Using Artificial Intelligence to Aid Vehicle Lightweighting in Crashworthiness with Aluminum

Christopher P. Kohar¹, Daniel S. Connolly¹, Timofei Liusko², and Kaan Inal¹, ²

¹Department of Mechanical and Mechatronics Engineering, University of Waterloo, Waterloo, Ontario, Canada, N2L 3G1
²Department of Systems Design Engineering, University of Waterloo, Ontario, Canada, N2L 3G1

Abstract. Significant efforts have been made in the automotive industry to reduce vehicle weight in order to improve vehicle fuel economy and reduce greenhouse gas emissions. New innovations in structural lightweight alloys and manufacturing techniques have allowed automakers to replace conventional steel with lighter aluminum structures. However, automakers have an enormous number of material and gauge thickness combinations to consider in the development process of the next generation production vehicle. Furthermore, the design combination of these materials and structures must not compromise the integrity of the vehicle during a vehicle collision. With the proliferation of inexpensive computational resources, automakers can now explore the effect of material selection on the crashworthiness of next-generation vehicles using computer simulations. While information from these simulations can be manually extracted, the vast amount of data lends itself to artificial intelligence (AI) techniques that can extract knowledge faster and provide more useful interpretations that can be convenient for designers and engineers. This work presents a framework for using artificial intelligence to aid the vehicle design cycle in crashworthiness using aluminum. Virtual experiments of a frontal crash condition of a pick-up truck are performed using finite element analysis to generate the data for this method. Different commercially available aluminum alloys and gauge thicknesses are varied in the virtual experiments. An advanced type of recurrent neural network is used to predict the time-series response of the occupant crash-pulse response, which is a key crashworthiness metric that is used for evaluating safety. This work highlights how automotive designs and engineers can leverage this framework to accelerate the development cycle of the next-generation lightweight vehicle.

1 Introduction

Over the last decade, significant attention has been placed on reducing greenhouse gas (GHG) emissions globally to combat the effects of climate change. The automotive and transportation industry is one of the major economic sectors that contributes substantially to GHG emissions. Many government mandates have been aimed at reducing GHG emissions. Structural lightweighting is a strategy that automakers employ to reduce energy consumption, which directly improves fuel economy, lowers operation costs, and overall reduces GHG emissions [1]. Automakers have pushed the manufacturing sector to develop new technologies in advanced manufacturing and lightweight materials engineering to be able to realize innovative designs for mass savings in production vehicles [2]. As a result, automakers now have several different material combinations and processes at their disposal to design and manufacture the lightweight vehicles of tomorrow.

However, automotive engineers must consider passenger safety in conjunction with vehicle lightweighting [3]. In the event of a vehicle collision, a crash pulse wave is generated and propagated throughout the vehicle. The shape and intensity of the crash pulse wave transmitted to the passenger directly correlates to the likeliness of a severe injury [4]. The transmission of the crash pulse wave can be mitigated through the systematic interaction of all vehicle components to either dissipate the energy through deformation of the structure or by strategically diverting the energy away from the passenger. This means that the development of new lightweight components cannot be generated in isolation. Although a designer can develop a new individual component from a new manufacturing process or material [5], this component may harm how the crash pulse wave propagates throughout the vehicle when integrated into the structure.

Therefore, automakers need to investigate how these innovative lightweight designs will affect the complete vehicle response during a collision. Although
prototypes of these new designs can be fabricated and tested, the time and cost associated with the trial-and-error process of development are no longer feasible for automakers in a rapidly evolving and competitive global economy. Automotive engineers now rely on numerical tools, such as finite element analysis, to perform virtual experiments of vehicles that are subjected to various crash scenarios before the prototyping and production stage.

