ARCHIVES OF MECHANICS
Arch. Mech. **76** (1-2), 61–92, 2024, DOI: 10.24423/aom.4454

Development of a machine learning-based design optimization method for crashworthiness analysis

A. BORSE, R. GULAKALA, M. STOFFEL

Institute of General Mechanics, RWTH Aachen University,
Eilfschornsteinstr. 18, 52062 Aachen, Germany,
emails: {borse, gulakala, stoffel}@iam.rwth-aachen.de

This article investigates design optimisation in the automotive field using machine learning (ML). A thin-walled crash box under axial impact is studied and the design parameters are optimised for front-impact crash tests. This study is based on geometrically and physically nonlinear shell theory, finite element analysis (FEA), dynamic buckling analysis and design optimisation using ML. An artificial neural network framework consisting of various ML methods is developed. A generative adversarial network is established for data generation and reinforcement learning is implemented to automate exploration of the design parameter. This ML framework is proven to determine optimal parameters under predefined crashworthiness constraints.

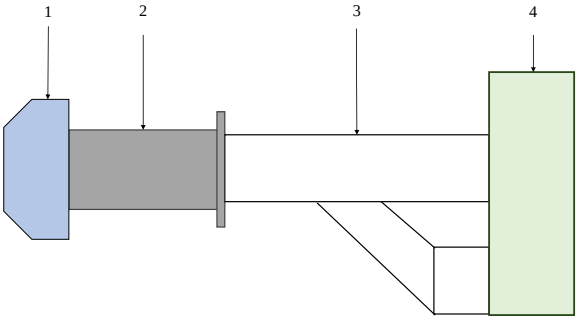
Key words: crashworthiness design, generative adversarial network, reinforcement learning, nonlinear shell theory.

1. Introduction

Design for crashworthiness is a key aspect of the vehicle design process. With 2.7 million recorded total road accidents in Germany in 2019 and 384,230 total injured [1, 2]. Therefore, stringent safety regulations, like United Nations vehicle regulations (UN-Rxx), the European New Car Assessment Program (EuroNCAP) in Europe, and Federal Motor Vehicle Safety Standards (FMVSS) in the USA, are established to evaluate the safety and reliability of the car and are required for legal sales in various parts of the world. Therefore, in the design process of the complete vehicle, crash simulations are performed to evaluate the performance of parts and/or assemblies and the level of safety provided by the vehicle to its occupants in a virtual crash test [3].

Due to the increasing complexity of automobiles and the intrinsic nonlinear dynamic nature of crashes, these simulations are resource-intensive and require specialized systems. Numerous simulations need to be performed to optimise the structural performance and crashworthiness of the part [4]. Due to the evolution

of strict safety regulations and focus on the lightweight design to control emission [5, 6], automobile manufacturers are subjected to strict restrictions. Besides, the manufacturability of complex design of parts and advancement in lightweight composite materials result in numerous parameters that could be optimised for each component. However, optimizing the complete vehicle for crashworthiness and satisfying market demands is resource-intensive and time-consuming. Computationally expensive simulations are needed to optimise every part with multiple parameters. This increases the preliminary design process, time and cost. Although various surrogate models (also known as meta-models or approximate models) exist, for a part optimization [4, 7–9]. But in the final stages, complete simulations are performed.



1) Front bumper member; 2) crash box; 3) front rails; 4) chassis

Fig. 1. Front-end structure of vehicle [10].

In automobiles, there are multiple energy-absorbing structures in the event of a crash. Special attention is given to such passive safety components in the crashworthiness design process [11]. Because frontal crashes account for one-quarter of all accidents [12], the optimisation of the crash box design is examined in this article. The crash box is an energy absorption device installed at the front end of all vehicles. It is located between the front bumper and chassis as shown in Fig. 1. It is typically made up of thin-walled structures from metal or composite materials. It helps in avoiding the direct transfer of energy to occupants and intrusion into the passenger area. It also manages safe deceleration and lowers the danger of injury [13].

In-depth research is done on designs and materials to optimise the crash box for crashworthiness. Various designs of the crash box including cylindrical [14–16], square/rectangular [17–19], tapered [20] and complex patterns [21–23] are proposed and studied in detail to understand their deformation pattern and energy absorption capabilities. Various configurations of these crash boxes under different design patterns like corrugations [15, 24], foam filling [18, 25], multi-cell [26, 27], segmented [28] and different geometric imperfection [29–31] have also

been assessed. Aside from alternative designs, various materials are being studied to improve crashworthiness design. Because lightweight design is also important in the design process, numerous lightweight alloys and composite materials are tested in crash box design. To enhance crashworthiness performance, customized materials are even created [32, 33]. Most of the composite materials are from carbon fibre composites [34, 35] and commonly steel or aluminium alloys have been studied for crash box design [24, 31].

The crash box has to undergo large deformation before any other body part in the event of the crash. Therefore, special attention is given to studying the deformation mechanism. The deformation pattern of the crash box is crucial to assess the total energy absorbed and impact forces experienced by occupants. Therefore, deformation in crash boxes with different designs and materials is studied [36]. It is noted that under axial loading, deformation can be either buckling or axial collapse or a combination of both patterns [37, 38]. This has an immediate impact on the total energy absorbed. The deformation mechanism is investigated to comprehend these deformation patterns. It assists in understanding and optimising the crash box design for controlled deformation. It has been noted that this deformation pattern depends on the thickness, slenderness ratio, length of the crash box, material strain rate sensitivity, and stress wave propagation [17, 38]. Optimizing the design of the crash box demands solving a multi-parameter optimization problem.

Modelling of shell buckling requires the correct simulation of the deformation of shell parts of the crash box. Therefore in the present study, the described mechanical background is modelled systematically by a nonlinear structural theory and a finite element model to develop a solution procedure for this transient axial stability problem under impact loading. However, these FEM simulations are computationally expensive and therefore, automation with a machine-learning framework is proposed in this article.

Machine learning is becoming prevalent in all fields in the engineering and technology industries, such as surrogate modelling [39, 40], medical engineering [41, 42] in biomechanics [43, 44], and structural mechanics [45] in the shell theory [46]. Due to continuous developments, ML frameworks are used to improve efficiency or replace conventional methods.

ML-based nonlinear regression models are used to achieve approximate solutions for computationally expensive simulations. These ML techniques are shown to help solve nonlinear structural problems [47–49]. Data analysis using ML is also performed to understand the appropriate parameters and is useful in the engineering fields, where numerous parameters need to be optimised [50].

That said, ML methods, especially supervised learning models need a lot of data to train and predict with adequate accuracy. This is where the generative learning models help in data augmentation and generate synthetic samples

of data that are physically admissible. Since the generator is trained indirectly through the discriminator, it learns the underlying data structure rather than mapping inputs to outputs or fitting to the curve. Numerous studies have been published in the literature using generative adversarial networks (GANs) to create synthetic data in the field of FEA and mechanics. In [51], the hybrid finite element method (FEM) and the GAN-based method were used to classify faults in rotor-bearing systems. TopologyGAN was proposed in [52] to perform topology optimization of structures with isotropic solid material behaviour. Simple body deformations with a simply supported structure loaded at the centre are modelled in [53]. Therefore, GANs have proven to be a useful tool in generating realistic-looking synthetic data which is leveraged to bolster the dataset for training the regression network inside of a reinforcement learning environment.

Due to recent accomplishments in various fields, like robotics [54], data management systems [55], and video games [56], reinforcement learning (RL) has gained popularity. RL is being used in a variety of problems from function approximation, automation and control problems. The growing popularity of RL methods is because RL agents can learn and develop by themselves. Therefore, an RL-based optimisation framework is developed and RL agents are trained to optimise the structural parameters of the crash box. The parameter exploration is constrained by the crashworthiness parameters defined by the user. For increasing the efficiency of the framework, a trained regression model is an integral part of the RL environment and a GAN is utilised for the data-driven part of the ML framework. The complete framework and individual components are described in Section 3.2. For predefined crashworthiness criteria, this framework has been observed to predict the design parameters of the crash box.

2. Deformation mechanism

In the thin-walled crash boxes, the energy can either be absorbed by axial and/or bending deformation, and this mode of deformation is influenced by imperfections and elastic-plastic buckling modes along with material and geometrical parameters [37, 57]. The bending is uncontrolled large deformation and can be dangerous in crashworthiness applications, while axial collapse deformation is desirable and helps in absorbing maximum energy in a controlled manner [38]. Therefore, numerous studies have been performed to understand these deformation mechanisms in various thin-walled structures and the factors influencing this deformation mode.

Initially, a simple collapse mode in the form of concertina is observed for thin cylindrical tubes and a simplified expression for the mean axial load is formulated for plastic rigid material [14]. Later square tubes of different cross-sections under static and dynamic loading are also experimentally studied [17]. For square

tubes, one symmetrical and two asymmetrical collapse modes of deformation are observed. Based on these deformation modes, a simplified kinematic model, as an extension of two fundamental folding elements is proposed [58–60]. Then the mean crushing force and crushing distance are calculated for each collapse mode. The effect of material strain rate sensitivity on the previous formulations has revised the equations for folding elements in square and circular crash boxes [61].

In the experimental study performed on mild steel square and circular tubes, it is observed that the dynamic deformation of columns is influenced by the transient response due to stress wave propagation, inertia effect, and strain-rate effects [37]. The initial collapse pattern is governed by the interaction of stress waves, their propagation, reflection and superposition, the slenderness ratio (length/width or length/diameter) and the critical length. Depending on these parameters, the column will fail by global bending or axial collapse [38, 62]. This effect of the interaction between stress wave propagation and inertia forces on the deformation mechanism and deformation pattern is further studied for square and cylindrical shells [38, 63, 64]. In this study, a model with elastic-plastic springs is used to define the phenomena of buckling and post-buckling [63] and concluded that the final buckling shapes also depend on the inertia properties of the striker and geometry of shells.

It is also noted that the column that starts to deform in progressive collapse may bend in later stages and detailed knowledge of stress distributions and plastic mechanisms is required in understanding this transition [37]. It is observed that lateral inertia forces assist in progressive buckling and as the deformation increases, inertia forces decrease and global buckling could occur even after progressive collapse [38]. Initial buckling pattern also depends on the impactor's mass and velocity, and then stress-strain histories [64].

