A data-driven methodology for the automated analysis and explanation of system behavior in crash simulations

Janis Mathieu[1,2], Michael Di Roberto[1], Michael Vielhaber[2]

[1]Porsche Engineering Group GmbH – Porschestraße 911, 71287 Weissach
[2]Institute of Engineering Design, Saarland University – Campus E2 9, 66123 Saarbrücken

Abstract

Attributable to model size and complexity of numerical crash simulations, it is not feasible for the engineers to analyze each area or component in detail, especially when these are not the core subject of investigation. In the field of occupant safety, the main explanatory objective is given by the signals of anthropometric test devices (ATD), as they are relevant for the fulfillment of legal regulations and consumer protection guidelines. Hence, this study proposes a data-driven methodology to automatically determine deviations in ATD behavior in a set of simulations and provide possible causes for the prevalence helping the engineer to understand simulations faster and to ensure quality. In the proposed methodology, sensor signals are used as a basis to describe the time-dependent system behavior providing a more superficial but mesh-invariant and faster computable description compared to approaches based on geometry data. A deviation score derived from cluster distances determined by a k-means algorithm is calculated for the full length and for sections of each signal in all simulations. The measurements describe local as well as global deviations and are subsequently used for detecting and quantifying anomalous system behavior. An event chain is reconstructed by cross correlating the time dependent deviation scores and the use of a domain knowledge database representing lag dependencies of deviations in different sensor signals. The validation of the approach is conducted by processing a dataset with frontal impact crash simulations, whereby the methodology was able to identify the main cause and give insights about secondary effects.

1 Introduction

The occupant protection in a wide range of load cases demanded by legal requirements and consumer protection institutions such as Euro-NCAP is a crucial part of modern vehicle development. These tests help to save lives and significantly reduce injuries in real world accident scenarios [1]. The challenges to meeting these requirements are constantly increasing as a consequence of rising system complexity due to variant diversity, stricter legal regulations, digitalization, and electrification of new vehicle generations. A reduction of development times as well as costs via virtual system validation using numerical crash simulations represents an integral aspect of modern development processes. These simulations allow the detailed investigation of mechanical system behavior, and subsequently reduce the number of physical prototypes significantly [2, 3]. Physical crash tests are therefore increasingly carried out in later development phases and serve to validate the simulation models. Attributable to the further reduction of physical crash tests as well as the increasing system complexity the number and intricacy of performed simulations is constantly rising causing complicated and time-consuming evaluation processes that have to be tackled by the engineers. The analysis of every subsystem or component up to the finest detail is therefore not feasible.

This specifies a need for automated data-driven simulation analysis tools to support engineers with the evaluation and interpretation of simulation results. Besides, widely used evaluation tools are still static and thus lack the possibility to use integrated data-driven methods such as machine learning algorithms for extracting context dependent information like anomalies or assessing a simulation result and suggest root causes for conspicuous behavior [4, 5]. Commercial postprocessor extensions such as DIFFCRASH [6] or SoS [7] provide the possibility for the algorithmic analysis of robustness behavior on the basis of a large simulation database. The limited automation and computational costs as a consequence of the primary focus on processing geometry data make these tools mainly suitable for large robustness campaigns but conditionally applicable in the typical daily evaluation processes, where only a small amount of data is available and relevant for the analysis and explanation of investigated system behavior. Therefore, this contribution discusses an automated evaluation approach for sparse crash simulation datasets to detect anomalous system behavior and additionally provide root causes for the occurrence.

The methodology is presented in section 3 after the discussion of different data-driven methods in the field of crash simulations with main focus being the optimizing of evaluation processes in section 2. To verify and validate the presented approach a dataset containing nine simulations of a frontal impact crash test according to FMVSS 208 is evaluated in section 4. In the last section a summary of the results as well as an outlook on possible extensions and future research is given.

2 Data-driven methods for numerical crash simulations

Due to the prospect of increasing system understanding, reducing the development time, and thus leveraging large cost potentials a variety of approaches based on purely data-driven methods such as machine learning algorithms for optimizing the crash simulation processes as shown in Fig. 1 are part of current research. The fundamental concept is to use suitable algorithms for either saving simulations as a whole, by using prediction models for relevant variables on one hand [8-13] or, on the other hand, speeding up the pre- or postprocesses by reducing the time effort for mesh generation [14], optimizing the computational resources [15], minimizing the evaluation time by the evaluation of simulation bundles [4-7, 16-20]. Another aspect discussed in [21, 22] is the storage, handling, and exploration of data and knowledge in data repositories. Since this contribution focusses on the optimization of simulation evaluation the literature survey emphases on the processing possibilities for the relevant simulation output datatypes and the specification of outliers and root causes.

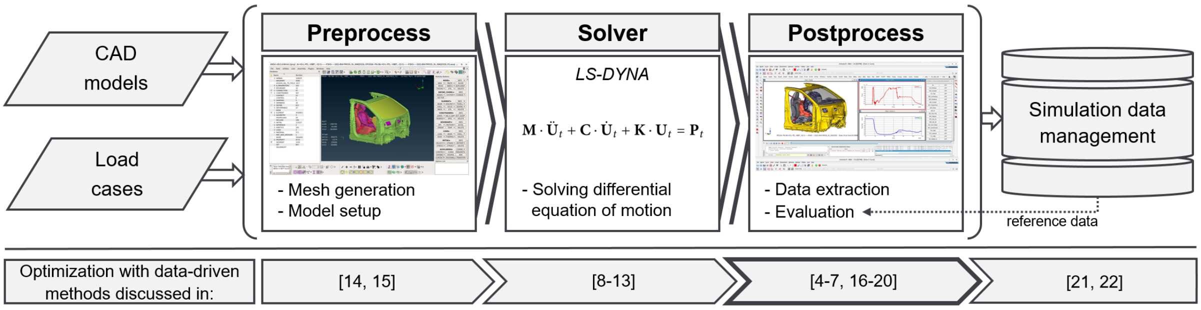

CAD models
Load cases
Preprocess
- Mesh generation
- Model setup
Solver
LS-DYNA
$\mathbf{M}\cdot\ddot{\mathbf{U}}_t + \mathbf{C}\cdot\dot{\mathbf{U}}_t + \mathbf{K}\cdot\mathbf{U}_t = \mathbf{P}_t$
- Solving differential equation of motion
Postprocess
- Data extraction
- Evaluation
Simulation data management
reference data
Optimization with data-driven methods discussed in:
[14, 15]
[8-13]
[4-7, 16-20]
[21, 22]