With the proliferation of inexpensive computational resources, computer-aided engineering (CAE) software can now explore the effect of the various manufacturing processes and lightweight materials on the crash pulse wave of next-generation vehicles. While CAE engineers can extract knowledge and intuition from these studies, the vast amount of data that is generated from this process lends itself to artificial intelligence (AI) techniques that can extract knowledge and information orders of magnitude faster than the conventional manual methods. This means that AI techniques could relate the desired material properties and manufacturing process of a component to the crash pulse wave transmitted to a passenger. Once an AI tool has been trained on this data, an engineer can exploit the AI tool to predict and explore new combinations of materials and processing conditions that simultaneously reduces weight while improving passenger safety. Recent advancements within the field of AI can enable scientists and engineers to capture the time-dependent behaviour of a highly nonlinear phenomenon, such as vehicle crash, to accelerate the development of next-generation vehicles.

This work presents an AI framework that can be used to enhance the development cycle in vehicle lightweighting with an application to aluminum. This framework relies on virtual designs of experiments for a frontal crash test of a pick-up truck using finite element analysis to generate the training data for the AI system. Different commercially available aluminum alloys and gauge thicknesses are varied in the virtual experiments. Artificial neural networks are used to predict the relationship between vehicle mass and thickness. An advanced type of recurrent neural network, known as a long-short term memory neural network (LSTM-NN), is used to predict the crash-pulse response of an occupant during the impact. Comparisons of the AI framework to the results obtained from the finite element analysis are presented.

2 Proposed Framework and Architecture

The proposed framework follows the method of virtual design of experiments through the use of finite element (FE) simulations to improve vehicle safety while reducing vehicle mass. The procedure of the framework is summarized as follows:

1. **Problem definition**: Generate a finite element model of the problem of interest. Define the input parameters/variables and the output response.
2. **Sampling of problem domain**: Select combinations of input parameters that are statistically representative of the entire domain of interest and simulate their response.
3. **Define AI system**: Create an AI architecture that maps the input parameters to the desired output response.
4. **Training of AI system**: Select data from the sample set to calibrate the AI system. This step is often conducted in conjunction with Step 3 to select an architecture that accurately captures the system response iteratively.
5. **Prediction of AI system**: Use remaining data for prediction with the AI system. This serves as a blind test and evaluates the ability of the AI system to generalize the problem.

The problem of interest is the lightweighting of a pick-up truck using aluminum. The material thickness and mechanical properties, through the stress-strain curve, are varied throughout the vehicle simulations. This study is limited to a single frontal impact crash scenario; however, the proposed framework can be extended to capture an arbitrary number of load cases. The vehicle mass and the crash pulse are the desired output response that is to be determined by the AI system.

2.1 Representation of Mass

The mass of a vehicle can be computed directly from a computer-aided design (CAD) model and related to changes in geometric parameters (i.e., material thickness, connectors, assemblies, etc.). However, some of these geometric relationships can be related in a nonlinear manner. If the CAD model is not available, the mass of a vehicle can still be computed through the constructed finite element model. However, a mathematical function that relates the mass to the geometric properties needs to be identified. This work employs a feed-forward artificial neural network to map the geometric parameters to the mass of the vehicle.

2.1.1 Feed-forward artificial neural networks

Feed-forward artificial neural networks (ANN) are a class of AI that is commonly used for determining input-output relationships, which is designed to mimic the structuring of the human brain [6]. The values of each input parameter, \( x_i \), are scaled by a corresponding weight, \( w_i \), applied with an offset, \( b \), and summed into a node called a neuron. Fig. 1 presents a schematic of a multi-weighted-input neuron.
Information through the weights [7]. The output of the neuron, $y$, is defined by an activation function, $\mathcal{F}$, such that

$$y = \mathcal{F}(w \cdot x_i + b)$$  \hspace{1cm} (1)

It should be noted that the index notation implies a summation operation. Typically, these activation functions are selected such that the output is bounded between 0 and 1 to simulate the “firing” response of a neuron within the brain. These include the hyperbolic tangent, $\sigma_t$, or sigmoidal, $\sigma_g$, functions.