In dynamic problems, stability and buckling behaviour are modelled as an eigenvalue problem leading to the determination of critical buckling load [65]. Therefore, the relationship between natural frequency and buckling load for simple elastic structures was examined [66]. It is noted that a nonlinear relation is obtained for imperfection-sensitive systems with elastic-plastic material. As the crash box deforms, different natural frequencies might be observed due to deformation and changes in dynamic load.

The bifurcation point defines the post-buckling in perfect shells. But in reality, due to unavoidable imperfections, the imperfect shells buckle before the bifurcation point [67]. This phenomenon must be considered when studying dynamically impacted thin-walled structures modelled as thin shells. Therefore, numerical imperfection, based on Koiter's initial post-buckling theory, is formulated in the stability of shells for realistic results [68]. As the actual imperfections in the structure are unknown, the numerical imperfection is calculated as a value

obtained through a linear combination of the eigenvectors [69]. This numerical imperfection is introduced as the perturbance in the mesh to define imperfect structure [70, 71]. Furthermore, defining the imperfection shape from the appropriate linear combination of eigenvectors is still an open research field. Therefore in the initial study, the first eigenmode is selected as the critical mode to define the numerical imperfection value for realistic imperfect thin-walled square tubes like in other studies [72] and to avoid the use of arbitrary imperfections. The impact simulation is carried out on this imperfect crash box.

Thus, multiple parameters, like slenderness ratio, critical length, strain rate, material hardening parameters, and impact velocity, are responsible for the complete deformation behaviour of thin-walled structures. This deformation pattern is essential in energy absorption and other crashworthiness characteristics.

3. Approach

3.1. Theoretical model

The thin-walled crash box (CB) is modelled using a geometrically and physically nonlinear shell model to develop an underlying structural model [46, 73]. The effects of inertia forces, external body forces and surface tractions are considered using the principle of virtual work. Undeformed and deformed configurations of the shell space along with base vectors are illustrated in Fig. 2. The variables $\boldsymbol{\theta_i}$ denote a curvilinear coordinate system that deforms together with the structure. The variables with a bar on top, like '$\overline{X}$', represent variables in deformed configurations. The base vector systems $\boldsymbol{a_i}$ and $\boldsymbol{g_i}$ are introduced at the mid-surface and in the shell space, respectively. For the transformation be-

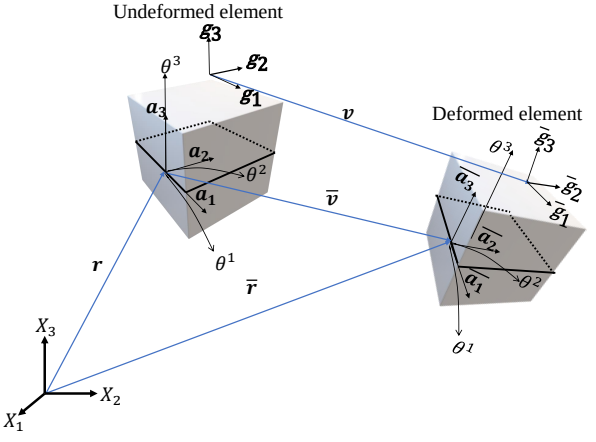


Fig. 2. Shell configuration – undeformed and deformed shell space.

tween these systems, we refer to [74]. Thus, the displacement vector in the shell space is represented by:

$$\boldsymbol{v} = v^{\alpha}\boldsymbol{a}_{\alpha} + v^{3}\boldsymbol{a}_{3}, \qquad \alpha = 1, 2. \tag{3.1}$$

The displacement vector at the mid-surface is given by

$$\overline{\boldsymbol{v}} = \overset{0}{v}{}^{\alpha}\boldsymbol{a}_{\alpha} + \overset{0}{v}{}^{3}\boldsymbol{a}_{3} \tag{3.2}$$

including displacement components as follows:

$$v_{\alpha} = \overset{0}{v}_{\alpha} + \theta^{3}\overset{1}{v}_{\alpha}, \tag{3.3}$$

$$v_{3} = \overset{0}{v}_{3}, \tag{3.4}$$

$$v_{3} = \overset{0}{v}_{3}. \tag{3.5}$$

Five degrees of freedom are considered, where $\overset{0}{v}_{\alpha}$, $\overset{0}{v}_{3}$ are denoting the displacements and $\overset{1}{v}_{\alpha}$ are standing for rotations. Here, the change in thickness during deformation is neglected. Therefore, the equations of the principle of virtual work can be written as follows:

$$\int_{V} \boldsymbol{S}^{ij}\delta\boldsymbol{E}_{ij}\,dV - \int_{V} \rho(\boldsymbol{B} - \boldsymbol{I})\delta\boldsymbol{v}\,dV - \int_{B_{\mathrm{s}}} \overset{\star}{\boldsymbol{S}}\delta\boldsymbol{v}\,dB_{\mathrm{s}} = 0. \tag{3.6}$$

Here, $\boldsymbol{S}^{ij}$ are components of the second Piola–Kirchhoff stress tensor, $\boldsymbol{E}^{ij}$ are the components of the Green–Lagrange strain tensor, $\boldsymbol{B}$ is the body force per unit mass, ρ is density, $\delta\boldsymbol{v}$ the virtual displacement vector, $\boldsymbol{I}$ is the inertia force vector and $\overset{\star}{\boldsymbol{S}}$ stands for surface tractions. The integration is carried out over the volume V and the shell boundary B_s.

From the above equations, for a shell mid-surface, using the first-order shear deformation theory (FOSD) with the von Karman hypothesis [73], the following equations are obtained:

$$\begin{aligned}
-\int_{M} \{&[(\overset{0}{L}{}^{\alpha\beta} - b^{\alpha}_{\lambda}\overset{1}{L}{}^{\alpha\beta})\,|_{\beta} - b^{\alpha}_{\beta}\overset{0}{L}{}^{\beta 3} - (\overset{0}{B}{}^{\alpha} - \overset{0}{I}{}^{\alpha} + \overset{0}{P}{}^{\alpha})]\,\delta\overset{0}{v}_{\alpha} \\
&+ [(\overset{0}{L}{}^{\beta 3} + \overset{0}{L}{}^{\alpha\beta}v_{3,\alpha})\,|_{\beta} - b_{\alpha\beta}(\overset{0}{L}{}^{\alpha 3} - b^{\alpha}_{\lambda}\overset{1}{L}{}^{\lambda\beta}) - (\overset{0}{B}{}^{3} - \overset{0}{I}{}^{3} + \overset{0}{P}{}^{3})]\,\delta\overset{0}{v}_{3} \\
&+ [(L^{1\alpha\beta} - b^{\alpha}_{\lambda}L^{2\alpha\beta})\,|_{\beta} - L^{0\alpha 3} - (\overset{1}{B}{}^{\alpha} - \overset{1}{I}{}^{\alpha} + \overset{1}{P}{}^{\alpha})]\,\delta\overset{1}{v}_{\alpha}\}\,dA \\
&+ \int_{C} \{[(\overset{0}{L}{}^{\alpha\beta} - b^{\alpha}_{\lambda}\overset{1}{L}{}^{\alpha\beta}) - \overset{\star 0}{L}{}^{\alpha\beta}]\,\delta\overset{0}{v}_{\alpha} + [(\overset{0}{L}{}^{\alpha\beta}v^{0}_{3,\alpha} + \overset{0}{L}{}^{\beta 3}) - \overset{\star 0}{L}{}^{3\beta}]\,\delta\overset{0}{v}_{3} \\
&+ [(\overset{1}{L}{}^{\alpha\beta} - b^{\alpha}_{\lambda}\overset{2}{L}{}^{\alpha\beta}) - \overset{\star 1}{L}{}^{\alpha\beta}]\,\overset{1}{v}_{\alpha}\}\,v_{\beta}\,ds = 0.
\end{aligned} \tag{3.7}$$

The forces, in-plane and transverse shear, and moments are expressed by $\overset{0}{L}{}^{\alpha\beta}$, $\overset{0}{L}{}^{\alpha 3}$ and $\overset{1}{L}{}^{\alpha\beta}$, respectively. Higher order terms, which are present due to geometrical nonlinearity, are denoted by $\overset{2}{L}{}^{\alpha\beta}$. Covariant and mixed components of the curvature tensor are written as b^{α}_{β} and $b^{\alpha\beta}$, respectively. The principle of virtual work obtained in a two-dimensional form is solved with integral over the shell mid-surface M and its boundary C. The outward unit vector is denoted by $\boldsymbol{v}$.

For the crash box loading, a specified boundary condition representing an impact is necessary. For this reason, external loading in Eq. (3.7) can be ignored. A solution of Eq. (3.7) in the form of wave propagation is assumed to lead to a non-trivial solution [65]. However, since the analytical approach is hardly possible, a numerical solution in two steps is chosen.

To begin with, the eigenmodes are determined numerically, being used as imperfection. In this way, a reproducible and systematic way of imperfection is introduced. The imperfection is calculated as the linear combination of the eigenmodes and is introduced in the crash box structure. In the next step, the transient problem is solved with the pre-deformed imperfect crash box. This procedure is supported by the evident superposition of eigenmodes to obtain a solution of inhomogeneous differential equations. FEA is used to solve this nonlinear dynamic system of equations.

3.1.1. FEA-preprocessing. Simulia ABAQUS (version 2021) [75] is used to model the problem. This dynamic impact problem is solved using the ABAQUS explicit solver including geometrical nonlinearity. The Python script is used for simulating various crash box configurations impacted with varying impact velocities.

The crash box is modelled as a shell structure. The impactor plate is modelled as a rigid 2D square plate with a side length of 200 mm, which moves with the initial velocity of impact. Additionally, a support plate identical to the impactor plate is secured to the rear of the crash box to stop the crash box from deforming excessively in any arrangement. Figure 3 represents the complete simulation

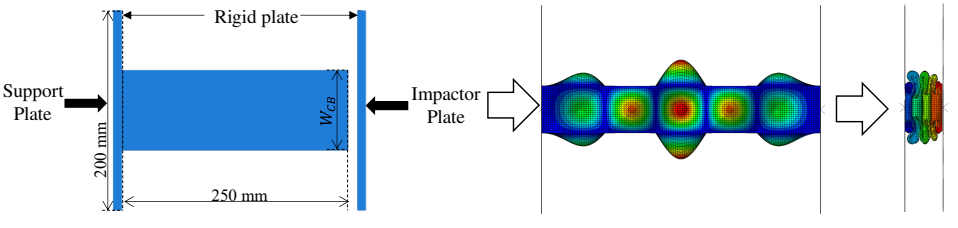


Fig. 3. Simulation method: assembly, linear buckling analysis and dynamic impact analysis (from left to right).

assembly. Multiple configurations of crash box input are simulated and input parameters, namely the width (W_{CB}) and thickness of the crash box (t_{CB}), and the impact velocity (V_{Impact}), are varied in a specific range, as illustrated in Table 1. However, the length of the crash box (L_{CB}) and the impactor mass (M_{Impactor}) are kept constant at 250 mm and 100 kg, respectively.

