID for plotting (optional)" is to filter out the correct elements from the "plotting.txt" file in case multiple parts are included in the connectivity matrix of that file. Finally, when pressing "Okay" all the inputs are checked for compliance and subsequently the AQUA analysis is launched.

As described in the description of the AQUA algorithm (see Section 2.1) the first step after loading the two csv-files is to do the DR. Before continuing with the next step the user is asked to confirm the recommended number of clusters identified by the algorithm. Because the reduced space is so important for the rest of the AQUA process it is important that the user agrees with the given reduced space. Another window pops up at this point, where the user can potentially change the DR technique, if the initially selected method is not satisfactory, as well as the clustering technique, the clustering space and how many clusters should be defined. The window is displayed in Figure 5.2.

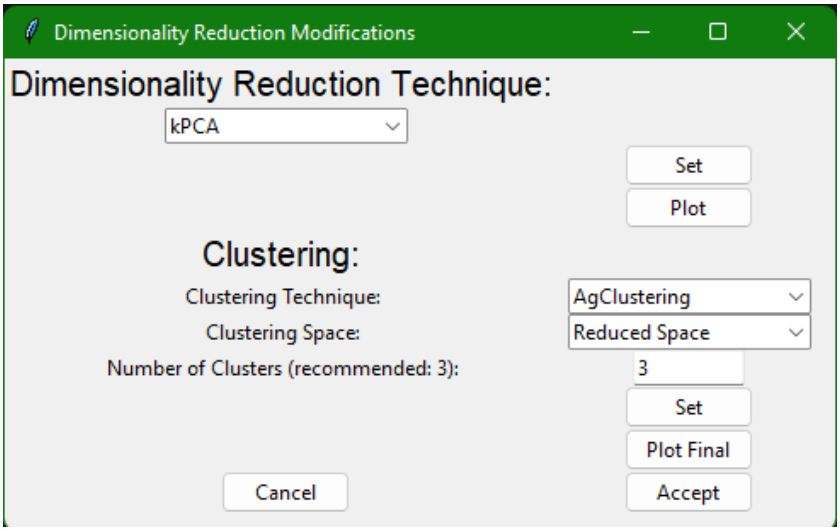



Figure 5.2: Pop-up window to confirm the DR and clustering techniques.

The dimensionality reduction technique can again be selected from a drop-down menu. When the first "Set" button is pressed the dimensionality reduction technique will be adjusted to the newly selected one. Below this button is a "Plot" button which allows the user to show the scatter plot of the reduced space without any clustering. Adjustments can therefore be made to the clustering method based on the generated reduced space. The clustering section has three input options, namely the clustering technique, the clustering space and the number of clusters with a recommendation from the algorithm. The clustering space can either be set to "Reduced Space" or "Full Space". When "Reduced Space" is chosen the clustering algorithm is run on the reduced space generated by the selected method. On the other hand when "Full Space" is selected the clustering is done based on the original full dimensional space. This is very useful to identify wrongly placed points in the reduced space, as was seen when using UMAP in Section 2.5.4. Additionally it is the method of choice when using the modified UMAP version, as then the clusters, as found in the full original space, are maintained in the reduced space as well. In the clustering section of the pop-up window a "Set" button is included as well, but more noticeably a "Plot Final" button which plots the final reduced space with the given clusters. When pressing "Accept" the algorithm will move on to

complete all the other steps in the analysis, as no further user input is required anymore.

It is worth noting that when selecting the modified version of UMAP called "testUMAP" in the UI the pop-up window changes to allow for the necessary adjustments needed for this method. The modified pop-up window is shown in Figure 5.3.

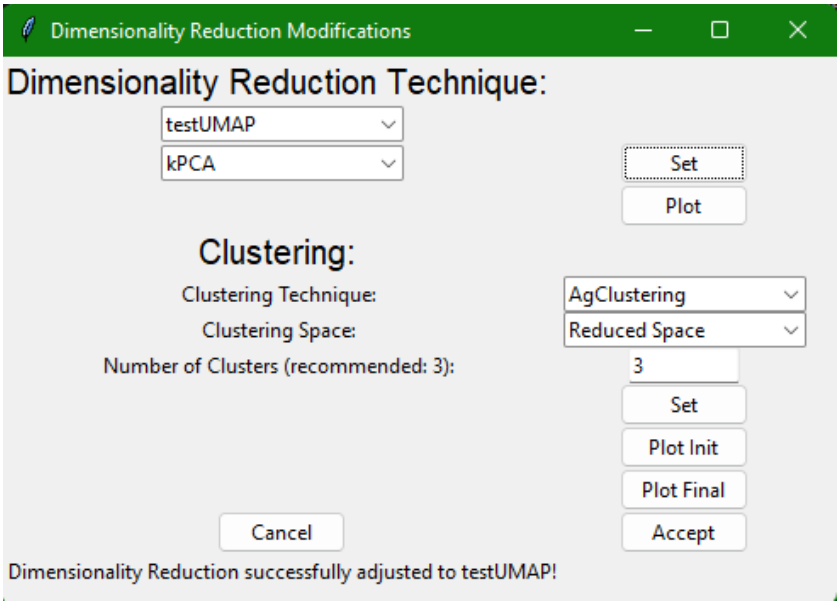



Figure 5.3: Modified pop-up window to confirm the DR and clustering techniques.

As can be seen a second drop-down menu appears below the first dimensionality reduction drop-down, and a "Plot Init" button appears in the clustering section. The second drop-down menu that appeared is in case the clustering space is set to "Reduced Space", as in this case the clusters as identified in the reduced space generated by the method of the second drop-down menu are used for the modified UMAP method. For this reason the "Plot Init" button is included, as this button will plot the reduced space of the method selected in the second drop-down menu with the clusters, to visualize what clusters are being enforced in the modified version of UMAP. However, if the clustering space is set to "Full Space", which is the default when selecting the modified version of UMAP, the second DR method is irrelevant, as it is not used to define the clusters. When pressing any of the plot buttons the scatter plot of the respective reduced space is shown with an additional window displaying the B-pillar. This second window displaying the B-pillar, or whichever part is being analysed, is what the "plotting.txt" file is needed for. It will plot the B-pillar with a color map of the respective values that are being analysed. In the case of the stamping process this plot would show the thickness of the B-pillar on the B-pillar and in the case of the side crash it would be the plastic strain. This color mapping on the B-pillar changes depending on what point the user clicks on in the scatter plot. Like so the user can easily analyse the reduced space by visualizing the different results of the data points in the scatter plot instantly.