A series of neurons can be connected to the same inputs with unique weights to form a layer with multiple outputs; these outputs can be connected to another layer of neurons with accompanying weights to form a multi-layer ANN. Fig. 2 presents a schematic of a multi-layered ANN where $y_i$, is the output vector of interest. This architecture is known as “feed-forward” because the scaled values of the inputs propagate forward to the output. This scheme provides a massively parallel architecture that lends itself to efficient vectorized computational operations.

Each weight, $w_{ij}$, and bias, $b_i$, in the network is unique and is determined through a learning procedure called a learning algorithm. The objective of the learning algorithm is to determine the values of weights that minimize an error function, $f(\epsilon_{tr})$, between the actual measurement of the desired output, $\hat{y}_i$, and the predicted response from the ANN, such that

$$\min(f(\epsilon_{tr})) = \min(f(\hat{y}_i - y_i))$$  \hspace{1cm} (2)

Common learning algorithms for parameter identification include the back propagation [8], Levenberg-Marquardt [9, 10], and the Adam [11] methods.

---

**2.2 Representation of Crash Pulse**

The crash pulse response that is experienced by a vehicle occupant is the result of highly nonlinear interactions of the various components and structures during deformation. This deformation is governed by the mechanics and physics of solids, which can be determined through a series of partial differential equations. Fundamentally, this nonlinear response can then be represented by a transfer function, $\mathbf{\Sigma}$, that relates the time-series output response, $y_t$, (i.e., the crash pulse response) to a time-series of inputs, $x_t$, and their corresponding history. This is represented schematically in Fig. 3. In the application of this work, the transfer function is a single mathematical model that describes the deformation behaviour of a full vehicle in a crashworthiness scenario to predict the crash pulse response. Although the time-series response can be represented using a series of ANNs for each time increment [12], this would require a substantial number of ANNs and the necessary parameters to be identified. Furthermore, this approach would not capture the time-dependence of the problem that may produce a solution with less generalizability. Fortunately, there exist AI techniques that can capture time dependence. Recurrent neural networks (RNN) are an architecture that can predict sequences of data, such as time series. Unlike feed-forward ANNs, RNNs use information from the previous evaluation in the sequence to predict the next value. As such, this work employs a particular architecture of RNN called long-short term memory neural networks.

**2.2.1 Long-short term memory neural networks**

Long-short term memory (LSTM) is a specific architecture of an RNN that has been effective in capturing long term temporal dependencies [13]. Fig. 4 presents the schematic of the LSTM cell. The cell receives the input parameter vector at the current time, $x_t \in \mathbb{R}^n$, and outputs the hidden vector, $h_t \in \mathbb{R}^h$, and the cell vector, $c_t \in \mathbb{R}^h$, where $n$ is the number of inputs at each timestep and $h$ is the number of hidden cells. It should be noted that the architecture of the LSTM cell requires the size of hidden and cell vectors to be the same. Furthermore, the cell receives the previous values of the hidden and cell vector, $h_{t-1}$ and $c_{t-1}$, respectively. The fundamental concept in an LSTM architecture is the series of memory cells that regulate the flow of historical information (i.e., hidden and cell values) that enters and exits the cell.
These include an input gate, $i$, forget gate, $f$, cell candidate gate, $g$, and an output gate, $o$, that are governed by

$$
\begin{align*}
    f &= \sigma_g(R_i h_{t-1} + W_f x_t + b_f) \\
    g &= \sigma_g(R_g h_{t-1} + W_g x_t + b_g) \\
    i &= \sigma_c(R_i h_{t-1} + W_i x_t + b_i) \\
    o &= \sigma_g(R_o h_{t-1} + W_o x_t + b_o)
\end{align*}
$$

