Proceedings of the ASME 2023 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference
IDETC-CIE 2023
August 13 – 16, 2023, Boston, MA

IDETC2023-114999

MACHINE LEARNING-BASED MODEL BIAS CORRECTION BY FUSING CAE DATA WITH TEST DATA FOR VEHICLE CRASHWORTHINESS

Jice Zeng, Ying Zhao
Department of Industrial and Manufacturing Systems Engineering
University of Michigan-Dearborn, Dearborn, MI, 48128

Guosong Li, Zhenyan Gao, Yang Li
Saeed Barbat
Vehicle Structure and Safety Research Department, Research & Advanced Engineering, Ford Motor Company, Dearborn, MI 48126, USA

Zhen Hu*
Department of Industrial and Manufacturing Systems Engineering
University of Michigan-Dearborn, Dearborn, MI, 48128

ABSTRACT

Physics-based simulation and analysis have emerged as promising techniques for optimizing the number of physical prototypes for vehicle crashworthiness evaluation in frontal impact with rigid barriers. Nonetheless, one of the hurdles for vehicle crashworthiness virtual certification is the potential differences between the computer simulation predictions and physical test results. In this regard, this study aims at improving the prediction capability of the Computer-Aided Engineering (CAE) model for crashworthiness performance evaluation at speeds beyond those defined by current regulations and public domain testing protocols. One way of achieving this is by integrating data from a number of physical crash tests with the CAE data using machine learning models. A novel approach is proposed in the displacement domain (deceleration vs. displacement) to enable data fusion to help recover missing physics associated with the CAE model. A nonlinear spring-mass model is used in this study to simulate rigid-barrier vehicle frontal impact. The deceleration response is transformed from a function of time to a function of displacement, and a Gaussian process regression (GPR) model is applied to capture the model bias of the nonlinear spring constant under a dynamic analysis scheme. The training data for the GPR model are split into multiple clusters by a Gaussian mixture model to capture bias patterns under different speed regimes. After clustering a GPR model is trained for each group of data. The optimal GPR model, trained by a specific cluster exhibiting the highest probability of new data belonging to it, is utilized for prediction. This selected GPR model is integrated with the original CAE model to predict vehicle deceleration under a new crash speed during a vehicle deceleration dynamic analysis. The proposed approach is validated using physical vehicle crash tests and demonstrated improved accuracy of CAE model results and predictions.

1. INTRODUCTION

Vehicle design involves evaluations against various attributes loading conditions such as crashworthiness, durability, NVH, styling flexibility, and others. A successful vehicle design hinges on the ability to fulfill various regulatory and non-regulatory attribute requirements. [1]. Designing optimized structures that lead to a controlled deformation remains a primary consideration to achieve structural crashworthiness design to help mitigate serious injuries and fatalities. [2].

Over the past several decades, researchers from academia and major automobile manufacturers have devoted considerable efforts towards optimizing the structural design of vehicles with the aim of improving vehicle crashworthiness. This is typically formulated with a set of design objectives and constraints. These include maximizing the structural energy absorption capability and controlling the peak deceleration of the vehicle during the collision. To achieve these objectives, researchers have adopted various optimization techniques, such as topology optimization, parametric optimization, and sensitivity analysis, in order to identify effective design solutions. Liu et al. [3] developed a collaborative optimization framework taking advantage of Latin hypercube design and response surface to enhance vehicle crashworthiness during a frontal impact. Wang et al. [4] proposed a reliability-based optimization method to enhance vehicle body crashworthiness, in which the copula function derived by the Bayesian method is

*Corresponding author: 2250 HPEC, University of Michigan-Dearborn, Dearborn, MI 48128, USA, Tel:+1-313-583-6312, Email: zhennhu@umich.edu

integrated to formulate complex parametric functions and probability distribution. Li et al. [5] also performed a six-sigma design optimization method to investigate the electric vehicle design with uncertainty, along with radial basis function and non-dominated sorting genetic algorithm II. Gu et al. [6] undertook a comparative investigation of various multi-objective optimization techniques in the context of frontal crash scenarios, with the aim of elucidating reliability and robustness concerns pertaining to vehicle crashworthiness design.

On the other hand, following the finalization of the design phase, the primary objective is to fulfill the Federal Motor Vehicle Safety Standards (FMVSS) by means of conducting prototype vehicle tests, thereby ensuring that novel vehicle designs align with these stipulations. The certification process for vehicle crashworthiness entails conducting numerous prototype vehicle tests. Recent advancements in high-fidelity computer simulation models, computational mechanics, and multi-body dynamics simulation have led to the emergence of certification by analysis (CBA) as a viable tool for reducing the required number of prototype vehicle tests. CBA involves supporting the certification of structural components through virtual physics-based simulations, such as CAE simulations [7]. Initially proposed in the aerospace sector [8, 9], the concept of CBA has since been extended to the certification of automotive structural components, offering an efficient way for vehicle manufacturers to develop new vehicles using a limited number of prototypes.

High-fidelity CAE simulations of vehicle crashworthiness are crucial in both the design stage and the production stage for improving crashworthiness and CBA. However, the utility of CAE models in vehicle crashworthiness can be constrained due to inaccuracies stemming from the exclusion of certain dynamics, numerical approximations, and assumptions made during the modeling process, as well as the inherently complex nature of vehicle crashes. Addressing the limitations of CAE models presents a continual research challenge, as discrepancies between simulated and real-world crash outcomes can affect structural design optimization during the early stages of vehicle development, as well as impede the adoption of computer-based analysis as a means of reducing the number of prototype vehicle tests in the post-design stage. Sub-optimal CAE predictions can give rise to additional prototypes and elongated development time.

In recent years, scholars have devised a variety of methods aimed at enhancing the precision of CAE simulations of crashworthiness, in order to more closely approximate the responses of real-world vehicle crash tests. Shi and Lin [10] developed an adaptive response surface and Gaussian process for model bias correction, first-order score function was also used to measure the sensitivities of variables to responses. Wang and Shi [11] introduced a Gaussian process regression model to capture a bias between model predictions and tested responses for vehicle crashworthiness design. Xi et al. [12] presented a copula-based bias correction method bypassing the dimensionality issue, in which copulas are used to formulate the statistical relations among model bias, model responses, and all design parameters. Despite making progress, current methods for correlating CAE crash models can still face limitations. One of the main challenges is the requirement for sufficient test data to ensure effective model correlation. Furthermore, many of the current methods are limited to correcting biases in CAE surrogate models using high-fidelity CAE simulations, rather than actual tests.

The objective of this paper is to address the limitations of existing methods in achieving a correlation between vehicle crash tests and high-fidelity CAE simulations. The ultimate goal is to facilitate model-based structural design for the improvement of crashworthiness and enable a reduction in the number of required crash tests through CBA. The rigid-barrier vehicle frontal impact in this paper is modeled as a spring-mass system with a nonlinear spring constant. The deceleration response is then transformed from the time domain (a function of time) to the displacement domain (deceleration as a function of displacement). A novel approach is proposed in the displacement domain (deceleration vs. displacement) to enable data fusion of the crash test data and CAE data for the recovery of the unmodeled physics in the CAE model using machine learning models. In order to collect data for the training of machine learning models, the model bias of the nonlinear spring constant is analyzed, by comparing the displacement responses of physical tests with their counterparts from CAE simulations under different test conditions (e.g., speed, and vehicle configurations). Since the model bias may vary with speed and vehicle parameters, a Gaussian mixture model is employed to split the obtained training data of model bias into multiple clusters. A GPR model is then constructed for each cluster of the training data. In order to integrate the GPR models of different clusters with the CAE model under a dynamic analysis scheme, for any given new inputs (e.g., speed, vehicle parameters), the probabilities that the new inputs belong to the clusters are first estimated. Based on the estimated probabilities, the predictions of the GPR models of the model bias are ensembled together using a weighted sum to be the predicted bias of the CAE model prediction of those inputs. This process repeats recursively over iterations in the displacement domain to dynamically recover the missing physics of the CAE model, and thus improve the prediction accuracy of the CAE model for speeds beyond those defined by current regulations and public domain testing protocols. The proposed approach is validated using physical vehicle crash tests and demonstrated its effectiveness.