Fig.1: Crash simulation process chain, based on [3]

2.1 Processing of simulation output data types in numerical crash simulations

In the majority of data-driven approaches **scalar data** is used for the prediction of variables describing subsystem behavior, particularly loads or injury criteria of anthropomorphic test devices [9-11] and not directly for optimizing the postprocess of full scale simulations. Due to being the most general data source research to date has focused on processing either **nodal or element data** describing the time dependent geometry of the finite element models [4-7, 16-20]. The major problem that arises in the algorithmic processing of mesh-based models is the uniform representation as a consequence of the geometric discretization due to the possible representation of identical geometries with different arrangements and numbers of nodes and elements. However, in literature a variety of methods for processing geometry data are proposed. *Diez et al.* [4] determine the geometric center of gravity along the longitudinal axis before the surrounding elements are projected onto the generated axis. This approach enables the processing of profile shaped components like a longitudinal member, but as it is stated in [5], the processing of planar components is conditionally feasible. Other methods are the use of a modified virtual spherical detector surface approach [20], the projection of mesh-data into a three-dimensional voxel grid [12, 19], the use of a coherent point cloud algorithm [13] or the mapping of different finite element meshes to one reference simulation [18, 21]. All methods are able to generate fixed size array geometry representations which can subsequently be processed with an algorithm, but additional memory and computational cost as well as a certain loss of information are relevant trade-offs that have to be considered. The geometry data is either used for the prediction of time dependent deformation behavior of academic structures with Long Short-Term Memory Networks as discussed in [12, 13] or for the optimization of the postprocess by analyzing feature importance of geometry parameters [16], determining correlations in deformation behavior of a spitwall in a full vehicle crash [6], rule mining approaches for robustness campaigns [4, 17, 18] and identifying suspicious system behavior with an automated outlier detection [5]. Aside from laborious and memory intensive to compute node and element data, time dependent system behavior is also described by the **time series data** simulation output. Hereby, behavior is described on a more superficial level comprising indirect information about geometric effects. In passive vehicle safety, time series data is typically generated by sensors either in a physical crash test or a numerical simulation and is thus typically available in discrete areas. With respect to occupant safety, time series data is particularly important as these are the basis for calculation of surrogate values like the HIC, NIC, chest intrusion or acceleration, relevant for legal regulations and consumer protection guidelines [23]. Subsequently, an unwanted change is considered critical, and the root cause for understanding the behavior is important. Pakiman et al. [22] used the internal energy curve of crash simulations for extracting features, for instance, the initial absorption time, the maximum energy absorption and the time where the maximum energy absorption occurs. These features are used to set up a knowledge discovery assistant based on a graphical database to cluster simulations according to their similarities over different vehicle development phases.

2.2 Outlier detection and root cause analysis

In general, an anomaly or outlier as defined in [24] can be considered as an observation that strongly deviates from a number of other observations, typically named as inliers, to raise suspicion that a significant difference in system behavior must have caused the appearance. Algorithms to identify outliers are classified into the categories supervised, semi-supervised and unsupervised learning [25]. Semi-supervised and supervised methods require prior knowledge in form of additional training datasets with which the outliers are identified. Unsupervised methods work on raw datasets and are able to identify structures in the data without prior knowledge. The potential of unsupervised methods regarding postprocess optimization for crash simulations is demonstrated in [5] with an automated method detecting outliers in the geometrical behavior of a longitudinal member in a frontal impact crash test, which folds depending on scattered wall thicknesses of various components in the model, either in the front, in the rear or in a mixed behavior. In the dimension-reduced representation of the element strains of the component, realized with a principal component analysis (PCA), mode-dependent clusters are visualized, and the outlier scores are calculated based on the distance to the k-th nearest neighbor. Aside from the automated method in [5], linear dimensionality reduction with PCA is also used for visualizing mechanical system behavior based on the geometry in a lower dimensional representation and thus realizing a manual definition of outliers [4, 6, 16-19]. Outlier detection is also possible by processing time series datasets, as it has been demonstrated in the analysis and feature learning of weather, sound, or medical data [26] such as algorithms for speaker identification in speech recognition or anomaly detection in single channel electroencephalography waveforms.

If an outlier has been identified the immediate question arises, if any reasons for the occurrence can be specified. Based on geometry data the “dPCA” method presented in [6] showed great potential in specifying the most important buckling component in a robustness campaign and differentiating between first and second order effects which can be used for causal analysis. *Diez et al.* [4] extracted rules for the mechanical behavior of a bumper in a frontal impact represented as a decision tree, which also can be used for identifying the root cause for the deformation mode of a specific sample. Causal associations can also be identified and quantified in large climate and cardiovascular time series datasets as shown by Runge et al. [27].

2.3 Derivation of the object of investigation

Summarizing the research activities to date, the algorithmic analysis of time series data has as of yet not been the main focus within the field of process optimization for numerical crash simulations. Subsequently, this study focusses on the automated analysis of time series data. As the change in either parameters or geometry will most likely affect the sensor signals, a computationally more efficient consideration with a minimal amount of data is realizable. The concept of the presented methodology is to provide the engineer an automatically generated pre-plausibility analysis, which can be used to find conspicuous areas and ensure the quality and consistency of the performed simulations.

3 Methodology for the automated analysis and explanation of system behavior

In this section the methodological architecture for detecting and specifying anomalous behavior as well as deriving possible root causes for the occurrence is discussed in detail. An overview of the presented methodology and processing of the data is given in Fig. 2.

