Vehicle Crash Simulation Models for Reinforcement Learning driven crash-detection algorithm calibration

Shahabaz Afraj (shabaz.afraj@gmail.com)
Technische Hochschule Ingolstadt https://orcid.org/0009-0003-8057-7407
Ondřej Vaculín
Technische Hochschule Ingolstadt
Dennis Böhmländer
Audi AG
Luděk Hynčík
ZCU v Plzni: Zapadoceska univerzita v Plzni

Research Article

Keywords: Virtual vehicle models, Crash tests, Crash simulations, Surrogate model, Crash-detection algorithm, Reinforcement Learning

Posted Date: June 29th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-3004299/v1

Title: Vehicle Crash Simulation Models for Reinforcement Learning driven crash-detection algorithm calibration

Authors:

Name: Shahabaz Afraj*

Contact: shahabaz.afraj@thi.de

Name: Ondrej Vaculin

Contact: Ondrej.Vaculin@thi.de

Name: Dennis Böhmländer

Contact: dennis.boehmlaender@audi.de

Name: Ludek Hyncik

Contact: hyncik@ntc.zcu.cz

*correspondence author

RESEARCH

Vehicle Crash Simulation Models for Reinforcement Learning driven crash-detection algorithm calibration

Shahabaz Afraj[1*], Ondřej Vaculín[1], Dennis Böhmländer[2] and Luděk Hynčík[3]

[*]Correspondence:
Shahabaz.Afraj@thi.de
[1]CARISSMA Institute of Safety in Future Mobility (C-ISAFE), Technische Hochschule Ingolstadt, Ingolstadt, Germany
Full list of author information is available at the end of the article

Abstract

The development of finite element vehicle models for crash simulations is a highly complex task. The main aim of these models is to simulate a variety of crash scenarios and assess all the safety systems for their respective performances. These vehicle models possess a substantial amount of data pertaining to the vehicle's geometry, structure, materials, etc., and are used to estimate a large set of system and component level characteristics using crash simulations. It is understood that even the most well-developed simulation models are prone to deviations in estimation when compared to real-world physical test results. This is generally due to our inability to model the chaos and uncertainties introduced in the real world. Such unavoidable deviations render the use of virtual simulations ineffective for the calibration process of the algorithms that activate the restraint systems in the event of a crash (crash-detection algorithm). In the scope of this research, authors hypothesize the possibility of accounting for such variations introduced in the real world by creating a feedback loop between real-world crash tests and crash simulations. To accomplish this, a Reinforcement Learning (RL) compatible virtual surrogate model is used, which is adapted from crash simulation models. Hence, a conceptual methodology is illustrated in this paper for developing an RL-compatible model that can be trained using the results of crash simulations and crash tests. As the calibration of the crash-detection algorithm is fundamentally dependent upon the crash pulses, the scope of the expected output is limited to advancing the ability to estimate crash pulses. Furthermore, the real-time implementation of the methodology is illustrated using an actual vehicle model.

Keywords: Virtual vehicle models; Crash tests; Crash simulations; Surrogate model; Crash-detection algorithm; Reinforcement Learning

Introduction

Developing passive safety systems for automotive applications is a vital and challenging task because it directly affects the safety of occupants during crucial events. In the event of a crash, to ensure the optimal safety performance of a new vehicle under development, it is necessary to subject every system to a comprehensive process of design evaluation, testing, and validation. The main aim is to optimize the system to reduce the impact of a collision on vehicle occupants. It can be achieved by designing the vehicle's structure to efficiently dissipate the crash energy and implementing systems that, when activated at the appropriate time, mitigate the impact forces on the occupants' bodies and prevent them from hard impacts with

interior parts or from ejecting out of the vehicle. Such cases need precise engineering design considerations and system-level computations.

The interaction of the crash loads with the vehicle structure mainly drives the crash dynamics and deformation behavior of the vehicle. The crash dynamics and deformation behavior are the complex system-level characteristics of the vehicle model which a series of virtual and physical crash tests must determine. Logic dictates that these tests must be designed such that the crash behavior is evaluated at every statistically significant and system-critical crash scenario. The obtained test results are measured with the help of various sensors and data is analyzed in different approaches depending on its aim and scope. For instance, the rate of deformation, crumpling of the vehicle body, etc can be deduced from strain gauges, and target tracking using a photogrammetry approach [1, 2], etc., the deceleration behavior can be recorded using accelerometers [3], the vehicle's global motion in translation, yaw, pitch, and roll can be determined by gyroscopic sensors and tracking using photogrammetry approach, etc. With advances in virtual modeling, stochastic analytical methodologies, and access to significantly extensive computational capabilities, there is a big advantage in increasing the dependency of passive safety development on virtual and data-driven techniques. There have been explorations in this direction by multiple approaches like big-data-based rule-finding analysis to establish IF-THEN format causality relationships in crash simulation results [4]. Machine learning approaches for component level deformation mode analysis using Long Short Term Memory (LSTM) Autoencoders are established in [5]. A recurrent neural network-based AI application is established by the authors for a non-iterative process to evaluate the behavior of finite element structures [6]. Despite this, it must be understood that variations from real-world tests may occur in the simulated and virtual results due to various reasons. These variations can hinder the full use of virtual methods in applications such as the calibration of crash-detection algorithms, a key focus of this research paper. The onboard, pre-calibrated algorithms that are capable of detecting and classifying a car crash and triggering the right combinations of restraint systems are hereby referred to as crash-detection algorithms.

Virtual models used for crash simulations are designed to compute complex crash behavior including deformations, stresses, energy dissipation, and overall vehicle kinematics. Therefore, the model must be defined very precisely and in great detail in order to accurately compute such a wide variety of results. However, when it comes to the calibration process of a crash-detection algorithm, it is only dependent on one output parameter: the acceleration crash pulse. While estimating crash pulses for a given crash scenario can be done straightforwardly using Finite Element Analysis, variations may occur between the crash pulses generated by simulation and those from real-world crash tests. A crash pulse prediction technique using regression analysis as presented in [7] could be a good alternative, but it is limited by the scope and quality of training data. If the calibration is done only using simulation results, these variations can lead to the misclassification of crashes and result in the misactivation of restraint systems (e.g. delayed triggering times for airbags). In an ideal world, one possible solution would be to analyze the root cause of such variations, improve the simulation models to be a more accurate representation of a real-world vehicle under test (VUT) and re-simulate all the crash pulses

to calibrate the crash-detection and classification algorithm. However, the process of establishing the root cause, changing the models, and validating the results in the current architecture, would be extremely time-consuming and would require a lot of computational power. Within the scope of this research, the authors propose a methodology to generate an optimally simplified virtual vehicle model that specifically targets the computation of one output parameter (i.e., crash pulse in frontal crash scenarios) and its mathematical dependencies, such that an AI-based reinforcement learning algorithm can be used to fine-tune the said model in order to improve the accuracy of its behavior using crash test observations of VUT. A calibrated model of this nature could be highly advantageous in increasing the range of scenarios that can be tested and evaluated for validating the crash-detection algorithm, as a favorable accuracy to computational cost could be achieved.

Methodology

The basic framework of the reinforcement learning technique is as shown in Fig. 1. There are four fundamental components that are necessary to adapt the vehicle crash simulation models into models that are compatible with reinforcement learning-based AI analysis:

1. Defining the environment: Definition of a simplified environment with goal-specific and minimum relevant dimensionality for an appropriate representation of the vehicle crash structure that allows for fast re-iterations.
2. State space modeling: Modelling of a system to evaluate the state of the environment due to policy-based actions.
3. Action space modeling: Modeling the possible set of actions that an agent can make to strategically alter the state of the environment.
4. Defining the reward function: Definition of a crash pulse-based reward function to evaluate the new state of the environment.

Figure 1 Fundamental framework of a reinforcement learning AI application

Defining the environment

The environment is a rule-based and (partially) deterministic formulation of a system, meaning that it follows a set of predetermined rules, and the outcomes of actions are calculated based on these rules [8]. The goal is to formulate the vehicle structure as an environment with independent physical parameters that can influence the crash pulse computation. Crash pulse is the measure of deceleration of the vehicle during a crash and the ultimate structural parameter that influences the crash pulse is the stiffness of the deforming car body. The stiffness of the car body in the event of a crash is studied deeply for analysis and reconstruction of accidents [9, 10] with considerations for the potential nonlinearities in the behavior [11].