The remainder of this paper is organized as follows: Section 2 briefly introduces the background of CAE model bias correction for a vehicle crash. Section 3 presents the proposed approach in the displacement domain for the fusion of CAE data with test data. An actual vehicle crash test is used to demonstrate the effectiveness of the proposed approach in Section 4, followed by conclusions in Section 5.

2. BACKGROUND

There are several reasons why the CAE model may vary from a real-world crash test. One potential reason is that a CAE

model is typically constructed based on existing knowledge of physical systems, which may involve subjective assumptions and simplifications. As a result, the reliability of a CAE model is heavily dependent on the engineer's level of understanding of the actual system. Another possible explanation is that the model parameters used in the current version of the CAE model may have worked well for previous tests but may fail to capture the information from new tests. This discrepancy between the CAE model and the actual test is commonly referred to as model bias or model uncertainty [13]. A substantial body of literature has emerged exploring the underlying factors that contribute to model uncertainty, as well as the mathematical methods used to characterize it [14-18]. Kennedy and O'Hagan [19, 20] proposed a mathematical model that is widely used to correlate CAE models with test data. The model is expressed as follows:

$$y_o(\mathbf{x}) = \rho g(\mathbf{x}, \boldsymbol{\theta}) + \delta(\mathbf{x}) + \varepsilon, \quad (1)$$

where $y_o(\mathbf{x})$ represents actual observations that are obtained for a specific set of inputs $\mathbf{x}$, while $y = g(\mathbf{x}, \boldsymbol{\theta})$ represents CAE model, which takes a set of inputs $\mathbf{x}$ and model parameters $\boldsymbol{\theta}$ as its inputs. The term ε represents measurement error, which is typically assumed to be Gaussian noise and independent from $\mathbf{x}$ and $\boldsymbol{\theta}$. Additionally, ρ is an unknown regression coefficient. The term $\delta(\mathbf{x})$ represents model bias, which can be caused by uncertainties in the model parameters or simplifications in the CAE model. It should be noted that the key difference between $\mathbf{x}$ and $\boldsymbol{\theta}$ is that $\mathbf{x}$ can represent controllable variables that execute different operation configurations, such as excitation or size and shape of a product, while $\boldsymbol{\theta}$ is a vector of model parameters that are random and uncontrollable, such as material properties, structural mass, and stiffness.

In this paper, a special form of Eq. (1) is employed by setting $\rho = 1$ and focusing on solely correcting the model bias. Therefore, the formulation in Eq. (1) is rewritten as

$$y_o(\mathbf{x}) = g(\mathbf{x}) + \delta(\mathbf{x}) + \varepsilon. \quad (2)$$

While the model correction formulation presented in Eq. (2) may appear straightforward, implementing it to correct a CAE model of vehicle crashworthiness is more complicated than suggested by the equation. In Section 3, we explore the proposed approach for correlating CAE and test data in the displacement domain.

3. PROPOSED APPROACH

3.1. Definitions of CAE model and variables

For the convenience, the model and variables in the proposed approach are defined. The deceleration during a vehicle crash is denoted as $a_d(t)$. In a discrete-time form, the deceleration at time instant t_i is represented as $a_{d,i}, i = 1, \cdots, N_t$, where N_t is the total number of time steps. In this study, initial crash speed v_0, vehicle front weight and rear weight (w_f, w_r) are considered as all input variables $\mathbf{x} = [v_0, w_f, w_r] \in \mathbb{R}^3$. Based on the definitions, the CAE model for crashworthiness analysis is given by

$$\mathbf{a}_s = G_s(\mathbf{x}) = G_s(v_0, w_f, w_r), \quad (3)$$

where $G_s(\mathbf{x})$ represents the CAE model with inputs $\mathbf{x}$, and $\mathbf{a}_s = [a_{s,1}, a_{s,2}, \cdots, a_{s,N_t}] \in \mathbb{R}^{N_t}$ are the CAE model-derived deceleration.

Assume that N_e groups of CAE simulation data are generated and N_c groups of experimental deceleration data are collected ($N_c \ll N_e$), we then have

- **CAE data**: $\mathbf{a}_s^{(j)}, j = 1, \cdots, N_e$, where $\mathbf{a}_s^{(j)} = [a_{s,1}^{(j)}, a_{s,2}^{(j)}, \cdots, a_{s,N_t}^{(j)}] = G_s(v_0^{(j)}, w_f^{(j)}, w_r^{(j)})$ are deceleration data generated using the *j*-th training sample $\mathbf{x}^{(j)} = [v_0^{(j)}, w_f^{(j)}, w_r^{(j)}], \forall j = 1, \cdots, N_e$ of the input variables.
- **Test data**: $\mathbf{a}_d^{(k)}, k = 1, \cdots, N_c$, where $\mathbf{a}_d^{(k)} = [a_{d,1}^{(k)}, a_{d,2}^{(k)}, \cdots, a_{d,N_t}^{(k)}]$ are deceleration data measured at the *k*-th crash test under testing condition $\mathbf{x}^{(k)} = [v_0^{(k)}, w_f^{(k)}, w_r^{(k)}], \forall k = 1, \cdots, N_c$.

3.2. Data fusion and model bias correction

In this section, we elaborately introduce how the CAE data and test data can be fused in the displacement domain with the purposed of model bias correction and improving the accuracy of model prediction. The vehicle crash test is assumed to be modeled as a spring-mass system, as illustrated in Fig. 1. The equation of motion for vehicle crash in Fig. 1 can be expressed based on system dynamics as follows

$$M \frac{dv}{dt} = -K(s)s(t), \quad (4)$$

where M is the vehicle mass; v is the velocity of the vehicle at the time t; $K(s)$ is a non-linear spring constant that varies with displacement s, and $s(t)$ is a displacement function relative to t.