3.1 Database definition

The underlying data structure for the simulation models is assumed to be represented by a model-tree, clarifying relations and documenting changes in the product development process. Subsequently, it is known from which base model a new one is derived, ensuring changes to be traceable especially in later development phases. The amount of data needed to learn or explain problems with data-driven algorithms is dependent on the complexity and size of the considered spaces [25]. Subsequently, the full statistical modeling and causal discovery of the problem as discussed in [27] for climate and cardiovascular data is not reasonably feasible in this context. For the presented approach a user defined simulation dataset with an expected size from up to ten datasets is assumed, which is significantly less than in the robustness campaigns discussed in [6, 17, 18]. The acquired datasets are thus not expected to be balanced as it is usually the case in the robustness campaigns. As outliers are dependent on the context [24], a well-documented model history is essential for the dataset definition as the engineer has to choose which models provide a proper context and thus form the basis of a meaningful comparison.

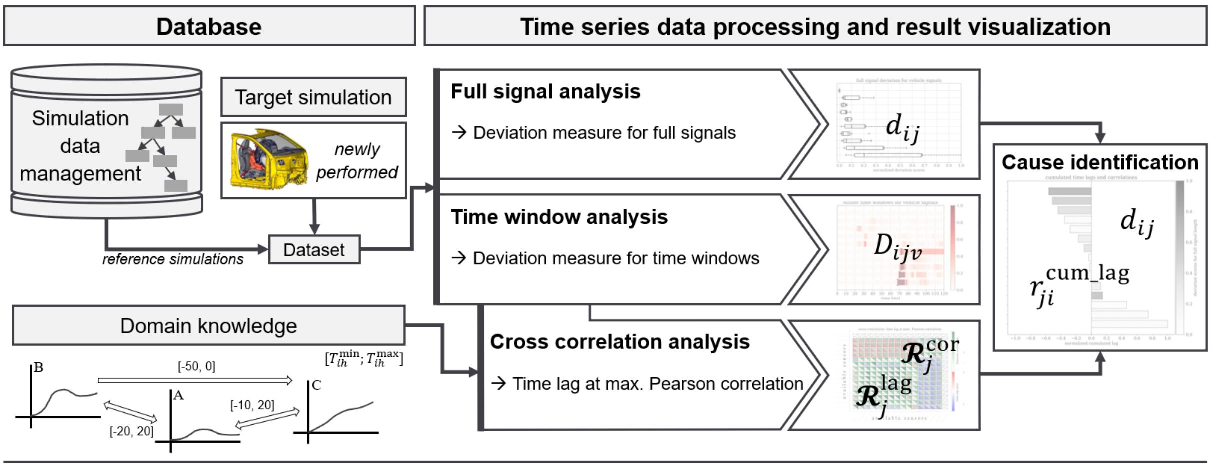

Database
Time series data processing and result visualization
Simulation data management
Target simulation
newly performed
reference simulations
Dataset
Domain knowledge
$[T_{ih}^{\min}; T_{ih}^{\max}]$
B
[-50, 0]
A
C
[-10, 20]
[-20, 20]
Full signal analysis
→ Deviation measure for full signals
d_{ij}
Time window analysis
→ Deviation measure for time windows
D_{ijv}
Cross correlation analysis
→ Time lag at max. Pearson correlation
$\mathcal{R}_j^{\mathrm{cor}}$
$\mathcal{R}_j^{\mathrm{lag}}$
Cause identification
d_{ij}
$r_{ji}^{\mathrm{cum\_lag}}$

Fig.2: Methodological architecture and data processing steps

3.2 Processing of time series data

The extracted time series data can be represented as a third order tensor S_{ijk} with i denoting the discrete points of evaluation (*sensors*), j the simulations (*samples*) and k the signal values over time (*features*). Since typical design changes from simulation to simulation are rather small and the available datasets are very limited, it is difficult to include prior knowledge as required for supervised learning methods. Furthermore, the outliers are highly dependent on the respective context, hence an unsupervised learning method, as also used in [4-6, 17, 19], is utilized in the proposed methodology. Hence, the full signals from all samples for each of the analyzed sensors is processed with a k-means clustering algorithm according to Fig. 2 step 3 and denoted in Eq. 1. The main objective for the k-means clustering algorithm is minimizing the within-cluster sum of squares (variance) J_i with the number of clusters $l = 1, 2, \dots, m$ with the respective cluster centers given as μ_{ilk}^{C}.

$$\min\{J_i\} = \min\left\{\sum_{l=1}^{m}\sum_{j=1}^{n} u_{lj}\left\|S_{ijk}-\mu_{ilk}^{\mathrm{C}}\right\|^2\right\} \quad \mathrm{s.t.}\ \ \mathrm{i)}\ u_{lj}\in\{0,1\}\ \forall\, l,j \quad \mathrm{ii)}\ \sum_{l=1}^{m} u_{lj}=1\ \ \forall\, j \tag{1}$$

Whereas the dimension k of the cluster center μ_{ilk}^{C} is given by the values per signal, the number of clusters m needs to be defined manually and is highly dependent on the data and thus has to be set individually for each sensor. The silhouette according to [28] is a measure of the quality of a clustering that is independent of the cluster count and can thus be used to specify the number of clusters. As the computation needs $O(n^2)$ for each sensor, the evaluation is more cost intensive than the k-means clustering itself. Hence, a simplified version referred to as medoid-based silhouette $q_{ij}(m)$ as discussed in [29] is used. For automatically determining the optimal number of clusters the k-means clustering is performed multiple times with the aim of maximizing the average silhouette for each sensor by manipulating the cluster number according to Eq. 2.

$$m_i^{\mathrm{opt}} = \underset{m}{\mathrm{argmax}}\left\{\frac{1}{n}\sum_{j=1}^{n} q_{ij}\right\} \tag{2}$$

Based on the optimal number of cluster centers m_i^{opt} for which the highest silhouette score is achieved the distance to the main cluster is calculated, whereby the main cluster μ_{ik}^{main} is the cluster inheriting the most samples and is thus assumed to display normal system behavior. μ_{ik}^{main} is calculated as denoted in Eq. 3.