As the scope of this research is limited to frontal crash scenarios, the methodology, and results are demonstrated using crash data from simulations and physical crash tests of these scenarios. To filter the region of interest, the car body can be split into two zones namely the passenger compartment and the frontal vehicle region as shown in Fig. 2 (a). The design principle of these zones is based upon

the strategy of mitigating the impacts on occupants. Hence, the frontal vehicle region is designed to deform and absorb the crash impact energies, and the passenger compartment is designed to stay undeformed so that there is a lower risk for occupants to make hard contact with the inner car surfaces [12]. This *a priori* knowledge can be used to simplify the model by defining the rigid zone as one perfectly rigid block and the deformable zone can be appropriately discretized into cells to constitute the local stiffness as shown in Fig. 2 (b). The authors in publications [13, 14, 15] discuss methodologies to discretize the vehicle body to formulate first and second-order mathematical models that are capable of representation of energy distribution patterns. These models are defined without any prior knowledge of the vehicle structure and assume a uniform ability of all the cells to absorb and transmit energy. In the proposed methodology of this paper, an in-depth dependency of the definition of cells with the vehicle structure is established, and the energy absorption/transmission parameters are initialized with due consideration to the geometrical and material properties of the encompassed components.

Figure 2 Generalized representation of remodeling and simplification process of virtual vehicle model

Each cell represents a specific volume on the vehicle frontal region and these cells must be parameterized to represent the stiffness behavior of the parts or section of parts that are enveloped by the respective cell. This is done in two steps:

- Spatial definition of cells
- Mathematical definition of the behavior of cells

In the pre-processing stage of FEM simulations, the virtual vehicle model is created by geometrically simplifying the CAD model and defining the properties like mesh, material models, contacts, etc. The spatial and mathematical definitions of the discretized cell model can be established using this virtual vehicle model. A detailed explanation of the methodology is as follows:

Spatial definition

The goal of the spatial definition for developing the discretized cell model is to create a grid of cells correlated to the vehicle structure as shown in Fig. 3.

Figure 3 Transformation of the VUT virtual model into a generalized cell structure

As mentioned, such a derivation can be done using the pre-processed FEM simulation model using the following steps:

Discretization of the model:

The discretization of the vehicle structure is limited to the region of interest as shown in Fig. 2. The discretization is constrained by certain physical parameters dependent upon the crash scenarios. The crash scenario defines the overlap of the frontal impact with the barrier during the crash test. The impact loads are computed and defined as an input for the model over the initial impact region. Hence the discretized cells must be structured such that the input forces can be accurately

Figure 4 Crash scenario overlap induced constraints for the discretization of the frontal vehicle region

defined over the complete cell region instead of partial regions. As the overlap is a percentage of width, this constrains the definition of some section planes in the y direction as shown in Fig. 4, where y_0 is the longitudinal axis, y_1 and $-y_1$ are overlaps in medium overlap crash tests and y_2 and $-y_2$ are overlaps in small overlap crash tests. The locations of acceleration sensors are represented using bright-colored dots.

To fully discretize the remaining region of interest, the following two methodologies can be utilized:

1 Uniform discretization: The general principle here is to divide the deformable zone uniformly into smaller cells. The constraints on the section planes perpendicular to the y-axis are considered and the refinement level is appropriately chosen. The advantage of this methodology is that an in-depth structural understanding of the components is not required for discretization. It is simple to compute and would result in an uncomplicated cell interaction model as cell boundaries are perfectly shared with the adjoining cells. However, such a discretization strategy leads to inefficiencies in material distribution between cells and also results in a complex cell behavior model. The complexity increase is due to an increase in the probability of the assignment of the components from different sub-assemblies into one cell. The computation and fine-tuning of the aggregate stiffness behavior in such cases would be moderately difficult.

2 Optimized discretization: In this approach, the discretization occurs at the sub-assembly level. Each sub-assembly is segmented into n cells such that the volume of each cell is approximately equal to a pre-defined target volume. After all the sub-assemblies are discretized, the region that is not enveloped by any cell is considered and these are discretized and defined as *null cells*. The *null cells* have no state equations and have no capacity to absorb or transmit any energy in their initial condition. This formulation simplifies the definition of the cell behavior model as there is a higher probability of coherence in properties like material, geometry, contacts, etc. The cell interaction model is also relatively simple due to the efficient modeling of empty space within the region of interest. Under the scope of this research work, the optimized discretization approach is utilized.

Extraction of geometric data:

The pre-processed FEM simulation model has the geometric information of all the parts, sub-assemblies, and assemblies in a vehicle structure. The embedded data follows the standard output architecture of the respective pre-processing simulation software tools. But irrespective of the development software, a general representation of the data modeling architecture is as shown in Fig. 5. The geometric data at the fundamental level is in the form of nodal coordinates. These nodes form different elements and the elements come together to form beam, shell, and solid components.

Figure 5 Data storage architecture of pre-processed FEM simulation model

The geometry can be effectively extracted by mining the spatial coordinates of nodes and the correlations between respective components, from the simulation models. For example, the nodes and their relation with respective elements, elements and their relation with respective beams, shells, & solids, etc. Therefore, the end result of this step is a data matrix as represented in the following equation:

$$N = \begin{bmatrix} N_1 & X_1 & Y_1 & Z_1 \\ N_2 & X_2 & Y_2 & Z_2 \\ : & : & : & : \\ N_i & X_i & Y_i & Z_i \end{bmatrix}, \qquad E = \begin{bmatrix} E_1 & t_1 & N_1 & N_2 & N_3 & . & . \\ E_2 & t_2 & : & : & : & : & : \\ : & : & : & : & : & : & : \\ E_j & t_j & N_a & N_b & N_c & . & . \end{bmatrix},$$

$$A_{sub} = \begin{bmatrix} P_1 & M^1 & E_1 & E_2 & E_3 & . & . \\ P_2 & M^2 & : & : & : & : & : \\ : & : & : & : & : & : & : \\ P_k & M^k & E_a & E_b & E_c & . & . \end{bmatrix}, \qquad A = \begin{bmatrix} A_{sub_1} \\ A_{sub_2} \\ : \\ A_{sub_l} \end{bmatrix}, \tag{1}$$

where
N is a $i \times 4$ matrix comprising of i nodes N_x and their cartesian co-ordinates X_i, Y_i, Z_i,
E is a $j \times m$ matrix comprising of j elements E_x, their thickness t_x (only for 2D elements) and their corresponding nodes $[N_a, N_b, N_c, ...]$. The number of nodes in each element depends upon the type of element,
A_{sub} is a $k \times n$ matrix comprising of k parts P_x, materials M^x and their corresponding elements $[E_a, E_b, E_c, ...]$. The number of elements in each part depends upon the size of the part and the refinement level of the mesh, and
A is a $l \times 1$ matrix comprising of l sub-assemblies A_{sub_x}.

Extraction of material data:
The pre-processed FEM simulation model also contains the material models that are assigned to individual parts and substantially drive their dynamic behavior under different loading conditions. As illustrated in Fig. 5, the material models can comprise a range of material properties convoluted with various mathematical models to represent the complex material behavior. The goal of this step is to extract the fundamental parameters that influence energy absorption and dissipation during the process of deformation in the event of a crash. But, as a cell can envelope multiple parts that could be made of different materials, an aggregate material behavior matrix must be derived. This matrix can be an approximate estimation as there is a scope for refinement during the iterations of the reinforcement learning phase. The data available in the material cards are extracted and some undefined/dependent parameters can be calculated if not explicitly defined. A general representation is

as follows:

$$M^x = \begin{bmatrix} \rho^x & Y^x & G^x & K^x & \sigma_y^x & \sigma_u^x & \sigma_f^x & \nu^x \end{bmatrix} , \quad (2)$$

where
ρ is the mass density,
Y, G, and K are the Young's, Shear, and Bulk moduli respectively,
σ_y, σ_u, and σ_f are yield strength, ultimate strength, and fracture point respectively, and
ν is the Poisson's ratio.

Mathematical definition

Every 3D cell defined in the previous step has a spatial association with a specific discretized volume on the vehicle frontal region (region of interest). A mathematical model must be defined and initialized to represent the behavior of each cell under the influence of loads. There are two main components of such behavior namely (i) cell behavior and (ii) cell interaction.

Cell behavior

The key objective is to establish a mathematical function that qualitatively represents the stiffness of the region enveloped by each cell. The function must also be compatible with the iteration and learning methodologies used during the training and testing phase of reinforcement learning. With due consideration to the goals and constraints, the cell's behavior is defined by a function based on the temporal relationship between input energy, energy dissipation, and energy transmission of cells under crash loads. The mathematical computations of crash energies in crash simulations and physical crash tests are illustrated in [16]. The key component of consideration from the total crash energy is the energy component spent on the deformation of the vehicle structure. This deformation energy is absorbed and eventually dissipated in various forms. Fundamentally each cell interacts with the deformation energy in two ways, a part of the energy is absorbed and the rest is dispersed into adjoining cells following the laws of conservation. This behavior is dependent on various factors like aggregate stiffness, reaction forces, etc for each cell. The modeling of the energies-based parameters and its use for defining the cell behavior is a utilization and extension of the technique presented in [17]. The cell behavior can generally be represented by a relation as shown in Fig. 6.

