Stapp Car Crash Journal
2023-01

Stapp Car Crash Journal, Vol. 67 (November 2023), pp. 1-13

Machine-Learning-Accelerated Simulations for the Design of Airbag Constrained by Obstacles at Rest

José E. Valenzuela del Río
Siemens Technology (United States)

Richard Lancashire, Karan Chatrath, Peter Ritmeijer
Siemens Industry Software (Netherlands)

Elena Arvanitis, Lucia Mirabella
Siemens Technology (United States)

ABSTRACT – Predicting airbag deployment geometries is an important task for airbag and vehicle designers to meet safety standards based on biomechanical injury risk functions. This prediction is also an extraordinarily complex problem given the number of disciplines and their interactions. State-of-the-art airbag deployment geometry simulations (including time history) entail large, computationally expensive numerical methods such as finite element analysis (FEA) and computational fluid dynamics (CFD), among others. This complexity results in exceptionally large simulation times, making thorough exploration of the design space prohibitive. This paper proposes new parametric simulation models which drastically accelerate airbag deployment geometry predictions while maintaining the accuracy of the airbag deployment geometry at reasonable levels; these models, called herein machine learning (ML)-accelerated models, blend physical system modes with data-driven techniques to accomplish fast predictions within a design space defined by airbag and impactor parameters. These ML-accelerated models are evaluated with virtual test cases of increasing complexity: from airbag deployments against a locked deformable obstacle to airbag deployments against free rigid obstacles; the dimension of the tested design spaces is up to six variables. ML training times are documented for completeness; thus, airbag design explorers or optimization engineers can assess the full budget for ML-accelerated approaches including training. In these test cases, the ML-accelerated simulation models run three orders of magnitude faster than the high-fidelity multi-physics methods, while accuracies are kept within reasonable levels within the design space.

KEYWORDS - Reduced order model, machine learning, neural networks, airbag deployment.

INTRODUCTION

A prime objective of vehicle restraint systems, like belts and airbags, is to mitigate the level of injuries of the vehicle occupants involved in an accident. These restraint systems lead to fewer fatalities and long-term disabilities. However, restraint systems may be less effective if not working as intended, potentially resulting in an increase in injuries.

The airbag deployment phenomenon and its effects on vehicle passengers are complex. Multiple disciplines such as fluid dynamics, multi-body solid rigid mechanics, passenger injury assessment and structural mechanics are involved in assessing airbag deployments.

Address correspondence to José E. Valenzuela del Rio, 755 College Rd E, Princeton, NJ 08540, USA. email: jose.valenzuela-del-rio@siemens.com.

The computational effort for each discipline varies, with fluid dynamics being the most computationally expensive if accuracy is paramount. Moreover, these disciplines interact with each other making the entire system (airbag and impactor) prediction an interwoven problem where the disciplines need to be coupled. This adds another significant burden of system prediction times. State-of-the-art numerical methods include mesh-based Eulerian CFD approaches. More popular are Lagrangian mesh-free approaches or particle methods – examples being the Corpuscular Particle method in LS-Dyna (Lin, 2014) and Finite Pointset method in VPS (Culière, 2003).

The long simulation run times for airbag deployment result in an even more prohibitive design time for airbag manufacturers and vehicle integrators. This is because there are multiple decisions to be made on the airbags and their integration into the vehicle, which require many supporting simulations. For instance, airbag size, material, initial folding, injection temperature, injection jet numbers and direction, porosity, number of straps and chambers, and position with respect to the occupant. Additionally, the deployment of airbags against occupants needs to be assessed for a multitude of scenarios and passenger types so that regulations are met and good star ratings on programs like European New Car Assessment Programme (Euro NCAP) are achieved. For instance, out-of-position (OOP) passenger, adults, children, and anthropometry, among others. The long simulation times may lead to sub-optimal airbag design and integration because new airbag and airbag update development times must be short to stay competitive and detailed thorough design explorations may not be affordable given the long simulation times. Injury risk-reducing airbag designs and vehicle integration with the restriction of short airbag development time can only be achieved by reducing the simulation time of airbag deployment geometries.

Reducing the fidelity of the disciplines involved in the simulation is a common approach to accelerate simulation times. For example, uniform pressure methods simulate gas flow in airbags, where the entire chamber at any given time is at a constant pressure throughout. This approach, much faster and numerically robust, is acceptable when the airbag is almost deployed before the first contact is made. However, it does not capture pressure gradients within the airbag chamber; thus, in situations where the airbag interacts with the environment or occupants during deployment, the accuracy loss in these approaches leads to simulation models with less predictive capabilities, and therefore of dubious value, especially for detailed design phases.

Applying response surfaces and metamodels is another common approach to accelerate simulation times. These approaches target to predict scalar or low-dimensional output of systems given the multi-dimensional input. Gaussian process predictions are common metamodeling techniques which assume a prior covariance across input variables (Williams, 2006) (Kennedy, 2001). Response surfaces are also prediction techniques that assume a functional relationship between input and output variables. Polynomial response surfaces have been employed to predict injury-related scalars such tension, compression, and head injury criterion (Tay, 2014). Also, the roll-over stability of capsules with airbags have been explored with response surface techniques (Horta, 2008). However, these approaches are not appropriate to field prediction such as airbag deployment geometry given their inability to handle large-dimensional outputs (airbag displacements are evaluated in meshes).

Other popular strategies to accelerate the simulations of multi-physics dynamical problems are reduced order models (ROM)s which reduce the dimension of the original high-dimensional output problem by leveraging the fact that the solution resides in a lower-dimension manifold (Audouze, 2013) (Higdon, 2008). A common ROM approach requires the projection of the high-dimensional equations on the low-dimensional space spanned by the bases (Berkooz, 1993). Therefore, this intrusive approach is hard to implement for multi-physics scenarios where solver implementations are complex. Recently, operator inference ROMs have been explored by several researchers with competitive accuracy and long time savings (Peherstorfer, 2016). However, these techniques have not been successfully scaled to the large-dimension design spaces where airbag designers and automotive integrators work. Other non-intrusive ROM methods employ interpolation to assess the basis coefficients. An example of this family is Proper Orthogonal Decomposition with Interpolation (PODI) whose success depends on the availability of data for the interpolation/regressor (Bui-Thanh, 2003).

Machine learning has advanced in the last decades driven by multiple industries such as e-commerce, computer vision, online advertising, and search engines, among others. The significant advances have spurred interest in science and engineering domains. A protein folding prediction machine-learning algorithm has proven to have competitive accuracy with experimental data (Jumper, 2021). Other examples are physics-informed machine learning techniques which combine physical first principles with data-driven approaches to solve forward and inverse problems involving nonlinear partial differential equations (Raissi, 2019). ROM methods such PODI have become more and more popular due to the formidable advance of high-performance computing (HPC) and ML in the last decade (Nguyen, 2022) (Audouze, 2013). Advancements in HPC result in faster and more data generation for computationally expensive simulations such as multi-physics ones. Machine learning has increased the regressor and interpolator quality for a given amount of data.

This work proposes to solve the multi-physics airbag deployment geometry against an obstacle with the PODI method to accelerate the simulation process, and hence, more thoroughly explore the available design space, and shorten the airbag development time and integration period into vehicles. The next section describes the method that creates the ML-accelerated simulation models and the related key performance indicators to assess the performance and quality of the method. Then, the results section explains the test cases for benchmarking the method. It is followed by the conclusion, recommendation, and future work for the proposed method.