$$\mu_{ik}^{\text{main}} = \mu_{ilk}^{\text{C}} : l = \underset{l}{\operatorname{argmax}} \left\{ \sum_{j=1}^{n} u_{lj} \right\} \tag{3}$$

The Euclidean main cluster distance d_{ij} is calculated according to Eq. 4.

$$d_{ij} = ||S_{ijk} - \mu_{ik}^{\text{main}}|| \tag{4}$$

For illustrating the procedure, a two-dimensional example is shown in Fig. 3a). In this particular case, k describes two dimensions. The Euclidian distance d_{ij} can thus be interpreted as an abstract deviation measure, quantifying how significant a certain outlier is compared to the normal system behavior as represented by the main cluster center. Seeing that the dataset is user-defined it is assumed, that most of the simulations in the dataset display normal behavior, otherwise if the dataset is highly contaminated, the assumption, that the main cluster shows inlier behavior is invalid. Since no dimensionality reduction is applied to signals, due to the low dimension especially in the analyzed time windows, a possible loss of information as a result of the dimensionality reduction is prevented [30].

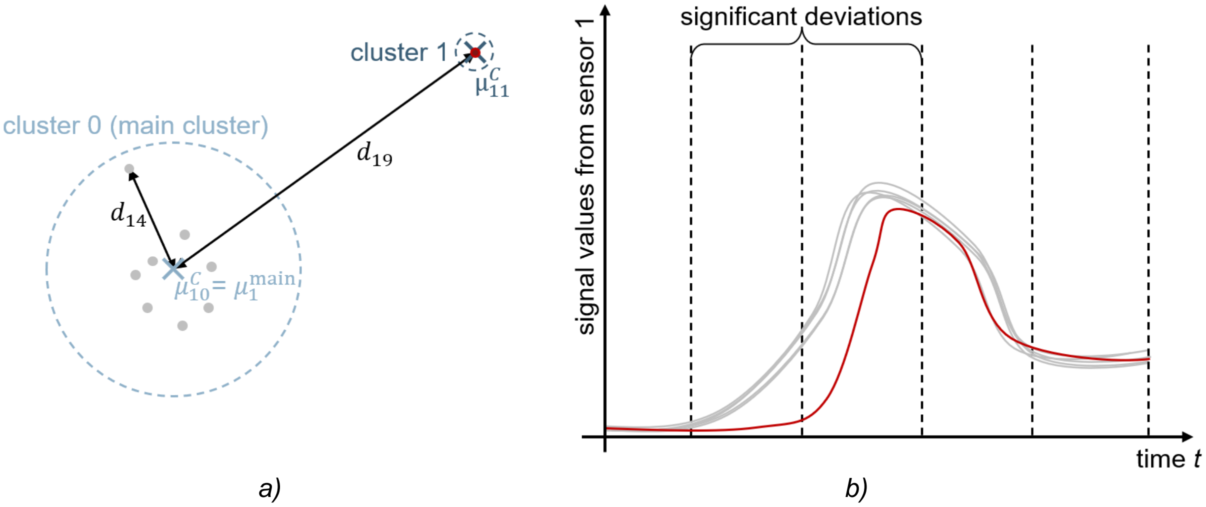


Fig.3: a) Main cluster distance illustrated on a two-dimensional example, b) time window analysis

According to Fig. 2 the procedure in Eq. 1 – Eq. 4 is applied on the full signals to identify anomalous signals indicating suspicious system behavior in the respective sensor region. The red signal exemplarily shown in Fig. 3b) has been identified as outlier, the gray ones as inliers. The second step of data processing is a window analysis to further quantify the deviation from a time perspective. The proposed procedure is subsequently applied on $v = 1,2,\ldots,w$ equally spaced sections of the signals for each sensor. The deviation scores for the time windows are thus denoted as D_{ijv}.

In the next step, the cross-correlation analysis according to [31] of the deviation scores over the time windows D_{ijv} is performed as denoted in Eq. 5. The cross-correlation matrix $\boldsymbol{\mathcal{R}}_j(\tau)$ contains the correlations between each of the deviation scores in the respective simulation depending on the time lag between the correlated vector pairs. Hence, a maximum correlation R_{jih}^{cor} as well as a corresponding time lag R_{jih}^{lag} denoted in Eq. 6 can be obtained. These specify in how far the deviations correlate and whether the deviations in one signal either lead or lag the deviations in another one. R_{jih}^{cor} and R_{jih}^{lag} have the same symmetrical format and the axis denoted by i and h are specified by the analyzed sensors.

$$\boldsymbol{\mathcal{R}}_j(\tau) = \text{CrossCorrelation}\{D_{ijv}\} \tag{5}$$

$$R_{jih}^{\text{lag}} = \underset{\tau}{\operatorname{argmax}}\{\boldsymbol{\mathcal{R}}_j(\tau)\}, \quad R_{jih}^{\text{cor}} = \underset{\tau}{\max}\{\boldsymbol{\mathcal{R}}_j(\tau)\} \tag{6}$$

To specify an event chain in the respective simulation a physically realistic time lag interval $[T_{ih}^{\min}; T_{ih}^{\max}]$ has to be predefined, since the cross-correlation matrix contains mathematical solutions for every vector pair. The idea here is to specify a physically meaningful threshold and not a global value for the dataset which highly depends on the contamination. For example, high time lags between deviations in a seatbelt section force near the ATD chest and the chest acceleration of the ATD itself are not expected since the spatial distance is relatively low. For the identification of main causes, the cumulated lag $r_{ij}^{\text{cum\_lag}}$ according to Eq. 7 is calculated. o denotes the number of sensors.

$$r_{ji}^{\text{cum\_lag}} = \sum_{h=1}^{o} R_{jih}^{\text{lag}} \quad : \quad R_{jih}^{\text{lag}} > T_{ih}^{\text{min}} \ \forall j \ \wedge \ R_{jih}^{\text{lag}} < T_{ih}^{\text{max}} \ \forall j \qquad (7)$$

3.3 Integration of domain knowledge

Due to the distinct use of time series data the knowledge of which sensors are needed to fully describe relevant system behavior is also necessary to avoid a loss of information and needs to be determined by expert knowledge for every problem statement individually, as the sensors might differ in regard to different load cases or systems. The causal dependencies are integrated according to Fig. 2 with a predefined array $[T_{ih}^{\text{min}}; T_{ih}^{\text{max}}]$ describing time lags in the deviations between the signals of the respective sensors.