Figure 6 Cell behavior: Relation between energy absorption and energy transmission, where E is input energy, E_a is absorbed energy, E_t is transmitted energy, E_x is the onset of transmission and E_s is the saturation point

In the interest of faster convergence, an approximate initialization of the cell behavior parameters is done based on the properties of the components enveloped by each cell as recorded in Eq. (2). The curvature and slopes of individual pieces of the curve are dependent on the defined parameters and during the reinforcement learning process, these parameters are iterated as *actions*.

Cell interaction

The transmitted energy component from the cell behavior model must be considered as an input for all adjoining cells. The transmission of energy between adjoining cells is modeled with the following boundary conditions:

1 The flow of energy is prohibited in the negative direction of the x-axis.
2 Cells that envelop empty space do not have the capacity for any energy transfer.
3 In neighboring cells that are along the principal direction of force (PDOF) and have a continuity of material, the energy distribution factor is set at a higher value than the lateral directions.
4 In cells adjacent to each other that have a continuity of material, the energy distribution factor is proportional to the ratio of cross-sectional areas at the boundary of the adjoining cells.
5 The summation of all factors of energy distribution should be equal to one.

A general representation is as shown in Fig. 7 (a). The factors of distribution associated with each direction are represented as follows:

$$E_i = k_i * E_{trans} \ , \tag{3}$$

where $i = \{x, y, y', z, z'\}$ such that $\sum k_i = 1$, y' and z' are negative y and z axes respectively and E_{trans} is the component of the total transmitted energy from a cell. However, the proportion of energy transmission in the five directions must be determined for each cell. The methodology employed can be explained using a discretized vehicle sub-assembly as shown in Fig. 7 (b).

Figure 7 Cell interaction for a discretized cell model using a vehicle sub-assembly.

With reference to Fig. 7, consider the following examples to illustrate the modeling of cell interaction factors with the defined boundary conditions:

- Cell $A1$: There are three adjoining cells to $A1$ and an interaction model must be defined for each pair. The cells $A1$ and $A2$ are connected in the progression of the x-axis and have continuity of the component between them. Generally speaking, the principal direction of force is along the x-axis, the factor of energy distribution in the x-axis would proportionately be much higher than in the other directions. Cells $A1$ and $B1$ are connected in the progression of the y-axis and have a continuity of material. However, cell $A1$ is located further than cell $B1$ and the direct crash energy interaction would be earlier for $B1$. With reference to the boundary conditions, a reverse flow is eliminated; hence the factor of energy distribution in the y-axis would be initialized to 0. Cells $A1$ and $B2$ are connected in the progression of the y-axis but do not have a continuity of material. Hence, there is no possibility of any transmission of the energies and the factor of energy distribution along the y-axis would be initialized to 0. Therefore, as per the boundary condition of the summation of factors (k_i), the factor k_x would be initialized to 1.
- Cell $B1$: There are two adjoining cells to $B1$. Cells $B1$ and $A1$ are connected in the progression of the negative y-axis (y') and have a continuity of material.

Cell $B1$ and $B2$ are connected in the progression of the x-axis but there is no continuity of material. Hence, the factor of transmission in the negative y-axis would be set to 1 and others would be set to 0.

- Cell $D1$: There are three adjoining cells to $D1$. As there is no continuity of the component between cells $D1$ and $D2$, the transmission factor between them is set to 0. Cells $C1$ and $E1$ both have a continuity of the component from cell $D1$. Hence, according to the boundary conditions, the respective factors of transmission should be proportional to the cross-sectional area of the component at the common cell boundary:

$$k_y = \frac{A_{D1E1}}{A_{D1C1} + A_{D1E1}} \ , \tag{4}$$

$$k_{y'} = \frac{A_{D1C1}}{A_{D1C1} + A_{D1E1}} \ , \tag{5}$$

where
A_{D1E1} is the cross-sectional area of the component at the cell boundary between cells $D1$ and $E1$,
A_{D1C1} is the cross-sectional area of the component at the cell boundary between cells $D1$ and $C1$,
k_y and $k_{y'}$ are the transmission factors in y and y' directions respectively.

State space modeling

The evaluation of the state of the environment is essential to learn the configuration of the cells that lead to a certain end result. The fundamental parameter that represents the state of the environment is the estimated crash pulse based on the configuration of the discretized cell model. A minimum of three acceleration sensors are installed in the VUT as shown in Fig. 2 which record three crash pulses at their respective positions. In order to retain an association with the reference signal the state equation is represented in terms of the y-intercept for every relative time reference to the time of impact (t_0). Hence the state vector $\mathbf{s}$ can be represented by Eq. (6), where $\pm d_1^x$ is the difference in the y-direction between the reference signal and the estimated signal at time t_1; i,j, and k are the number of recorded time intervals in the respective crash pulse; a, b, and c are indicators for crash pulses recorded from three different acceleration sensors:

$$\mathbf{s} = [(\pm d_1^a \ , \pm d_2^a \ , \pm d_3^a \ ....., \pm d_i^a), \ (\pm d_1^b \ , \pm d_2^b \ , \pm d_3^b \ ....., \pm d_j^a), \\ (\pm d_1^c \ , \pm d_2^c \ , \pm d_3^c \ ....., \pm d_k^c)] \ . \tag{6}$$

Action space modeling

The means of interaction and learning for a Reinforcement learning agent is to observe the environment and take some policy-based actions [8]. According to the modeling architecture of the environment, every cell has a total of seven configurable parameters. If there are n cells in an environment, then a matrix of all configurable parameters can be written as follows:

$$\mathbf{A} = \begin{bmatrix} E_{x_1} & E_{s_1} & f_{x_1} & f_{y_1} & f_{y'_1} & f_{z_1} & f_{z'_1} \\ E_{x_2} & E_{s_2} & f_{x_2} & f_{y_2} & f_{y'_2} & f_{z_2} & f_{z'_2} \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ E_{x_n} & E_{s_n} & f_{x_n} & f_{y_n} & f_{y'_n} & f_{z_n} & f_{z'_n} \end{bmatrix} . \quad (7)$$

It is evident that the possible permutations of all possible actions in the action space for an algorithm to test are colossally enormous. Hence, efficient environment modeling is highly critical. An active attempt is done to avoid defining cells in regions with no components, transmission factors are hard set to 0 where energy transfer is not feasible, etc.

Secondly, a criterion is defined for the agent that forbids taking actions on the cells that are not in the primary or secondary region of impact. For example, in the initial duration of a driver-side small overlap crash test scenario, the cells in the front of the vehicle on the passenger's side would not be relevant for the end results. Hence, no attempt will be made to fine-tune those parameters.

Finally, the preference for parameter fine-tuning shifts from cell behavior parameters to cell interaction parameters, once the saturation point has been reached as all the input energy would be transmitted to the adjoining cells capable of reception. Hence, the effect on the output would be high. Therefore, the action space and the actual matrix of configurable parameters could be effectively diminished and convergence could be efficiently achieved.

Reward function

The output parameter evaluated from the environment is the estimated crash velocity signal. The measured reference signal during the crash simulations and crash tests is the acceleration signal. However, in the current implementation, the first integral of a filtered crash acceleration signal dimensioned as velocity (v) is used as a reference. Hence, the key mathematical computation is to quantify the similarity between the reference signal and the estimated signal from the environment and reward the configuration/actions that improve the similarity of the estimated signal to the reference signal.

There is a range of mathematical techniques that can be used to quantify the similarities between two curves, as illustrated in [18]. But, there are some specific considerations for the reward function in the current scenario. Consider the curves shown in Fig. 8. In all the curves shown, one possible method to quantify the similarity could be the *area between the curves*. It would be a positive scalar quantity computed using Eq. (8), where $Q(x)$ is the estimated signal, $P(x)$ is the reference signal, a and b are the limits within which the curve segments exist. Although the area between curves is a good indicator of the similarity, it has certain drawbacks. The area between the curves shown in Fig. 8 (a), (b), and (c) is quantitatively almost the same, but the curves are qualitatively very different. To evaluate the effectiveness of different actions in different states and to learn the best course of

Figure 8 Illustration of curve comparison requirements and possible suitable quantification models

action given the state of the environment, the reward function must be able to effectively distinguish the desirable outcomes from the undesirable ones:

$$A = \int_a^b |Q(x) - P(x)| \, \mathrm{d}x \ . \tag{8}$$

Hence, an additional mathematical function is considered i.e. the Discrete Fréchet Distance (DFD). If the loci of two points that are connected with a leash, were to simultaneously trace the two curves without any backward movements, the DFD can be defined as the minimum length of leash required that would allow both points to fully trace the curves. Fréchet distance is computed using Eq. (9) where $F(Q, P)$ is the Fréchet distance of two curves Q (estimated signal), and P (reference signal), q and p are reparameterizations of Q and P to be continuous and non-decreasing, t could be thought of as the 'time' variable that only allows forward propagation along the curves, and d is the distance function of the metric space S [18]. The DFD, which is used in the modeling of the reward function is computed as illustrated by [19] using the following equations:

$$F(Q, P) = \inf_{q,p} \max_{t \in [0,1]} \Big(d\Big(Q\big(q(t)\big), P\big(p(t)\big)\Big)\Big) \tag{9}$$

$$d\,F(Q, P) = \min \|L\| \tag{10}$$

where $d\,F$ is the discrete Fréchet distance, L is a coupling between Q and P, and length $\|L\|$ of the coupling L is the length of the longest link L.