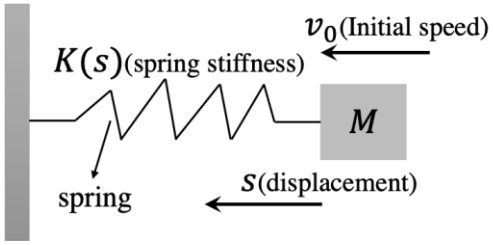


Fig. 1. Spring-mass model for vehicle

When it comes to solving differential equation (4), numerical integration is a commonly used approach to estimate dynamic properties such as displacement, deceleration, and velocity. This is achieved by setting the initial displacement and deceleration to zero, while specifying an initial velocity value

v_0. Assuming Δt is a time moment at a certain velocity or deceleration, N_t is the total number of time steps, we have

$$\begin{aligned} &i=0, s_0=0, v_0=v_0, a_0=0,\\ &\text{For } i=0 \text{ to } N_t:\\ &\quad i=i+1,\\ &\quad s_{i+1}=s_i+v_i\Delta t,\\ &\quad v_{i+1}=v_i+a_i\Delta t,\\ &\quad a_{i+1}=\underbrace{-\frac{1}{M}K(s_i)s_i}_{\text{Unknow or unable to model accurately}}, \end{aligned} \tag{5}$$

As evident from Eq. (5), the deceleration of the vehicle is influenced by a highly nonlinear parameter which poses a significant challenge in accurately determining its value through theoretical analysis. This challenge stems from the intricate nature of the vehicle crash, rendering it difficult to obtain the dynamic properties via solving differential equations directly. To address this challenge, we harness the powerful capabilities of both CAE simulation and machine learning models to capture the unknown function of the nonlinear spring constant with respect to displacement. By leveraging these techniques, we can more accurately model the behavior of the vehicle and ultimately enhance our understanding of its dynamic properties. a_{i+1} in Eq. (5) is estimated as

$$a_{i+1}=G_a(s_i, v_0, w_f, w_r)+\delta(s_i, v_0, w_f, w_r), \tag{6}$$

where $G_a(s_i, v_0, w_f, w_r)$ represents predicted deceleration by a CAE model given inputs s_i and v_0, w_f, w_r. $\delta(s_i, v_0, w_f, w_r)$ represents a machine learning model with the aim of accounting for unmodeled deceleration with respect to displacement, which is not captured by CAE model.

To properly train a machine learning model $\delta(s_i, v_0, w_f, w_r)$, the identification of the appropriate training data in the displacement domain is required. In order to do so, we must initially convert the time-domain data (accleration vs. time) into displacement domain data (displacement vs. accleration), which will then allow us to gather the necessary training data required for the successful training of $\delta(s_i, v_0, w_f, w_r)$. For the k-th crash test, where $k=1,\cdots,N_c$, the test condition with input variables $\mathbf{x}^{(k)}=[v_0^{(k)}, w_f^{(k)}, w_r^{(k)}]$ results in deceleration rsponse $\mathbf{a}_d^{(k)}=[a_{d,1}^{(k)}, a_{d,2}^{(k)}, \cdots, a_{d,N_t}^{(k)}], \forall k=1,\cdots,N_c$. Assume the initial velocity $v_{d,0}^{(k)}=v_0^{(k)}$ for the k-th crash test, the displacement domain data can be derived from structural dynamics

$$\begin{aligned} &s_{d,0}^{(k)}=0; v_{d,0}^{(k)}=v_0^{(k)};\\ &\text{For } i=1 \text{ to } N_t:\\ &\quad s_{d,i}^{(k)}=s_{d,i-1}^{(k)}+v_{d,i-1}^{(k)}\Delta t,\\ &\quad v_{d,i}^{(k)}=v_{d,i-1}^{(k)}+a_{d,i-1}^{(k)}\Delta t, \end{aligned} \tag{7}$$

where $s_{d,i}^{(k)}, i=1,\cdots,N_t$ is the displacement of the k-th crash test at time step t_i. Through the Eq. (7), the time domain data can be readily converted into displacement domain data, as follows

$$\begin{aligned} &\mathbf{s}_d^{(k)}=[s_{d,1}^{(k)},\cdots,s_{d,N_t}^{(k)}]; \mathbf{v}_d^{(k)}=[v_{d,0}^{(k)}, v_{d,1}^{(k)},\cdots,v_{d,N_t}^{(k)}];\\ &\mathbf{a}_d^{(k)}=[a_{d,1}^{(k)},\cdots,a_{d,N_t}^{(k)}]; \forall k=1,\cdots,N_c. \end{aligned} \tag{8}$$

Similarly, taking $\mathbf{x}^{(k)}=[v_0^{(k)}, w_f^{(k)}, w_r^{(k)}]$ as inputs in CAE model, the deceleration response $\mathbf{a}_s^{(k)}=G_s(\mathbf{x}^{(k)})=G_s(v_0^{(k)}, w_f^{(k)}, w_r^{(k)})$ for the k-th crash test in CAE model is obtained, corresponding displacement domain data can be derived using the same process in Eq. (7)

$$\begin{aligned} &\mathbf{s}_s^{(k)}=[s_{s,0}^{(k)}, s_{s,1}^{(k)},\cdots, s_{s,N_t}^{(k)}]; \mathbf{v}_s^{(k)}=[v_{s,0}^{(k)}, v_{s,1}^{(k)},\cdots,v_{s,N_t}^{(k)}];\\ &\mathbf{a}_s^{(k)}=[a_{s,1}^{(k)},\cdots,a_{s,N_t}^{(k)}]; \forall k=1,\cdots,N_c. \end{aligned} \tag{9}$$

It is important to emphasize that in the time domain, where data points are plotted as a function of time versus deceleration, there exists a one-to-one mapping relationship between the data points obtained through both the CAE simulation and actual testing methods, during the same time period. However, when the time domain is converted to the displacement domain, where data points are plotted as a function of displacement versus deceleration, this one-to-one mapping relationship may no longer hold, as observed in Fig. 2.

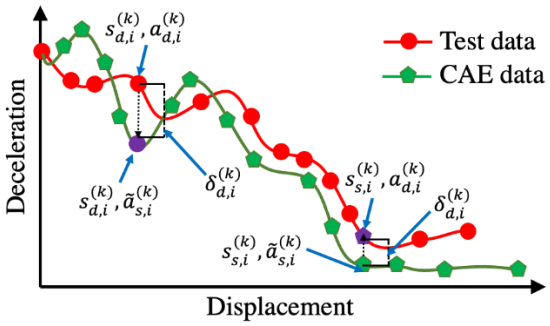


Fig. 2. Model bias analysis

The reason for this is that differential equations, which involve numerical integration with respect to time, can result in variations between the data points for the CAE simulation and actual test. Consequently, it is essential to consider this potential variability when performing data analysis in the displacement domain. Therefore, direct subtraction of the data in Eq. (9) from that in Eq. (8) does not allow us to obtain the necessary training outputs, that is, model bias data, for training the $\delta(s_i, v_0, w_f, w_r)$, such as

$$\delta(s_{s,i}^{(k)}, v_0^{(k)}, w_f^{(k)}, w_r^{(k)}) \neq a_{d,i}^{(k)} - a_{s,i}^{(k)};$$
$$\forall k = 1, \cdots, N_c; i = 1, \cdots, N_t. \quad (10)$$

A bijective projection scheme has been devised to establish a precise correspondence in the displacement domain and effectively account for any inherent model inaccuracies between CAE predictions and actual tests. To achieve this, two distinct scenarios have been explored based on the maximum displacement in CAE and test data, as depicted in Fig. 2: (1) projecting from CAE predictions to actual tests; and (2) projecting from actual tests to CAE predictions. Both of these scenarios enable a bijective mapping to be established, facilitating a more accurate and reliable comparison between simulation results and physical measurements.