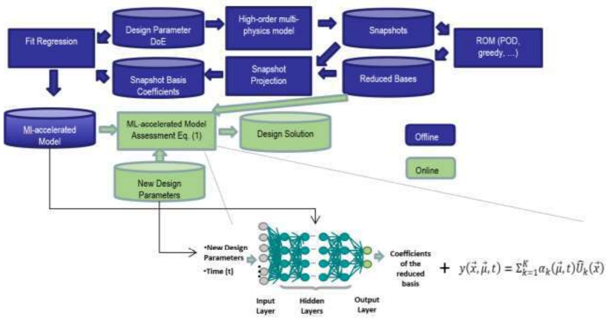

Figure 1. PODI with Neural Networks overview.

METHODS

The authors propose the creation of PODI models which combine the fidelity of system modes and the time-efficiency of data-driven approaches. Proper Orthogonal Decomposition (POD) techniques are employed to capture the airbag displacement bases; whereas neural networks (NN)s bring the computational efficiency required to accelerate the complex simulation.

The proposed method involves two parts: an offline part for data generation and interpolator building, and an online part for unseen evaluations across the design space that corresponds to training parameter space. Note that by "unseen evaluations" authors meant evaluations different from the training ones.

Offline phase

First, a Design of Experiments (DoE) in the design parameter space generates multiple airbag-impactor systems. Specifically, Latin Hypercube Sampling (LHS) methods will plan and conduct a set of evaluations with high-order multi-physics, detailed models to generate the necessary training data for the presented method. Several LHSs with different number of experiments will be performed to study the method accuracy versus the amount of data available. The multi-physics models are built and solved with the commercially available Simcenter Madymo solver (Siemens, 2022). HPC is employed to accelerate this data generation step for the offline phase. The solution, y, at each time step for each system, is called a Snapshot; these Snapshots are stored in a database/matrix $\boldsymbol{Y}$.

$$\boldsymbol{Y} = \begin{bmatrix} \boldsymbol{y}(\vec{\boldsymbol{x}},\overrightarrow{\boldsymbol{\mu_1}},\boldsymbol{t_1}),\ldots\boldsymbol{y}(\vec{\boldsymbol{x}},\overrightarrow{\boldsymbol{\mu_1}},\boldsymbol{t_T}), \\ \boldsymbol{y}(\vec{\boldsymbol{x}},\overrightarrow{\boldsymbol{\mu_2}},\boldsymbol{t_1}),\ldots,\boldsymbol{y}(\vec{\boldsymbol{x}},\overrightarrow{\boldsymbol{\mu_{N_{DOE}}}},\boldsymbol{t_T}) \end{bmatrix}$$

Where $\vec{\boldsymbol{x}} \subset \mathbb{R}^3$ are the initial Euclidean coordinates for the initial position of the airbag nodes, $\overrightarrow{\boldsymbol{\mu_i}} \in \mathbb{R}^{\boldsymbol{P}}$ is the *Design Parameter* values for DOE member $\boldsymbol{i}$, $\boldsymbol{N_{DOE}}$ is the number of DOE members, and $\boldsymbol{T}$ is the number of time steps for the simulated system. Note that $\boldsymbol{Y} \in \mathbb{R}^{\boldsymbol{N x L}}$ with $\boldsymbol{N}$ the number of cells of the grid used in the high-fidelity solver and $\boldsymbol{L} = \boldsymbol{T} \cdot \boldsymbol{N_{DOE}}$. Note that $\boldsymbol{y}$ is a vector where each component represents the solution at a grid node. For notation simplicity, this vectorial nature of $\boldsymbol{y}$ is not included in the equations.

The *Reduced Bases* for the parametric system are assessed by POD (however, other techniques such as greedy approaches could easily be used as well); it is done by calculating the Singular Value Decomposition of matrix $\boldsymbol{Y}$. These bases are stored in a database for easy future access.

$$\boldsymbol{Y} = \boldsymbol{U}\,\boldsymbol{\Sigma}\,\boldsymbol{V}^{\boldsymbol{T}}$$

where the matrix $\boldsymbol{U} \in \mathbb{R}^{\boldsymbol{NxR}}$, the matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{\boldsymbol{RxR}}$, the matrix $\boldsymbol{V} \in \mathbb{R}^{\boldsymbol{LxR}}$, and $\boldsymbol{R}$ is the rank of $\boldsymbol{Y}$. Notice that the matrix $\boldsymbol{U}$ holds the left-singular vectors of $\boldsymbol{Y}$. Given that these vectors are orthonormal, the $\boldsymbol{U_i} \in \mathbb{R}^{\boldsymbol{N}}$columns form the *Reduced Bases* for efficiently representing any solution of the target variable $\boldsymbol{y}$; if keeping only the $\mathbf{K}$ first bases of $\mathbf{U}$, $\widehat{\boldsymbol{U}} \in \mathbb{R}^{\boldsymbol{NxK}}$ is built. Note that $\boldsymbol{K}$ is the dimension of the reduced space which is taken to be $\boldsymbol{K} \ll \boldsymbol{R}$. In other words, the target variable can be expressed in terms of:

$$\boldsymbol{y}(\vec{\boldsymbol{x}},\vec{\boldsymbol{\mu}},\boldsymbol{t}) = \boldsymbol{\Sigma}_{\boldsymbol{k=1}}^{\boldsymbol{K}}\boldsymbol{\alpha_k}(\vec{\boldsymbol{\mu}},\boldsymbol{t})\widehat{\boldsymbol{U}}_{\boldsymbol{k}}(\vec{\boldsymbol{x}}) \qquad (\mathbf{1})$$

where $\boldsymbol{\alpha_k}(\vec{\boldsymbol{\mu}},\boldsymbol{t})$ are the basis coefficients for the given time and *Design Parameters* $\vec{\boldsymbol{\mu}}$. Note that $\widehat{\boldsymbol{U}}_{\boldsymbol{k}}(\vec{\boldsymbol{x}}) \in \mathbb{R}^{\boldsymbol{N}}$ on the Euclidean coordinates while $\boldsymbol{\alpha_k}(\vec{\boldsymbol{\mu}},\boldsymbol{t})$ depends only on the *Design Parameters* and time.

To build the PODI interpolators that map from the design parameter space and time to the basis coefficients, these coefficients for the training data must be calculated. It is achieved by projecting the snapshots $\boldsymbol{Y}$ into the *Reduced Bases* $\widehat{\boldsymbol{U}}$ which leads to the *Snapshots Bases Coefficients* $\mathbf{A} \in \mathbb{R}^{\boldsymbol{KxN_{DOE}}}$ for the training data.

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{\alpha_1}(\overrightarrow{\boldsymbol{\mu_1}},\boldsymbol{t_1}) & \ldots & \boldsymbol{\alpha_1}(\overrightarrow{\boldsymbol{\mu_{N_{DOE}}}},\boldsymbol{t_T}) \\ \ldots & \ldots & \ldots \\ \boldsymbol{\alpha_K}(\overrightarrow{\boldsymbol{\mu_1}},\boldsymbol{t_1}) & \ldots & \boldsymbol{\alpha_K}(\overrightarrow{\boldsymbol{\mu_{N_{DOE}}}},\boldsymbol{t_T}) \end{bmatrix} = \widehat{\boldsymbol{U}}^{\boldsymbol{T}}\mathbf{Y} \quad (\mathbf{2})$$

Thus, the PODI regression/interpolation model is as follows:

$$\boldsymbol{\alpha_k(\vec{\mu}, t) = NN_k(\vec{\mu}, t)\ for\ k = 1, .. K} \quad \mathbf{(3)}$$

The $\boldsymbol{NN_k}$, a neural network regressor for the basis coefficient $\boldsymbol{k}$, is fit with the full-model data $\boldsymbol{Y}$ and the corresponding coefficients $\boldsymbol{A}$ (see Equation 2) at the evaluation points $\boldsymbol{\vec{\mu_i}\ for\ i = 1, \ldots N_{DOE}}$, and $\boldsymbol{t_l\ for\ l = 1, \ldots, T}$. The $\boldsymbol{K}$ regressors, *ML-accelerated Models* in Figure 1, are stored in the corresponding database. All the ML models shown are trained with the Adam algorithm, and *epochs*=15,000. Each epoch is done with just one batch, given that the largest dataset contains 20,000 data points (maximum number of airbag-impactor system $\boldsymbol{N_{DOE} = 400}$, each with $\boldsymbol{T = 50}$ time steps).