In conjunction with *area between the curves*, DFD would be effective in distinguishing the curve similarity quantification between the curves in Fig. 8 (a) and (b). The DFD of the former would be smaller than that of the latter. In addition to the aforementioned methodologies, root means squared euclidean error (RMSEE) is also used as an indicator of the similarity between two curves and computed as follows:

$$RMSEE = \sqrt{\frac{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_n - q_n)^2}{n}} \ , \tag{11}$$

where p_x and q_x are the discrete points from the curves P and Q at time t_x.

An episode can be defined as the set of policy-based actions an RL agent makes to interact with the environment and eventually reaches a termination condition. The reward factor R for the generated estimations for each iteration in an episode

can be defined by Eq. (12), where ΔA, ΔDFD, and $\Delta RMSEE$ are the differences in the area between the curves, Discrete Fréchet Distance and root mean squared euclidean error respectively, of the current and the previous episode. The factors a, b, and c can be determined using a simple regression analysis:

$$\begin{aligned} R &= f(\Delta A,\ \Delta DFD,\ \Delta RMSEE) \\ &= \frac{a}{\Delta A} + \frac{b}{\Delta DFD} + \frac{c}{\Delta RMSEE}\ . \end{aligned} \tag{12}$$

Conditions of Termination

Termination of an episode is necessary to be defined in three conditions:

1. The fine-tuning of the environment parameters would ideally lead to a state where a perfectly close fit of the estimated crash pulse with the reference signal is reached. However, realistically an acceptable range of accuracy must be defined so that the agent avoids reaching deep into the regions of diminishing returns. Hence, a termination condition must be set to end the process once the curve similarity parameters achieve a pre-defined threshold.
2. During the learning process, the RL agent may reach a highly undesirable state due to multiple bad actions and it becomes very inefficient to let the agent keep trying to recover. In such cases, it's beneficial to terminate the current episode and begin a new one. Hence, the curve similarity parameters must also have a pre-defined higher limit threshold that must result in termination.
3. In every episode of the training exercise, a maximum limit of iterations must be defined, after which the episode must be forced to terminate. It avoids potential looping scenarios and acts as a positive reinforcement for the agent to learn to reach the desired outcomes faster.

Results

The methodology proposed in this research paper is fully implemented using virtual vehicle models and analyzed using the simulation results. To illustrate the implementation process and to explain the methodology with reference to a practical example, individual steps are presented in this section. The illustrations mainly concentrate on the process of adaptation of a conventional FEM simulation model into a reinforcement learning compatible surrogate model with the aim of estimating the crash pulses for calibration of the crash-detection algorithm.

Defining the environment

The virtual vehicle model used for the creation of the reinforcement learning environment is an ESI Virtual Performance Solution pre-processing stage model. The details are as shown in Table 1. The nodes, elements, shells, solids, parts, subassemblies, and assembly matrices are extracted by following the respective standard software architecture. The vehicle model is cropped for separating the region of interest (ROI). The front vehicle region is considered for further discretization to complete the spatial definition of the cells and Table 2 shows some of the basic parameters used. Fig. 9 (a) shows the transition of the virtual vehicle model from

Table 1 Technical details of the used virtual vehicle model and crash scenario used for reference signals

Parameter	Value
Vehicle type	B-segment
Vehicle mass	1600 kg
Order of number of nodes	10^6
Order of number of 2D elements	10^6
Order of number of 3D elements	10^5
Crash scenario	
Velocity	40 kph
Overlap	100 %
Relative angle	0 deg

pre-processing FEM software environment to the Matlab environment using the extracted matrices.

Fig. 9 (b) shows the development of individual cells to represent discretized volumes of the components in the frontal vehicle region. The mass of the material enveloped by each cell and their aggregate material properties are stored in association with individual cells. The interaction models are also defined using the boundary conditions.

Table 2 Parameters used for discretization and definition of the environment

Parameter	Value
ROI crop limit	1.20 m
ROI Volume	$2.15 * 10^9$ mm^3
Threshold cell volume	$3 * 10^5$ mm^3
Uniform discretization	7165 cells
Optimised discretization	531 cells

Figure 9 (a) Virtual vehicle model used for illustrations and cropping the model to separate ROI. (b) Process of the spatial definition of cells by discretization of the individual structural components.

State space

The state of the environment is computed based on the estimated crash pulse generated as a result of the respective configuration. For the purpose of simplicity, only one acceleration sensor is assumed for illustration. During the discretization process, the acceleration sensors are clustered individually as separate cells such that the energy transmitted to the sensor locations can be computed accurately. The resultant relative discrete displacement is used to generate the estimated crash pulse. The sensors mounted on the VUT operate at a frequency of $2000Hz$. Hence, each iteration has a time interval of $0.5ms$ and the state space equation is a vector of discrete delta values between reference and estimated signals as shown in Fig. 10.

Figure 10 Evaluation of the state of the environment based on the difference between estimated and reference velocity signals.

Reward function

The evaluation of the effectiveness of an *action* in a particular state is quantified by the curve similarity factors: (i) Area between the curves, (ii) Discrete Fréchet

Distance, and (iii) Root Mean Squared Euclidean Error. Table 3 shows an example of the computation of curve similarity factors for six consecutive iterations of an episode and the respective reward factors. In Eq. (12), the factors a, b, and c are initialized to one for the sake of these calculations. If one considers the base configuration's estimation as $I0$, then the reward factor for iteration $I1$ is evaluated with reference to its previous state. A relative increase in the reward factor is desirable and is rewarded and a relative decrease is undesirable and is punished. The *actions* taken to change the state from $I4$ to $I5$ show a largely positive shift in the reward factor hence gaining a high reward. In contrast, the *actions* from $I5$ to $I6$ suffer large punishment. Therefore a policy is formulated as the RL-agent interacts with the environment over a large set of episodes, learning about the *state-action* combinations that result in large rewards.

Table 3 Comparison of the curve similarity parameters between iterations of a reinforcement learning episode

	I1	I2	I3	I4	I5	I6
Area between the curves	0.0259	0.0311	0,0363	0,0189	0,0015	0,0415
Discrete Fréchet Distance	0,423	0,455	0,487	0,411	0,335	0,519
RMSEE	0,2585	0,2737	0,2889	0,22	0,1511	0,3041
Reward factor	44,84	38,00	33,06	59,89	676,27	29,31

Conclusion and Open Problems

The research paper presented a methodology to adapt the crash simulation models into a reinforcement learning-compatible architecture, with the goal of estimating the crash pulses for calibration of the crash-detection algorithm. The modeling of fundamental blocks of reinforcement learning architecture viz., Environment, State, Actions, and Rewards are illustrated in detail. A wide range of established reinforcement learning agents could be used to interact with the environment using the current architecture. The results from a variety of crash simulations can be used as a reference during the learning phase and further expanded using the results from physical crash tests. Hence a policy could be learned that can modify the virtual environment model to behave in accordance with known and accurate real-world observations. This opens up a great opportunity to develop surrogate virtual models that are calibrated using simulations and physical test results with the help of AI. These models would be highly useful in calibrating the standard crash-detection algorithm and also validating its performance in a large set of statistically significant crash scenarios. As the scope of this paper is limited to presenting an adaptation technique, the results presented are intended to demonstrate the methodology in action. The initial implementations show good compliance with the expected outcomes.

Future work will be focused on the extended validation of the method to get more insights, particularly into its limitations. The coherence of aggregating complex structural and material properties to represent the energy absorption and transmission behavior of a cell is being tested for possible limitations or drawbacks. A detailed study of the accuracy of estimations and the computing power and time required to achieve them is ongoing. The non-convergent cases are to be studied to identify new boundary conditions to improve the process efficiency.

Acknowledgements
The authors are grateful to AUDI AG, for sponsoring this research project.

Funding
Open access funding is enabled and organized by Technische Hochschule Ingolstadt.