For the scenario (1) where the maximum displacement in CAE data is larger than that in test data, CAE data is used for interpolation. For the *k*-th crash test at time step t_i , the estimated deceleration by using CAE data is given by

$$\tilde{a}_{s,i}^{(k)} = f_{\text{int}}(s_{d,i}^{(k)}; \mathbf{s}_s^{(k)}, \mathbf{a}_s^{(k)}), \forall k = 1, \cdots, N_c; i = 1, \cdots, N_t, \quad (11)$$

where $f_{\text{int}}(s_{d,i}^{(k)}; \mathbf{s}_s^{(k)}, \mathbf{a}_s^{(k)})$ is a function to interpolate $s_{d,i}^{(k)}$ based on data $\mathbf{s}_s^{(k)}, \mathbf{a}_s^{(k)}$ given in Eq. (9). The model bias therefor is calculated through simple mathematical operation that subtracts $\tilde{a}_{s,i}^{(k)}$ from corresponding actural test $a_{d,i}^{(k)}$, that is, $\delta_{d,i}^{(k)} = a_{d,i}^{(k)} - \tilde{a}_{s,i}^{(k)}$.

Similarly, model bias for all $s_{d,i}^{(k)}$ from the N_c groups of test data are obtained. Finally, all inputs and outputs (model bias) for training in scenario (1) are summarized as

$$[\mathbf{x}_d; \boldsymbol{\delta}_d] = \begin{cases} \text{Inputs}: \mathbf{x}_{d,i}^{(k)} = [s_{d,i}^{(k)}, v_0^{(k)}, w_f^{(k)}, w_r^{(k)}] \\ \text{Bias}: \delta_{d,i}^{(k)}, \end{cases}, \quad (12)$$
$$\forall k = 1, \cdots, N_c; i = 1, \cdots, N_t.$$

For the scenario (2) where the maximum displacement in test data is larger than that in CAE data, test data is used for interpolation. The deceleration corresponding to $x_{s,i}^{(k)}$ is estimated as

$$\tilde{a}_{d,i}^{(k)} = f_{\text{int}}(s_{s,i}^{(k)}; \mathbf{x}_d^{(k)}, \mathbf{a}_d^{(k)}), \quad (13)$$

where $f_{\text{int}}(s_{s,i}^{(k)}; \mathbf{x}_d^{(k)}, \mathbf{a}_d^{(k)})$ is a function to interpolate for CAE deceleration at $x_{s,i}^{(k)}$ using test data $\mathbf{x}_d^{(k)}, \mathbf{a}_d^{(k)}$, resulting in model bias corresponding to $x_{s,i}^{(k)}$, given by $\tilde{\delta}_{d,i}^{(i)} = \tilde{a}_{d,i}^{(k)} - a_{s,i}^{(k)}$. From calculated model bias, we have all inputs and bias outputs for $s_{s,i}^{(k)}$ $\forall k = 1, \cdots, N_c; i = 1, \cdots, N_t$ as

$$[\mathbf{x}_s; \boldsymbol{\delta}_s] = \begin{cases} \text{Inputs}: \mathbf{x}_{s,i}^{(k)} = [s_{s,i}^{(k)}, v_0^{(k)}, w_f^{(k)}, w_r^{(k)}] \\ \text{Bias}: \tilde{\delta}_{d,i}^{(k)}, \end{cases}, \quad (14)$$
$$\forall k = 1, \cdots, N_c; i = 1, \cdots, N_t.$$

Once the interpolation for two scenarios is complete, all inputs and outputs are combined for training machine learning model $\delta(s_i, v_0, w_f, w_r)$ as

$$\mathbf{x}_{\delta,t} = \{\mathbf{x}_d \cup \mathbf{x}_s\},$$
$$\boldsymbol{\delta}_t = \{\boldsymbol{\delta}_d \cup \boldsymbol{\delta}_s\}. \quad (15)$$

Upon gathering the necessary training data comprising all relevant inputs and model bias outputs, a machine learning model may be developed to accurately capture the intricate relationships between the bias of deceleration and relevant input variables such as displacement, vehicle weights, and initial crash speed. In this study, Gaussian Process Regression (GPR) is utilized to model the unknown relationship between deceleration bias and inputs in the displacement domain.

In GPR, the goal is to estimate the unknown parameters of the Gaussian process that best fit the training data. Given a set of training data points represented by inputs $\mathbf{x}_\delta = [s, v_0, w_f, w_r]$ and outputs $y_\delta = \delta(s, v_0, w_f, w_r)$ approximated by a Gaussian process $\hat{G}_\delta(\mathbf{x}_\delta)$, the mean and covariance are given by [21]

$$E\left[\hat{G}_\delta(\mathbf{x}_\delta)\right] = \mathbf{h}^T(\mathbf{x}_\delta)\boldsymbol{\beta},$$
$$Cov\left[\hat{G}_\delta(\mathbf{x}_\delta), \hat{G}_\delta(\mathbf{x}'_\delta)\right] = \sigma^2 R(\mathbf{x}_\delta, \mathbf{x}'_\delta), \quad (16)$$

where the term of $\mathbf{h}^T(\mathbf{x}_\delta)$ denotes a vector of known regression functions, such as constant or linear functions, that are used to approximate the relationship between the input variables and the output variables. The column vector $\boldsymbol{\beta}$ represents the coefficients related to the polynomial regression of $\mathbf{h}^T(\mathbf{x}_\delta)$ and is used to estimate the unknown parameters of the model. The constant σ^2, commonly referred to as the error term, captures the variability of the response that is not explained by the predictor or the regression functions. The function $R(\mathbf{x}_\delta, \mathbf{x}'_\delta)$ denotes the correlation between the responses at two points $\mathbf{x}_\delta$ and $\mathbf{x}'_\delta$, which is typically modeled using a kernel function that captures the similarity between the inpus.

$R(\mathbf{x}_\delta, \mathbf{x}'_\delta)$ is widely defined as Gaussian kernel function as follows [22]

$$R(\mathbf{x}_\delta, \mathbf{x}'_\delta) = \prod_{i=1}^{4} \exp\left[\rho_i (x_{\delta,i} - x'_{\delta,i})^2\right], \quad (17)$$

where the vector $\boldsymbol{\rho} = [\rho_1, \rho_2, \rho_3, \rho_4]^T$ represents a set of correlation parameters that quantify the change in correlation between $\hat{G}_\delta(\mathbf{x}_\delta)$ and $\hat{G}_\delta(\mathbf{x}'_\delta)$ as the difference between the corresponding input variables $\mathbf{x}_\delta$ and $\mathbf{x}'_\delta$ increases. The hyperparameters $\boldsymbol{\beta}$, σ^2 , and $\boldsymbol{\rho}$ determine the overall scale, noise level, and length scale of the Gaussian process, respectively. These hyperparameters, along with the mean function and the covariance function, fully specify the GPR model and can estimated by the maximum likelihood estimation (MLE) method [23], which involve finding the values that maximize the likelihood of observing the data given the model. Once the hyperparameters are estimated, the GPR model can be used to predict the response values at new input locations and

quantify the uncertainty in the predictions using the covariance function.