Online phase

Once the $\boldsymbol{NN_k}$ are available, the basis coefficients for any unseen airbag-impactor $\boldsymbol{\vec{\mu}}$ at any time $\boldsymbol{t}$ will be assessed based on Equations 3; note that the inputs for this step are $\boldsymbol{\vec{\mu}}$, $\boldsymbol{t}$, and $\boldsymbol{NN_k\ for\ k = 1, .. K}$. Then, the high-order solution will be assessed based on Equation 1 with the only added input of the basis $\boldsymbol{\widehat{U}_k}$. Notice that the computational expense of these two online operations is much smaller than those for assessing the high-dimensional solution where CFD and FEA solvers need to be solved in a combined way.

ML-accelerated Model Including Forces

The shapes of airbags during inflation against a physical obstacle depend on the contact forces between the two entities (airbag and obstacle). Therefore, it makes sense from the physical perspective to include the forces at the contact points (contact forces) at time $\boldsymbol{t-1}$ as inputs to the ML-accelerated approach when assessing the airbag geometry at time $\boldsymbol{t}$. Specifically for complex systems such as free obstacles where the impactor location varies during the inflation, it is more difficult to correctly predict the airbag deployment process. The main challenge of including the contact forces as inputs is that it is a vectorial field whose domain is the whole airbag. For instance, if the airbag is discretized with a mesh of 10,000 nodes, the dimension of the contact force is 30,000 which cannot be managed efficiently with the PODI approach. This is because the input layer of $\boldsymbol{NN_k}$ would need to increase accordingly resulting in exceptionally large NN which would require a big data set and a long training time.

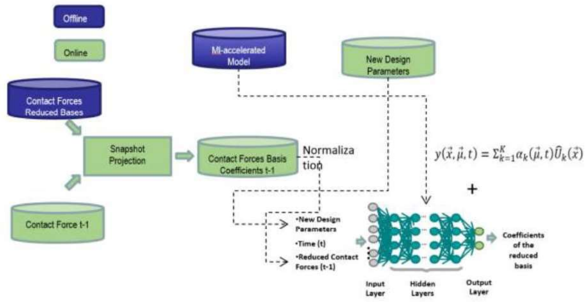

Figure 2. Augmented approach to include Contact Forces information from the previous Time Step (t-1).

This work proposes to reduce the previous step *Contact Forces* by assessing their POD *Reduced Bases*, $\boldsymbol{V_m} \in \mathbb{R}^{\boldsymbol{N}}$ $\mathbf{for}$ $\boldsymbol{m = 1, \ldots M}$, and expanding them up to the first *M Contact Forces Basis Coefficients*, $\boldsymbol{\beta_{m,c}(\vec{\mu}, t)}$ for $\boldsymbol{m = 1, \ldots M; c \in \{f_x, f_y, f_z\}}$, i.e. the contact forces are expanded as follows:

$$\boldsymbol{f_c(\vec{x}, \vec{\mu}, t) = \Sigma_{m=1}^{M} \beta_{m,c}(\vec{\mu}, t)\widehat{V}_{m,c}(\vec{x})\ \ for\ c \in \{x, y, z\}} \mathbf{(4)}$$

Then, the *Contact Forces Basis Coefficients* for the earlier time step $\boldsymbol{t-1}$, $\boldsymbol{\beta_{m,c}(\vec{\mu}, t-1)}$ are input to the PODI NN. Figure 2 shows this additional process. During the offline phase, for each contact force component, $\boldsymbol{c}$, the Euclidean norm for all the training snapshots assessed by the high-fidelity multi-physics solver can be seen as samples of a distribution. Each of these distributions has an expected value $\boldsymbol{\mu}$ and standard deviation $\boldsymbol{\sigma}$ that can be estimated based on these samples. Distribution outliers outside the range $\boldsymbol{\mu +/- n_{std}\sigma}$ of the contact force distributions are removed for each of the three components for the contact force distribution. After these removals, the *Contact Forces Reduced Bases* and the corresponding coefficients $\boldsymbol{\beta_{m,c}(\vec{\mu}, t)}$ for the three $\boldsymbol{c}$ vector components will be assessed with the POD methods for the remaining cleaned training data. Means $\boldsymbol{\mu_{c,m}^{\beta}}$ and standard deviations $\boldsymbol{\sigma_{c,m}^{\beta}}$ are calculated for each distribution of *Contact Force Basis Coefficients*.

The added inputs for the PODI NN are the $\boldsymbol{\beta_{m,c}(\vec{\mu}, t)}$ coefficients at the earlier time $\boldsymbol{t-1}$. Before training these coefficients are normalized into the range [0,1] to ensure that all the inputs to the PODI NN have similar ranges, and thus, the training is smoother. This normalization is achieved by transforming these basis coefficients at $\boldsymbol{t-1}$ with the Normal Cumulative Function Distribution (NCFD) of the corresponding $\boldsymbol{(c, m)}$ distribution of the *Contact Forces Basis Coefficients* of the training data, $\boldsymbol{\Phi(\mu = \mu_{c,m}^{\beta}, \sigma = \sigma_{c,m}^{\beta})}$. The offline training can then be performed. Note that the *New Design Parameters* (including impactor and airbag geo-physical parameters) and time, as previously, are the remaining inputs to the PODI NN.

During the online phase, the previous *Contact Force* at $\boldsymbol{t-1}$ are projected into the *Contact Forces Reduced Bases* calculated during the offline phase to obtain the testing/unseen *Contact Forces Basis Coefficients* at $\boldsymbol{t-1} \quad \boldsymbol{\beta_{m,c}(\vec{\mu}, t)} \quad \boldsymbol{for\ m = 1, \ldots M; c \in \{f_x, f_y, f_z\}}$ where $\boldsymbol{c}$ is the component of the contact force and $\boldsymbol{m}$ the basis coefficient. Once these $\boldsymbol{\beta_{m,c,t-1}}$ values are available, the evaluation/inference of the unseen airbag-impactor system can be accomplished.

Performance Metrics. This work pursues the acceleration of airbag deployment geometry evaluations with respect to full-order solutions derived from traditional numerical methods that involve CFD and FEA solvers. It is also important to understand the accuracy of the ML-accelerated methods, so potential users can safely decide when to use the proposed methods. Therefore, the following metrics are employed to evaluate the PODI-based solution:

1. ML-accelerated inference time: time to assess the behavior of an unseen airbag-impactor system. The interest is in the acceleration with respect to the full-order model.