As an example these two plots for the dummy side crash analysis are shown in Figure 5.4.

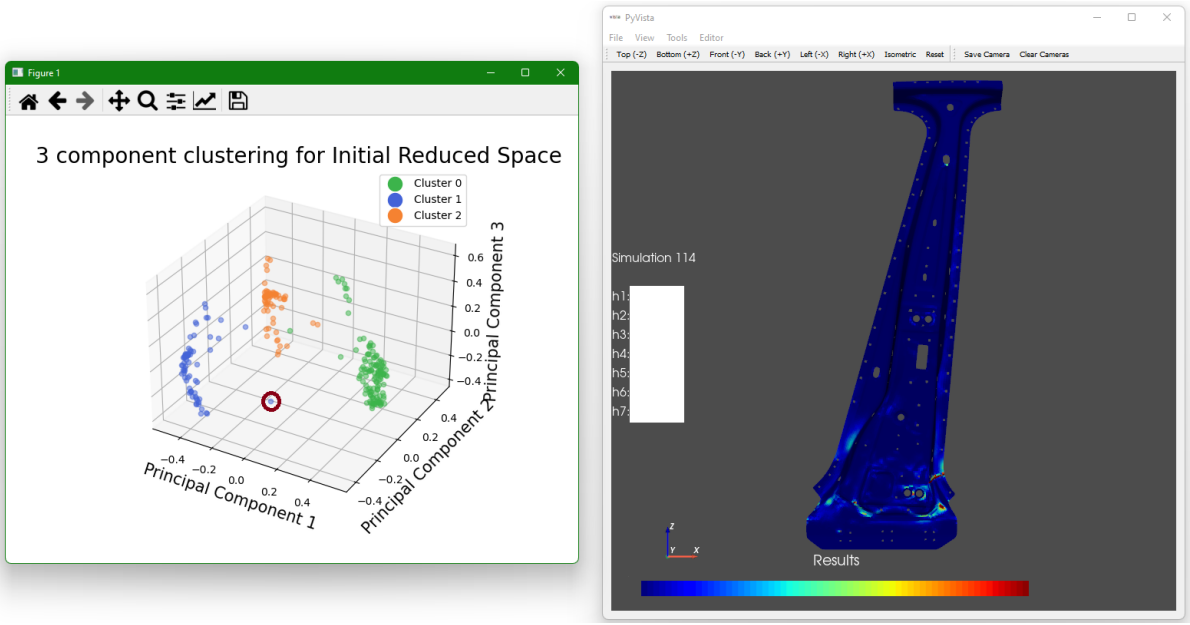



Figure 5.4: Scatter plot and data visualization window for the analysis of the reduced space.

In the example shown in Figure 5.4 the scatter point encircled in red was clicked on, and then the plastic strain mapping of this point is shown on the B-pillar in the other window. The B-pillar plot also shows which simulation was selected and what parameters were used for that simulation. However, due to confidentiality concerns the parameter values are not shown here.

While the other steps of the AQUA algorithm are being run, the terminal will display information at specific checkpoints within the code. These terminal outputs could for example look like shown in Figure 5.5. Due to confidentiality the exact cluster occurrences cannot be shown.

```
Number of clusters set to 3.
Cluster occurences: [          ].
Wrong certainty confidence: 0.853204537488641
Right certainty confidence: 0.6720825618766018
Classification score: 0.8509615384615383
Threshold value: 0.853204537488641
Surrogate error: 0.04955088623772226
8.36% of values are undefined!
```

Figure 5.5: Terminal outputs of an example AQUA analysis.

The information displayed in the terminal includes the number of clusters used in the final analysis, the number of cluster occurrences, the right and wrong certainty confidence of the classification model, as well as the classification score and the threshold value used to classify the undefined points. It also includes information on the surrogate error and the percentage of undefined points found in the MC simulation.

Once all of the steps in the AQUA analysis are complete a final window pops up which gives the user all the possible visualization options and tools to analyse the outcome accordingly. This final window is shown in Figure 5.6.

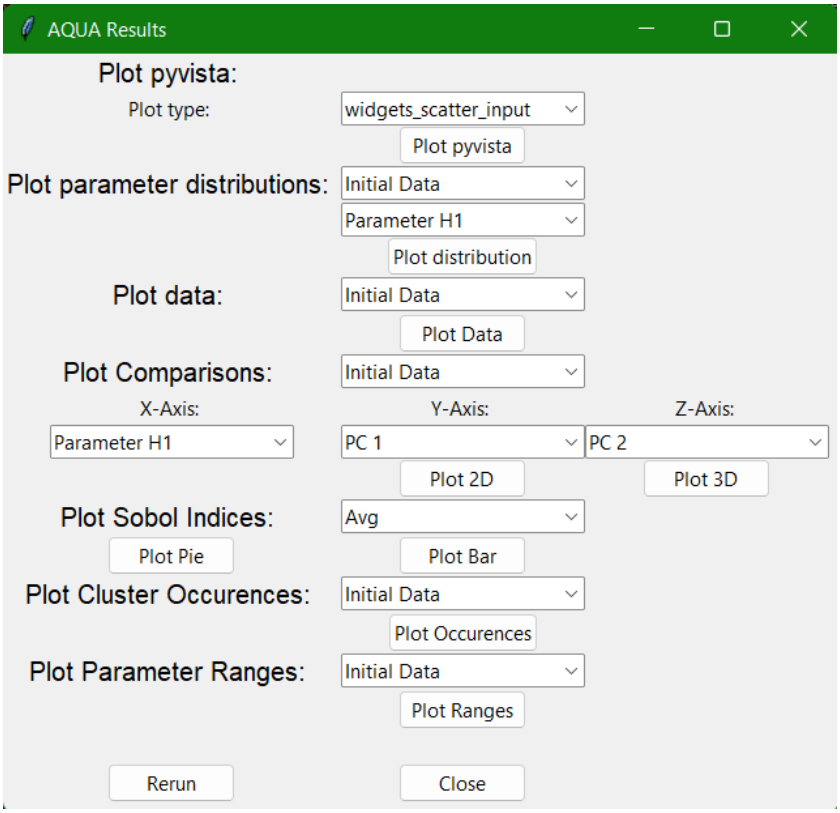



Figure 5.6: Final pop-up window to analyse the results of the AQUA algorithm.