A Gaussian process with mean $\mathbf{h}^T(\mathbf{x}_\delta)\boldsymbol{\beta}$ and covariance matrix $\sigma^2 R(\mathbf{x}_\delta, \mathbf{x}'_\delta)$ is used to describe the $\hat{G}_\delta(\mathbf{x}_\delta)$, the likelihood funtion of $\boldsymbol{\delta}_t$ is written as

$$L_{GP}(\boldsymbol{\beta}, \sigma, \boldsymbol{\rho}) = (2\pi\sigma^2)^{-\frac{N_\delta}{2}} |\mathbf{R}|^{-\frac{1}{2}} \times \exp\left[-\frac{1}{2\sigma^2}(\boldsymbol{\delta}_t - \mathbf{H}\boldsymbol{\beta})^T \mathbf{R}^{-1}(\boldsymbol{\delta}_t - \mathbf{H}\boldsymbol{\beta})\right], \tag{18}$$

where N_δ is the total number of training points. $\mathbf{H} = [\mathbf{h}^T(\mathbf{x}_{\delta,t}^{(1)}), \mathbf{h}^T(\mathbf{x}_{\delta,t}^{(2)}), \cdots, \mathbf{h}^T(\mathbf{x}_{\delta,t}^{(N_\delta)})]^T$, $\mathbf{R}$ is a $N_\delta \times N_\delta$ matrix whose elements are $R(\mathbf{x}_{\delta,t}^{(i)}, \mathbf{x}_{\delta,t}^{(j)}), \forall i, j = 1, \cdots, N_\delta$, in which $\mathbf{x}_{\delta,t}^{(i)}$ is the i-th sample in $\mathbf{x}_{\delta,t}$. The optimal hyperparameters $\hat{\boldsymbol{\beta}}$, $\hat{\sigma}^2$, and $\hat{\boldsymbol{\rho}}$ can be obtained through the Eq. (18). The prediction at new points $\mathbf{x}_\delta^{new} = [s^{new}, v_0^{new}, w_f^{new}, w_r^{new}]$ by trained GPR model is given by

$$\hat{G}_\delta(\mathbf{x}_\delta^{new}) \sim N(\mu_\delta(\mathbf{x}_\delta^{new}), \sigma_\delta^2(\mathbf{x}_\delta^{new})), \tag{19}$$

where $N(\cdot,\cdot)$ is a standard Gaussian distribution, $\mu_\delta(\mathbf{x}_\delta^{new})$ and $\sigma_\delta^2(\mathbf{x}_\delta^{new})$ are derived as

$$\begin{aligned} \mu_\delta(\mathbf{x}_\delta^{new}) &= \mathbf{h}^T(\mathbf{x}_\delta^{new})\hat{\boldsymbol{\beta}} + \mathbf{r}^T(\mathbf{x}_\delta^{new})\mathbf{R}^{-1}(\boldsymbol{\delta}_t - \mathbf{H}\hat{\boldsymbol{\beta}}), \\ \sigma_\delta^2(\mathbf{x}_\delta^{new}) &= \sigma^2 \begin{Bmatrix} 1 - \mathbf{r}^T(\mathbf{x}_\delta^{new})\mathbf{R}^{-1}\mathbf{r}(\mathbf{x}_\delta^{new}) \\ +[\mathbf{H}^T\mathbf{R}^{-1}\mathbf{r}(\mathbf{x}_\delta^{new}) - \boldsymbol{\delta}_t]^T(\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}) \\ \times[\mathbf{H}^T\mathbf{R}^{-1}\mathbf{r}(\mathbf{x}_\delta^{new}) - \boldsymbol{\delta}_t] \end{Bmatrix}, \end{aligned} \tag{20}$$

where the term of $\mathbf{r}^T(\mathbf{x}_\delta^{new})$ is a $N_\delta \times 1$ vector, of which the i-th element is $R(\mathbf{x}_\delta^{new}, \mathbf{x}_{\delta,t}^{(i)}), \forall i = 1, \cdots, N_\delta$. For more details on GPR method, please refer to the references [24, 25].

Before training a GPR model using available training data, all training data is clustered using Gaussian mixture model (GMM) in order to more explicitly capture data features. Consequently, multiple GPR models are obtained based on each cluster of training data. GMM is a powerful unsupervised learning technique that has found widespread use in a variety of fields [26-28]. GMM assumes that the data is generated by a mixture of several Gaussian distributions, each with a different mean and variance. The goal of the algorithm is to estimate these parameters by maximizing the likelihood of the data under the mixture model. In GMM, each sample is assigned to a group with a probability, allowing for a probabilistic representation of the data. This means that the GMM can not only group data points, but can also quantify the degree of uncertainty in the assignment of each data point to its corresponding group.

The fundamental formulation in GMM is given by

$$p(\mathbf{x}) = \sum_{w=1}^{W} \pi_w N(\mathbf{x} \mid \boldsymbol{\mu}_w, \boldsymbol{\Sigma}_w), \tag{21}$$

where $\mathbf{x}$ is a variable vector, $p(\cdot)$ is the operator of marginal probability distribution. W is the total number of Gaussian components or clusters, π_w is the k-th mixing coefficient or weights, $\sum_{w=1}^{W} \pi_w = 1$, $\boldsymbol{\mu}_w$ and $\boldsymbol{\Sigma}_w$ are the mean vector and covariance matrix of the k-th cluster. In this study, the variable vector is four-dimensional, including initial vehicle speed, fontal and rear weight and displacement, that is, $\mathbf{x}$ is equal to $\mathbf{x}_\delta = [s, v_0, w_f, w_r]$.

Once the training data is split into multiple groups by GMM, the multiple GPR models are trained using training data in individual group. The optimal GPR model, trained with a specific cluster exhibiting the highest probability of new data $\mathbf{x}_\delta^{new} = [s^{new}, v_0^{new}, w_f^{new}, w_r^{new}]$ belonging to, is used for prediction. This selected GPR model is finally used to approximate the nonlinear function of deceleration in Eq. (6).

$$a = G_a(s, v_0, w_f, w_r) + \hat{G}_\delta(\mathbf{x}_\delta^{new}), \tag{22}$$

The deceleration can be calculated in an iterative way given any new vehicle speed v_0^{new} and new vehicle weights, w_f^{new}, w_r^{new}, based on structural dynamics as follows

$$\begin{aligned} &i = 0, s_i^{new} = 0, v_i^{new} = v_0^{new}, a_i^{new} = 0, \\ &\text{For } i = 1 \text{ to } N_t: \\ &\quad s_i^{new} = s_{i-1}^{new} + v_{i-1}^{new}\Delta t, \\ &\quad \mathbf{x}_\delta^{new} = [s_i^{new}, v_0^{new}, w_f^{new}, w_r^{new}], \\ &\quad a_i^{new} = G_a(s_i^{new}, v_0^{new}, w_f^{new}, w_r^{new}) + \hat{G}_\delta(\mathbf{x}_\delta^{new}), \\ &\quad v_i^{new} = v_{i-1}^{new} + a_i^{new}\Delta t. \end{aligned} \tag{23}$$

It is noteworthy that the quantity of $\hat{G}_\delta(\mathbf{x}_\delta^{new})$ is a random variable, as specified in Eq. (22). To incorporate the associated uncertainty in the prediction, a Monte Carlo simulation method must be employed iteratively in accordance with the dynamic analysis scheme outlined in Eq. (23). Fig. 3 depicts the overall procedures of proposed model bias correction framework. Initially, the vehicle crash is represented using a spring-mass system model. However, due to the nonlinearity of the spring constant, this model fails to capture certain physical phenomena. To overcome this limitation, the deceleration responses with respect to time in both CAE model and actual crash tests are converted to displacement responses. To establish a one-to-one mapping between the two sets of responses and to calculate the model bias between the CAE model and actual tests, a bijective projection is developed. This projection ensures that the transformed responses can be accurately compared and that any discrepancies between the CAE model and actual tests are accounted for. When all training data is available, the GMM is applied to cluster all data with the purpose of capturing explicit data features, leading to multiple GPR models trained by individual cluster. The GPR model with the highest probability of assigning new inputs to a particular cluster is selected for prediction. To start the model prediction, the system's displacement and acceleration are initialized, along with a new speed. At each time step, the resulting displacement is used to

compute a deceleration response without bias correction from CAE model. To account for any discrepancies between the CAE model and actual crash tests, the model bias is estimated using the selected GPR model.