2. Mean root mean squared error (RMSE): the mean RMSE across all times $\boldsymbol{T}$, and all $\boldsymbol{N\_test}$ unseen airbag-impactor systems used for testing.

$$\overline{\boldsymbol{RMSE}} = \frac{\mathbf{1}}{\boldsymbol{N_{test}\cdot T}} \sum_{i,l=1}^{N_{test},T} \boldsymbol{RMSE_{i,l}(\vec{x}, \vec{\mu_\iota}, t_l)} = \frac{\mathbf{1}}{\boldsymbol{N_{test}\cdot T}} \sum_{i,l=1}^{N_{test},T} \sqrt{\frac{\mathbf{1}}{\boldsymbol{N}} \sum_{j=1}^{N} \left(\boldsymbol{y_{gt,j}(\vec{x}, \vec{\mu_\iota}, t_l) - y_j(\vec{x}, \vec{\mu_\iota}, t_l)}\right)^2} \quad (5)$$

where $\boldsymbol{y_{gt,j}}$ is the ground-truth for the full-order model solution and $\boldsymbol{y_j}$ is the ML-accelerated solution given by Equation 1, both given at the node $\boldsymbol{j}$. Note that $\boldsymbol{RMSE}$ and $\overline{\boldsymbol{RMSE}}$ have a minimum value of zero which imply perfect accuracy; all the remaining values of these metrics are positive; the larger the values the lower the accuracy. Additionally, $\boldsymbol{RMSE}$ could be seen as samples of a distribution with mean $\overline{\boldsymbol{RMSE}}$**.**

3. ML-accelerated training time: the time needed to fit the NN regressor given by Equation 3. This is an upfront time that needs to be incurred once for a given design space. However, it will be crucial for airbag designers and integrators to decide whether the ML-accelerated model is suitable for their design, optimization, and robustness analysis. All the offline training times models covering the parameter spaces are done on the same Intel(R) laptop.

It is worth mentioning that the ML-accelerated simulation depends on several parameters. Specifically, the above performance metrics' sensitivity to the following two parameters will be studied: $\boldsymbol{N_{DOE}}$ the number of airbag-obstacle systems, and $\boldsymbol{K}$ the dimension of the reduced space. Sensitivity along other algorithmic parameters such as the number of layers and neurons in the NN will not be studied in this work.

RESULTS AND DISCUSSION

For benchmarking the proposed ML-accelerated simulation model a Simcenter Madymo validated model has been created consisting of an unfolded single-chamber driver airbag resting on a flat rigid structure. Above this flat structure an impactor is located. Via define-variables the main parameters effecting the airbag and impactor model responses can be varied: e.g. permeability, venting hole sizes, mass flow rate and jet directions of the inflator, airbag shape via tether/strap lengths but also the mass, initial position, and velocity as well as the shape and size of the impactor can be easy varied.

Two cases with increasing complexity are tested:

1. Locked flexible obstacle with a force-based contact characteristic: The purpose for this first test case is to assess the ML-accelerated approach with an obstacle with different contact stiffness but locked. The number of parameters for the airbag design and obstacle choice is $\boldsymbol{P=6}$. The parameters list and their ranges are shown in the Table 1, and instances in Figure 3.

2. Free rigid obstacle: The goal is to extend to a more complex test case where the obstacle is not restricted and therefore free to move. The number of parameters and their ranges are the same as the ones in Table 1, except that the impactor object is effectively rigid ('Contact Charact. Slope' is held constant to a value of $\boldsymbol{2\cdot 10^6 N/m}$**),** and an additional variable 'Impactor Orientation around the axis X', defined in the range [0°, 45°], is included.

Parameters	Min	Max
Airbag Permeability (-)	0.02	0.035
Inflator Temperature (°K)	500	800
Jet Direction (degrees)	30	60
Impactor Radius (m)	0.1	0.25
Ell Degree (-)	2.0	4.5
Contact Charact. Slope (N/m)	1000.0	10^6

Table 1: Airbag design parameters and impactor parameters.

The ML-accelerated models are tested in the above two test cases: 1) airbag deployment geometry against a locked, flexible obstacle and 2) airbag deployment geometry against a free, rigid obstacle. For these two cases, the Performance Metrics described in the corresponding subsection are assessed and analyzed. Note that both models are tested against unseen airbag-impactor systems which are within the training parameter space.

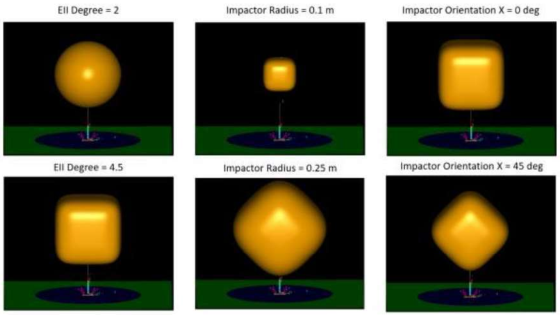

Figure 3. Maximum and minimum instances of the obstacle space.

Airbag deployment geometry against a locked flexible obstacle with a force-based contact characteristic

The first performance metric to study is the inference time for the airbag displacements. The NN_k for each basis coefficient k is of the feedforward type with 5 layers and the corresponding number of neurons [7, 25, 45, 15, 1]; the learning rate is set to $3 \cdot 10^{-4}$. For an ML-accelerated model hyperparameters with $N_{DOE} = 400$ and $K = 12$ the inference run times are shown in Table 2. The numbers in Table 2 are more than three orders of magnitude lower than the multi-physics evaluation run time of about 3,930 s provided by Simcenter Madymo. Note that both evaluation run times have been assessed in the same computer (see Subsection 'Performance Metrics'). Note that the inference time dependency on the hyperparameters N_{DOE} and K is not shown in Table 2 because they are not significant: 1) in the range $N_{DOE} = [100, 400]$ the used NN sizes were the same, and 2) the displacement inference times scales approximately linearly with K. It is worth remembering that these ML-accelerated models are parametric, thus, these modes can evaluate variations of airbag designs and obstacles within the training design space.

Task	Time(s)
X-Model Eval	0.29
Y-Model Eval	0.26
Z-Model Eval	0.27
Models Loading	0.12

Table 2: Inference relevant times. ML-accelerated model. Airbag inflation against a locked obstacle with varying stiffness.

Next, the ML-accelerated model airbag inflation with hyperparameters $N_{DOE} = 400$ and $K = 12$ for an unseen airbag impactor system is compared in Figure 11 in Appendix A with that of the ground truth provided by a validated Simcenter Madymo model. The inflation process is captured quite well both on global and local level by the ML-accelerated model with the associated tremendous run time evaluation advantage of more than three orders of magnitude with respect to that of Simcenter Madymo. Figure 11 in Appendix A shows also the $RMSE$ for the z-displacement at the six selected times, t. Among the shown times in Figure 11 in Appendix A, the largest z-displacement $RMSE$ is at $t = 30\,ms$ where discrepancies between the two results are visible at both the bottom and top part of the cross-section.

Next, the accuracy of the ML-accelerated model is quantitatively analyzed with the previously-introduced performance metric $\overline{RMSE}$ for the z-displacement and its influence with respect to the N_{DOE} and K hyperparameters, see Figure 4. Note that 50 unseen airbag-impactor system inflations are used for testing purposes. As expected, increases in both parameters result in lower $\overline{RMSE}$. Figure 4 also shows that there is a saturation of around $K = 10$.

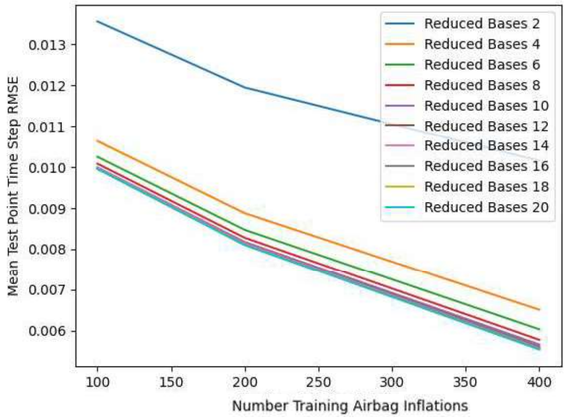

Figure 4. z-displacement $\overline{RMSE}$ (m) vs K (number of modes) for $N_{test} = 50$ unseen airbag-impactors. Reference length is the airbag strap of $0.28\,m$. Airbag inflation against a locked obstacle with varying stiffness.