The main visualization options are split into the sections "Pyvista" ([51]), "Parameter Distributions", "Data", "Comparisons", "Sobol Indices", "Cluster Occurences" and "Parameter Ranges". All of the possible options will not be shown here, as it is too many possible visualization possibilities, however the main sections will be explained. The "Pyvista" section is used to visualize the results in the full dimension as a mapping on the B-pillar, similar to the window shown on the right side of Figure 5.4. The most interesting plot type here is the "widgets_scatter_input" type, which will include sliders in the window to adjust the parameter values to be visualized and the mapping is adapted on the fly using the backward mapping of the AQUA algorithm. Additionally,

the scatter plot is shown with a red dot indicating where in the reduced space the parameter combination would end up in. This plot is shown in Figure 5.7. The parameter values and the values of the scalar bar are not shown again due to confidentiality reasons.

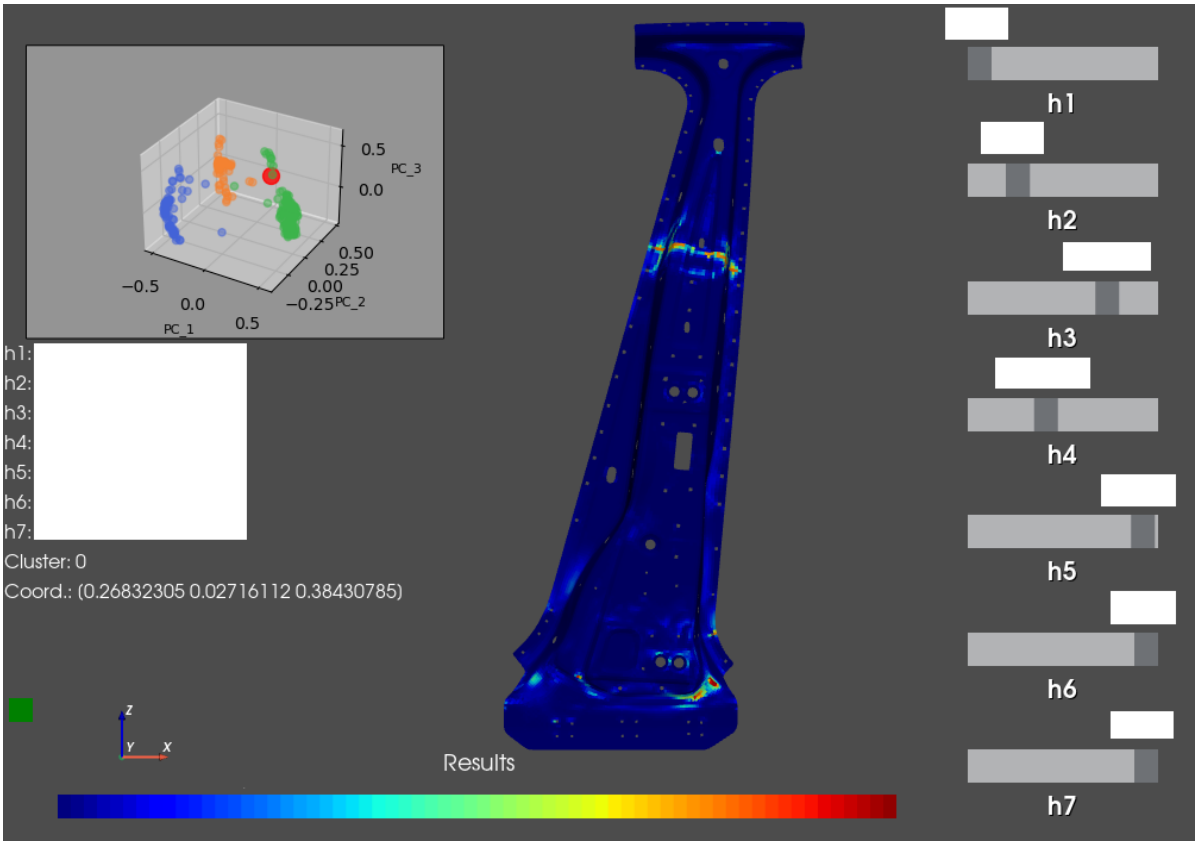



Figure 5.7: Visualization with "Pyvista" to show backward mapping on B-pillar.

The next section "Parameter Distributions" can show the distributions of the parameters and the PC of the reduced space for the "Initial Data", which is the data of the data set passed to the AQUA algorithm as input, or the "Predicted Data", which is the data generated by the MC simulation. The "Data" section allows the user the plot the reduced space of the "Initial Data" or the "Predicted Data". Additionally the option of including the undefined points cluster in the predicted data can be selected.

The "Comparisons" section allows the user to plot different two or three dimensional scatter plots with different selectable axis values. Instead of plotting the three PC like in the standard scatter plot of the reduced space, this plot allows the user to plot the data based on the values of certain parameters giving many more options to find correlations in the data. This option was used for example in Figure 2.9 in Section 2.2. Like for the "Data" section the user can again choose between plotting the "Initial Data", the "Predicted Data" or the "Predicted Data with Undefined Points".

In the next section, the "Sobol Indices" section, the Sobol indices can be plotted as a pie chart or as a bar plot. Notably the Sobol indices can be shown as the average of all three PCs or they can be shown for each PC component individually. It is worth noting that the second order Sobol indices are only displayed in the bar plot.

Very important for mode identification is the section for plotting the "Cluster Occurrences". A pie chart is plotted showing the division of data points between all the clusters. This can again be done for the "Initial Data", the "Predicted Data" and the "Predicted Data with Undefined Points".

Lastly the "Parameter Ranges" section allows the user to plot the range of values for each parameter in every cluster, to identify any patterns in the data and see if a cluster with certain characteristics in the results is obtained with specific parameter values only.