The estimated model bias is subsequently added to the deceleration response, enabling model bias to be corrected. Throughout the prediction process, the displacement, velocity, and acceleration are updated iteratively, reflecting the evolving state of the system. By incorporating this methodology, the resulting model can more accurately predict the behavior of the system in real-world crashes, enhancing the design and testing of vehicle safety measures.

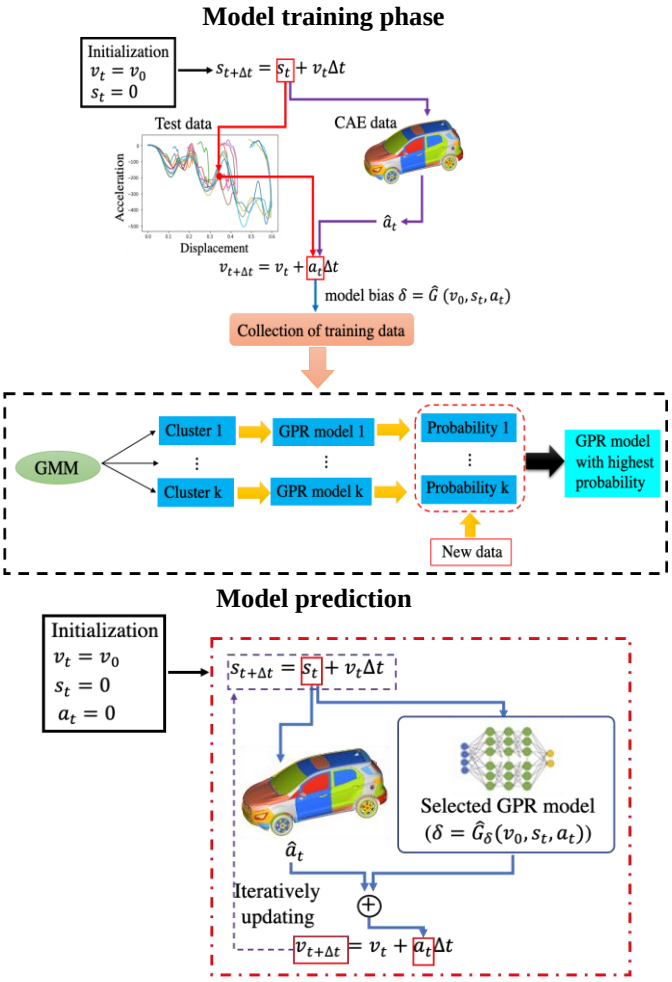


Fig. 3. The workflow of proposed model bias

3.3. Model validation for vehicle crashworthiness

Model validation is for evaluating the degree to which a model accurately represents a real-world system [29]. The selection of appropriate validation metrics is an essential aspect of this process, as they should be able to quantitatively measure the correlation or discrepancy between the model prediction and experimental data. For this study, where we are analyzing time series data, we adopt the ISO (International Organization for Standardization) metric to evaluate the level of agreement between computational and experimental data. The ISO metric is a robust approach that incorporates various metrics to assess the correlation between two time series data reliably [30]. The ISO metric includes several sub-metrics, such as the corridor score, phase score, magnitude score, and slope score. These sub-metrics contribute to an overall ISO metric rating, which can help us objectively evaluate the performance of our model and compare its predictions to experimental data. A comprehensive description of the detailed computation procedures for each score metric and the overall ISO rating can be found in references [30] and [31].

4. CASE STUDY

Fig. 4 shows the physical test and a CAE model for the full-frontal impact of a vehicle. In the current study, our focus is directed towards the prediction of crash pulse or deceleration, as various impact speeds generate distinctive pulses.

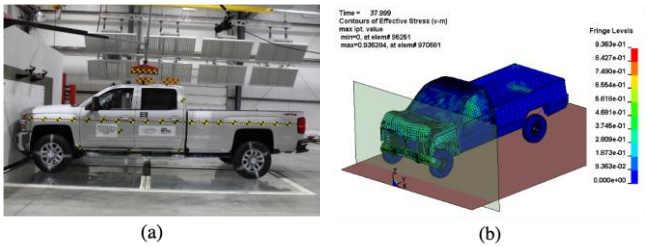


Fig. 4. Vehicle crashworthiness: (a) actual test; (b) CAE model of a vehicle

To implement the proposed approach, the initial step entails the gathering of CAE data obtained from a CAE simulation model. For training inputs, a total of 1009 sets of speed and weights are generated, based on which 1009 sets of pulses are collected from the CAE simulations. The output sequence for each pulse is the vehicle acceleration, with a data duration of 100 ms and a sampling frequency of 12.5 kHz. Only 11 sets of test data for different speeds, weights, and configurations are currently available, as illustrated in Table 1.

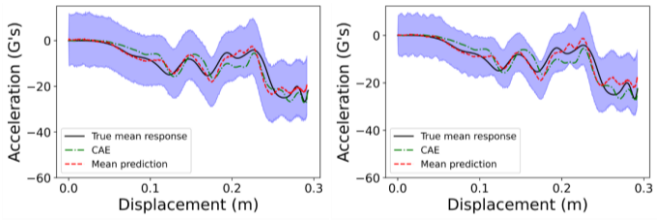
Table 1 Configuration of vehicle-to-rigid barrier full frontal test design

No.	Speed (v_0, mph)	Front Weight (w_f, lbs)	Rear Weight (w_r, lbs)	P/T Configuration	FWD/AWD
1	8.1	1994.7	1400	Type I	AWD
2	12.1	2163.7	1592.9	Type II	AWD
3	16.7	2120.1	1600.2	Type I	FWD
4	22	2237.4	1765.3	Type II	AWD
5	24.9	2100.4	1554.6	Type I	FWD
6	25.2	2155.4	1707.1	Type II	AWD
7	25.2	2058.9	1507.9	Type I	FWD
8	35.2	2179.3	1745.5	Type II	AWD
9	35.1	2107.7	1556.6	Type I	FWD
10	35.2	2166.9	1737.2	Type II	AWD
11	35.2	2168.9	1694.7	Type II	AWD

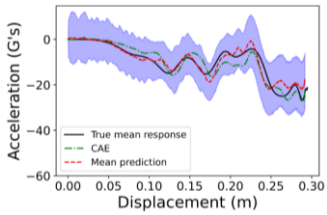
4.1. Results of model bias correction

To ensure a one-to-one correlation between the 10 sets of test data available for training and their corresponding CAE data, a bijective projection method is adopted as discussed above. Notably, it is imperative to highlight that the remaining test data is utilized for validation purposes. Upon the availability of all training data, we apply GMM to categorize the training data sharing similar features. In this study, we explore three scenarios comprising single cluster, two clusters and three clusters. It is important to note that over-specifying the number of clusters can lead to a decrease in the training data associated with each cluster. This, in turn, can adversely affect the accuracy of the trained model. The study then proceeds to train multiple GPR models (i.e., single, two, and three GPR models correspond to three scenarios) based on the training data in each cluster. In the process of predicting new inputs via GPR model, the optimal GPR model for prediction deployment is the one trained with a specific cluster that demonstrates the highest probability of the new data belonging to that cluster.