4 Verification and validation of the approach

The presented methodology is verified and validated on a dataset of nine crash simulations describing a full overlap frontal impact crash test against a rigid barrier with an ATD on the co-driver position and an initial velocity of 56 km/h (FN56-P3) according to FMVSS 208 [32]. For a simulative analysis of the frontal impact crash test typically sled models as shown in Fig. 4a) are used, where the body-in-white is accelerated with a pre-defined acceleration from a full crash simulation or a physical crash test [11].

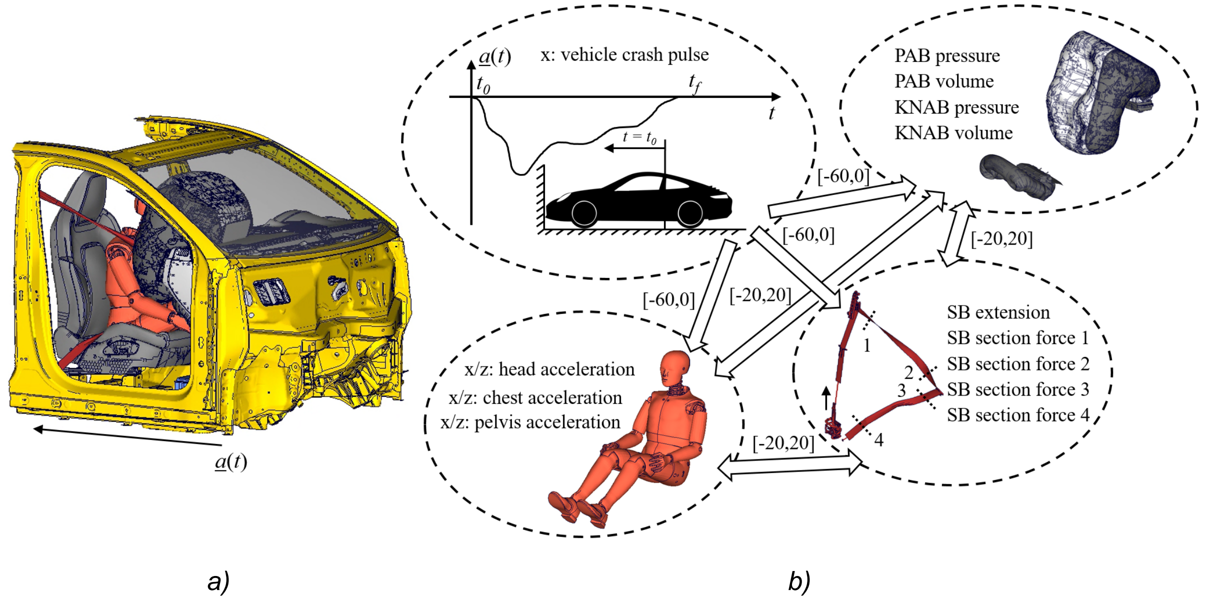


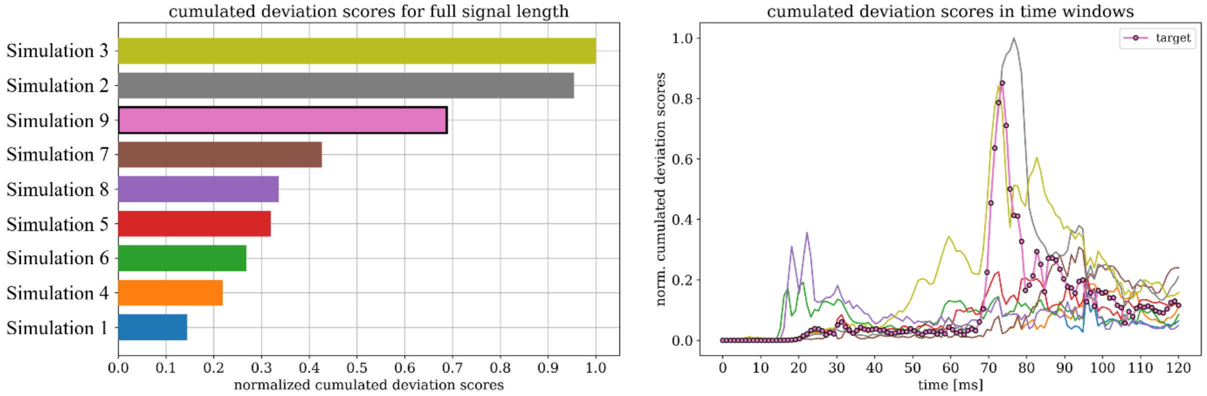
Fig.4: a) LS-DYNA simulation model, b) Domain knowledge

4.1 Use-Case: FN56-P3 with ATD on co-driver position

The change in the simulation which is analyzed in the following is a different seatbelt force-level switching time. In the scenario changes are especially expected in the chest acceleration due to the strong correlation with seatbelt forces and switching time. For describing the ATD movement, pelvis, chest, and head accelerations in the three spatial directions are used. Other signals can be easily added; for simplicity, the use of the reduced signals is sufficient for this purpose. The vehicle behavior is described with the vehicle deceleration in x-direction (crash pulse), volume and pressure of the knee and passenger airbag as well as the seatbelt extension and four seatbelt section forces in the seatbelt. In terms of the considered use-case first order effects for a change in ATD behavior are possibly located either in the vehicle configuration and movement or in the restraint system. Second order effects can occur between the restraint system and vehicle movement or between the components of the restraint system themselves. As a preset motion is applied to the model, neither the ATD behavior nor the restraint system has any influence on this. All sensors and their relations as well as the allowed time lags between deviations are illustrated in Fig. 4b).