Figure 5 shows more details of the $RMSE$ distribution: 90% and 10% percentiles together with $\overline{RMSE}$; it is shown that the $RMSE$ distribution becomes thinner (smaller dispersion) and the $\overline{RMSE}$ lower with the increase of the number of training airbag-impactor systems N_{DOE}. A smaller dispersion for the test data means that the method accuracy varies less with respect to the $\overline{\boldsymbol{RMSE}}$ value, leading to more consistency in the results.

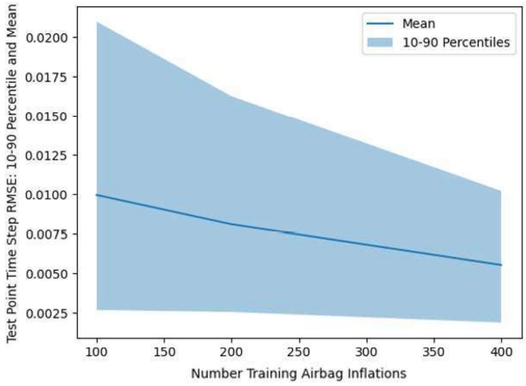

Figure 5. z-displacement $\overline{\mathrm{RMSE}}$ (m) distribution for $\mathrm{N_{test}} = 50$ unseen airbag-impactors. Reference length is the airbag strap of 0.28 m. Airbag inflation against a locked obstacle with varying stiffness.

Table 3 shows the ML-accelerated offline times for $\boldsymbol{K = 12}$, and $\boldsymbol{N_{DOE}} \in [\mathbf{100, 400}]$. This offline time includes data generation, POD assessment and PODI NN training. As expected, the total training time (POD plus NN) increases (faster than linearly) with the number of training airbag-impactor systems. This time increase is more significant in the POD step than in the NN training. It is also worth highlighting that the POD and NN training times are of the same order of magnitude as generating just the high-fidelity, multi-physics solution for one airbag-impactor system. Therefore, for the studied $\boldsymbol{N_{DOE}}$ range the data generation is the longest time step for building the ML-accelerated model.

$\boldsymbol{N_{DOE}}$	Gener. Time (s)	POD (s)	NN Training (s)	Total Trn. Time(s)
100	393000	93	2376	2469
200	786000	495	3348	3843
400	1572000	1254	5832	7086

Table 3: Offline phase times. ML-accelerated model K = 12. Airbag inflation against a locked obstacle with varying stiffness.

Table 3 shows the ML-accelerated offline times for $\boldsymbol{K = 12}$, and $\boldsymbol{N_{DOE}} \in [\mathbf{100, 400}]$. This offline time includes data generation, POD assessment and PODI NN training. As expected, the total training time (POD plus NN) increases (faster than linearly) with the number of training airbag-impactor systems. This time increase is more significant in the POD step than in the NN training. It is also worth highlighting that the POD and NN training times are of the same order of magnitude as generating just the high-fidelity, multi-physics solution for one airbag-impactor system. Therefore, for the studied $\boldsymbol{N_{DOE}}$ range the data generation is the longest time step for building the ML-accelerated model.

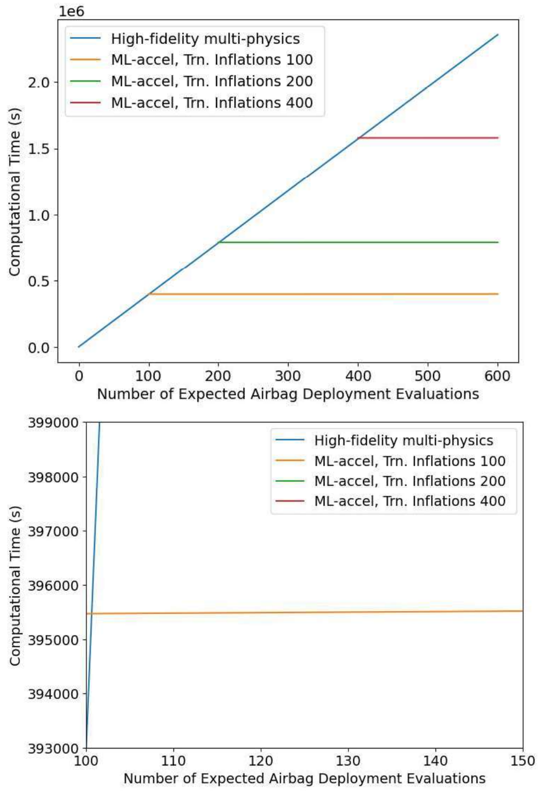

Figure 6. Evaluation time needed as a function of number of expected airbag deployment evaluations. Top: in the range $\mathrm{N_{DOE}} = [0, 600]$. Bottom: Zoomed in around $\mathrm{N_{DOE}} = 100$. Airbag inflation against a locked obstacle with varying stiffness.

Figure 6 shows the time comparison between the high-fidelity multi-physics simulations and the ML-accelerated models as a function of the number of expected airbag-impactor evaluations. The former is more performant for analysis with low number of expected evaluations. The latter has an overhead due to the data generation; the training time overhead for the studied $\boldsymbol{N_{DOE}}$ is negligeable, as previously commented it is the same order of magnitude as generating one airbag-impactor system evaluation. However, once $\boldsymbol{N_{DOE}}$ reaches certain values, the ML-accelerated method becomes more time efficient than traditional methods and is a competitive alternative provided their accuracy fits the study requirements.

Airbag deployment geometry against a free rigid obstacle

Given the higher complexity of modelling the airbag inflation against a free obstacle, two PODI input approaches are used to accelerate these simulations:

1. use only the geophysical parameters of the airbag-impactor system and time; $\boldsymbol{NN_k}$ of the feedforward type with 5 layers and the corresponding number of neurons [7, 30, 40, 10, 1].

2. augment the first approach with contact force information where only the first three modes for the contact force component z are used, i.e. $\boldsymbol{M = 3}$, see Subsection 'ML-accelerated Model including Forces'; $\boldsymbol{NN_k}$ of the feedforward type with 5 layers and the corresponding number of neurons [10, 30, 40, 10, 1].

The learning rate is set to $\boldsymbol{1 \cdot 10^{-3}}$ for both studies.

Figure 7 shows the comparison of the three displacement $\overline{\boldsymbol{RMSE}}\boldsymbol{s}$ for the two PODI approaches. The PODI approach that includes contact forces as inputs leads not only to more accurate results but also to more stable training results,

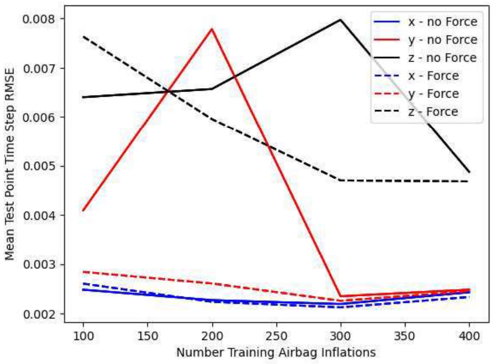

Figure 7. $\overline{\text{RMSE}}$(m) Comparison ML-accelerated Models 1) no Forces (solid) and b) with Forces (dashed). Reference length is the airbag strap of 0.28 m. Airbag inflation against a free rigid obstacle.

i.e. the approach with no contact forces results in an unexpected high $\overline{\boldsymbol{RMSE}}$ for the y component at $\boldsymbol{N_{DOE} = 200}$ and for the z component at $\boldsymbol{N_{DOE} = 300}$. Therefore, in this section the PODI approach that includes the contact forces as inputs is employed.