All of these visualization tools in combination give the user the ability to analyse all of the data in many possible ways to be able to identify any patterns and characteristics in the results. Data visualization is essential for data analysis, as without it is extremely hard to find any kind of information in the data. The given tools should allow the user to visualize every relevant information for the kind problems like the ones in this thesis. All of the plots and graphs in this thesis were generated with these tools.

## 5.2 The Iterative Algorithm

When running the iterative version of the AQUA algorithm the goal is to generate a data set consisting of the input description **H** and output description **X**, by running the iterative cycle described in Section 2.6. To do this, the "Cycle" check box needs to be ticked in the main starting window of the algorithm as shown in Figure 5.8. When the iterative cycle option is selected it is important to include and additional sub-folder called "reference_files" in the selected main folder containing all the files for the analysis. This sub-folder needs to contain all of the files required to launch the desired simulations as well as any other files to modify the parameters in the simulation model. Notably when starting the iterative cycle for the first time it is not necessary to have the "H.csv" and "X.csv" file in the main folder, as these two files will be generated in the first cycle.

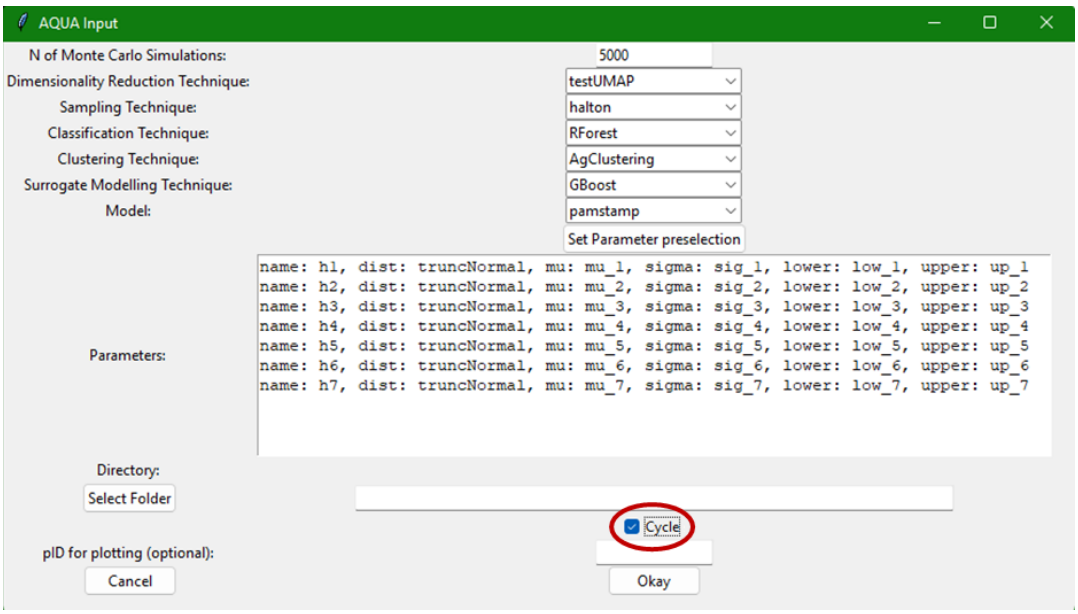



Figure 5.8: Main window to launch the iterative version of the AQUA algorithm.

When clicking "Accept" the iterative cycle begins and unlike the base version of the algorithm no further user inputs are demanded. The only time user input is requested is after the first 60 simulations of the Halton points or in the first cycle when an existing data set is to be extended. In this case the user needs to check the reduced space and tell the algorithm if dynamic clustering is to be performed, or if a predefined number of clusters should be used throughout the whole process. This is especially relevant when no clusters are detected, as the cluster detection algorithm cannot tell if no clusters are found, as it starts the analysis with a minimum of two clusters always. If no clusters are detected, like in the case of the stamping process, the user needs to set the dynamic clustering to off, and set the predefined number of clusters to one. This can also be done if clusters are found, however it is not recommended to do so. When dynamic clustering is on, which is the default, the AQUA algorithm will identify the ideal number of clusters automatically.

After every cycle, when the base AQUA analysis is run again with the new extended data set to determine the next samples to be added, the final pop-up window (see Figure 5.6) will show up in case the user wants to visualize the data between the cycles. This window is automatically closed after a certain amount of time to continue with the iterative cycle. Finally this is the only necessary input from the user. The iterative algorithm will use the input data from the main starting window to generate or extend the "H.csv" file with the new samples to be simulated. Once the new samples that should be simulated are determined, the simulations are launched automatically from within the code and the algorithm will wait until the simulations are finished by scanning the output folders for the desired files. Once all the simulations are finished the algorithm will export all the necessary information from the output files and store the

data in the "X.csv" file. After all the data from the new simulations has been extracted for this cycle, the base version of the AQUA algorithm is run again to determine the next samples to be simulated and the cycle repeats until the convergence condition is met. The user does not need to be attentive of the algorithm during this time as the algorithm can be left to run in the background independently, while the user is left free to do other tasks.

After every cycle all of the information of the AQUA analysis for that cycle is saved in a file called "info.csv", which can be looked at by the user to get an idea of where the iterative algorithm is currently at and how close to the convergence condition the results have already gotten.

While the base version of the algorithm can easily be used for other simulations (or even completely different data sets) without any modifications, the iterative version of the algorithm is specific to each model, as the simulation files are stored in various different formats and launched in different ways. If a new model needs to be run with the iterative version of the algorithm some modifications and preparations of the simulation files are necessary.