4.2 Automated analysis and report generation

A global overview of the deviation scores in the dataset is given in Fig. 5, whereby Fig. 5a) contains the cumulated deviation scores for the full signals of the respective simulation and Fig. 5b) shows the cumulated deviation scores of each time window. The analyzed simulation corresponds to the newest one in the model tree and is highlighted. Compared to the other simulations in the dataset the deviations are on the higher end, whereby the fact, that the deviations mainly occur between 65 ms and 100 ms is specifically interesting and gives the engineer a time window to explicitly look at.



a) *b)*

Fig.5: a) Cumulated deviation scores for the full signals, b) Cumulated deviation scores in each time window

Fig. 6 gives more detailed information about the target simulation. The plot consists of a horizontal box plot illustrating the deviation scores for the full-length signals as well as a horizontal bar plot displaying the deviation scores in the respective time window, whereby both of them share the same y-axis containing the sensor names. The box plot on the left represents the deviation scores for the full signals from all simulations at the respective sensor, whereby the analyzed simulation, here number 9, is marked with an “x”. This provides global information on how the signals from the individual sensors differ from each other.

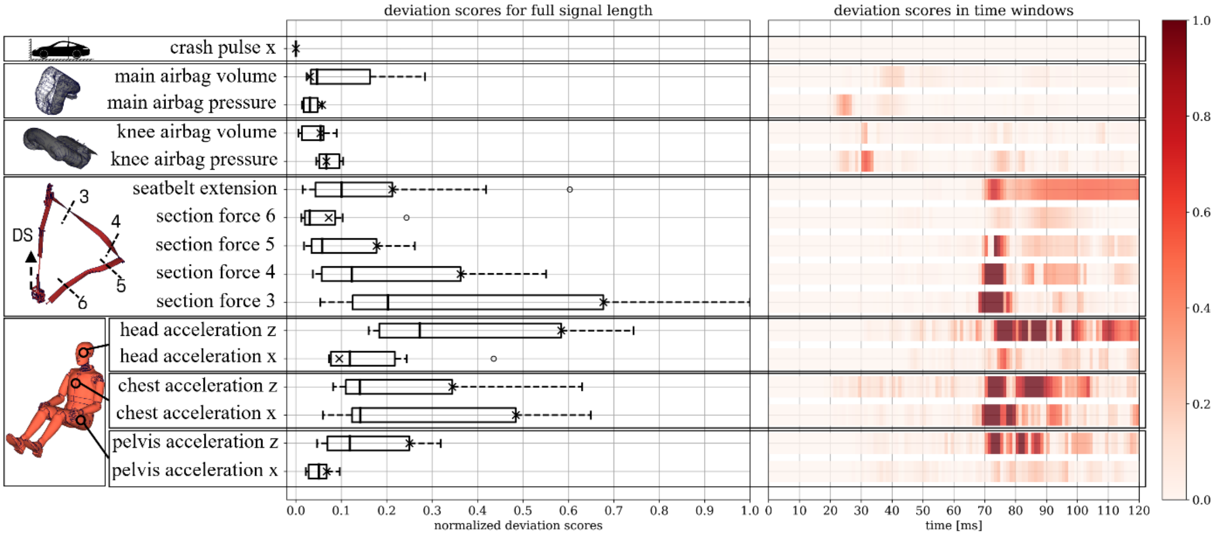
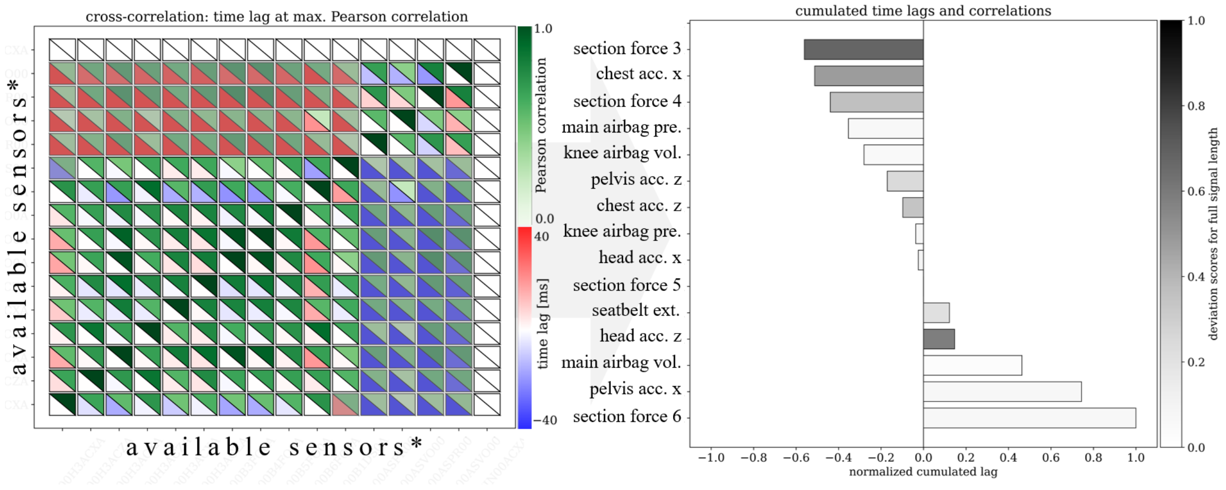


Fig.6: Deviation scores for the simulation with a change in the seatbelt force-level switching time

The bar plot on the right provides further information about the signals in the target simulation and visualizes at which points in time the deviations occur. Each time window has a size of 1 ms and the coloring is according to the amplitude of the deviation score. In the ATD signals the pelvis x-acceleration and head x-acceleration only display minor deviations. The remaining ATD signals display high deviations in a time interval of 70-100 ms. The longest deviations occur in the head z-acceleration and the earliest in the chest x-acceleration. In the signals describing the vehicle lower deviations occur in the main and knee airbag volume and pressure between 20-30 ms. High deviations occur shortly after 65 ms in the section force 3 which is earlier than the subsequent changes in the ATD load and the deviations in the remaining seatbelt section forces. The seatbelt section force 6 only displays marginal deviations as the section is located near the end fitting. This behavior is reasonable since the seatbelt section force 3 is most likely to show deviations when changing either seatbelt force levels or the switching time. The only vehicle signal displaying deviations after 100 ms is the seatbelt extension. No deviations are found to be in the vehicle crash pulse since an identical one was used for all of the simulations.