As with the first test case, first the inference time for the airbag displacements is studied. For an ML-accelerated model hyperparameters are $\boldsymbol{N_{DOE} = 400}$ and $\boldsymbol{K = 12}$ the times are shown in Table 4. The numbers in Table 4 are more than three orders of magnitude lower than the multi-physics evaluation time of 3750 s provided by Simcenter Madymo.

Task	Time(s)
X-Model Eval	0.36
Y-Model Eval	0.33
Z-Model Eval	0.34
Models Loading	0.08

Table 4: Inference relevant times. ML-accelerated model with Forces. Airbag inflation against a free rigid obstacle.

Figure 12 in Appendix A shows the airbag cross-sections during inflation at six time points to compare the ML-accelerated model with hyperparameters $\boldsymbol{N_{DOE} = 400}$ and $\boldsymbol{K = 12}$, and the ground truth provided by Simcenter Madymo. Again, the airbag deployment geometry looks quite well captured by the ML-accelerated model to the naked eye. Figure 12 in Appendix A shows also the $\boldsymbol{RMSE}$ for the $\boldsymbol{z}$-displacement at the six selected times, $\boldsymbol{t}$. Among the shown times in Figure 12 in Appendix A, the largest $\boldsymbol{RMSE}$ is at $\boldsymbol{t = 15\,ms}$ where $\boldsymbol{z}$-displacement discrepancies between the two results are visible at both the top and left part of the cross-section.

Figure 9 shows the $\boldsymbol{z}$-displacement $\overline{\boldsymbol{RMSE}}$ and its influence with respect to the $\boldsymbol{N_{DOE}}$ and $\boldsymbol{K}$ hyperparameters. Again, this metric is computed with 50 unseen airbag-impactor system inflations used for testing purposes. For this second test cases saturations are seen for the two hyperparameters. The gain from moving from 300 to 400 airbag-impactor system inflations is negligible. Figure 8 shows the $\boldsymbol{RMSE}$ distribution; similarly to the previous test case, the $\boldsymbol{RMSE}$ distribution becomes thinner with the increase of the number of training airbag-impactor systems $\boldsymbol{N_{DOE}}$.

Note that for the unseen/testing airbag-impactor systems, this section assumes that the contact forces are known - as opposed to the airbag nodal displacements which are unknown and assessed by the ML-accelerated model. Future work will include a separate multi-body impactor modelling which will provide the contact forces to the ML-accelerated model, therefore, the pre-requisite for knowing the forces in advance will be removed.

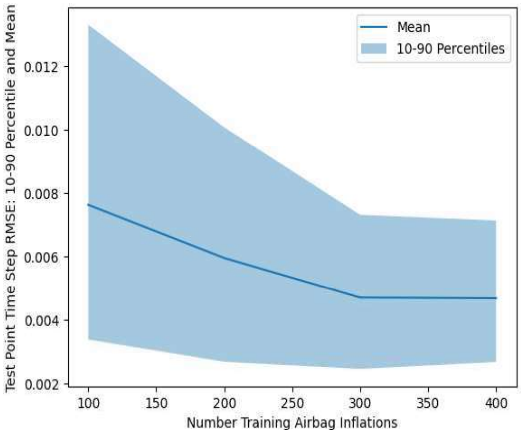

Figure 8. z-displacement $\overline{\text{RMSE}}$ (m) distribution for $\text{N}_{\text{test}} = 50$ unseen airbag-impactors. Reference length is the airbag strap of 0.28 m. Airbag inflation against a free rigid obstacle.

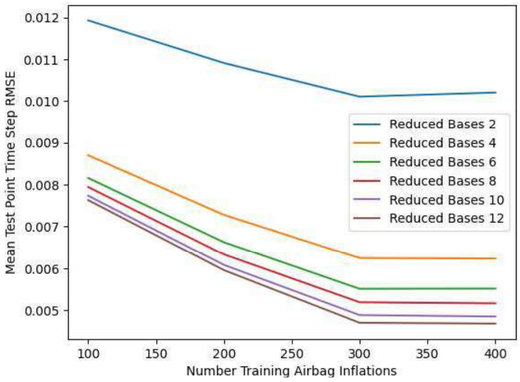

Figure 9. z-displacement $\overline{\text{RMSE}}$ (m) vs K (number of modes) for $\text{N}_{\text{test}} = 50$ unseen airbag-impactors. Reference length is the airbag strap of 0.28 m. Airbag inflation against a free rigid obstacle.

Table 5 shows the ML-accelerated offline times for $\boldsymbol{K = 12}$ and $\boldsymbol{N_{DOE}} \in [\mathbf{100}, \mathbf{400}]$. As in the first test case, the total training time increases (faster than linearly) with the number of training airbag-impactor systems. Again, this time increase is more significant in the POD step than in the NN training. Also, the training times (not data generation times) for this second test case are higher than for the first fixed-obstacle fixed case. This is because the NN used has higher capacity since there are more inputs to the NN and the displacement modes are more complex in this test case where the obstacle is free to move during the inflation. As it was found in the first test case, the POD and NN training times are of the same order of magnitude as just one high-fidelity multi-physics solver. Therefore, for the studied $\boldsymbol{N_{DOE}}$ range, the data generation is much more time-intensive than the remaining offline steps.

$\boldsymbol{N_{DOE}}$	Gener. Time (s)	POD (s)	NN Training (s)	Total Trn. Time(s)
100	375000	105	3420	3525
200	750000	492	3960	4452
400	1500000	1722	6912	8634

Table 5: Offline phase times. ML-accelerated model with Forces K = 12. Airbag inflation against a free rigid obstacle.

Figure 10 shows the time comparison between the high-fidelity multi-physics simulations and the ML-accelerated models as a function of the number of expected airbag-impactor evaluations. The trends are like the ones in the first test cases: the former is more performant for analysis with low number of expected evaluations; whereas the latter becomes more time efficient than traditional methods as the number of expected evaluation increases and becomes a competitive alternative provided their accuracy fits the study requirements.

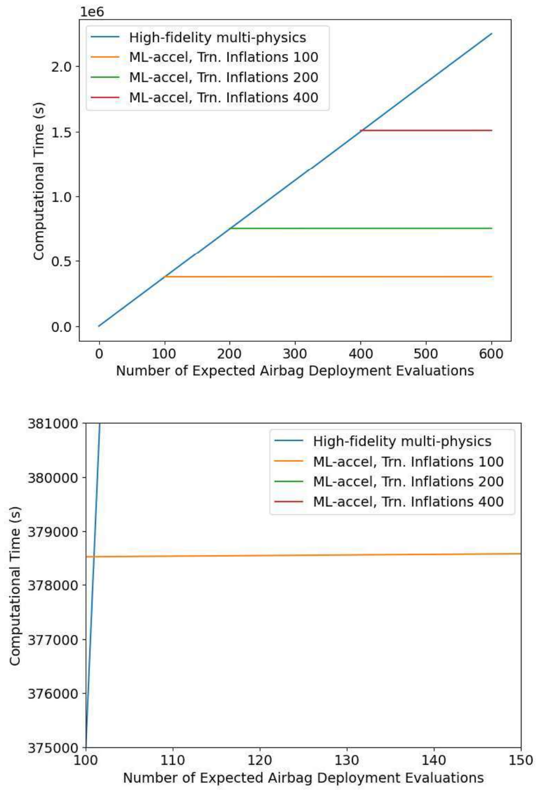

Figure 10. Evaluation time needed as a function of number of expected airbag deployment evaluations. Top: in the range $\text{N}_{\text{DOE}} = [0, 600]$. Bottom: Zoomed in around $\text{N}_{\text{DOE}} = 100$. Airbag inflation against a free rigid obstacle.