TABLE 1. Arbitrary development of structural parameters of crash box design.

Structural parameters	W_{CB} [mm]	t_{CB} [mm]	V_{Impact} [m/s]
Lower bound	40	1	5
Upper bound	80	2	10

The contact is assumed to be frictional with a friction coefficient of 0.25. The impactor and support plate have meshed with rigid elements (R3D4) with a size of 10 mm, and the crash box body has meshed with shell elements (S4R) with a size of 2 mm. The size and element selection are based on the mesh sensitivity research done in the preliminary work, and it has been found that this element size produces accurate results with minimal computing effort required.

TABLE 2. Material properties.

Density [$\mathrm{kg \cdot m^{-3}}$]	Young's modulus [GPa]	Poisson ratio –
2700	70	0.33

TABLE 3. Plasticity properties.

Yield stress [MPa]	Plastic strain –
80	0
173	0.174

The material is assumed to be aluminium 6063-T5 with bilinear elastoplasticity with a von Mises yield surface and strain-rate independent isotropic hardening [76]. The elastic and plastic properties of the material are illustrated in Tables 2 and 3, respectively.

3.1.2. Impact simulations for crashworthiness analysis. In the impact analysis, symmetry buckling does not occur even when the bifurcation point is reached; therefore, a small numerical imperfection is introduced to trigger the buckling. The imperfection is a numerical imitation of the imperfection in the structure and the deformation characteristics are affected by this imperfection. As a result,

a realistic solution is obtained [70]. Therefore initially, the linear bucking analysis (LBA) is performed with Abaqus/Standard. Output from the LBA is used to introduce imperfection in the dynamic impact simulation. Only the first mode is assumed to define the imperfection. The imperfection is defined by scaling the lowest eigenmode. Thus, the imperfection of 1% of the thickness of the crash box in each configuration is introduced. The goal of this method is to perturb the mesh to achieve the correct deformation pattern in the early part of the impact analysis without significantly reducing the external work done.

Following LBA, dynamic impact simulations are performed for each configuration. The impactor is impacted on the fixed crash box and the deformation is analysed. To optimise the crash box, the total energy absorbed is evaluated. The energy absorbed (EA) by the crash box is defined as the crash energy dissipated by the crash box in plastic deformation, as shown in Eq. (3.8). Apart from the EA, a few more metrics are evaluated in crashworthiness. This article uses initial peak crushing force (IPCF), mean contact force (P_{m}), and maximum deformed length ($d_{\max}$), as crashworthiness evaluation metrics. IPCF is the maximum reaction force observed during the initial stages of impact and it should be controlled so other neighbouring structures would not yield before the crash box starts deforming. Here, P_{m} is the average reaction force experienced by the crash box over the $d_{\max}$, it can be computed with the EA as shown in Eq. (3.9) [77]. These crashworthiness metrics are evaluated for all the configurations. The mass of the crash box (M_{CB}) is furthermore considered due to the increasing importance of lightweight design in the automotive industry [78]:

$$EA = \int_0^{d_{\max}} F(s)\, ds, \tag{3.8}$$

$$P_{\mathrm{m}} = \frac{EA}{d_{\max}} = \frac{\int_0^{d_{\max}} F(s)\, ds}{d_{\max}}, \tag{3.9}$$

where $F(s)$ is the contact force observed over the deformed length s. Subsequently, a database of various configurations and respective crashworthiness metrics values is assembled. The database consists of the structural parameter of the crash box and analysis parameters, W_{CB}, L_{CB}, t_{CB}, V_{Impact}, M_{Impactor} and crashworthiness metrics, IPCF, EA, P_{m}, and $d_{\max}$, M_{CB}. This database is used in the data-driven part of the machine learning framework and the training of the proposed DCGAN.

3.1.3. Optimisation problem. In impact simulations from the input variables, the energies, reaction forces and deformation are computed. From this data, the crashworthiness metrics values are determined. The goal of this study is to

optimise the input variables for the user-defined crashworthiness metrics values. Therefore, the inverse optimisation problem is solved. The five components I_k of an input vector can be summarised as

$$\boldsymbol{I} = (I_1\ I_2\ I_3\ I_4\ I_5)^T. \tag{3.10}$$

The I_k are five input variables, W_{CB}, L_{CB}, t_{CB}, V_{Impact}, M_{Impactor}. Thus the complete problem can be expressed as the following Eq. (3.11) and every crashworthiness metric is represented as a separate function ($f_j(\boldsymbol{I})$). The approach is proposed in studies on multi-objective optimisation [79]. The optimisation function ($T(f_j(\boldsymbol{I}))$) can be described as the combination of the functions of crashworthiness metrics and the constraint can be represented in terms of user-specified crashworthiness metrics values, like $\overline{\mathrm{IPCF}}, \overline{EA}, \overline{d}_{\max}, \overline{P}_{\mathrm{m}}, \overline{M}_{\mathrm{CB}}$:

$$\min T(f_j(\boldsymbol{I})) = F(f_1(\boldsymbol{I}), f_2(\boldsymbol{I}), f_3(\boldsymbol{I}), f_4(\boldsymbol{I}.f_5(\boldsymbol{I})) \quad \forall\, j = 1,5, \tag{3.11}$$

Subject to constraints:

$$\begin{aligned} f_1(\boldsymbol{I}) &= \mathrm{IPCF}, & 0 \le \mathrm{IPCF} \le \overline{\mathrm{IPCF}},\\ f_2(\boldsymbol{I}) &= EA, & 0 \le EA \approx \overline{EA},\\ f_3(\boldsymbol{I}) &= d_{\max}, & 0 \le d_{\max} \le \overline{d}_{\max},\\ f_4(\boldsymbol{I}) &= P_{\mathrm{m}}, & 0 \le P_{\mathrm{m}} \le \overline{P}_{\mathrm{m}},\\ f_5(\boldsymbol{I}) &= M_{\mathrm{CB}}, & 0 \le M_{\mathrm{CB}} \le \overline{M}_{\mathrm{CB}}. \end{aligned}$$

3.2. Machine learning framework

Due to the complex nature of the crash box design, all parameters ($\boldsymbol{I}$) must be carefully assessed and an extensive method is required to automate this design-

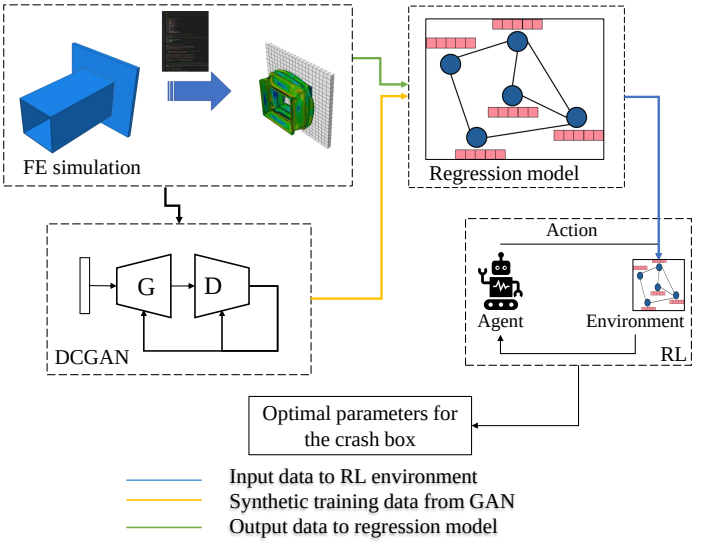


FIG. 4. Complete machine learning framework.

optimisation process. As a result, several machine learning methods interact ingeniously to produce the final results. The complete machine learning framework is displayed in Fig. 4. This section describes the components of the architecture as well as the complexities of the individual components. Initially, an artificial database is created using the Deep Convolutional Generative Adversarial Network. This database is similar to the data acquired from impact simulations using FEM. This database is used in the regression network training. This regression network is used in the RL environment, which automates the exploration of design parameters.

3.2.1. Deep convolutional generative adversarial network. Given the complexity and computational expense of crash simulations, it is understandable that the existing studies so far have settled on strategies that result in the best accuracy for fewer data samples. Generating a larger number of samples for training an FE surrogate for the RL leads to excess computational effort and time. To overcome this, a generative-based approach is proposed, employing the Deep Convolutional Generative Adversarial Network (DCGAN) to generate synthetic data required for training the regression network which acts as a solver for the environment for RL thereby saving time and computational expense. The data generated includes the t_{CB}, V_{Impact} and crashworthiness metrics pairs of the crash box configuration.