## 5.3 The Coupled Algorithm

The coupled version of the AQUA algorithm is specifically designed for the coupling of the stamping process of the B-pillar with the dummy side crash simulation. Similar to the iterative version of the algorithm, the coupled version is specific to the coupling of the stamping process of the B-pillar and the dummy side crash simulation. If another model needs to be coupled, further modifications and preparations of the simulation files are required. The main window for the coupled version of the algorithm can be seen in Figure 5.9

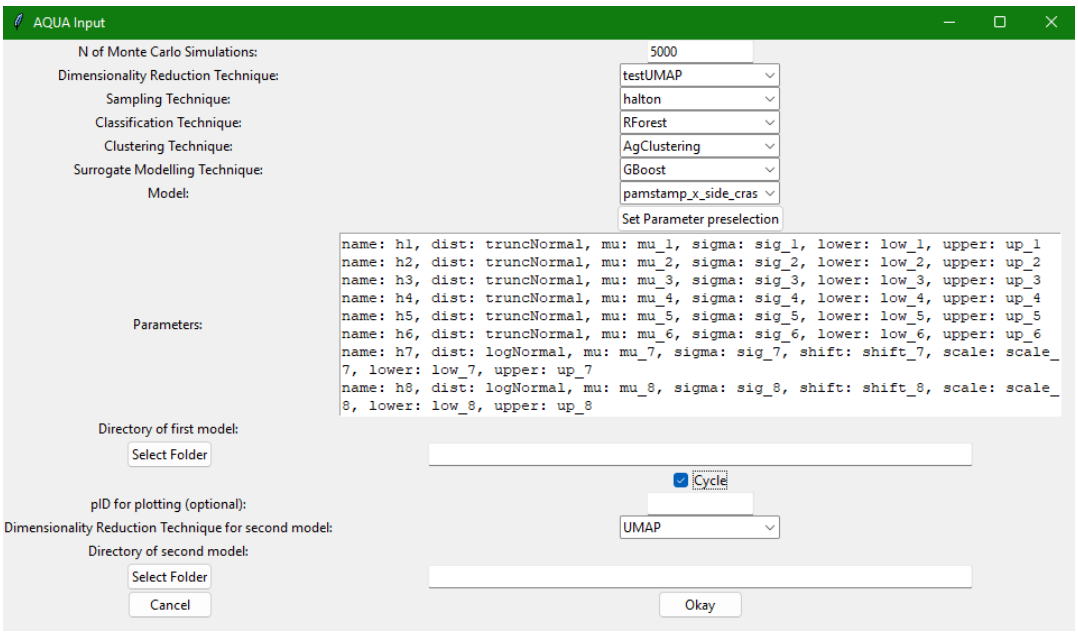



Figure 5.9: Main window to launch the coupled version of the AQUA algorithm.

Because two different models are used in this version of the algorithm, it is necessary to give an additional input to the algorithm containing the location of the data set for the stamping process. The directory of the first model is therefore the directory for the data of the stamping process, and the directory of the second model would in this case be the directory for the data of the coupled side crash analysis. Additionally, different DR methods can be used for each method, as it is not a given that the same DR method is suitable for both methods. Once all of the inputs are defined and the "Okay" button is pressed the first step of the algorithm is to run the base version of the AQUA algorithm on the data set of the stamping process. This is done to obtain the SM that will be used to generate the new thickness mappings for the side crash simulations. After the SM of the stamping process is trained, the iterative version of the AQUA algorithm is launched for the dummy side crash simulation. However, unlike the iterative version of the not coupled side crash simulation, in the coupled version the parameters are split into the ones for the stamping process and for the side crash simulation. The SM of the stamping process is used to generate the thickness mappings for the corresponding parameter combinations. These thickness mappings are then included in the side crash simulations. The rest of the algorithm is exactly the same as the iterative version of the AQUA algorithm.

Most notably this version of the algorithm only needs to be used as a substitute to the iterative version of the algorithm when the coupled simulation is to be run. Once the data set is generated, the base version of the algorithm can be used to analyse the results.

# 6 Conclusion and Outlook

Safety standards in car manufacturing are evolving constantly over time and expectations for car manufacturers to improve their safety equipment grows as a consequence. To proof their safety measures to customers, crash tests are performed by independent companies testing the vehicles in certain crash scenarios in the form of crash tests. Car manufacturers use simulations to test the effectiveness of new design concepts in the vehicles development, because emulating crash tests is extremely expensive. On the other hand, the production process of a vehicle includes many different parts and steps that are subject to tolerances, which can lead to uncertainties in the results. However, with numerous crash scenarios being very complex and the requirements to pass these tests becoming ever more detailed, it is necessary to account for possible uncertainties of the results in the simulations. For this reason research is being conducted in the field of uncertainty quantification to account for possible manufacturing and production tolerances in the simulations of the crash tests to give a range of possible outcomes. Additionally, it is also of interest to include as much detail as necessary about the parts in the simulations, as unrealistic assumptions can influence the results significantly. For this reason effort is being put into including realistic representations of important vehicle parts in the simulations, leading to the idea of coupling the production processes of parts and the final crash test simulation for a more realistic representation.

Since crash test simulations are computationally expensive and time consuming it is not possible to run simulations for all possible parameter combinations that need to be analysed. Consequently, it is necessary to create detailed Surrogate Models (SMs) that are able to generate new simulation results at a fraction of the cost of conventional simulations. With machine learning algorithms revolutionising the capabilities of computers to make accurate predictions for a given input it is compelling to connect this technology to finite-element simulations. The Artificial Quantification for Uncertainty Anomalies (AQUA) algorithm introduced by Dr. Marc Rocas [1] builds on the idea of combining different machine learning techniques to obtain a SM of the crash test simulation and with that run a sensitivity analysis to quantify the uncertainty in the crash test scenario.

This thesis includes various chapters that solve different problems related to the uncertainty quantification of the side crash test and a more realistic representation of the B-pillar thickness. The first chapter focuses on the AQUA algorithm itself and introduces new improvements made to the algorithm to address some problems found in the initial analysis of a dummy side crash simulation. The improvements include a Mixture of Experts (MoE) approach that introduces clustering and classification into the algorithm to train a different SM for each detected cluster, which represents a certain

outcome behaviour, or mode, of the simulation. This approach allows for a more accurate MC analysis, as the predictions maintain the clustered structure of the original data, as well as mode detection and the identification of probabilities for a random parameter combination to fall into a certain cluster.