(3)

where $R = \begin{bmatrix} R_f, R_g, R_i, R_o \end{bmatrix}$, $W = \begin{bmatrix} W_f, W_g, W_i, W_o \end{bmatrix}$ and $b = \begin{bmatrix} b_f, b_g, b_i, b_o \end{bmatrix}$ are weights and biases to be identified. The output vector size of all gate operators must be of size, $h$, such that $f, g, i, o \in \mathbb{R}^h$ and the size of the weights and biases vectors are $W_j \in \mathbb{R}^{h \times h}$, $R_j \in \mathbb{R}^{h \times h}$, $b_j \in \mathbb{R}^h$. It should also be mentioned that the weights and biases do not evolve with time. The cell and hidden vector are updated according to

$$
\begin{align*}
    c_t &= f \odot c_{t-1} + g \odot i \\
    h_t &= o \odot \sigma_c(c_t)
\end{align*}
$$

(4)

where $\odot$ is the Hadamard product (element wise) operator. Finally, the desired output response of the time-series is obtained by decoding the hidden vector through the use of a feed-forward ANN. Fig. 5 presents a schematic of the LSTM-NN. Similarly, the weights in the LSTM cell and the weights in the feed-forward ANN do not evolve with time.

![Fig. 5. Schematic of an LSTM-NN.](image)

### 3 Problem Formulation

#### 3.1 Finite Element Model

The full vehicle finite element (FE) model used in this study is a pick-up truck that was developed by George Washington University National Crash Analysis Center as part of several United States Department of Transportation (US-DOT) – National Highway Traffic Safety Administration (NHTSA) programs. Fig. 6 presents a schematic of the FE model and the setup of the pick-up truck. This FE model is constructed to replicate the testing conditions outlined in the United States New Car Assessment Program (US-NCAP) [14] for frontal collision into a concrete barrier. Each crash simulation was analyzed using a non-linear explicit dynamic formulation of the commercial FE software LS-DYNA – R9.3.0 [15] with double precision utilizing $16 \times 2.7$GHz Intel Xeon E5-2680 processors and parallel processing techniques. On average, each FE simulation required a wall clock of 2 hours 39 minutes 20 seconds (9560 seconds).

Fig. 6 also presents a typical deformed response of the FE model. The initial velocity of the vehicle in each simulation was 15.6 m/s [56.3 km/h]. The total simulation time was 150 milliseconds to allow for sufficient settling time after impact. The crash pulse was evaluated at the nodes of two solid elements (highlighted in Fig. 6b) that represent the location of an accelerometer for the occupant in the driver and passenger position in the vehicle. Each crash pulse measurement was evaluated in three-axes (x, y, z) that was sampled at 200 $\mu$s (5000Hz) and filtered using the SAE CFC 60 standard [16].

In total, the FE model consists of 875 groups of parts. In reality, an automaker would need to specify the material of all the components in the vehicle. For the purpose of this study, the components of the vehicle were grouped into four categories based on their structural functionality within a frontal crash scenario.

![Fig. 6. a) Isometric View, b) View of Accelerometers, c) Representative Deformed Response for Pick-up Truck Model.](image)
where is governed by an associative flow rule defined as Poisson ratio, which can be described by the elastic modulus, elastic strain rate through the Cauchy stress tensor, such that elastic strain rate, the total deformation rate, that is governed by an assumed to follow an elasto-viscoplastic formulation. If yielding occurs, the plastic strain rate, defined by

\[
\dot{\varepsilon}^p = \frac{\partial \Phi}{\partial \sigma} \quad \dot{\varepsilon}^p \geq 0
\]

The plastic flow stress model is assumed to follow a Voce-type plasticity law [18] with linear hardening contribution defined by

\[
\sigma_y = \sigma_0 [U_0 - (U_0 - 1) \exp(-D \dot{\varepsilon}^p)]
\]

where \(\sigma_0\) is the material quasi-static yield stress, \(U_0 \geq 1\) is the ratio of the saturation stress to quasi-static yield stress, and \(D\) is the Voce hardening saturation coefficient. Eqs. (5) - (11) form a series of simultaneous equations that can be incrementally solved using Newton-Raphson to identify \(\Delta \dot{\varepsilon}^p\).