The overall architecture of the proposed DCGAN is shown in Fig. 5. GAN are a class of deep learning algorithms that can generate synthetic data from latent space. GANs provide a unique ability to generate a new multidimensional vector space corresponding to points in the problem domain forming a compressed representation of the data distribution. The GAN framework consists

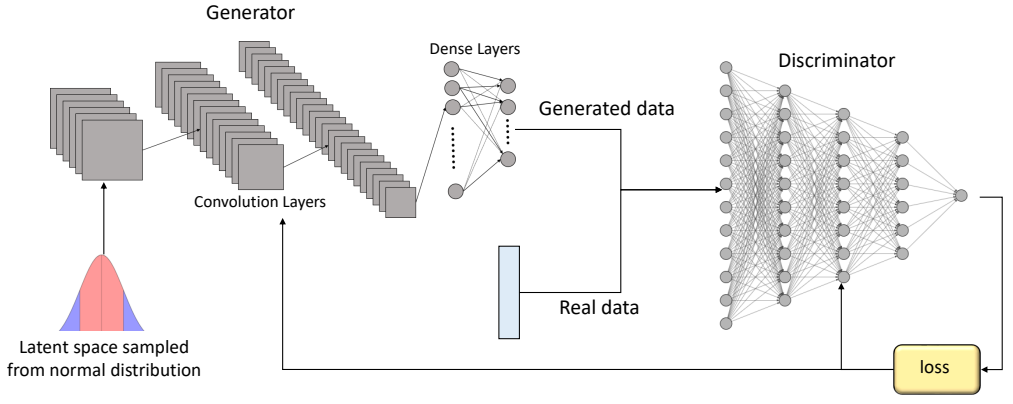


Fig. 5. Architecture of the DCGAN.

of two deep networks namely, a generator and a discriminator. These two neural networks compete with each other in a min-max game or a zero-sum game, where one network's gain is the other network's loss. To begin with, a latent vector is passed to the Generator which in turn generates synthetic samples of data represented by $G(z)$. This generated data is passed to the discriminator which outputs a probability of the given data to be real or synthetic data which is represented by $D(G(z))$. Real data is also passed into the discriminator whose output is denoted by $D(x)$. The networks are then trained on these resulting outputs where we aim to maximize the loss of discriminator and minimize the loss of generator. In other words, we train the Generator to produce realistic synthetic data that is indistinguishable from the real data for the discriminator. This implies, that in an ideal training scenario, the discriminator gives a probability of 50% for both real and synthetic data. Therefore to achieve this, it is crucial to have the right loss function, the value of which the generator tries to minimise and the discriminator maximizes. The loss function is expressed by

$$\min_G \max_D V(G,D) = E_x[\log(D(x))] + E_z[\log(1 - D(G(z)))]. \tag{3.12}$$

Here, G represents the generator, D represents the discriminator, $D(x)$ represents the output of the discriminator for real input, E_x is the target or actual output over all the real data instances, $G(z)$ is the output generated by the generator with the given noise z. The output of the discriminator for a generated input for a sample z is denoted by $D(G(z))$ and E_z is the expected value over all random inputs to the generator. The LHS of the above equation represents the two-player game where $V(G,D)$ represents the loss function and the generator tries to minimize Min_G the loss for fake outputs while the discriminator tries to maximize Max_D the loss. A lot of generator and discriminator architectures have been proposed since their inception. In the present study, a generator with residual connections is proposed.

Usually, in sequential networks, the output of the current layer n^{th} is connected to the next layer $(n+1)^{\text{th}}$ as an input, with x_i being the data at layer i, where $i = 1, k$ with k being the final layer. Consider an arbitrary layer, n, as the current layer and $n+1$ as the succeeding layer, which results in the transformation $x_{n+1} = H(x_n)$ in a classical convolutional network. Where $H()$ is a convolution operator that maps the input to output based on the number of convolutional filters and filter stride. The output data shape after a convolution, for an input shape of $W_{\text{out}} \times H_{\text{out}}$ is given by $\frac{(W \times H)_{\text{in}} - K + 2P}{S} + 1$. In the current study, a residual connection is added between the non-linear transformation, also known as a skip connection, which transforms $x_{n+1} = H(x_n)$ to $x_{n+1} = H(x_n) + x_n$. The skip connection is fundamentally an identity mapping,

where the input from the previous layer is added directly to the output of the other layer. These residual connections help in treating the vanishing gradient problem which plagues the deep neural networks. In the case of the discriminator, we employ a densely connected sequential network to differentiate between real and generated samples.

Unlike the usual application of GANs where the generator tries to train based on the likelihood of data i.e., probabilities, we apply the GANs for curve fitting. This is a novel implementation of GANs and requires unconventional loss functions [80]. The loss functions for the generator and discriminator with m number of samples are given by:

$$\mathcal{L}_{\text{Gen}} = \alpha \frac{1}{m} \sum_{i=1}^{m} \log(1 - D(G(z^{(i)}))) + \beta \frac{1}{m} \sum_{i=1}^{m} (Z^{(i)} - d^{(i)})^2, \tag{3.13}$$

$$\mathcal{L}_{\text{Disc}} = \frac{1}{m} \sum_{i=1}^{m} \log(D(x^{(i)})) + \log(1 - D(G(z^{(i)}))). \tag{3.14}$$

3.2.2. Regression network. A regression model is developed to expedite the performance of the RL framework. The trained regression model is a key component in the RL environment, which is explained in Section 3.2.3. As mentioned in Section 3.2.1, this regression model is trained and validated using data from FEM simulations and DCGAN-generated data.

For this regression model, the crash box's structural parameters are chosen as input parameters and crashworthiness metrics are chosen as output parameters. As the length of the crash box will be fixed for the specific vehicle and the crashworthiness tests are predefined norms for a given vehicle class, the L_{CB} and M_{Impactor} are kept constant and are not trained in the regression model. Pandas [81] and seaborn [82] libraries are used for data preprocessing and the scikit-learn [83] library is used to construct and train the regression model.

Data analysis shows that W_{CB}, t_{CB}, and V_{Impact} have a considerable effect on crashworthiness metrics, as shown the top right corner square (5×5) in Fig. 6. It is observed that W_{CB} and t_{CB} have a major effect on the IPCF with the relative factor of 0.7 and 0.5, respectively, and while V_{Impact} has a major effect on the $d_{\max}$ shown with the relative factor of 0.6. Here, L_{CB} and M_{Impactor} are shown to not affect the crashworthiness metrics because in the database they are unchanged for all the simulation configurations. Therefore, the structural parameters of the crash box, W_{CB}, t_{CB}, and V_{Impact} are chosen as input factors for this regression model, whereas crashworthiness criteria, namely IPCF, EA, P_{m}, $d_{\max}$ and M_{CB}, are selected as output parameters. Table 4 shows the network's architecture and chosen hyperparameters for training the network.

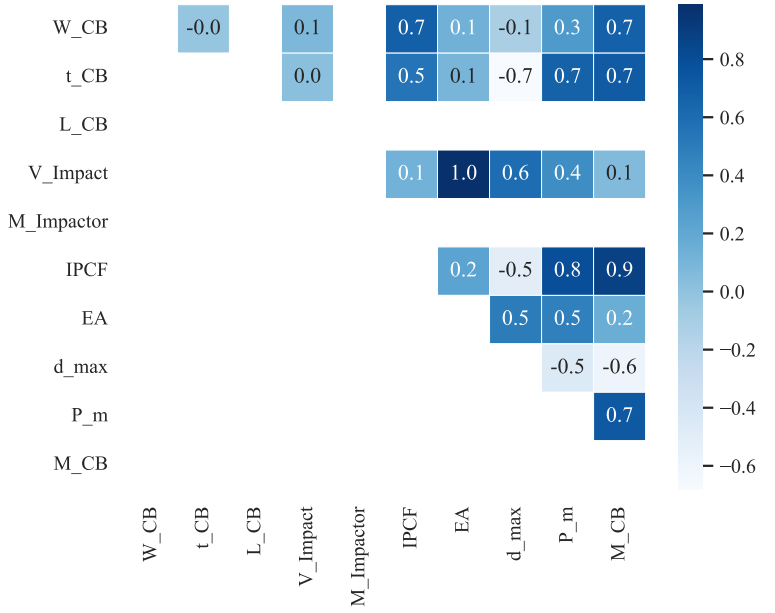


FIG. 6. Correlation matrix between structural parameters and crashworthiness criteria.

TABLE 4. Hyperparameters of the SLNN network.

Hyperparameters	Parameter value
Network architecture	3-40-40-40-40-5
Train- test split	0.8- 0.2
Optimiser	Adam [84]
Learning rate	0.001
Epochs	500
Batch size	16
Loss	MSE
Metrics	MAE, RMSE

3.2.3. Reinforcement learning environment. In this study, a machine learning framework based on reinforcement learning is proposed for the optimization process. To automate the exploration of the structural parameters of the crash box, a model-free RL environment is created. The environment is in OpenAI's Gymnasium format [85]. User-defined structural parameters are analysed in the RL environment and required structural parameters are identified that fulfil the crashworthiness metrics value specified by the user.

For the RL environment, initially, two input arrays are provided. The first array consists of user-defined structural parameters initial thickness (t_{Initial}), W_{CB} and V_{Impact} and the second array is made up of crashworthiness metrics

value specified by the user, namely IPCF, EA, P_{m}, $d_{\max}$ and M_{CB}. The t_{Initial} is required as the engineer's first guess to initialize the RL environment. This choice could help RL agent compute the optimal thickness (t_{opt}) quickly if the t_{Initial} is near the t_{opt} as later shown in the results Section 4.4.

W_{CB} and V_{Impact} are regarded as constant parameters specified by the user and the RL agents cannot modify them. This is decided to provide consistent results every time and not a pair of width and thickness, like a pair of (W_{opt}, t_{opt}), every time agents are used. Also, it is assumed the V_{Impact} will be selected beforehand for specific vehicles and crash tests and is regarded as consistent and due to space constraints the W_{CB} can be fixed beforehand. Furthermore, as the regression model is trained on different t_{CB}, W_{CB} and V_{Impact}, the same RL environment can be implemented for different W_{CB} and V_{Impact} given by the user. The RL agents analyse the t_{Initial} for given W_{CB} and V_{Impact} based on the user-defined crashworthiness metric value. After analysing t_{Initial} the agent can incrementally change the thickness to t_{opt} fulfilling the crashworthiness metrics value.

The model-free environment is required to estimate the required thickness as there is no established model of optimization environment. The model in the RL refers to the dynamics which establish a connection between the change in the thickness for achieving the user-specified crashworthiness metrics value. The state (s) in the RL environment refers to the current value of the thickness of the crash box (starting with t_{Initial} and can later be changed by the agent's action/step) for given W_{CB} and V_{Impact} and action (a) refers to RL agent's step to either increase, decrease or keep the thickness as it is in that state. After every action of the RL agent the state of the RL environment is changed.

The RL agent attempts to analyse the consequence of the agent's action and take action in the direction to fulfil the user-defined crashworthiness metrics value. The RL environment compares the crashworthiness metrics value of the current state and user-defined crashworthiness metrics value and computes the rewards for the agent for taking the action. The RL agents try to find the policy that maximizes the reward as shown in Eq. (3.16). The RL agent takes the action in any state under the policy function. The policy is the function that returns the appropriate action in the current state. Initially, the agent wanders in the RL environment and gains experience by following a random policy then evaluating and incrementally improving the policy.