Fig. 7a) shows the maximum Pearson correlations and the corresponding time lags between the deviation scores over the windows in the investigated simulation. Fig. 7b) displays the resulting cumulated lag values. The grayed-out areas in Fig. 7a) correspond to values that are not considered in the summation. The visualization of the cumulated values displays a more global overview, and the distinct values give more insights if necessary. Possible main causes and secondary effects can directly be obtained in the visualization of the cumulated values in Fig. 7 since the deviations in the signal with the highest negative time lag, on average, lead the deviations in the other signals. In this particular case the seatbelt section force 3 can be identified as the main cause for deviations in both ATD and vehicle signals, which correctly corresponds to the ground truth since the section force 3 is particularly sensitive to changes in the force-level switching time. The deviations in the chest x-acceleration also lead on average which is reasonable due to an almost direct geometrical coupling to the seatbelt section force 3. Based on the knowledge about the changes made in the simulation input the deviations in the remaining signals are most likely to be secondary effects. As shown secondary effects occur in other parts of the restraint system such as the passenger airbag pressure or the seatbelt forces located nearer to the end fitting, but also in the ATD itself. The further specification of the signals and effects need to be conducted by the engineer, as numerical or robustness issues cannot directly be quantified with the presented methodology.



a) b)

Fig.7: a) Lags at maximum Pearson correlation, b) Cumulated time lags for each sensor

5 Conclusion and outlook

The proposed methodology based on the automated analysis of the time series data output from numerical crash simulations is able to generate targeted information for the engineer which can help to quantify and explain the simulation results. The generated result plots for the analyzed simulation contain information about the dataset as a whole, the full-length signal deviations and time window deviation as well as information regarding the time lags of the window deviations. The methodology also highlights the possibilities of combining domain knowledge and data-driven approaches, which also works on relatively sparse datasets as relations do not need to be learned based on large balanced datasets. Due to the self-optimizing properties of the methodology only a physically explainable time lag threshold is required to be set for the highest and lowest allowed lags between deviations by the user. A Power-Point report based on the visualizations is generated in around a minute on a standard workstation, which makes the methodology applicable to daily work since no immense computing power is needed.

Although only the time series data output is used, the description of the system was accurate enough to highlight all relevant areas and reconstruct the event chain for pre-plausibility check a simulation based on previous performed ones. This clearly emphasizes the potential of the time series data analysis in the context of numerical crash simulation as these are mesh-invariant and significantly faster computable than geometry-based approaches. Build upon the generated results, the engineer can mainly focus on highlighted areas and the quality for the simulations is automatically ensured. Especially in the discipline of occupant safety in which the time series data output is highly important due to the relevance for legal regulations and consumer protection guidelines. Since the engineers spend a substantial amount of time analyzing and interpreting the data the presented methodology shows a significant potential for increase time and cost efficiency.

Limitations of the presented approach are especially the potential blind spots due to the lack of sensor signals or the superficial information level the time series data provides. Another aspect is the wrong setting of lag thresholds and dependencies, which requires results to be further analyzed by the engineers. As a result of the assumptions made for the cluster distance analysis and the focus on deviations in the signals the approach not suitable for large datasets containing simulations with strongly deviant behavior.

Consequently, in further investigations it is necessary deliberating to what extent the method can also contribute to post-process optimization also in other load cases for occupant protection such as the side impact, structural crash simulations but also the comparison of physical test data and crash simulations.

6 Appendix

* Order of sensors in Fig. 7a) from left to right: pelvis acceleration x, pelvis acceleration z, chest acceleration x, chest acceleration z, head acceleration x, head acceleration z, section force 3, section force 4, section force 5, section force 6, seatbelt extension, knee airbag pressure, knee airbag volume, main airbag pressure, main airbag volume, crash pulse x.

7 Literature

[1] H.-H. Braess and U. Seiffert, Eds., Vieweg Handbuch Kraftfahrzeugtechnik: mit 122 Tabellen, 6., Aktualisierte und erw. Aufl. in ATZ/MTZ-Fachbuch. Wiesbaden: Vieweg + Teubner, 2011.

[2] B. Bender and K. Gericke, Eds., Pahl/Beitz Konstruktionslehre: Methoden und Anwendung erfolgreicher Produktentwicklung. Berlin, Heidelberg: Springer Berlin Heidelberg, 2021.

[3] B. Klein, FEM: Grundlagen und Anwendungen der Finite-Element-Methode im Maschinen- und Fahrzeugbau. Wiesbaden: Springer Fachmedien Wiesbaden, 2015.

[4] C. Diez, P. Kunze, D. Toewe, C. Wieser, L. Harzheim, and A. Schumacher, “Big-Data Based Rule-Finding for Analysis of Crash Simulations”, in Advances in Structural and Multidisciplinary Optimization, A. Schumacher, T. Vietor, S. Fiebig, K.-U. Bletzinger, and K. Maute, Eds., Cham: Springer International Publishing, 2018, pp. 396–410.

[5] D. Kracker, R. K. Dhanasekaran, A. Schumacher, and J. Garcke, “Method for automated detection of outliers in crash simulations”, International Journal of Crashworthiness, vol. 28, no. 1, pp. 96–107, Jan. 2023.

[6] C.-A. Thole, L. Nikitina, I. Nikitin, and T. Clees, “Advanced Mode Analysis for Crash Simulation Results”, 9th LS-DYNA Forum, Bamberg, 2010.