In this paper, prediction results of test No. 3-5 and No. 10 are presented for the sake of explanation. Figs. 5-8 show the comparison results of model prediction obtained using proposed approach given different GPR models, as well as CAE prediction without bias correction. Since GPR model enables to probabilistically make prediction for crash test, with the 95% confidence interval indicated by the blue shaded area in the accompanying figures.



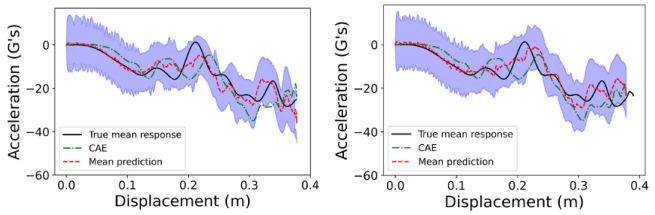
(a) single GPR model (b) two GPR models



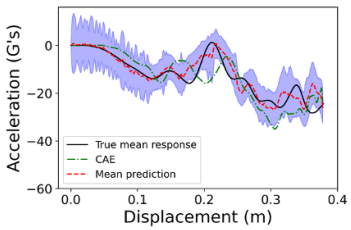
(c) three GPR models

Fig. 5. Model prediction for test #3

The results demonstrate a satisfactory level of agreement between the mean predictions generated by the proposed approach and the actual results from the crash tests for all scenarios, including the use of single, two, and three GPR models. These findings suggest that the bias or discrepancy between the CAE) model and the true system has been substantially reduced. Furthermore, while the proposed approach exhibits a consistent and robust predictive performance across a range of GPR models, the selection of the appropriate number of GPR models is critical for optimizing model prediction accuracy and reducing uncertainty. Specifically, it has been observed that the utilization of a single GPR model, where all available training data are used for model training without clustering, yields predicted pulse curves that are less congruent with actual test data when compared to predictions generated through the use of two or three GPR models. This phenomenon is evidenced in the case of test #4, where the use of adjacent predictions at a distance of 0.2 m is found to be most accurate when leveraging multiple GPR models. Moreover, it is visually apparent that the shaded region identified by more than one GPR model is considerably narrower, especially for tests #4, #5, and #10. This observation indicates a greater level of confidence in the model predictions as opposed to utilizing a single GPR model. The notable improvement in performance achieved by leveraging multiple GPR models can be attributed to the utilization of the GMM, which enables the grouping of training data with comparable features. By clustering data in this manner, the model can more accurately capture the intricacies of the data. In doing so, the selected GPR model is better tailored to predicting responses for new inputs that share similarities with the training data of this selected model.



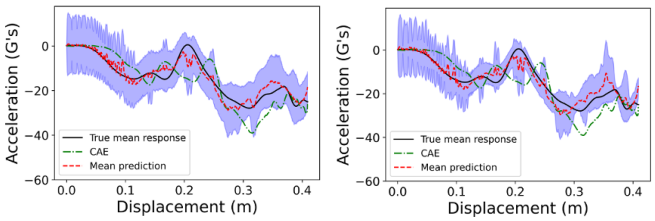
(a) single GPR model (b) two GPR models



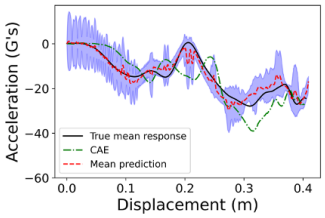
(c) three GPR models

Fig. 6. Model prediction for test #4

It is noteworthy that the prediction of tests #3 and #4 presents an additional challenge. As previously outlined in Table 1, the training data utilized for predicting these tests, which consists of 10 sets of test data excluding #3 or #4, does not contain similar information, such as vehicle speed, to these specific tests. In other words, the data pertaining to the crash test to be predicted falls outside the domain of the training data. Nevertheless, the proposed approach is capable of mitigating any potential model bias by leveraging the correlation between CAE data and test data to improve prediction accuracy.

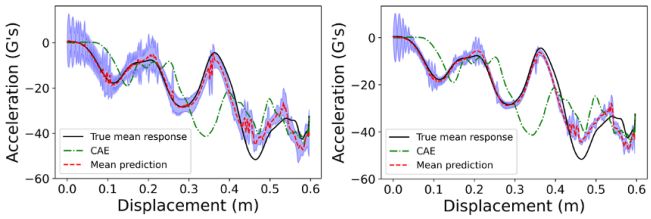


(a) single GPR model (b) two GPR models

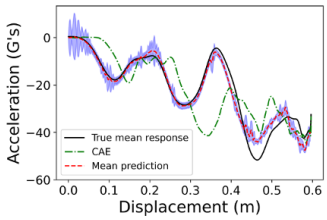


(c) three GPR models

Fig. 7. Model prediction for test #5



(a) single GPR model (b) two GPR models



(c) three GPR models

Fig. 8. Model prediction for test #10

4.2. Quantitative accuracy measurement

Table 2 summarizes the overall ISO score of the proposed model bias approach for three periods, namely 0-20 ms, 0-40 ms, and 0-60 ms using different number of GPR models. The findings suggest that the ISO scores derived from the proposed approach are consistently higher than those obtained from the CAE model across all time periods in the three scenarios. Furthermore, the analysis revealed that employing three Gaussian Process Regression (GPR) models yielded the highest ISO scores, suggesting that the optimal number of clusters in our case is three. Notably, the ISO scores for test #10 were found to be the highest, indicating a higher level of accuracy compared to the other tests. This observation can be explained by the fact that the training data for predicting test #10 also includes relevant information pertaining to the test, such as vehicle speed. Therefore, the superior prediction performance for test #10 is not surprising.

Table 2 Comparison of different number of GPR models by ISO metric

Test No.	Time period	ISO Score (0, 20) ms	ISO Score (0, 40) ms	ISO Score (0, 60) ms
3	CAE	0.774	0.768	0.822
	Single GPR	0.863	0.828	0.822
	Two GPR	0.867	0.839	0.801
	Three GPR	**0.868**	**0.841**	**0.805**
4	CAE	0.651	0.658	0.681
	Single GPR	0.745	0.762	0.730
	Two GPR	0.776	0.772	0.742
	Three GPR	**0.801**	**0.808**	**0.750**
5	CAE	0.477	0.565	0.656
	Single GPR	0.722	0.786	0.743
	Two GPR	0.737	0.764	0.725
	Three GPR	**0.770**	**0.873**	**0.754**
10	CAE	0.647	0.617	0.541
	Single GPR	0.945	0.894	0.875
	Two GPR	0.964	0.919	0.889
	Three GPR	**0.965**	**0.922**	**0.904**

5. CONCLUSION

In this study, a machine learning-based model bias correction approach is proposed in displacement domain to enhance the predictive performance for vehicle crashworthiness by fusion CAE data with test data. In order to address the variation between CAE simulation data and actual test data, a bijective projection method is proposed, which enables the analysis of differences between the CAE model and the actual tests. Upon completion of the training data collection, the GMM is employed to automatically cluster the training data, with each cluster being utilized to train a GPR model. This process yields multiple GPR models based on the identified clusters. The selection of the most suitable GPR model is contingent on its training by a cluster that demonstrates the highest probability of new data inputs belonging to. This selected GPR model is subsequently integrated with the CAE model in a dynamic analysis framework, thereby facilitating the prediction of crash response under a new speed. To validate the effectiveness of the proposed approach, an industrial application of a vehicle crashworthiness case was conducted. The results indicate that the proposed approach outperformed the original CAE model by providing better crash pulse response predictions.