As the RL agents are motivated by the reward function, this makes the definition of reward challenging. Therefore, the trained regression network established in Section 3.2.2 is used. A reward for the RL agent is determined at every step after the action suggested by the agent is taken. The RL environment and trained regression model calculate the reward for the RL agent and the environment's subsequent state. With the output of the regression network, a consistent reward function is defined. Only IPCF and $d_{\max}$ values are considered in determining

the t_{opt} and defining the continuous reward function for the agent's action. The other crashworthiness metrics values, namely EA, P_{m} and M_{CB}, are provided as the information at the end of every step for the current thickness value. This is considered as all of the crashworthiness metrics are interdependent and therefore only IPCF and $d_{\max}$ are chosen to determine the required parameters. This reward function is based on positive and negative reinforcement and grants the agent a reward for every step in the right direction to fulfil the criteria. The environment stops the optimization process if the RL agent takes more than 500 steps or if the difference between the expected and predicted output variables is less than 0.2% and grants additional reward points.

Initially, the governing policy (π) of the RL agent is chosen at random. Then by using the state-action value also referred to as the Q-value ($Q^{\pi}(s,a)$) defined by the following Eq. (3.15), policy evaluation and improvements are iteratively carried out. The Q-value estimates the expected rewards in the current state s by taking action a under π [86]

$$Q^{\pi}(s,a) = R(s,a) + \gamma \sum P(s'|s,a)V^{\pi}(s,a), \tag{3.15}$$

where $Q^{\pi}(s,a)$ is the Q-value for a current state (s) and action (a) according to policy (π), $R(s,a)$ is a current reward, (γ) is the discount factor for future rewards, ($P(s'|s,a)$) is the Markov chain probability matrix for current s and a and the next state (s'), and $V^{\pi}(s,a)$ is the state value function for current s and a. The agent's task is to carry out the policy improvements to collect the highest rewards [86] and determine the optimal policy as defined $\pi^*(s)$ to carry out the optimal action ($a^{\pi*}(s)$) in s as stated in Eq. (3.16) [86, 87]

$$a^{\pi*}(s) = \arg\max_{a} Q^{*}(s,a), \tag{3.16}$$

where $Q^{*}(s,a)$ is the optimal state-action value. It is the expected reward in s and carrying a according to π^* [87].

To get model-free estimates of the $Q^{\pi}(s,a)$ for all states and actions, all s–a pairs are needed to be tested. However, as the choice of π is initialized as a random event and then incrementally improved, it is possible that under some chosen π some actions might not be taken in certain states. Therefore, ϵ-greedy policy exploration is carried out. It offers the probabilistic choice to take a under π or choose another action. The probability is calculated from the ϵ value. It helps in balanced exploration and converging $Q^{\pi}(s,a)$ to the true Q-value [86].

Because the RL environment is model-free, only a restricted number of agents can be used. Therefore, pre-existing REINFORCE [88, 89] from OpenAI's Gymnasium documentation and deep Q-learning with ϵ-greedy policy (DQN) [90]

and proximal policy optimization (PPO) from Keras documentation [91] agents have been assessed. These agents have been adjusted to be used with the custom environment and then trained. The following Section 4.4 summarises the training's outcomes.

4. Results

4.1. Impact simulation results

Figure 7 shows the complete framework developed for the crash box design optimisation. LBA and dynamic impact simulations are performed on different crash box configurations, which serve as a simulation database. Then, structural parameters of the crash box, impact test specification, and crashworthiness metric values are extracted from these FEM simulations, as described in

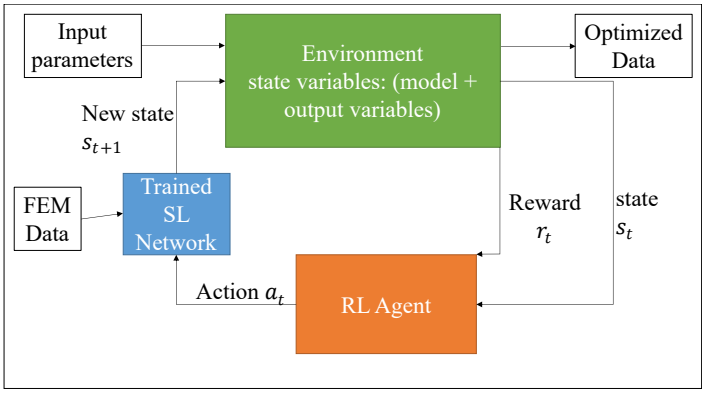


Fig. 7. Reinforcement learning framework.

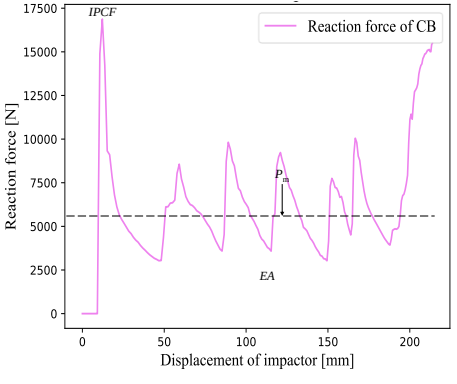


Fig. 8. Reaction force-deformation graph for axially impact loaded crash box.

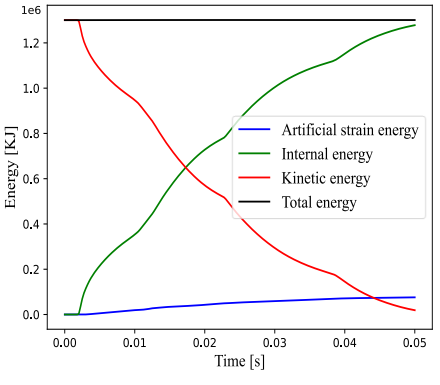
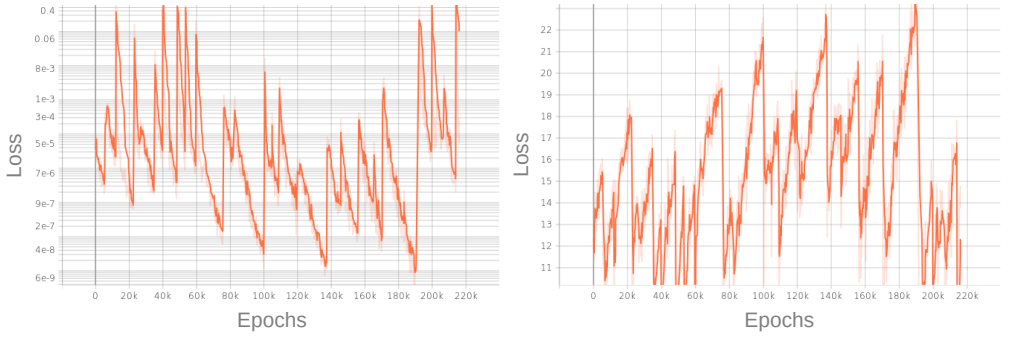


Fig. 9. Energies over deformed length graph for one of the configurations.

Section 3.1.2. For all configurations, graphs are captured for all the energies and the reaction forces over deformation, as shown in Figs. 8 and 9 respectively. Figure 9 shows the energy absorbed by the crash box (green curve) and Fig. 8 represents the deformation pattern. The initial peak represents the initial buckling in Fig. 8 which also denotes IPCF and each subsequent peak shows the local bucking phenomenon which denotes the wavy deformation pattern as displayed in Fig. 3. From this Fig. 8, crashworthiness metrics mentioned in Section 3.1.2 IPCF, $d_{\max}$, P_{m} can be calculated.

4.2. DCGAN results

Figure 10 shows the loss of the discriminator and generator during training. It can be observed that the losses are approximate mirror images of each other. This is because the generator and discriminator engage in a zero-sum game where they compete with each other. This makes it hard to evaluate the training from



(a) Epochs vs loss for generator. (b) Epochs vs loss for discriminator.

FIG. 10. Loss plot of generator and discriminator.

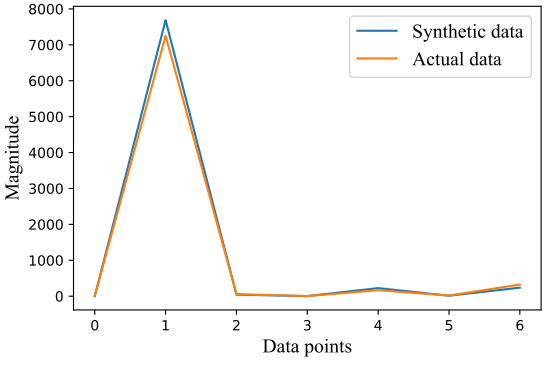


FIG. 11. Comparison of the synthetic sample with the nearest actual data sample.

the losses similar to other neural network frameworks. Therefore, the likelihood of the synthetic data to resemble the actual data is used as a measure of training.

In Fig. 11, a synthetic sample generated by the proposed DCGAN, which is present in the data set is compared to the closest actual sample. It can be observed that the general trend of the variables of the synthetic sample conforms to the actual sample suggesting the success of the proposed DCGAN in creating realistic synthetic samples that are not present in the dataset.

4.3. Regression network results

A regression network is trained using the above-mentioned database of FEM simulations and data generated by using the DCGAN. This trained regression model can estimate the IPCF, EA, P_{m}, $d_{\max}$ and M_{CB} for a crash box with W_{CB} and t_{CB} and impacted at V_{imapct}. This trained regression model is an essential part of the RL environment in computing the next state of the environment as per the agent's action and the corresponding reward for RL agents. Therefore, the accuracy of this regression model is crucial.

As the training progresses the final loss is observed at 0.0014 and 0.0224 for the training and validation dataset, respectively. The trained model has an accuracy of 98.11% and 94.05% for training and the validation data set, respectively. In Fig. 12, the loss and accuracy for the training and validation dataset are illustrated but the losses are close at the final stages of training as it approaches 0. The total training and validation time for this network is 4 min 21 sec for 500 epochs.

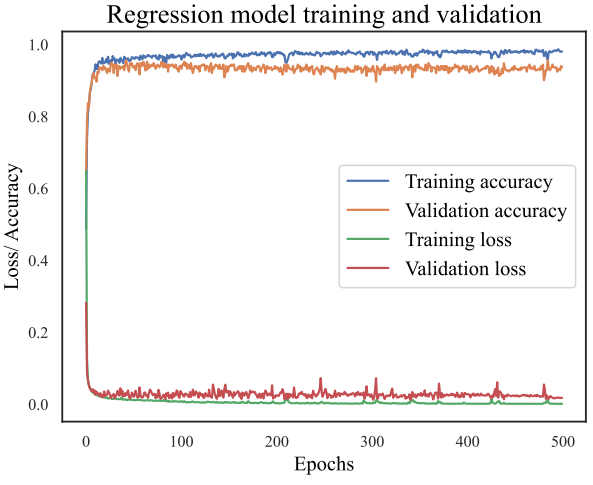


Fig. 12. Loss over the training of regression model.