Abbreviations
RL: Reinforcement Learning,
LSTM: Long Short Term Memory,
VUT: Vehicle Under Test,
AI: Artificial Intelligence,
FEM: Finite Element Method,
CAD: Computer-Aided Design
PDOF: Principle Direction Of Force,
DFD: Discrete Fréchet Distance,
RMSEE: Root Mean Squared Euclidean Error,
ROI: Region Of Interest.

Availability of data and materials
The simulation models used in this study are available from AUDI AG, but restrictions apply to their availability, which was used under license for the current study, and so are not publicly available. Datasets are however available from the authors upon reasonable request and with permission of AUDI AG.

Competing interests
The authors declare that they have no competing interests.

Consent for publication
All authors approved the final submitted manuscript and consent for its publication.

Authors' contributions
Authors SA, OV, and DB conceptualized the presented hypothesis. SA developed and tested the algorithm. LH, VO, and DB supervised and handled the project administration. All authors contributed to finalizing the manuscript.

Author details
[1]CARISSMA Institute of Safety in Future Mobility (C-ISAFE), Technische Hochschule Ingolstadt, Ingolstadt, Germany. [2]Development of partner protection/accident detection, Audi AG, Ingolstadt, Germany. [3]Department of applied sciences, University of West Bohemia, Pilsen, Czech Republic.

References
1. Mcclenathan, R.V., Nakhla, S.S., Mccoy, R.W., Chou, C.C.: Use of photogrammetry in extracting 3d structural deformation/dummy occupant movement time history during vehicle crashes. SAE Technical Paper Series (2005). doi:10.4271/2005-01-0740
2. Rentschler, W., Uffenkamp, V.: Digital photogrammetry in analysis of crash tests. SAE Technical Paper Series (1999). doi:10.4271/1999-01-0081
3. Varat, M.S., Husher, S.E.: Vehicle impact response analysis through the use of accelerometer data. SAE Technical Paper Series (2000). doi:10.4271/2000-01-0850
4. Diez, C., Kunze, P., Toewe, D., Wieser, C., Harzheim, L., Schumacher, A.: Big-data based rule-finding for analysis of crash simulations. In: Schumacher, A., Vietor, T., Fiebig, S., Bletzinger, K.-U., Maute, K. (eds.) Advances in Structural and Multidisciplinary Optimization, pp. 396–410. Springer, Cham (2018)
5. Hahner, S., Iza-Teran, R., Garcke, J.: Analysis and prediction of deforming 3d shapes using oriented bounding boxes and lstm autoencoders. In: Artificial Neural Networks and Machine Learning–ICANN 2020: 29th International Conference on Artificial Neural Networks, Bratislava, Slovakia, September 15–18, 2020, Proceedings, Part I 29, pp. 284–296 (2020). Springer
6. Koeppe, A., Bamer, F., Markert, B.: An efficient monte carlo strategy for elasto-plastic structures based on recurrent neural networks. Acta Mechanica **230**(9), 3279–3293 (2019). doi:10.1007/s00707-019-02436-5
7. Sequeira, G.J.A., Konda, A.R., Lugner, R., Jumar, U., Brandmeier, T.: Crash pulse prediction using regression algorithm with gradient descent optimization method for integrated safety systems. SAE International Journal of Transportation Safety **10**(2) (2022). doi:10.4271/09-10-02-0009
8. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction / Richard S. Sutton and Andrew G. Barto, Second edition edn. Adaptive computation and machine learning. The MIT Press, Cambridge, Massachusetts (2018)
9. Neptune, J.A., Flynn, J.E.: A method for determining accident specific crush stiffness coefficients. SAE Technical Paper Series (1994). doi:10.4271/940913
10. Varat, M.S., Husher, S.E., Kerkhoff, J.F.: An analysis of trends of vehicle frontal impact stiffness. SAE Technical Paper Series (1994). doi:10.4271/940914
11. Nusholtz, G.S., Xu, L., Shi, Y., Di Domenico, L.: Vehicle mass and stiffness: Search for a relationship. SAE Technical Paper Series (2004). doi:10.4271/2004-01-1168
12. Mills, N.: Chapter 9.3.4 - seating case study - crash safety. In: Mills, N. (ed.) Polymer Foams Handbook, pp. 205–233. Butterworth-Heinemann, Oxford (2007). doi:10.1016/B978-075068069-1/50010-6. https://www.sciencedirect.com/science/article/pii/B9780750680691500106
13. Harmati, I., Rovid, A., Varlaki, P.: Energy absorption modelling technique for car body deformation. In: SACI 2007, pp. 269–272. IEEE, Piscataway NJ (2007). doi:10.1109/SACI.2007.375523
14. Harmati, I., Várlaki, P.: Identification of energy distribution for crash deformational processes of road vehicles. Acta Polytechnica Hungarica **4**(2), 19–28 (2007)

15. Harmati, I., Varlaki, P.: Estimation of energy distribution for car-body deformation. In: ISCIII '07, pp. 229–232. IEEE, Piscataway NJ (2007). doi:10.1109/ISCIII.2007.367394
16. Afraj, S., Böhmländer, D., Vaculin, O., Hynčik, L.: Quantification methodology for crash behavior comparison between virtual crash simulations and real-time crash tests. In: FISITA World Congress 2021 (2021). doi:10.46720/f2021-pif-072
17. Harmati, I., Rovid, A., Szeidl, L., Varlaki, P.: Identification and reconstruction of car body deformation applying tensor product models. In: 2006 International Conference on Intelligent Engineering Systems, pp. 98–101 (2006). doi:10.1109/INES.2006.1689349
18. Mothershed, D.M., Lugner, R., Afraj, S., Sequeira, G.J., Schneider, K., Brandmeier, T., Soloiu, V.: Comparison and evaluation of algorithms for lidar-based contour estimation in integrated vehicle safety. IEEE Transactions on Intelligent Transportation Systems **23**(5), 3925–3942 (2022). doi:10.1109/TITS.2020.3044753
19. Eiter, T., Mannila, H.: Computing discrete fréchet distance. Technical Report CD-TR 94/96, Technical University of Vienna, Vienna, Austria (Apr. 25, 1994)