### 3.2.2 Application to Commercially Available Aluminum Sheet

As mentioned earlier, automakers can choose from a variety of sheet metals, such as aluminum, magnesium, and steel, to construct a vehicle. Various grades exist within each family of materials based on their alloying elements and material processing conditions. In particular, Fig. 8 presents a representation of the various automotive grades of aluminum alloys that are commercially available to vehicle designers. Each of these materials has fundamental differences in alloying elements and manufacturing processing that generates unique deformation behaviours that ultimately affect the mechanical response in the form of the stress-strain curve. These differences can be beneficial or detrimental to the response of a vehicle in a crashworthiness application. In addition, the manufacturing complexity and cost associated with producing these materials and integrating them into a vehicle is also a factor that automakers must carefully consider in their development.
3.3 Sampling of Component Grouping Thickness and Material Properties

Various combinations of part thicknesses and material parameters can be used to create virtual experiments of the frontal crash scenario, which can be used for data generation and machine learning. The original FE model assumed the majority of these components were fabricated using various grades of automotive sheet steel. Now, the direct substitution of the steel components with automotive grade aluminum will accomplish the goal of mass reduction because of the difference in density ($\rho_{\text{steel}} \approx 7800 \text{ kg/m}^3$ vs. $\rho_{\text{al}} \approx 2700 \text{ kg/m}^3$). However, the general strength differential between automotive grade aluminum and steel can have an adverse effect on the overall crashworthiness performance of the vehicle. Furthermore, the difference in elastic properties will also affect other vehicle performance metrics, such as vehicle stiffness and noise, vibration, and harshness (NVH). Nevertheless, the gauge thickness of the aluminum components can be increased to compensate for this reduction in strength. This enhancement in the geometric cross section will have a non-linear effect that can offset the added mass. As such, the sheet thickness of a part, $i$, within a component category, $p$, are scaled by the ratio of the density

$$t_{i}^{(n,p)} = (\rho_{\text{steel}}/\rho_{\text{al}})t_{i}^{(p)}$$

(12)

where $t_{i}$ is the initial thickness of the part defined in the FE model. Similar to material selection, an automaker would also need to specify the material thickness of each component. In the data generation, the thickness of each component category is sampled according to

$$t_{i}^{(n,p)} = \eta(n,p)t_{i}^{(p)} = \eta(n)(\rho_{\text{steel}}/\rho_{\text{al}})t_{i}^{(p)}$$

(13)

where $n$ is a sample in the data generation and $\eta(n,p)$ scales the thickness of a component category, $p$. This scaling parameter is constrained, such that

$$0.80 \leq \eta(n,p) \leq 1.25$$

(14)

Next, the presented flow stress model has five unique parameters to describe the evolution of the stress-strain response of the material. Smerd et al. [19] have reported that automotive aluminum alloys generally exhibit low strain-rate sensitivity at room temperatures. Yet, aluminum alloys start to exhibit positive rate sensitivity at moderate and high strain-rates due to a change in temperature through adiabatic heating [3, 20]. As such, the Cowper-Symonds coefficients of $C = 9.39 \times 10^{13} \text{s}^{-1}$ and $P = 10.55$ are used for all materials in this study [21]. This reduces the number of free material parameters to three. Now, some of these parameter combinations will produce materials that are non-physical (i.e., a material with a low yield and no hardening or a high yield with too high hardening). To constrain the domain for this study, a sample combination of the material parameters must produce a stress-strain curve that is monotonically increasing by imposing

$$U_{i}^{(n,p)} \geq 1, \quad D^{(n,p)} > 0$$

(15)

The combination of material parameters must generate a stress-strain curve that lies within the upper and lower limits of the stress-strain responses presented in Fig. 8. The upper limits are formed by the AA6111-T6 response, while the lower limits are formed by AA5754 and AA6060-T6. Therefore, any sample must satisfy

$$\min \left( \sigma_{\text{AA6060-T6}}^{A^{p}} \right) \leq \sigma_{\text{AA5754}}(\sigma_{0}^{(n,p)}, U_{i}^{(n,p)}, D_{i}^{(n,p)}) \leq \max \left( \sigma_{\text{AA6111-T6}}^{A^{p}} \right) \forall x \in [0.4]$$

(16)

This creates a convex hull of material parameters that satisfy these constraints. A multi-dimensional Latin hypercube algorithm is used to sample the convex hull of material parameters to generate 400 unique combinations of material coefficients for the 20 distinct parameters ($3 \times \text{material parameters} + 1 \times \text{thickness coefficient}) \times 5$ component categories).