Next is the implementation of non-linear Dimensionality Reduction (DR) techniques to better capture complex non-linear behaviour in the simulation results. The two introduced methods have different characteristics, with one method, namely Multidimensional Scaling (MDS), maintaining the original distances of all points in the reduced space, and the other method , Uniform Manifold Approximation and Projection (UMAP), favoring local distances to naturally form clusters in the reduced space. For the latter method a modification was made to help with wrong classifications of intermediate points and defining clearer cluster boundaries. MDS is the best representation of the original data, as the original distances are maintained in the reduced space, making this method more suitable for the initial analysis of the generated space and identifying outliers. Additionally, it is the only method suitable for the sensitivity analysis, as the other methods are not representative of the real variance in the points. However, the generated structure with this method can be chaotic and hard to understand, especially when analysing the results of the Monte-Carlo (MC) simulation. When analysing the MC results and trying to generate a backward mapping of new predictions it is recommended to use the modified UMAP method, as it is easier to differentiate between the different clusters and the clear more separated cluster boundaries result in more accurate and detailed backward mappings.

Lastly, an optimized sampling method is developed using and iterative cycle approach with the goal of specifically targeting areas of low certainty in the MC analysis to enrich these areas in the next cycle. The number of simulations required to properly train the machine learning algorithms is significantly reduced, as a more evenly spread data set across clusters is obtained and parameter combinations that were identified as problematic for the machine learning models are simulated and added in the next cycle. Consequently, classifying points as undefined during the MC analysis using the certainty of the classification model significantly improves the accuracy of the considered MC samples, as potentially wrong classifications are filtered out to avoid a negative influence on the results of the AQUA analysis. Finally, the iterative cycle approach allows for the use of a different convergence condition based on the percentage of undefined points in the MC analysis. This gives a good indication whether enough data points are included in each cluster to properly train the machine learning algorithms and if the predictions are accurate enough.

Chapter 3 is about the stamping process of the B-pillar to obtain a more realistic representation of the thickness. As was found in the analysis of the dummy side crash simulation the thickness of the B-pillar is an important factor for the results, which is why the desire to have a more realistic representation arose. The objective is to apply the AQUA algorithm to the stamping process to identify the most important parame-

ters affecting the process and generate a SM that can be used to generate new thickness mappings for use in the side crash model. The AQUA analysis shows a continuous behaviour in the results, without any discontinuities or clusters forming in the data. This is quite different to the results of the dummy side crash, as a linear relationship to the parameters is found. The most significant parameter is the initial sheet thickness, as it dictates the starting point of the part. Even though the initial sheet thickness is the most important parameter and a linear relationship is found between the final results and the parameter combinations, the AQUA analysis is still useful. The thickness mappings resulting from the stamping process are anything but uniform, as was previously assumed in the dummy side crash analysis. Instead, variations are present especially around the edges and areas with high deformations like corners. Additionally, the AQUA analysis yields a SM that can be used to generate new thickness mappings for different parameter combinations without having to run a new simulation.

In the subsequent chapter, all the previous findings are combined to obtain a coupled AQUA analysis of the dummy side crash simulation. With the SM of the stamping process new thickness mappings of the B-pillar can be generated instantly for the parameter combinations defined using the iterative cycle method. These thickness mappings are then included in the side crash simulation for a more realistic representation of the B-pillar. Instead of having the uniform thickness of the B-pillar as a parameter for the side crash, other parameters for the stamping process are used to obtain the thickness mapping. The additional detail acquired with this coupling results in a significantly different outcome of the AQUA analysis compared to the initial analysis. Several of the results that were considered as outliers and extremely unlikely in the initial analysis turned out to be a lot more common, as the thickness mapping of the B-pillar has thinner and thicker areas that are more prone to failure or more robust than when assuming a uniform thickness. Even completely new results were found that had not previously existed in the initial analysis. This shows that the additional detail gained by coupling the stamping process with the dummy side crash simulation paid off and significantly improved the results to be more in line with reality.

The last chapter showcases the AQUA algorithm and its application for the different use cases. It is explained how the base algorithm works for analysing existing data sets with all of the newly introduced methods. It is also explained how the iterative cycle algorithm and the coupled version of the algorithm are launched. All of this is done exclusively with a User Interface developed for this application, which is demonstrated in detail.

Multiple improvements and analyses were made in this thesis with satisfactory results. However, there are still more problems to be tackled and improvements to be made. The first aspect that needs a more in depth analysis and testing is the definition of the threshold value $\alpha$ used in the optimized sampling approach. While the concept of classifying points as undefined to filter out wrong classifications is valid, the exact point at which a point should be filtered out is not entirely well defined. As of now the cumulative

probability of the right and wrong classifications from the train-test-split is used. However, the train-test-split consists of the undefined points of the previous cycles, which means the results obtained from this train-test-split are not completely representative of the accuracy for the MC simulation. In addition to that, the classification certainties for every cluster are different. Although the current implementation works extremely well, more research needs to be done to better understand the correlation between the percentage of undefined points and the accuracy of the classification model to optimize the threshold value $\alpha$ and potentially define a different threshold value for each cluster. Additionally, for the Mixture of Experts approach other classification and clustering methods with different hyper-parameters could be analysed and compared to the ones used in this thesis.

For the stamping process it would be interesting to not only extract the thickness of the B-pillar, but also the hardness. As was found in the initial analysis of the dummy side crash the hardness of the B-pillar has a significant influence on the results. In addition to extracting the thickness it would be desirable to also extract the hardness. Since it would not be efficient to run to separate AQUA algorithms for the same simulation to obtain different results, both of the desired outputs should be extracted simultaneously and the parameter combinations for the next cycle could consist of a combination of points from the analysis of the thickness and from the analysis of the hardness. This way both analysis can profit from all of the simulations run for the stamping process and two SMs are generated with the same data set. With the two SMs using the same inputs the coupling to the side crash could then include a thickness mapping and a hardness mapping of the B-pillar for an even more realistic implementation.

Lastly, to improve the coupled side crash analysis even more production processes of significant parts could be coupled to the simulation. In addition to coupling the B-pillar as was done in this thesis, other parts could be coupled as well, like for example the retractor support. It might also be appealing to apply other machine learning techniques like reinforcement learning to this process, for example to optimize the shape or position of certain parts in the model.

# Bibliography


[1] Marc Rocas Alonso. *Quantifying uncertainty in complex automotive crashworthiness computational models : development of methodologies and implementation in VPS/Pamcrash*. PhD thesis, UPC, Escola Tècnica Superior d'Enginyers de Camins, Canals i Ports de Barcelona, Jul 2021. URL http://hdl.handle.net/2117/351116.