Figures

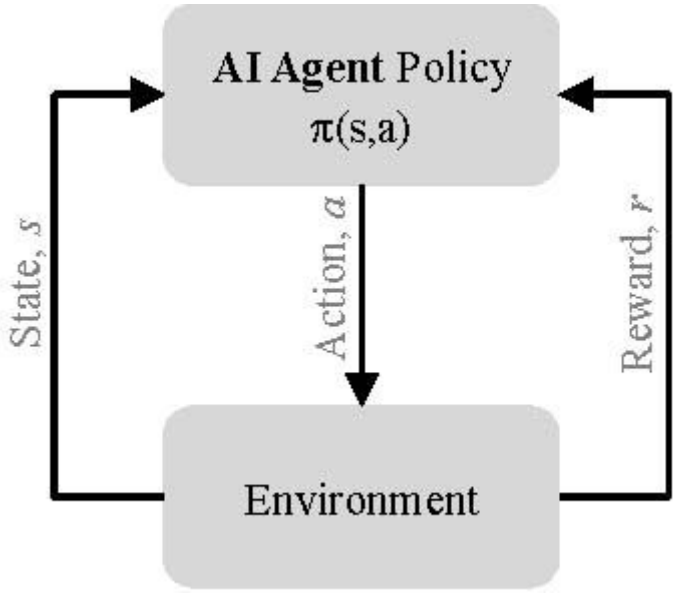

Figure 1

Fundamental framework of a reinforcement learning AI application

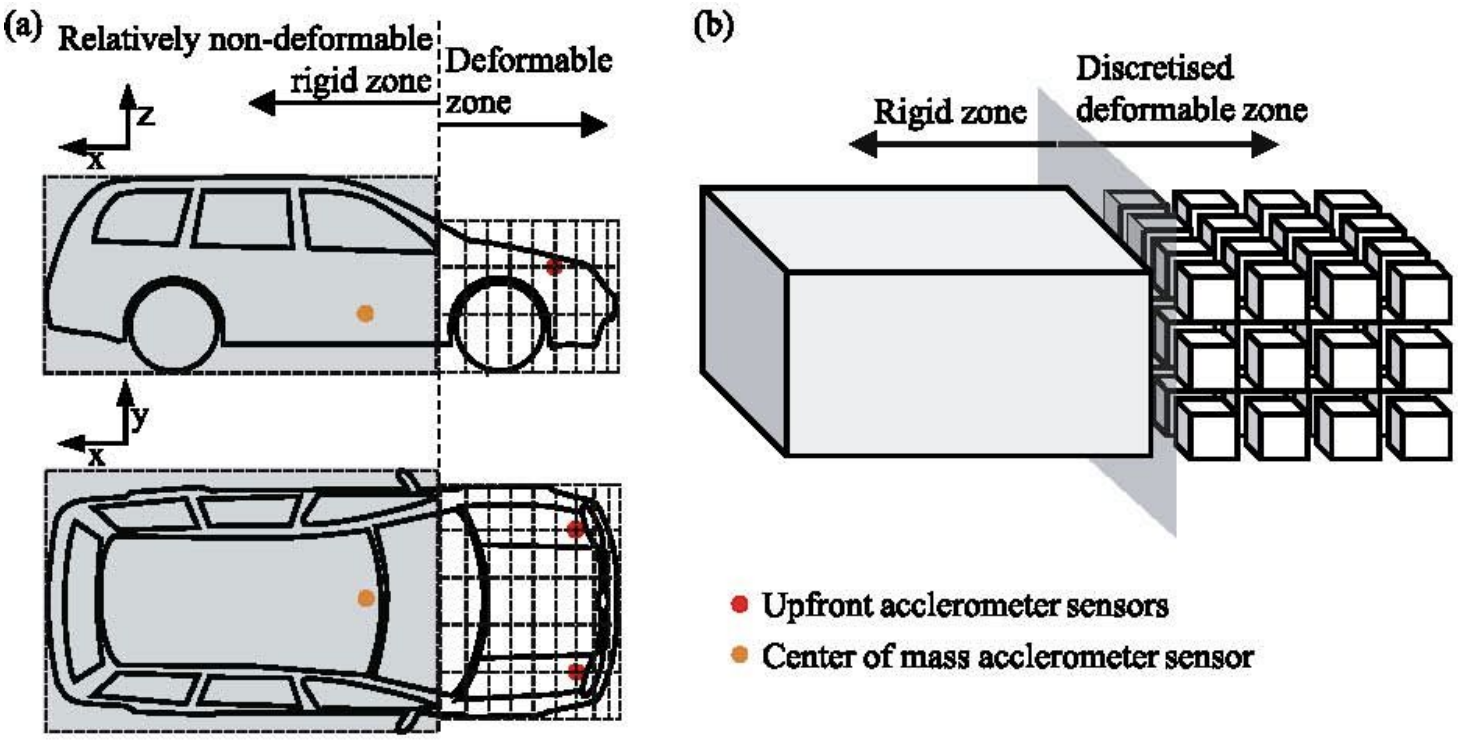

Figure 2

Generalized representation of remodeling and simplification process of virtual vehicle model

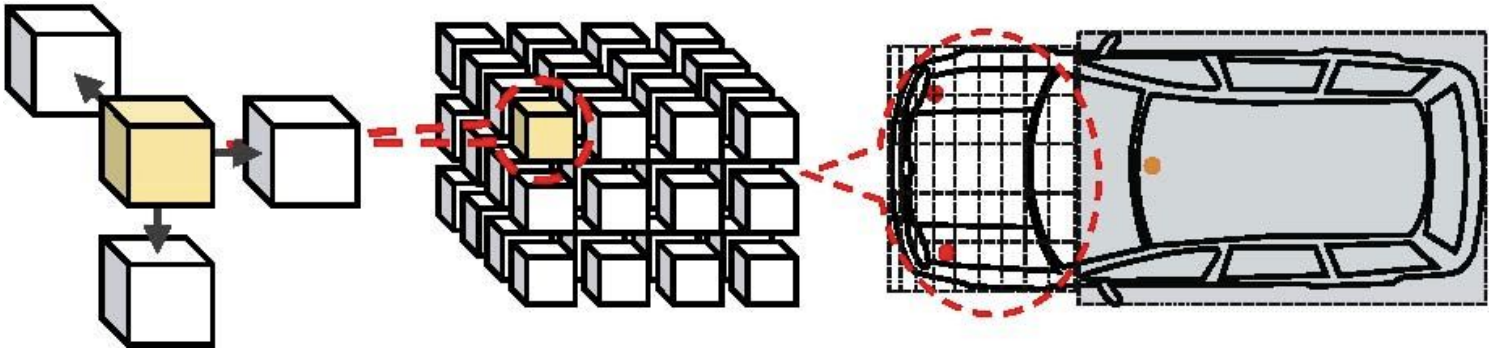

Figure 3

Transformation of the VUT virtual model into a generalized cell structure

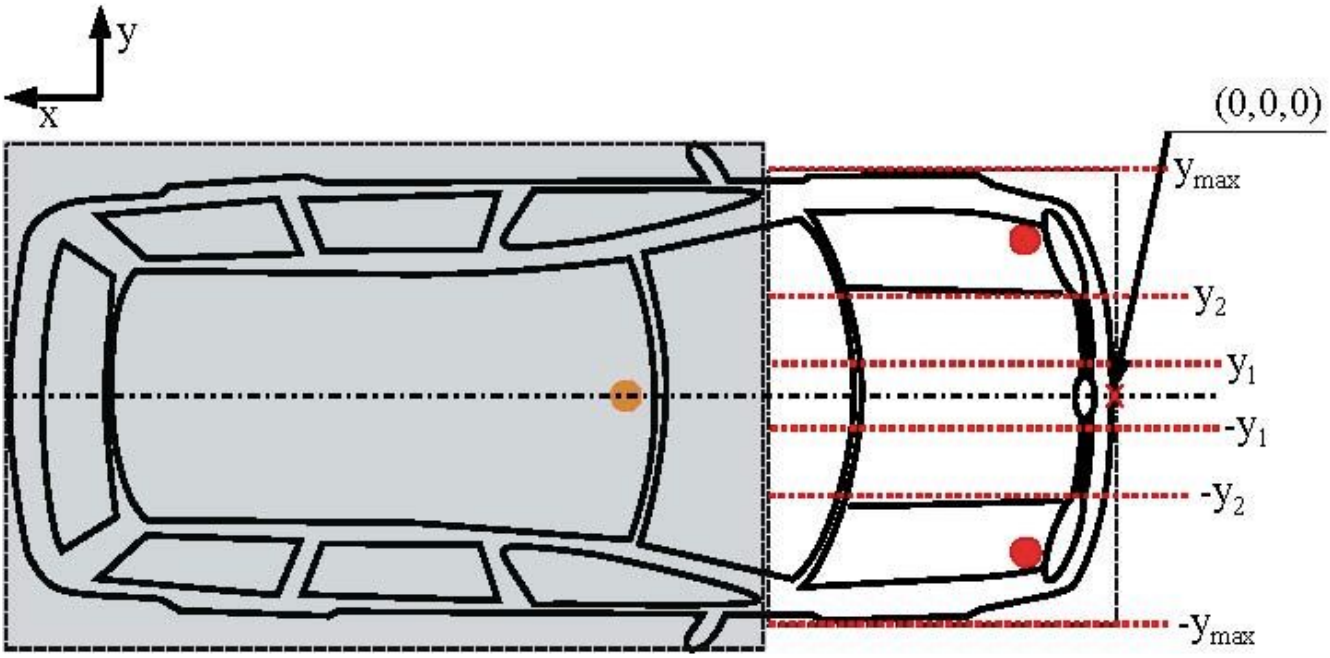

Figure 4

Crash scenario overlap induced constraints for the discretization of the frontal vehicle region