4 Results and Discussion

4.1 Network Architecture and Training

Table 2 and Table 3 presents the network architectures. All models use a training/validation split of 80/20, where 320 samples are randomly selected for determining the weights in each model. All models used the Adam optimizer with an initial learning rate of 0.001 that converged using a mean squared error (MSE) function. The tanh activation function was used in the ANNs of each model. All models were trained using the Keras/Tensorflow libraries for Python using 2 x 1.4GHz nVidia Titan Xp (12GB GDDR5) GPUs with an nVidia GTX SLI HB Bridge.

Table 2. Architecture of ANN Model for Mass.

| Dropout | 0.00 |
|---------|------|
| Batch Size | 32 |
| # Layers in ANN | 2 |
| # Neurons in ANN | [40, 1] |
| # of Weights | 281 |
| # of Epochs | 5000 |
| Training Time [sec] | 44.8 |

Table 3. Architecture of LSTM-NN Model for Accelerometers.

| # Hidden Units | 1536 |
| Dropout | 0.25 |
| Batch Size | 16 |
| # Layers in ANN | 2 |
| # Neurons in ANN | [192, 6] |
| # of Weights | $9.8 \times 10^4$ |
| # of Epochs | 1000 |
| Training Time [sec] | $3.8 \times 10^4$ |
Fig. 9 presents the convergence plots for the ANN and LSTM-NN models that were used for representing the vehicle mass and accelerometers, respectively. The training of the ANN model for mass converges quickly to low MSE magnitude ($>10^{-5}$) with a relatively low number of neurons and depth. The validation error is also low with having a similar order of magnitude as the training error. This means that the model for mass should have excellent predictive capabilities. This is a result of the degree of nonlinearity in the mass being minor while varying only the section thickness [22]. However, this response can change significantly if the topology of the structure is varied. The training error of the LSTM-NN model for the accelerometers converged to an MSE magnitude of ~0.01. Although the MSE of the LSTM-NN is higher than the ANN model, the error is still rather low. It should be reiterated that the degree of nonlinearity is much higher in predicting the acceleration response. Nevertheless, the LSTM-NN model still shows good predictive capabilities in capturing the time-series response in acceleration for all axes and locations of interest.

To demonstrate this capability, a representative sample was taken from the validation set for prediction of the driver and passenger accelerations via the AI framework. Fig. 10 presents the acceleration response predicted by the AI framework for both the driver and passenger in all three axes. This sample has an MSE value of 0.0139. This figure shows the capability of the AI framework to simultaneously predict the response of the two nodes with good fidelity. The total time to predict the acceleration response of the driver and passenger using the AI framework was 6.77 milliseconds; this is significantly faster ($1.41 \times 10^6$) than the time required to generate the result via the FE method. Now, the total number of nodes in the FE model is $7.34 \times 10^5$, which accounts for significant overhead for the FE solution. This methodology can be scaled to predict the acceleration of all nodes within the FE model. The ability of this AI framework to well predict the nodal accelerations offers an opportunity to predict the resulting displacement of all nodes through a simple integration of time. In turn, this suggests the prospect of predicting deformation, such as strain (i.e., the spatial gradient of nodal displacements), of a complete structure. This approach could allow for the real-time prediction of the deformation of structures that can be useful for intrusion measurements and occupant safety. However, it should be noted that this requires the use of the GPU to achieve these computational gains. Nevertheless, this significant reduction in computational time can aid scientists and engineers in developing lightweight solutions when coupled with an intelligent optimization framework [3].