In the present study, the impact of damping on the prediction of vehicle crashworthiness has not been incorporated in the equation of motion, therefore, future research endeavors will focus on the inclusion of damping for vehicle crashworthiness analysis.

ACKNOWLEDGEMENT

Funding for this work was provided by Ford Motor Company through the University Research Program. The support is gratefully acknowledged.

REFERENCES

[1] J. Happian-Smith, *An introduction to modern vehicle design*. Elsevier, 2001.

[2] J. Fang, G. Sun, N. Qiu, N. H. Kim, and Q. Li, "On design optimization for structural crashworthiness and its state of the art," *Structural and Multidisciplinary Optimization*, vol. 55, no. 3, pp. 1091-1119, 2017.

[3] X. Liu, R. Liang, Y. Hu, X. Tang, C. Bastien, and R. Zhang, "Collaborative optimization of vehicle crashworthiness under frontal impacts based on displacement oriented structure," *International journal of automotive technology*, vol. 22, pp. 1319-1335, 2021.

[4] Q. Wang, Z. Huang, and J. Dong, "Reliability-based design optimization for vehicle body crashworthiness based on copula functions," *Engineering Optimization*, vol. 52, no. 8, pp. 1362-1381, 2020.

[5] Z. Li, L. Duan, A. Cheng, Z. Yao, T. Chen, and W. Yao, "Lightweight and crashworthiness design of an electric vehicle using a six-sigma robust design optimization method," *Engineering Optimization*, vol. 51, no. 8, pp. 1393-1411, 2019.

[6] X. Gu, G. Sun, G. Li, L. Mao, and Q. Li, "A comparative study on multiobjective reliable and robust optimization for crashworthiness design of vehicle structure," *Structural and Multidisciplinary Optimization*, vol. 48, no. 3, pp. 669-684, 2013.

[7] M. Guida, A. Manzoni, A. Zuppardi, F. Caputo, F. Marulo, and A. De Luca, "Development of a multibody system for crashworthiness certification of aircraft seat," *Multibody System Dynamics*, vol. 44, no. 2, pp. 191-221, 2018.

[8] F. Caputo, G. Lamanna, D. Perfetto, A. Chiariello, F. Di Caprio, and L. Di Palma, "Experimental and numerical crashworthiness study of a full-scale composite fuselage section," *AIAAJ*, vol. 59, no. 2, pp. 700-718, 2021.

[9] G. Olivares, J. F. Acosta, and V. Yadav, "Certification by Analysis I and II," in *The Joint Advanced Materials and Structures Meeting*, 2010.

[10] L. Shi and S.-P. Lin, "A new RBDO method using adaptive response surface and first-order score function for crashworthiness design," *Reliability Engineering & System Safety*, vol. 156, pp. 125-133, 2016.

[11] X. Wang and L. Shi, "A new metamodel method using Gaussian process based bias function for vehicle crashworthiness design," *International Journal of Crashworthiness*, vol. 19, no. 3, pp. 311-321, 2014.

[12] Z. Xi, P. Hao, Y. Fu, and R.-J. Yang, "A copula-based approach for model bias characterization," *SAE Int. J. Passeng. Cars-Mech. Syst*, vol. 7, no. 2, pp. 781-786, 2014.

[13] C. Jiang, Z. Hu, Y. Liu, Z. P. Mourelatos, D. Gorsich, and P. Jayakumar, "A sequential calibration and validation framework for model uncertainty quantification and reduction," *Computer Methods in Applied Mechanics and Engineering* , vol. 368, p. 113172, 2020/08/15/ 2020.

[14] P. D. Arendt, D. W. Apley, and W. Chen, "Quantification of model uncertainty: Calibration, model discrepancy, and identifiability," 2012.

[15] J. Zeng and Y. H. Kim, "Identification of Structural Stiffness and Mass using Bayesian Model Updating Approach with Known Added Mass: Numerical Investigation," *Journal of Structural Stability and Dynamics*, vol. 20, no. 11, 2020.

[16] J. Zeng, M. D. Todd, and Z. Hu, "Probabilistic damage detection using a new likelihood-free Bayesian inference method," *Journal of Civil Structural Health Monitoring*, pp. 1-23, 2022.

[17] A. Thelen *et al.*, "A comprehensive review of digital twin—part 2: roles of uncertainty quantification and optimization, a battery digital twin, and perspectives," *Structural and multidisciplinary optimization*, vol. 66, no. 1, p. 1, 2023.

[18] J. Zeng and Y. H. Kim, "Probabilistic Damage Detection and Identification of Coupled Structural Parameters using Bayesian Model Updating with Added Mass," *Journal of Sound and Vibration*, p. 117275, 2022.

[19] M. C. Kennedy and A. O'Hagan, "Bayesian calibration of computer models," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 63, no. 3, pp. 425-464, 2001.

[20] D. Higdon, M. Kennedy, J. C. Cavendish, J. A. Cafeo, and R. D. Ryne, "Combining field data and computer simulations for calibration and prediction," *SIAM Journal on Scientific Computing*, vol. 26, no. 2, pp. 448-466, 2004.

[21] E. Schulz, M. Speekenbrink, and A. Krause, "A tutorial on Gaussian process regression: Modelling, exploring, and exploiting functions," *Journal of Mathematical Psychology*, vol. 85, pp. 1-16, 2018.

[22] J. Quinonero-Candela and C. E. Rasmussen, "A unifying view of sparse approximate Gaussian process regression," *The Journal of Machine Learning Research*, vol. 6, pp. 1939-1959, 2005.

[23] J.-S. Park and J. Jeon, "Estimation of input parameters in complex simulation using a Gaussian process metamodel," *Probabilistic engineering mechanics* , vol. 17, no. 3, pp. 219-225, 2002.

[24] A. Sobester, A. Forrester, and A. Keane, *Engineering design via surrogate modelling: a practical guide*. John Wiley & Sons, 2008.

[25] C. E. Rasmussen and H. Nickisch, "Gaussian processes for machine learning (GPML) toolbox," *The Journal of Machine Learning Research*, vol. 11, pp. 3011-3015, 2010.

[26] G. J. McLachlan and S. Rathnayake, "On the number of components in a Gaussian mixture model," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 4, no. 5, pp. 341-355, 2014.

[27] C. Fraley, A. E. Raftery, T. B. Murphy, and L. Scrucca, "mclust version 4 for R: normal mixture modeling for

model-based clustering, classification, and density estimation," Technical report, 2012.
[28] J. Zeng and Z. Hu, "Automated operational modal analysis using variational Gaussian mixture model," *Engineering Structures*, vol. 273, p. 115139, 2022.
[29] L. E. Schwer, "An overview of the PTC 60/V&V 10: guide for verification and validation in computational solid mechanics," *Engineering with Computers*, vol. 23, no. 4, pp. 245-252, 2007.
[30] S. Barbat, Y. Fu, Z. Zhan, R.-J. Yang, and C. Gehre, "Objective rating metric for dynamic systems," *Enhanced Safety of Vehicles, Seoul, Republic of Korea*, vol. 2, no. 3, 2013.
[31] I. O. f. Standardization, "Road Vehicles—Objective Rating Metric for Non-Ambiguous Signals," ed: ISO/TS 18571: 2014, 2014.