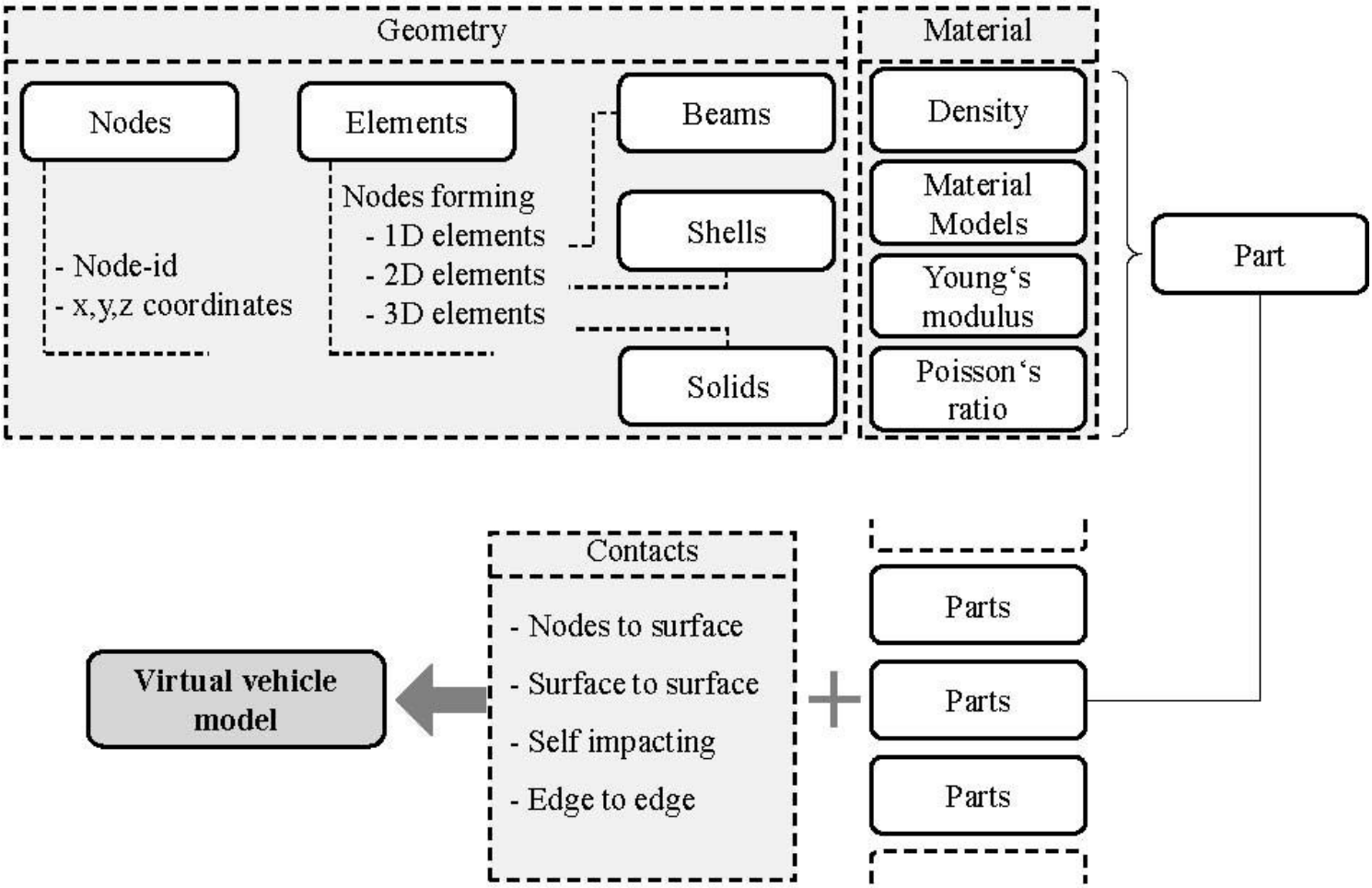

Figure 5

Data storage architecture of pre-processed FEM simulation model

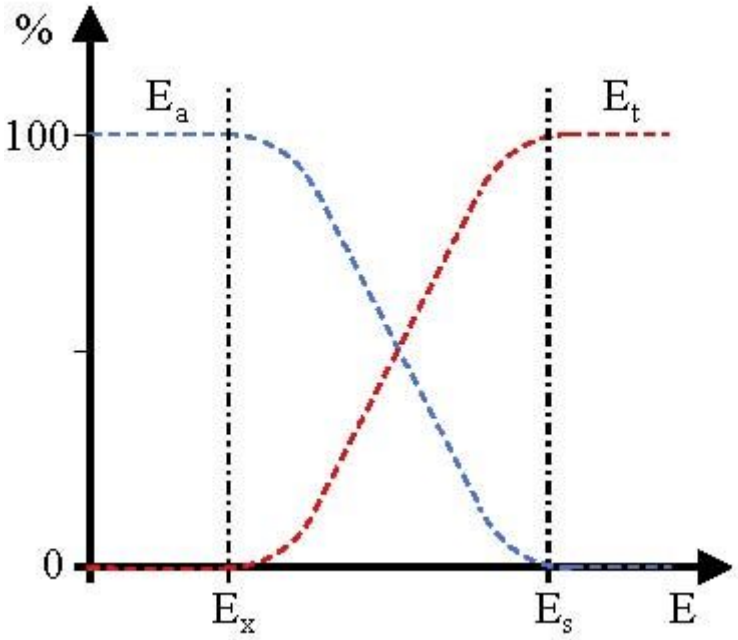

Figure 6

Cell behavior: Relation between energy absorption and energy transmission, where E is

input energy, Ea is absorbed energy, Et is transmitted energy, Ex is the onset of transmission and Es is the saturation point