The residual distribution and fluctuation of prediction for the test dataset are computed to observe the accuracy of all the output variables. A slightly

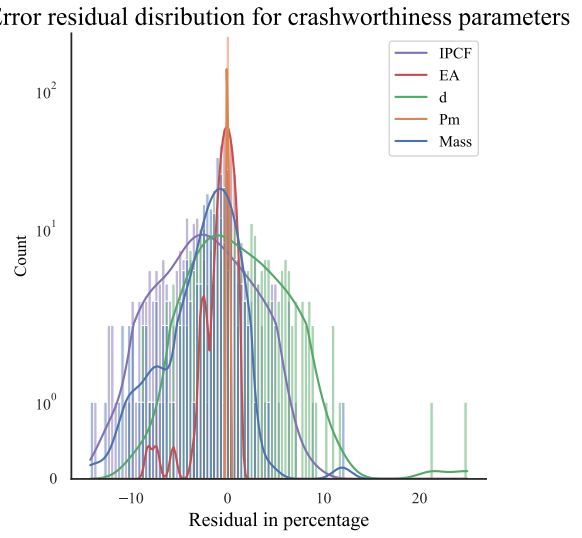


FIG. 13. Residual percentage error in prediction of regression model for validation dataset.

negative-biased distribution for predicted values of IPCF, EA, $d_{\max}$, P_{m} M_{CB} and is shown, as illustrated in Fig. 13. From the above loss graph and this residual distribution, it is observed that the model is well-trained and predicts the output accurately with the residual error between $+/-15\%$ and most of the predictions lie in $+/-10\%$. Therefore, this regression model is employed in the RL environment.

4.4. Reinforcement learning results

4.4.1. Reinforcement learning training. REINFORCE, DQN and PPO agents are tested in the RL environment. The reward over the training graph for each agent is shown in Figs. 14–16, respectively. The structural parameters, t_{initial}, W_{CB} and V_{Impact}, for each agent are 1 mm, 45 mm, 5.1 m/s, repectively. The values for required crashworthiness criteria IPCF and $d_{\max}$ are 19.56 kN and 128.22 mm, respectively and are used to determine t_{opt} and compute the reward for the RL agents. The other crashworthiness metrics EA, P_{m} and M_{CB} are 1.18 kJ, 9.23 kN and 170 g, respectively. The training format for all the agents is different as agents are developed by different organizations. The training format is chosen so the training time and number of iterations for each agent are loosely comparable.

The REINFORCE agent is observed to take a significant amount of training time yet is still unable to predict t_{opt} consistently, but the DQN and PPO can predict the required thickness of given W_{CB} and V_{Impact}. Unlike comparable

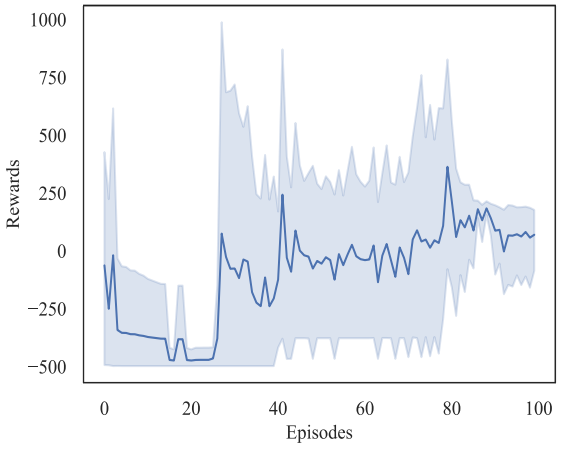


FIG. 14. Reinforce agent learning.

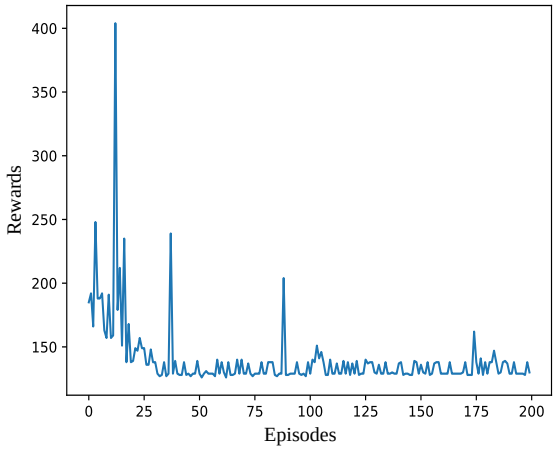


FIG. 15. DQN agent learning.

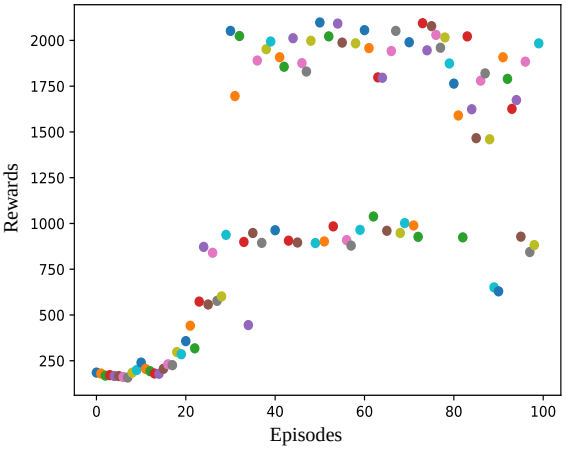


FIG. 16. PPO agent learning.

TABLE 5. RL agents training format and learning time.

Agent	Learning time [min]	Training format [–]
REINFORCE	79.41	5×100
DQN	3.12	200×1
PPO	14.73	100×250

RL-related research, the rewards in this one only increase up to a particular level. At each step, the agents have been encouraged to take intelligent moves or actions to estimate t_{opt}. To maximize the reward and predicted t_{opt}, the agents perform an appropriate number of actions from t_{initial} and not get stuck in an unwanted loop of achieving rewards by oscillating. In any state, the most beneficial course of action is to strive toward satisfying the crashworthiness metrics value while earning rewards. As a result, throughout the training, the agent is encouraged to act fittingly.

The DQN agent is observed to predict the optimal thickness quicker than the PPO agent, as indicated in Table 5. As a result, additional investigations are done to understand the DQN results and sensitivity to the input parameters. Also, it is observed that the DQN agent can predict the optimal parameters quickly as shown by the rewards reaching the stagnation point after episode 50. After this point, rewards are almost identical for the remaining episodes. However, the DQN is still trained for 200 episodes to ensure adequate training of the underlying model and increase robustness. It is also advantageous that the saved model can be generally used for various configurations as shown in Section 4.4.2. Table 6 shows the DQN can determine t_{opt} which is close to t_{CB}. As both DQN agents determine identical t_{opt} the RL agent predicted properties are the same and are near the user-specified/expected crashworthiness metrics values.

Also as observed in Table 6, while the DQN agent can predict the t_{opt} and the t_{initial} determines the time required to reach completion of agents. The accuracy

TABLE 6. DQN agents training for different initial thickness values.

Property	DQN-Agent 1/2
t_{Initial} [mm]	1/1.2
t_{CB} [mm]	1.4
t_{opt} [mm]	1.38
Expected IPCF [kN]	19.56
Predicted IPCF [kN]	19.44
Expected $d_{\max}$ [mm]	128.22
Predicted $d_{\max}$ [mm]	131.45
Time required [min]	3.03/2.14

of the regression network in the RL environment also influences the final result and time required for the optimisation process. The fluctuations of rewards in episodic training settings towards the completion of the RL training are observed and are characteristic of RL agents. It is suggested that the RL agents can be trained for a longer time to reduce the fluctuations. Regardless, the framework can determine the t_{opt} for pre-defined crashworthiness criteria for a user-defined consistent W_{CB}, V_{Impact}. The framework can be used to optimise the thickness of a crash box of 250 mm length and the square cross-section of varied width impacted at different impact velocities, as well as specified crash tests and crashworthiness metrics with minor changes in the RL environment.

4.4.2. Reinforcement learning validation. From Table 7, it can further be observed that the saved DQN agent can further be used for finding optimal thickness. It can assess the input arrays and determine the optimal thickness for given crashworthiness criteria and the time required is a few seconds.

Table 7. Saved DQN agents optimises for different structural parameters and crashworthiness criteria.

Structural parameters	Crashworthiness criteria	t_{opt} [mm]	t_{CB} [mm]	Time required [s]
45-1.0-5.1	19.56-1.18-128.22-9.23-170	1.39	1.4	1.76
65-1.0-5.4	35.43-1.38-85.71-16.19-316	1.79	1.8	2.87
45-1.5-9.3	33.20-4.04-202.43-20.00-243	1.97	2.0	3.60
40-1.5-7.2	26.08-2.41-142.81-16.88-205	1.83	1.9	1.95

In this article, design optimization with the machine learning method is explored. Different ML methods are employed to work together and it is demonstrated to be feasible. Since shape optimization is a nonlinear polynomial fit, machine-learning methods can be used to find the solution.

Data augmentation using DCGAN is performed in this study to boost the data set for training the regression network and subsequently the RL agent. The DCGAN has successfully generated synthetic samples of the crash box's structural parameters and the crashworthiness metrics pairs from latent noise.

By using FEM data and DCGAN, a data-driven regression model is employed in the RL environment. The RL agent is being used for parameter exploration and optimization in the shape optimization. Because of its automated exploration capabilities, RL may be employed in a variety of mechanical optimization methods. This could reduce the amount of computational and manual effort required. The trained regression model in the environment is used by the RL in this study to understand the shape optimization model. However, the RL envi-

ronment can be used directly in the FEM solver to fully automate the process in the simulation workspace.

As REINFORCE directly optimises the policy for maximum Monte-Carlo return, hyperparameter tuning and an increased number of episodes might be needed for a good-performing REINFORCE agent. However, due to the model-free nature of the environment, DQN and PPO can be directly used for the current problem. Also, the tuning and training time required for DQN and PPO is considerably shorter than REINFORCE. Although RL can predict the most suitable design parameters, a validation study using FEA may be necessary. Because RL training appears to have fluctuating variation even after training, engineers should utilize the RL results in combination with professional experience in the same field to assess suitable configurations. Nonetheless, the current machine learning framework can be utilised for the square crash box of 250 mm length. Varied widths and impact velocities can be used with the current framework and the optimal thickness can be obtained for user-defined crashworthiness standards.