CONCLUSION

Significant inference time gains in the range of more than three orders of magnitude (up to 3750 times faster) have been demonstrated for the two test cases involving airbag deployment geometries against impactors. Additionally, these simulation time speed-up gains come with accuracies up to $\mathbf{98}\%$ (defined as the $\left(\boldsymbol{l_{ref}} - \overline{\boldsymbol{RMSE}}\right)\big/\boldsymbol{l_{ref}}$ where $\boldsymbol{l_{ref}}$, the reference measure for the calculation, is set to the airbag strap length because it represents a magnitude of the same order of the maximum airbag displacements during the deployment) with respect to Simcenter Madymo. The ML-accelerated simulation models are valid in the training parameter spaces; therefore, these models are suitable for inferring airbag-impactor system behaviors within these training parameter spaces provided by Table 1 (instances are shown in Figure 3). The two test cases demonstrated have parameter spaces of dimension six.

The time-efficiency during inference, parametric capability within the training parameter space, and accuracy of the ML-accelerated simulation approach make them suitable for optimization, robustness checking and sensitivity analysis of airbag deployment geometry simulations against obstacles during early and preliminary design phases. These models can be used by airbag and car designers alike, and can, therefore, reduce airbag development time. Additionally, the proposed ML-accelerated models are not intrusive to fully-fledged implementations as opposed to other speed-up mechanisms. Finally, it is worth noting that these ML-accelerated simulation approaches complement existing fully-fledged multi-physics models, since the former require the latter for data generation.

Offline phase times (including POD calculation and NN training) represent an upfront cost that is negligible with respect to the data generation time for the studied $\boldsymbol{N_{DOE}}$. Therefore, in optimization and uncertainty quantification applications, these ML-accelerated models are beneficial provided that the number of expected evaluations is higher than necessary number of training deployments $\boldsymbol{N_{DOE}}$ for the ML-accelerated model to achieve the desired accuracy (see Figure 6 and Figure 10). The accuracy can be controlled with the hyperparameters $\boldsymbol{N_{DOE}}$ and $\boldsymbol{K}$ (Figure 4 and Figure 9); and data generation and offline times can be controlled with the hyperparameter $\boldsymbol{N_{DOE}}$ (Figure 6 and Figure 10) as shown in the two test cases; ML-accelerated model users should capitalize on these trade-offs to further accelerate their airbag-impactor system explorations if their tasks do not require extremely high accuracy.

The dominant factor limiting the scalability of the method to larger dimension parameter spaces is an increase of POD calculation time due to the expected larger number of snapshots $\boldsymbol{N_{DOE}} \cdot \boldsymbol{T}$ need for larger parameter spaces. Alternatives to further scale this approach are to include more computing power (better computers or distributed clusters) and/or employ randomized SVD (Halko, 2011). This alternative should allow the proposed method to be extended to parameter spaces with dimensions double or triple the ones tested in this work. The need to bring larger NNs should not be limiting since in the second test the NN input space was augmented by $\boldsymbol{M = 3}$ and the resulting NN training and inference time increased only modestly.

The presented method is still in its infancy given the shown test cases in this work. Even though the time evolution of airbag deployment geometry (focus of the current work) is a crucial ingredient to assess occupants' accelerations, and hence, the safety, biomechanics and injury problems, the proposed method still needs to interface with multi-body models/algorithms to comprehensively assess safety and biomechanics. Additionally, the current work has been tested against relatively simple airbag-impactor systems (unfolded, one-chamber, parametric simple obstacle made of one component, …), and design spaces of dimension six.

FUTURE WORK

The model extension to complete the impactor modeling and its acceleration assessment, and therefore, the associated anthropomorphic test device (ATD) injury assessment, is needed to completely produce a comprehensive simulation that can serve current airbag design and integrator workflows. Authors intend to integrate the proposed ML-accelerated simulation models for airbag deployment geometry with multi-body simulators to assess occupants' metrics such as accelerations, forces, and moments, and hence, injury and biomechanics assessments. This will be realized as follows: at each point of time, the proposed ML-accelerated model will output airbag displacements that will be input to the multi-body simulators which, in turn, will output the contact forces that will be fed to the ML-accelerated model in the next time step. Questions such as time resolution needed for ML-model training, communication frequency, scaling, and interpolation robustness across the training spaces for this integration need to be answered before the proposed method can be employed by airbag designers.

Additionally, studies investigating the scalability of the approach with a high dimension of the design space will be the focus of future work as previously mentioned. Other investigations to mature the method and make it production-environment ready are benchmarking analyses with more complex test cases including but not limited to several chambers, initially folded airbags, belted ATDs, multiple airbags, and human body/mannequins' models composed of several components. Other interesting future work is a proper study to determine whether the proposed method has any limited extrapolation capability given its expressivity is constrained within the reduced basis.

ACKNOWLEDGMENTS

We are grateful to Emmanuel Motheau and Scott Kolb for their automation and infrastructure tasks, respectively. Their valuable work enabled the research presented in this paper.

REFERENCES

Audouze, C. F. (2013). Nonintrusive reduced-order modeling of parametrized time-dependent partial differential equations. *Numerical Methods for Partial Differential Equations* 29.5, 1587-1628.

Berkooz, G. P. (1993). The proper orthogonal decomposition in the analysis of turbulent flows. *Annual review of fluid mechanics* 25.1, 539-575.

Bui-Thanh, T. M. (2003). Proper orthogonal decomposition extensions for parametric applications in compressible aerodynamics. In 2. A. conference (Ed.), *21st AIAA applied aerodynamics conference*.

Culière, P. e. (2003). Finite point set method: a mesh free approach to model airbag inflation. *SAE Technical Paper.*

Halko, N. P.-G. (2011). Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review* 53.2, 217-288.

Higdon, D. e. (2008). Computer model calibration using high-dimensional output. *Journal of the American Statistical Association*, 570-583.

Horta, L. G. (2008). *On the Application of a Response Surface Technique to Analyze Roll-Over Stability of Capsules With Airbags Using LS-Dyna.* NASA.

Jumper, J. e. (2021). Highly accurate protein structure prediction with AlphaFold. *Nature* 596.7873, 583-589.

Kennedy, M. C. (2001). Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 425-464.

Lin, C.-H. Y.-P. (2014). Evaluation of ls-dyna corpuscular particle method for side impact airbag deployment applications. *13th International LS-DYNA Users Conference*. Dearborn, US.

Nguyen, M.-N. a.-G. (2022). An efficient PODI method for real-time simulation of indenter contact problems using RBF interpolation and contact domain decomposition. *Computer Methods in Applied Mechanics and Engineering* 388.

Peherstorfer, B. a. (2016). Data-driven operator inference for nonintrusive projection-based model reduction. *Computer Methods in Applied Mechanics and Engineering* 306, 196-215.

Raissi, M. P. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics* 378, 686-707.

Siemens. (2022). *Simcenter Madymo*. Retrieved from https://www.plm.automation.siemens.com/global/en/products/simulation-test/occupant-safety-simulation.html

Tay, Y. Y. (2014). A response surface methodology in predicting injuries to out-of-position occupants from frontal airbags. *ASME International Mechanical Engineering Congress and Exposition*. American Society of Mechanical Engineers.

Williams, C. K. (2006). *Gaussian processes for machine learning*. Cambridge, MA, USA: MIT Press.