![Fig. 9. Convergence plots for a) ANN model for Mass and b) LSTM-NN for Accelerometers.](image1)

![Fig. 10. Representative Sample of a) Driver and b) Passenger Acceleration Response in the i) X-direction, ii) Y-direction, iii) Z-direction from LS-DYNA and AI predictions.](image2)
5 Conclusion

This work presented an AI framework that can accelerate the development cycle in vehicle lightweighting. This framework was applied to the application of lightweighting a pick-up truck with the use of aluminum. This approach relied on a virtual design of experiments for a frontal crash test of the pick-up truck. The data for training the AI system was collected from finite element simulations of the vehicle under this loading condition. Different material properties that lie within the range of commercially available aluminum alloys along with various gauges of thickness were used for generating this data. An ANN was used to predict the mass of the vehicle, while an LSTM-NN was used to predict the crash pulse response that would be subjected to a driver and occupant. The AI approach was able to predict the occupant acceleration response up to $1.41 \times 10^6$ times faster than the conventional FE solution with minimal error (MSE < 0.01). Coupling this approach into an optimization framework would aid in the accelerated development of lightweight solutions. This framework also shows promise in the capability of predicting the entire deformation response of a vehicle.

This research was enabled in part by the support provided by Calcul Québec (www.calculquebec.ca) and Compute Canada (www.computecanada.ca). The authors also gratefully acknowledge Dr. HuiZhong Lu at Compute Canada - University of Sherbrooke for his technical support.

References

1. H. Kim, G. Keoleian, S. Skerlos, J Ind Ecol 15(1), 64080 (2011).
2. A. Sachdev, R. Mishra, A. Mahato, A. Alpas, ICAA13 Pittsburgh, Springer, Cham, 609-622, (2012).
3. C. Kohar, A. Zhumagulov, A. Brahme, M. Worswick, R. Mishra, K. Inal, Int J Impact Eng 95, 17-34 (2016).
4. A. Kullgren, M. Krafft, Å. Nygren, C. Tingvall, Accident Anal Prev 32(2), 197-205 (2000).
5. C. Kohar, M. Mohammadi, R. Mishra, K. Inal, Thin Wall Struct 93, 64-87 (2015).
6. K. Hornik, M. Stinchombe, W. Halbert, Neural Networks 3, 551-560 (1990).
7. I. Aleksander, M. Helen, An introduction to neural computing: 2nd edition, (International Thomson Computer Press, London, 1995).
8. D. Rumelhart, G. Hinton, R. Williams, Nature 323, 533-536 (1986).
9. K. Levenberg, Q Appl Math 2, 164-168 (1944).
10. D. Marquardt, J Soc Ind Appl Math 11(2), 431-441 (1963).
11. D. Kingma, J. Ba, arXiv preprint arXiv:1412.6980 (2014).
12. L. Lanzi, C. Bisagni, S. Ricci, Comput Struct 82(1), 93-108 (2004).
13. K. Greff, R. Srivastava, J. Koutník, B. Steunebrink, J. Schmidhuber, IEEE T Neur Net Lear 28(10), 2222-2232 (2016).
14. L. Hershman, SAE Technical Paper, 2001-06-0245 (2001).
15. J. Hallquist, Livermore Software Technology Corporation, Livermore, California (2006).
16. Society of Automotive Engineers, SAE Technical Report, J211-1 (2007).
17. G. Cowper, P. Symonds, Providence, Rhode Island (1957).
18. E. Voce, Metallurgia 51(307), 219-226 (1955).
19. R. Smea, S. Winkler, C. Salisbury, M. Worswick, D. Lloyd, M. Finn, Int J Impact Eng 32, 541-560 (2005).
20. C. Kohar, A. Brahme, J. Imbert, R. Mishra, K. Inal, Int J Solid Struct 128, 174-198 (2017).
21. S. Shu, N. Jones, Int J Crashworthiness 9(2), 195-217 (2004).
22. G. Sun, G. Li, S. Zhou, H. Li, S. Hou, Q. Li, Struct Multidiscip O 44(1), 99-110 (2011).