5. Conclusions

The article examines dynamic crash box deformation, data augmentation through generative adversarial networks, and structural multicriteria optimization using reinforcement learning. The crash box design and deformation mechanism are investigated after the theoretical shell model is established. The shell stability phenomenon under dynamic axial loading is also investigated in this structural model. The FE simulation is used to develop and simulate the numerical approximation model. To help with the data-driven ML component, DCGAN-based data is generated in addition to FEM-derived data. A framework for structural optimization based on RL is designed to optimise the thickness of the square crash box under predetermined crashworthiness regulations. The length and cross-section of the crash box are fixed in this study, and only the thickness is optimised. Following this investigation, comparable ML frameworks for optimizing various automotive structural components can be built. This could aid in reducing the amount of time and computational resources required for prototype development.

The study's key findings are as follows:

- A polynomial regression model is used as a metamodel in the optimisation process.
- Data is generated using DCGAN for computationally expensive simulations.
- A ML framework based on RL is developed for multiobjective design parameter exploration in the optimisation procedure.

- A model-free custom RL environment is utilised with preexisting RL agents in the optimisation process.
- The saved RL agent is used for design optimisation of various configurations of the crash box.

The current study has demonstrated the prospect of machine learning (ML) in the crashworthiness analysis. However, additional investigation is required to optimize multiple parameters simultaneously for predefined crashworthiness criteria. This can be a substitute for conventional methods, making it more efficient. A comparative study is also necessary to evaluate the robustness and generalizability of the ML framework over conventional methods. This comparison will justify the use of an ML framework in industries. Nevertheless, this study's results show that model-free RL can be promising in industrial cases, and primitive RL agents (DQN and PPO) could be useful in a variety of cases. Moreover, newer RL agents may solve additional complicated problems and be more efficient than DQN/PPO.

Acknowledgements

The authors gratefully acknowledge the financial support provided by Deutsche Forschungsgemeinschaft Priority Programme: SPP 2353 (DFG Grant No. STO 469/16-1).

References

1. Destatis, *Accidents registered by the police by type of damage/location*, 2023, Database access in March, 2023.

2. A. Blackburn, *Road safety in the UNECE region*, July 2022, Database access in April, 2023.

3. S. Yadav, S.K. Pradhan, *Investigations into dynamic response of automobile components during crash simulation*, Procedia Engineering, **97**, 1254–1264, 2014, 12th Global Congress on Manufacturing and Management, GCMM, 2014.

4. F. Duddeck, E. Wehrle, *Recent advances on surrogate modeling for robustness assessment of structures with respect to crashworthiness requirements*, 10th European LS-DYNA Conference, 06, 2015.

5. S. Zhang, P. Zhu, W. Chen, *Crashworthiness-based lightweight design problem via new robust design method considering two sources of uncertainties*, Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, **227**, 7, 1381–1391, 2013.

6. T. Valayil, J.Ch. Isaack, *Crash simulation in ANSYS LS-DYNA to explore the crash performance of composite and metallic materials*, International Journal of Scientific and Engineering Research, **4**, 8, 2013.

7. P. Zhu, F. Pan, W. Chen, S. Zhang, *Use of support vector regression in structural optimization: Application to vehicle crashworthiness design*, Mathematics and Computers in Simulation, **86**, 21–31, 2012, The 7th International Symposium on Neural Networks + The Conference on Modelling and Optimization of Structures, Processes and Systems.
8. M. Rayamajhi, S. Hunkeler, F. Duddeck, *Efficient robust shape optimization for crashworthiness*, 10th World Congress on Structure and Multidisciplinary Optimization, 05, 2013.
9. J. Forsberg, L. Nilsson, *Evaluation of response surface methodologies used in crashworthiness optimization*, International Journal of Impact Engineering, **32**, 5, 759–777, 2006, International Symposium on the Crashworthiness of Light-weight Automotive Structures.
10. L. Morello, L. Rosti Rossini, G. Pia, A. Tonoli, *The Automotive Body Volume II: System Design*, 1st ed., Mechanical Engineering Series, Springer, Dordrecht, 2011.
11. G. Lu, T. Yu, *1 – Introduction*, in: *Energy Absorption of Structures and Materials*, G. Lu, T. Yu [eds.], Woodhead Publishing Series in Metals and Surface Engineering, Woodhead Publishing, 2003, pp. 1–24.
12. M. Kuehn, T. Hummel, J. Bende, *Small-overlap frontal impacts involving passenger cars in Germany*, 23rd Enhanced Safety of Vehicles Conference, 05, 2013, pp. 1495–1504.
13. N.S.B. Yusof, S.M. Sapuan, M.T.H. Sultan, M. Jawaid, M.A. Maleque, *Design and materials development of automotive crash box: A review*, Ciência & Tecnologia dos Materiais, **29**, 3, 129–144, 2017.
14. J. M. Alexander, *An approximate analysis of the collapse of thin cylindrical shells under axial loading*, The Quarterly Journal of Mechanics and Applied Mathematics, **13**, 1, 10–15, 1960.
15. M. Altin, S. Halis, H. Yücesu, *Investigation of the effect of corrugated structure on crashing performance in thin-walled circular tubes*, International Journal of Automatic Science and Technology, **1**, 12, 1–7, 2017.
16. D. Karagiozova, M. Alves, *Transition from progressive buckling to global bending of circular shells under axial impact – Part I: Experimental and numerical observations*, International Journal of Solids and Structures, **41**, 5, 1565–1580, 2004.
17. W. Abramowicz, N. Jones, *Dynamic axial crushing of square tubes*, International Journal of Impact Engineering, **2**, 2, 179–208, 1984.
18. W. Chen, T. Wierzbicki, S. Santosa, *Bending collapse of thin-walled beams with ultralight filler: Numerical simulation and weight optimization*, Acta Mechanica, **153**, 183–206, 2002.
19. D.C. Han, S.H. Park, *Collapse behavior of square thin-walled columns subjected to oblique loads*, Thin-Walled Structures, **35**, 3, 167–184, 1999.
20. C. Qi, S. Yang, F. Dong, *Crushing analysis and multiobjective crashworthiness optimization of tapered square tubes under oblique impact loading*, Thin-Walled Structures, **59**, 103–119, 2012.
21. X. Zhang, G. Cheng, Z. You, H. Zhang, *Energy absorption of axially compressed thin-walled square tubes with patterns*, Thin-Walled Structures, **45**, 9, 737–746, 2007.
22. J. Wang, Y. Zhang, N. He, C.H. Wang, *Crashworthiness behavior of Koch fractal structures*, Materials & Design, **144**, 229–244, 2018.

23. N. Tanlak, F.O. Sonmez, *Optimal shape design of thin-walled tubes under high-velocity axial impact loads*, Thin-Walled Structures, **84**, 302–312, 2014.

24. H. Ghasemnejad, H. Hadavinia, D. Marchant, A. Aboutorabi, *Energy absorption of thin-walled corrugated crash box in axial crushing*, SDHM Structural Durability and Health Monitoring, **4**, 1, 29–46, 2008.

25. F. Djamaluddin, *Review: Deformation and optimisation crashworthiness method for foam filled structures*, Latin American Journal of Solids and Structures, **16**, 7, e213, 2019.

26. B. Jafarian, M.J. Rezvani, *An experimental investigation on energy absorption of thin-walled bitubal structures by inversion and axial collapse*, International Journal of Mechanical Sciences, **126**, 270–280, 2017.

27. J. Song, *Numerical simulation on windowed tubes subjected to oblique impact loading and a new method for the design of obliquely loaded tubes*, International Journal of Impact Engineering, **54**, 192–205, 2013.

28. S.W. Astuti, W.A. Wirawan, A. Zulkarnain, D.T. Istiantara, *Comparison of energy absorption and pattern of deformation material crash box of three segments with bilinear and Johnson Cook approach*, Journal of Physics: Conference Series, **1273**, 1, 012078, 2019.

29. N.N. Hussain, S.P. Regalla, Y.V.D. Rao, *Comparative study of trigger configuration for enhancement of crashworthiness of automobile crash box subjected to axial impact loading*, Procedia Engineering, **173**, 1390–1398, 2017, Plasticity and Impact Mechanics.

30. C.N. Nguyen, T. Dirgantara, L. Gunawan, I. Putra, H. Ly, *Analytical prediction of square crash box structure with holes due to impact loading*, Regional Conference on Mechanical and Aerospace Technology, 11, 2013.

31. X.W. Zhang, H. Su, T.X. Yu, *Energy absorption of an axially crushed square tube with a buckling initiator*, International Journal of Impact Engineering, **36**, 3, 402–417, 2009.

32. M. Bambach, L. Conrads, M. Daamen, O. Güvenç, G. Hirt, *Enhancing the crashworthiness of high-manganese steel by strain-hardening engineering, and tailored folding by local heat-treatment*, Materials & Design, **110**, 157–168, 2016.

33. A. Quadfasel, M. Teller, M. Madivala, C. Haase, F. Roters, G. Hirt, *Computer-aided material design for crash boxes made of high manganese steels*, Metals, **9**, 7, 2019.

34. A. Djerrad, F. Fan, X. Zhi, Q. Wu, *Experimental and FEM analysis of AFRP strengthened short and long steel tube under axial compression*, Thin-Walled Structures, **139**, 9–23, 2019.

35. M. Engul, F. Oz, N. Ersoy, *Experimental and numerical investigation of the crushing process of composite crash box*, 21th International Conference on Composite Materials, 08, 2017.

36. N.A.Z. Abdullah, M.S.M. Sani, M.S. Salwani, N.A. Husain, *A review on crashworthiness studies of crash box structure*, Thin-Walled Structures, **153**, 106795, 2020.

37. W. Abramowicz, N. Jones, *Transition from initial global bending to progressive buckling of tubes loaded statically and dynamically*, International Journal of Impact Engineering, **19**, 5, 415–437, 1997.

38. N. JONES, *Several phenomena in structural impact and structural crashworthiness*, European Journal of Mechanics – A/Solids, **22**, 5, 693–707, 2003, General and Plenary Lectures from the 5th EUROMECH Solid Mechanics Conference.

39. S.B. TANDALE, M. STOFFEL, *Spiking recurrent neural networks for neuromorphic computing in nonlinear structural mechanics*, Computer Methods in Applied Mechanics and Engineering, **412**, 116095, 2023.