APPENDIX A

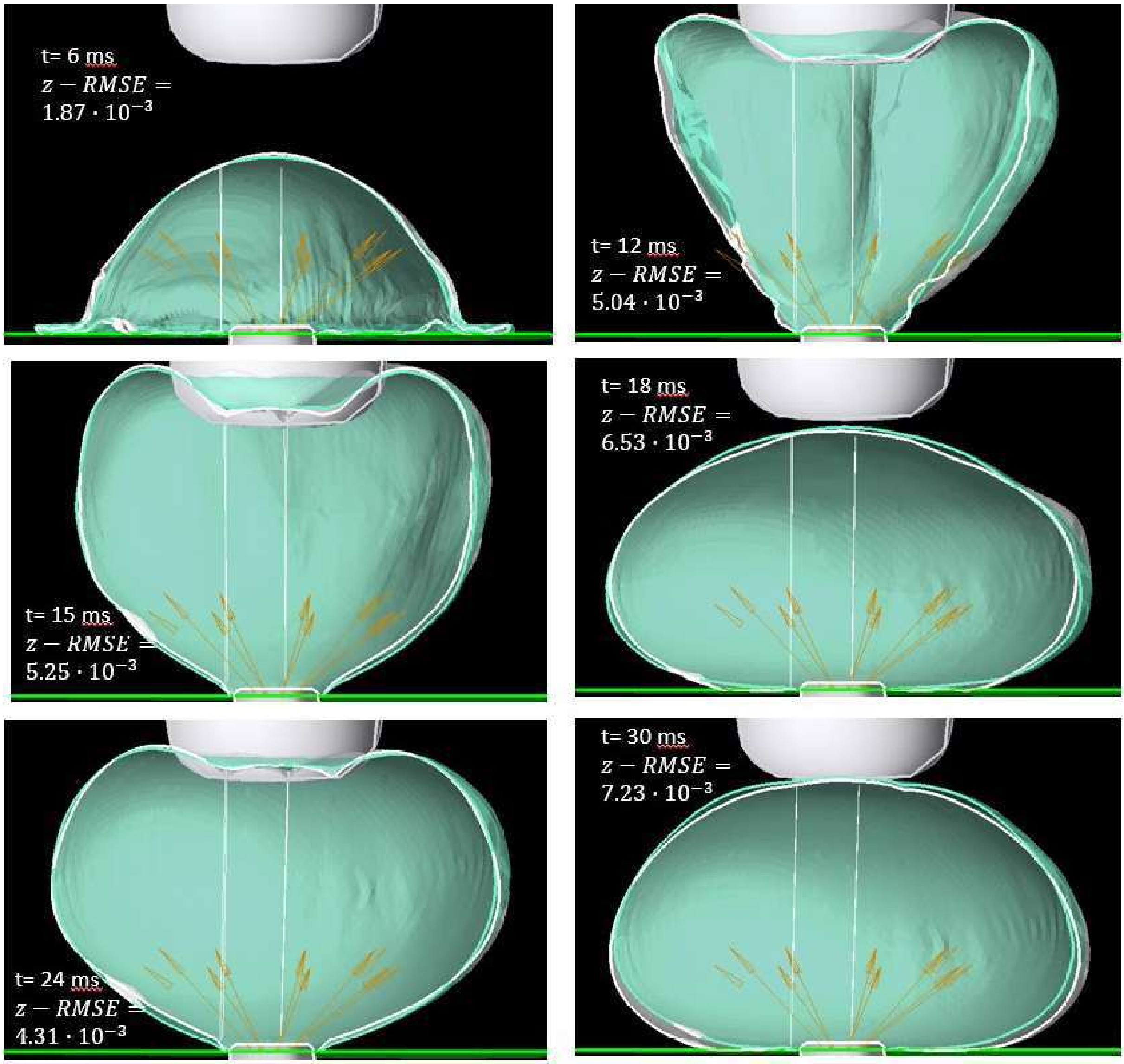

Figure 11. Airbag inflation cross-section of the Ground truth/Simcenter Madymo (white) and ML-accelerated model (green) at various times against a locked obstacle with a force-based contact characteristic for an unseen/test airbag-impactor system.

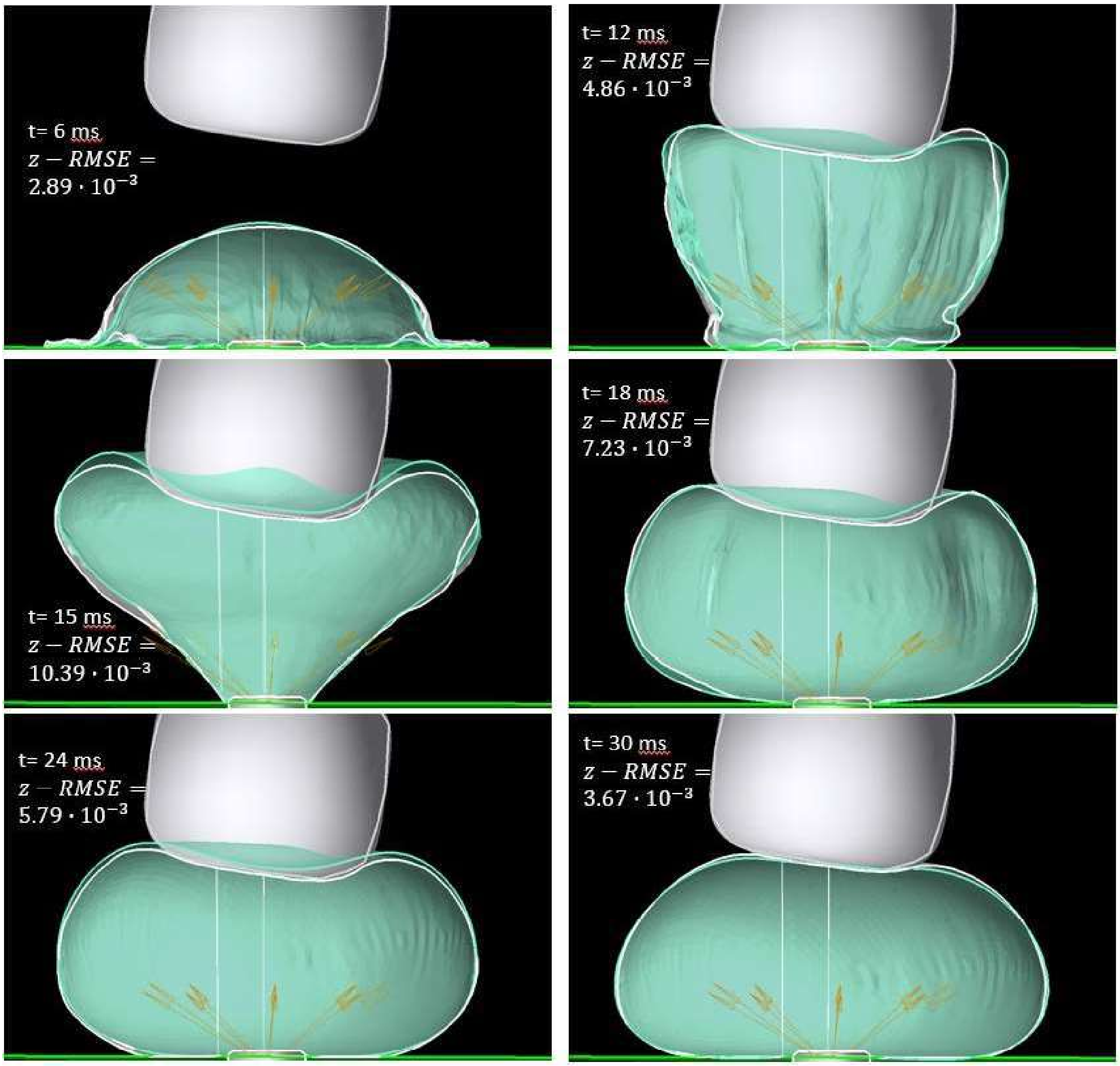

Figure 12. Airbag inflation cross-section of the Ground truth/Simcenter Madymo (white) and ML-accelerated model (green) at various times against a free rigid obstacle for an unseen/test airbag-impactor system.