[7] S. Wolff and C. Bucher, “Recent Developments for Random Fields and Statistics on Structures”, Weimar Optimization and Stochastic Days, Weimar, 2012.

[8] M. K. Belaid, M. Rabus, and R. Krestel, “CrashNet: an encoder–decoder architecture to predict crash test outcomes”, Data Mining and Knowledge Discovery, vol. 35, no. 4, pp. 1688–1709, Jul. 2021.

[9] H. Joodaki, B. Gepner, and J. Kerrigan, “Leveraging machine learning for predicting human body model response in restraint design simulations”, Computer Methods in Biomechanics and Biomedical Engineering, vol. 24, no. 6, pp. 597–611, Apr. 2021.

[10] M. Rabus, M. K. Belaid, S. A. Maurer, and S. Hiermaier, “Development of a model for the prediction of occupant loads in vehicle crashes: introduction of the Real Occupant Load Criterion for Prediction (ROLCp)”, Automotive and Engine Technology, vol. 7, no. 3–4, pp. 229–244, Dec. 2022.

[11] M. Thiele and H. Mullerschön, “Optimization of an Adaptive Restraint System Using LS-OPT and Visual Exploration of the Design Space Using D-SPEX”, 9th International LS-DYNA Users Conference, Detroit, 2006.

[12] C. P. Kohar, L. Greve, T. K. Eller, D. S. Connolly, and K. Inal, “A machine learning framework for accelerating the design process using CAE simulations: An application to finite element analysis in structural crashworthiness”, Computer Methods in Applied Mechanics and Engineering, vol. 385, p. 114008, Nov. 2021.

[13] L. Greve and B. P. van de Weg, “Surrogate modeling of parametrized finite element simulations with varying mesh topology using recurrent neural networks”, Array, vol. 14, p. 100137, Jul. 2022.

[14] Z. Zhang, Y. Wang, P. K. Jimack, and H. Wang, “MeshingNet: A New Mesh Generation Method Based on Deep Learning”, in Computational Science – ICCS 2020, V. V. Krzhizhanovskaya, G. Závodszky, M. H. Lees, J. J. Dongarra, P. M. A. Sloot, S. Brissos, and J. Teixeira, Eds., in Lecture Notes in Computer Science, vol. 12139. Cham: Springer International Publishing, 2020, pp. 186–198.

[15] J. Büttner, S. Schwarz, and A. Schumacher, “Reduzierung des numerischen Ressourcenbedarfs für die multidisziplinäre Optimierung”, ATZ Elektronik, vol. 15, no. 12, pp. 54–59, Dec. 2020.

[16] S. Ackermann, L. Gaul, T. Hambrecht, and M. Hanss, “Principal Component Analysis for Detection of Globally Important Input Parameters in Nonlinear Finite Element Analysis”, Weimar Optimization and Stochastic Days, Weimar, 2008.

[17] B. Bohn et al., “Analysis of Car Crash Simulation Data with Nonlinear Machine Learning Methods”, Procedia Computer Science, vol. 18, pp. 621–630, 2013.

[18] R. Iza-Teran, “Enabling the Analysis of Finite Element Simulation Bundles”, International Journal of Uncertainty Quantification, vol. 4, no. 2, pp. 95–110, 2014.

[19] D. Kracker, J. Garcke, and A. Schumacher, “Automatic Analysis of Crash Simulations with Dimensionality Reduction Algorithms such as PCA and t-SNE”, presented at the 16th International LS-DYNA Users Conference, Virtual Event, 2020.

[20] T. Sprügel, T. Schröppel, and S. Wartzack, “Generic approach to plausibility checks for structural mechanics with deep learning” in Proceedings of the 21st International Conference on Engineering Design (ICED 17) Vol 1: Resource Sensitive Design, Design Research Applications and Case Studies, Vancouver, 2017, pp. 299–308.

[21] J. Garcke and R. Iza-Teran: “Machine Learning Approaches for Repositories of Numerical Simulation Results”, 10th European LS-DYNA Conference, Würzburg, 2015.

[22] A. Pakiman, J. Garcke, and A. Schumacher, “Knowledge discovery assistants for crash simulations with graph algorithms and energy absorption features”, Applied Intelligence, 2023.

[23] Arbeitskreis Messdatenverarbeitung Fahrzeugsicherheit, „Crash Analyse Kriterien“, Version 1.6.2, 2005.

[24] C. C. Aggarwal, “An Introduction to Outlier Analysis”, in Outlier Analysis, Cham: Springer International Publishing, 2017, pp. 1–34.

[25] G. Paaß and D. Hecker, Künstliche Intelligenz: Was steckt hinter der Technologie der Zukunft? Wiesbaden: Springer Fachmedien Wiesbaden, 2020.

[26] M. Längkvist, L. Karlsson, and A. Loutfi, “A review of unsupervised feature learning and deep learning for time-series modeling”, Pattern Recognition Letters, vol. 42, pp. 11–24, Jun. 2014.

[27] J. Runge, P. Nowack, M. Kretschmer, S. Flaxman, and D. Sejdinovic, “Detecting and quantifying causal associations in large nonlinear time series datasets”, Science Advances, vol. 5, no. 11. American Association for the Advancement of Science (AAAS), Nov. 2019.

[28] P. Rousseeuw, “Silhouettes: a graphical aid to the interpretation and validation of cluster analysis”, Journal of Computational and Applied Mathematics, vol. 20, pp. 53-65, 1987.

[29] M. Van der Laan, K. Pollard, and J. Bryan, “A new partitioning around medoids algorithm”, Journal of Statistical Computation and Simulation, vol. 73, no. 8, pp. 575–584, Aug. 2003.

[30] I. Jolliffe., “Principal Component Analysis”, Springer Series in Statistics. New York: Springer-Verlag, 2002.

[31] W. Menke, “Detecting and understanding correlations among data”, Environmental Data Analysis with MatLab® or Python. Elsevier, pp. 277–317, 2022.

[32] Code of Federal Regulations, Title 49 Chapter V Part 571-208, Occupant Crash Protection.