[2] Insurance Institute for Highway Safety. Fatality facts 2018: Yearly snapshot, 2019. URL https://www.iihs.org/topics/fatality-statistics/detail/yearly-snapshot#trends.

[3] European New Car Assessment Programme. Far side occupant test & assessment procedure, 2017. URL https://cdn.euroncap.com/media/32284/euro-ncap-far-side-test-and-assessment-protocol-v10.pdf.

[4] European New Car Assessment Programme. Side mobile barrier, 2020. URL https://www.euroncap.com/en/vehicle-safety/the-ratings-explained/adult-occupant-protection/lateral-impact/side-mobile-barrier/.

[5] European New Car Assessment Programme. 2020 seat leon test report, 2020. URL https://cdn.euroncap.com/media/62101/euroncap-2020-seat-leon-datasheet.pdf.

[6] European New Car Assessment Programme. 2020 seat leon test results, 2020. URL https://www.euroncap.com/en/results/seat/leon/41397f.

[7] SEAT S.A. Internal source from SEAT S.A., 2022.

[8] Frank Nielsen. *Introduction to HPC with MPI for Data Science*. Springer International Publishing, 09 2016. ISBN 978-3-319-21902-8.

[9] Jan de Leeuw. Convergence of the majorization method for multidimensional scaling. *Journal of Classification*, 5:163–180, 02 1988. doi: 10.1007/BF01897162.

[10] Insurance Institute for Highway Safety. About our tests: Side crash tests, 2022. URL https://www.iihs.org/ratings/about-our-tests#side-crash-tests.

[11] Jerome Sacks, William J. Welch, Toby J. Mitchell, and Henry P. Wynn. Design and Analysis of Computer Experiments. *Statistical Science*, 4(4):409 – 423, 1989. doi: 10.1214/ss/1177012413. URL https://doi.org/10.1214/ss/1177012413.

[12] W.E. Walker, P. Harremoës, J. Rotmans, J.P. van der Sluijs, M.B.A. van Asselt, P. Janssen, and M.P. Krayer von Krauss. Defining uncertainty: A conceptual basis for uncertainty management in model-based decision support. *Integrated Assessment*, 4(1):5–17, 2003. doi: 10.1076/iaij.4.1.5.16466. URL https://doi.org/10.1076/iaij.4.1.5.16466.

[13] Marc C. Kennedy and Anthony O'Hagan. Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(3): 425–464, 2001. doi: https://doi.org/10.1111/1467-9868.00294. URL https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/1467-9868.00294.

[14] S. H. Lee and W. Chen. A comparative study of uncertainty propagation methods for black-box-type problems. *Structural and Multidisciplinary Optimization*, 37 (239), 2008. doi: 10.1007/s00158-008-0234-7. URL https://doi.org/10.1007/s00158-008-0234-7.

[15] Russel E. Caflisch. Monte carlo and quasi-monte carlo methods. *Acta Numerica*, 7: 1–49, 1998. doi: 10.1017/S0962492900002804.

[16] Sang Yong Chung, S. Venkatramanan, Hussam Eldin Elzain, S. Selvam, and M.V. Prasanna. Chapter 4 - supplement of missing data in groundwater-level variations of peak type using geostatistical methods. In Senapathi Venkatramanan, Mohan Viswanathan Prasanna, and Sang Yong Chung, editors, *GIS and Geostatistical Techniques for Groundwater Science*, pages 33–41. Elsevier, 2019. ISBN 978-0-12-815413-7. doi: https://doi.org/10.1016/B978-0-12-815413-7.00004-3. URL https://www.sciencedirect.com/science/article/pii/B9780128154137000043.

[17] G. K. Robinson. That BLUP is a Good Thing: The Estimation of Random Effects. *Statistical Science*, 6(1):15 – 32, 1991. doi: 10.1214/ss/1177011926. URL https://doi.org/10.1214/ss/1177011926.

[18] Saltelli A, Ratto M, Andres T, Campolongo F, Cariboni J, Gatelli D, Saisana M, and Tarantola S. *Variance-Based Methods*, chapter 4, pages 155–182. John Wiley & Sons, Ltd, 2007. ISBN 9780470725184. doi: https://doi.org/10.1002/9780470725184.ch4. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/9780470725184.ch4.

[19] European New Car Assessment Programme. Side impact mobile deformable barrier testing protocol, 2021. URL https://cdn.euroncap.com/media/67261/euro-ncap-side-protocol-ae-mdb-v813.pdf.

[20] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction, 2018. URL https://arxiv.org/abs/1802.03426.

[21] Oded Maimon and Lior Rokach. *Data Mining and Knowledge Discovery Handbook, 2nd ed.* Springer International Publishing, 01 2010. ISBN 9780387098227.

[22] Harakhun Tanatavikorn and Yoshiyuki Yamashita. Improving data reliability for process monitoring with fuzzy outlier detection. In Krist V. Gernaey, Jakob K. Huusom, and Rafiqul Gani, editors, *12th International Symposium on Process Systems Engineering and 25th European Symposium on Computer Aided Process Engineering*, volume 37 of *Computer Aided Chemical Engineering*, pages 1595–1600. Elsevier, 2015. doi: https://doi.org/10.1016/B978-0-444-63577-8.50111-X. URL https://www.sciencedirect.com/science/article/pii/B978044463577850111X.

[23] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, page 226–231. AAAI Press, 1996.

[24] Kalidas Yeturu. Chapter 3 - machine learning algorithms, applications, and practices in data science. In Arni S.R. Srinivasa Rao and C.R. Rao, editors, *Principles and Methods for Data Science*, volume 43 of *Handbook of Statistics*, pages 81–206. Elsevier, 2020. doi: https://doi.org/10.1016/bs.host.2020.01.002. URL https://www.sciencedirect.com/science/article/pii/S0169716120300225.

[25] Khadija El Bouchefry and Rafael S. de Souza. Chapter 12 - learning in big data: Introduction to machine learning. In Petr Škoda and Fathalrahman Adam, editors, *Knowledge Discovery in Big Data from Astronomy and Earth Observation*, pages 225–249. Elsevier, 2020. ISBN 978-0-12-819154-5. doi: https://doi.org/10.1016/B978-0-12-819154-5.00023-0. URL https://www.sciencedirect.com/science/article/pii/B9780128191545000230.