40. R. GULAKALA, B. MARKERT, M. STOFFEL, *Graph neural network enhanced finite element modelling*, Proceedings in Applied Mathematics and Mechanics, **22**, 1, e202200306, 2023.

41. R. GULAKALA, B. MARKERT, M. STOFFEL, *Generative adversarial network based data augmentation for CNN based detection of Covid-19*, Scientific Reports, **12**, 11, 19186, 2022.

42. R. GULAKALA, B. MARKERT, M. STOFFEL, *Rapid diagnosis of Covid-19 infections by a progressively growing GAN and CNN optimisation*, Computer Methods and Programs in Biomedicine, **229**, 107262, 2023.

43. M. STOFFEL, D. WEICHERT, R. MÜLLER-RATH, *Modeling of articular cartilage replacement materials*, Archives of Mechanics, **61**, 1, 69–87, 2009.

44. C. GAMEZ, B. SCHNEIDER-WALD, A. SCHUETTE, M. MACK, L. HAUK, A. UL MAULA KHAN, N. GRETZ, M. STOFFEL, K. BIEBACK, M.L. SCHWARZ, *Bioreactor for mobilization of mesenchymal stem/stromal cells into scaffolds under mechanical stimulation: Rreliminary results*, PLOS ONE, **15**, 1, e0227553, 2020.

45. M. STOFFEL, F. BAMER, B. MARKERT, *Artificial neural networks and intelligent finite elements in non-linear structural mechanics*, Thin-Walled Structures, **131**, 102–106, 2018.

46. M. STOFFEL, T.D. VU, R. SCHMIDT, D. WEICHERT, *A comparative study of deformations of elastic-viscoplastic plates using different structural hypotheses*, Proceedings in Applied Mathematics and Mechanics, **12**, 1, 185–186, 2012.

47. C. CZECH, M. LESJAK, C. BACH, F. DUDDECK, *Data-driven models for crashworthiness optimisation: Intrusive and non-intrusive model order reduction techniques*, Structural and Multidisciplinary Optimization, **65**, 190, 2022.

48. S. DONG, T. JING, J. ZHANG, *A few-shot learning-based crashworthiness analysis and optimization for multi-cell structure of high-speed train*, Machines, **10**, 8, 696, 2022.

49. Z. FANG, K. ROY, Y. DAI, J.B.P. LIM, *Effect of web perforations on end-two-flange web crippling behaviour of roll-formed aluminium alloy unlipped channels through experimental test, numerical simulation and deep learning*, Thin-Walled Structures, **179**, 109489, 2022.

50. B. BOHN, J. GARCKE, R. IZA-TERAN, A. PAPROTNY, B. PEHERSTORFER, U. SCHEPSMEIER, C.-A. THOLE, *Analysis of car crash simulation data with nonlinear machine learning methods*, Procedia Computer Science, **18**, 621–630, 2013, 2013 International Conference on Computational Science.

51. Y. GAO, X. LIU, H. HUANG, J. XIANG, *A hybrid of fem simulations and generative adversarial networks to classify faults in rotor-bearing systems*, ISA Transactions, **108**, 356–366, 2021.

52. Z. NIE, T. LIN, H. JIANG, L.B. KARA, *TopologyGAN: Topology optimization using generative adversarial networks based on physical fields over the initial domain*, Journal of Mechanical Design, **143**, 3, 031715, 2021.

53. Z. Wang, S. Rosa, L. Xie, B. Yang, S. Wang, N. Trigoni, A. Markham, *Defonet: Learning body deformation using generative adversarial networks*, ArXiv: Computer Science, Robotics, 2440–2447, 2018.

54. J. Kober, J.A. Bagnell, J. Peters, *Reinforcement learning in robotics: A survey*, The International Journal of Robotics Research, **32**, 11, 1238–1274, 2013.

55. M. Schaarschmidt, A. Kuhnle, B. Ellis, K. Fricke, F. Gessert, E. Yoneki, *Lift: Reinforcement learning in computer systems by learning from demonstrations*, arXiv: abs/1808.07903, 2018.

56. V. Mnih, K. Kavukcuoglu, D. Silver, A. Rusu, J. Veness, M. Bellemare, A. Graves, M. Riedmiller, A. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, D. Hassabis, *Human-level control through deep reinforcement learning*, Nature, **518**, 529–533, 2015.

57. D.C. Han, S.H. Park, *Collapse behavior of square thin-walled columns subjected to oblique loads*, Thin-Walled Structures, **35**, 3, 167–184, 1999.

58. R.J. Hayduk, T. Wierzbicki, *Extensional collapse modes of structural members*, Computers & Structures, **18**, 3, 447–458, 1984.

59. T. Wierzbicki, W. Abramowicz, *On the crushing mechanics of thin-walled structures*, Journal of Applied Mechanics, **50**, 4a, 727–734, 1983.

60. W. Abramowicz, T. Wierzbicki, *A kinematic approach to crushing of shell structures*, [in:] 3rd International Conference on Vehicle Structural Mechanics (1979), SAE International, 1979.

61. W. Abramowicz, N. Jones, *Dynamic progressive buckling of circular and square tubes*, International Journal of Impact Engineering, **4**, 243–270, 1986.

62. Ø. Jensen, M. Langseth, O. Hopperstad, *Transition between progressive and global buckling of aluminium extrusions*, Conference on Structures under Shock and Impact, VII, pp, 269–277, 2002.

63. D. Karagiozova, N. Jones, *Dynamic elastic-plastic buckling of circular cylindrical shells under axial impact*, International Journal of Solids and Structures, **37**, 14, 2005–2034, 2000.

64. D. Karagiozova, N. Jones, *Dynamic buckling of elastic-plastic square tubes under axial impact – Part II: Structural response*, International Journal of Impact Engineering, **30**, 2, 167–192, 2004.

65. S. Gao, H. Tan, *Study of dynamic plastic buckling of cylindrical shell impacted by sudden constant load*, Technische Mechanik – European Journal of Engineering Mechanics, **12**, 4, 209–211, 2019.

66. T. Alexander, *The relationship between the buckling load factor and the fundamental frequency of a structure*, Structure Congress 2005, pp. 1–17, 04, 2005.

67. J.W. Hutchinson, W.T. Koiter, *Postbuckling theory*, Applied Mechanics Review, **23**, 12, 1353–1366, 1970.

68. N. Jones, *Structural Impact*, 2nd ed., Cambridge University Press, Cambridge, 2011.

69. J. Arbocz, *Post-buckling Behaviour of Structures Numerical Techniques for More Complicated Structures*, Springer, Berlin, Heidelberg, pp. 83–142, 1987.

70. H. SCHMIDT, *Stability of steel shell structures: General report*, Journal of Constructional Steel Research, **55**, 1, 159–181, 2000.

71. Dassault Systemes, Buckling of a simply supported square plate, 2023, Access in February, 2023.

72. Y. DAI, K. ROY, Z. FANG, B. CHEN, G.M. RAFTERY, J.B.P. LIM, *A novel machine learning model to predict the moment capacity of cold-formed steel channel beams with edge-stiffened and un-stiffened web holes*, Journal of Building Engineering, **53**, 104592, 2022.

73. M. STOFFEL, *Methode experimentale pour la validation de modeles mecaniques*, PhD Thesis, Aachen, 2007.

74. M. STOFFEL, R. SCHMIDT, D. WEICHERT, *Shock wave-loaded plates*, International Journal of Solids and Structures, **38**, 42, 7659–7680, 2001.

75. M. SMITH, *ABAQUS/Standard User's Manual, Version 6.9*, Dassault Systèmes Simulia Corp, United States, 2009.

76. ASM aerospace specification metals Inc, Aluminum 6063-t5, 2023, Database access in March, 2023.

77. N.S. HA, G. LU, *Thin-walled corrugated structures: A review of crashworthiness designs and energy absorption characteristics*, Thin-Walled Structures, **157**, 106995, 2020.

78. A. BAROUTAJI, M. SAJJIA, A.-G. OLABI, *On the crashworthiness performance of thin-walled energy absorbers: Recent advances and future developments*, Thin-Walled Structures, **118**, 137–163, 2017.

79. J. LIM, C. YOU, I. DAYYANI, *Multi-objective topology optimization and structural analysis of periodic spaceframe structures*, Materials & Design, **190**, 108552, 2020.

80. R. GULAKALA, B. MARKERT, M. STOFFEL, *Generative learning-based model for the prediction of 2D stress distribution*, Proceedings in Applied Mathematics and Mechanics, **23**, 4, e202300201, 2023.

81. The pandas development team, pandas-dev/pandas: Pandas, February, 2020.

82. M.L. WASKOM, *Seaborn: Statistical data visualization*, Journal of Open Source Software, **6**, 60, 3021, 2021.

83. F. PEDREGOSA, G. VAROQUAUX, A. GRAMFORT, V. MICHEL, B. THIRION, O. GRISEL, M. BLONDEL, P. PRETTENHOFER, R. WEISS, V. DUBOURG, J. VANDERPLAS, A. PASSOS, D. COURNAPEAU, M. BRUCHER, M. PERROT, E. DUCHESNAY, *Scikit-learn: Machine learning in Python*, Journal of Machine Learning Research, **12**, 2825–2830, 2011.

84. D.P. KINGMA, J. BA, *Adam: a method for stochastic optimization*, arXiv: 1412.6980, 2017.

85. G. BROCKMAN, V. CHEUNG, L.PETTERSSON, J. SCHNEIDER, J.SCHULMAN, J. TANG, W. ZAREMBA, *OpenAI gym*, arXiv: 1606.01540, 2016.

86. R.S. SUTTON, A.G. BARTO, *Reinforcement Learning: An Introduction*, MIT Press, 2018.

87. OpenAI, Josh Achiam, Part 1: Key concepts in RL, 2024, OpenAI spinning up documentation access in Jan, 2024.

88. R.J. WILLIAMS, *Simple statistical gradient-following algorithms for connectionist reinforcement learning*, Machine Learning, **8**, 229–256, 2004.

89. S. Chandrasekar, *Training using reinforce for mujoco*, 2022, Documentation accessed in March, 2023.

90. C. Watkins, P. Dayan, *Technical note: Q-learning*, Machine Learning, **8**, 279–292, 1992.

91. J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, *Proximal policy optimization algorithms*, CoRR, arXiv: abs/1707.06347, 2017.

Received December 1, 2023; revised version February 2, 2024.
Published online March 6, 2024.