[26] C.D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. An Introduction to Information Retrieval. Cambridge University Press, 2008. ISBN 9780521865715. URL https://nlp.stanford.edu/IR-book/.

[27] Trevor Hastie. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York: Springer, 01 2009. ISBN 9780387848570. doi: 10.1007/978-0-387-84858-7.

[28] L Breiman. Random forests. *Machine Learning*, 45:5–32, 10 2001. doi: 10.1023/A:1010950718922.

[29] Wei-Yin Loh. Tree-structured classifiers. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2:364 – 369, 05 2010. doi: 10.1002/wics.86.

[30] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[31] Audi MediaCenter. Multifaceted personality: predictive active suspension in the a8 flagship model, 2019. URL

https://www.audi-mediacenter.com/en/press-releases/multifaceted-personality-predictive-active-suspension-in-the-a8-flagship-model-1

[32] Ingwer Borg and Patrick Groenen. *Modern Multidimensional Scaling: Theory and Applications (Springer Series in Statistics)*. New York: Springer, 08 2005. doi: 10.1007/978-1-4757-2711-1.

[33] J.B. Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29:1 – 27, 03 1964. doi: 10.1007/BF02289565. URL https://doi.org/10.1007/BF02289565.

[34] Patrick J. F. Groenen and Michel van de Velden. Multidimensional scaling by majorization: A review. *Journal of Statistical Software*, 73(8):1–26, 2016. doi: 10.18637/jss.v073.i08. URL https://www.jstatsoft.org/index.php/jss/article/view/v073i08.

[35] Patrick J.F. Groenen, Rudolf Mathar, and Willem J. Heiser. The majorization approach to multidimensional scaling for minkowski distances. *Journal of Classification*, 12(1):3–19, March 1995. ISSN 0176-4268. doi: 10.1007/BF01202265.

[36] William Ford. Chapter 6 - orthogonal vectors and matrices. In William Ford, editor, *Numerical Linear Algebra with Applications*, pages 103–118. Academic Press, Boston, 2015. ISBN 978-0-12-394435-1. doi: https://doi.org/10.1016/B978-0-12-394435-1.00006-5. URL https://www.sciencedirect.com/science/article/pii/B9780123944351000065.

[37] Fred E. Szabo. C. In Fred E. Szabo, editor, *The Linear Algebra Survival Guide*, pages 47–77. Academic Press, Boston, 2015. ISBN 978-0-12-409520-5. doi: https://doi.org/10.1016/B978-0-12-409520-5.50010-2. URL https://www.sciencedirect.com/science/article/pii/B9780124095205500102.

[38] P.G. Goerss and J.F. Jardine. *Simplicial Homotopy Theory*. Modern Birkhäuser Classics. Birkhäuser Basel, 2009. ISBN 9783034601894. URL https://books.google.es/books?id=ED1bVh5K-5YC.

[39] J.P. May. *Simplicial Objects in Algebraic Topology*. Chicago Lectures in Mathematics. University of Chicago Press, 1992. ISBN 9780226511818. URL https://books.google.es/books?id=QGjwV0gyQnIC.

[40] Greg Friedman. Survey Article: An elementary illustrated introduction to simplicial sets. *Rocky Mountain Journal of Mathematics*, 42(2):353 – 423, 2012. doi: 10.1216/RMJ-2012-42-2-353. URL https://doi.org/10.1216/RMJ-2012-42-2-353.

[41] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003. doi: 10.1162/089976603321780317.

[42] Bin Luo, Richard C. Wilson, and Edwin R. Hancock. Spectral embedding of graphs. *Pattern Recognition*, 36(10):2213–2230, 2003. ISSN 0031-3203. doi: https://doi.org/10.1016/S0031-3203(03)00084-0. URL https://www.sciencedirect.com/science/article/pii/S0031320303000840.

[43] Alexander Modell, Ian Gallagher, Joshua Cape, and Patrick Rubin-Delanchy. Spectral embedding and the latent geometry of multipartite networks, 2022. URL https://arxiv.org/abs/2202.03945.

[44] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In Yves Lechevallier and Gilbert Saporta, editors, *Proceedings of COMPSTAT'2010*, pages 177–186, Heidelberg, 2010. Physica-Verlag HD. ISBN 978-3-7908-2604-3.

[45] Yoshimasa Tsuruoka, Jun'ichi Tsujii, and Sophia Ananiadou. Stochastic gradient descent training for L1-regularized log-linear models with cumulative penalty. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 477–485, Suntec, Singapore, August 2009. Association for Computational Linguistics. URL https://aclanthology.org/P09-1054.

[46] Wei Xu. Towards optimal one pass large scale learning with averaged stochastic gradient descent, 2011. URL https://arxiv.org/abs/1107.2490.

[47] Tong Zhang. Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *Proceedings of the Twenty-First International Conference on Machine Learning*, ICML '04, page 116, New York, NY, USA, 2004. Association for Computing Machinery. ISBN 1581138385. doi: 10.1145/1015330.1015332. URL https://doi.org/10.1145/1015330.1015332.

[48] Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987. ISSN 0377-0427. doi: https://doi.org/10.1016/0377-0427(87)90125-7. URL https://www.sciencedirect.com/science/article/pii/0377042787901257.

[49] R.E. Bellman. *Dynamic Programming*. Dover Books on Computer Science Series. Dover Publications, 2003. ISBN 9780486428093. URL https://books.google.cl/books?id=fyVtp3EMxasC.

[50] R. Bellman, R.E. Bellman, and Karreman Mathematics Research Collection. *Adaptive Control Processes: A Guided Tour*. Princeton Legacy Library. Princeton University Press, 1961. ISBN 9780691079011. URL https://books.google.es/books?id=POAmAAAAMAAJ.

[51] C. Bane Sullivan and Alexander A. Kaszynski. Pyvista: 3d plotting and mesh analysis through a streamlined interface for the visualization toolkit (vtk). *Journal of Open Source Software*, 4(37):1450, 2019. doi: 10.21105/joss.01450. URL https://doi.org/10.21105/joss.01450.