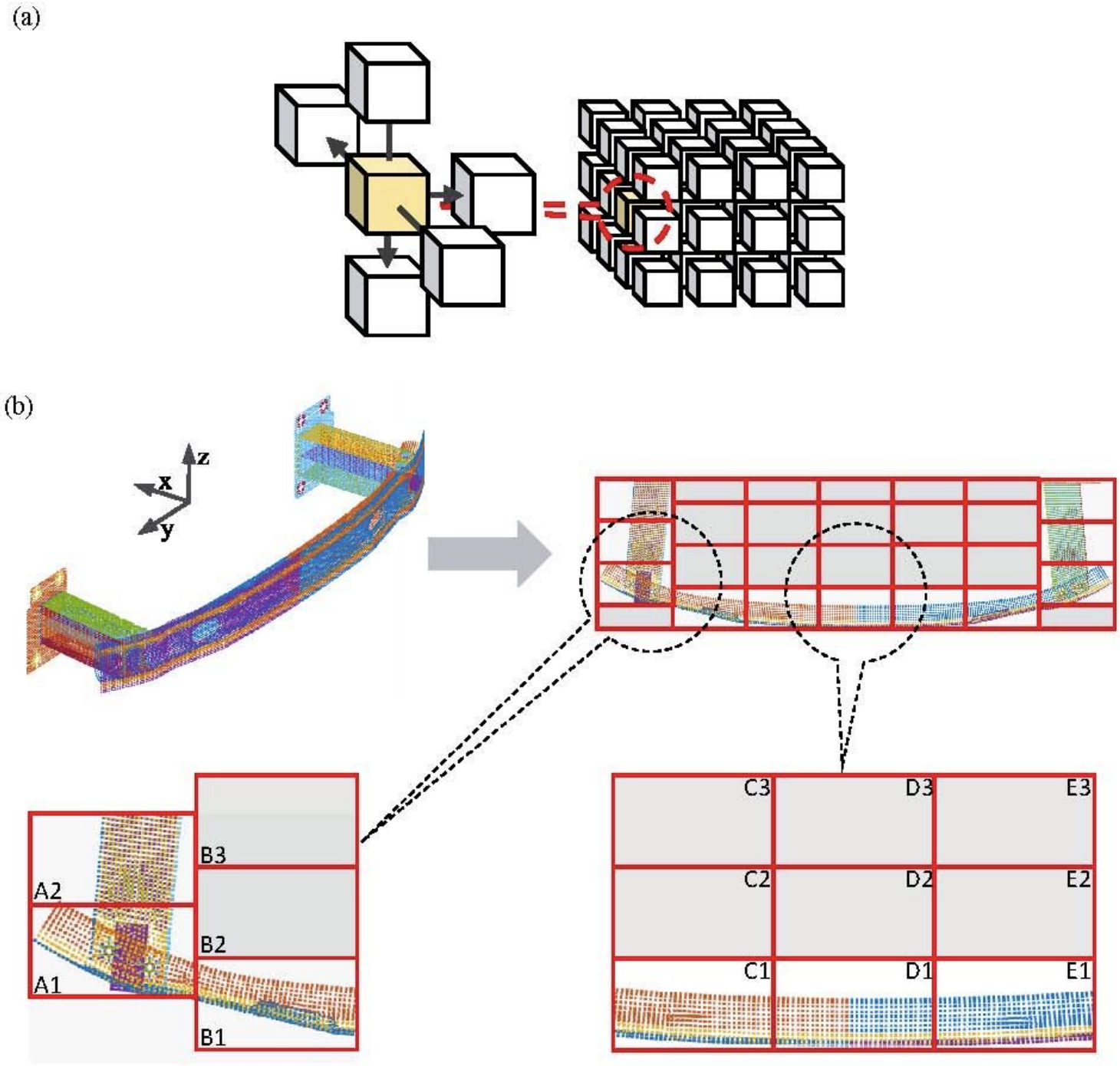

Figure 7

Cell interaction for a discretized cell model using a vehicle sub-assembly

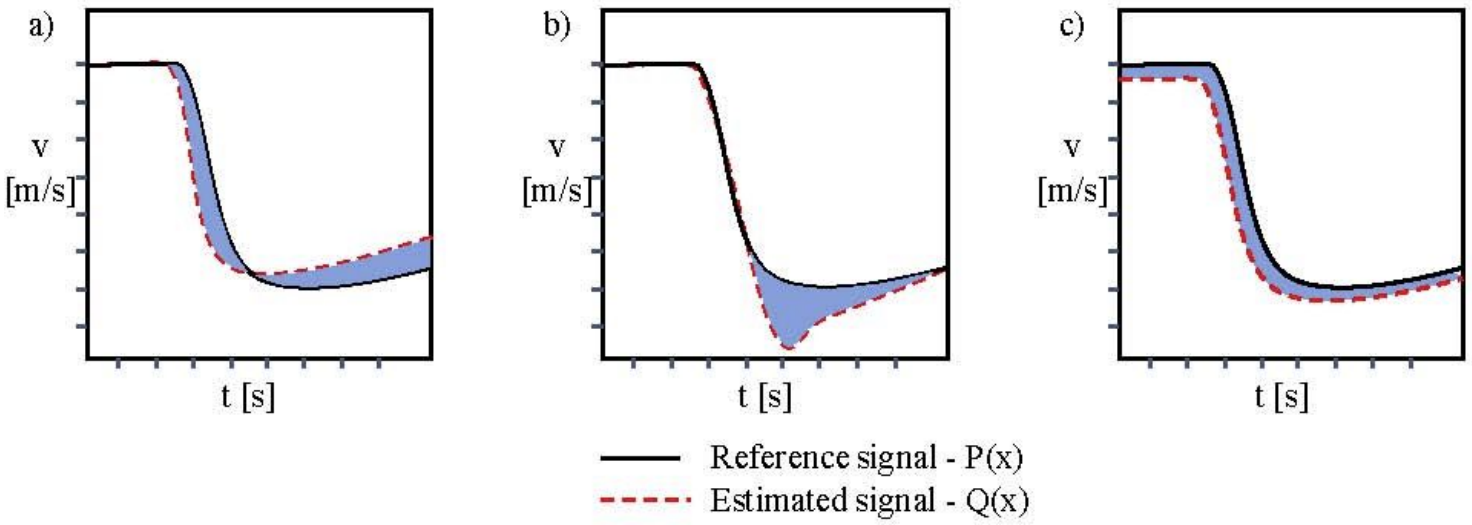

Figure 8

Illustration of curve comparison requirements and possible suitable quantification models

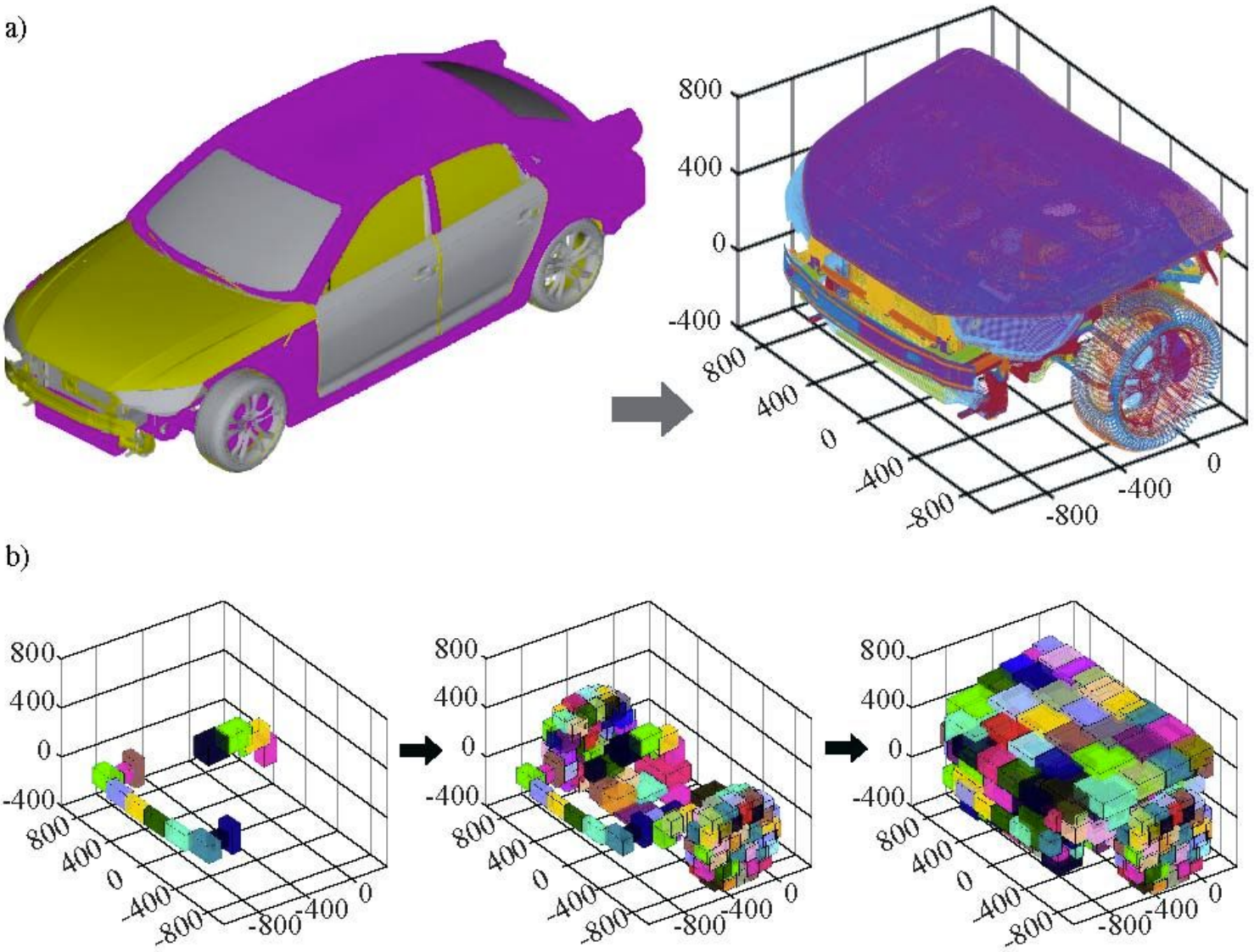

Figure 9

(a) Virtual vehicle model used for illustrations and cropping the model to separate ROI.

(b) Process of the spatial definition of cells by discretization of the individual structural components

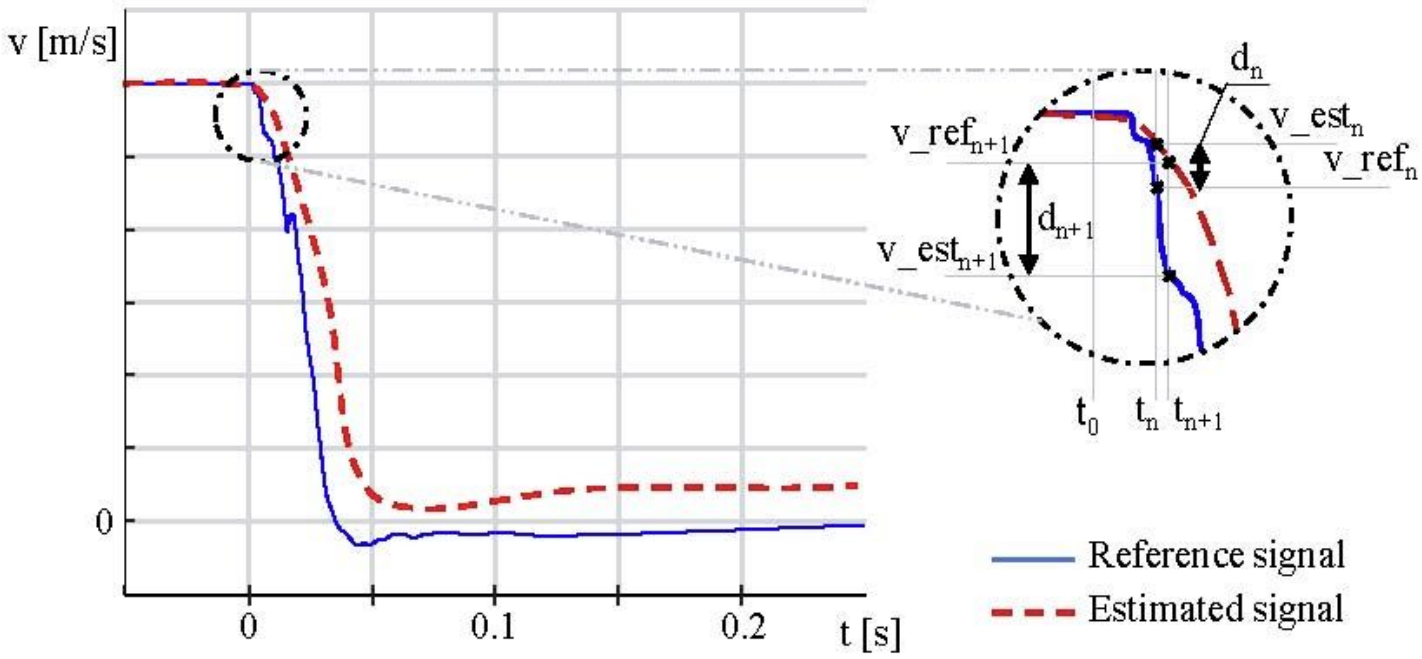

Figure 10

Evaluation of the state of the environment based on the difference between estimated and reference velocity signals

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- SuplimentaryManuscriptforrefShahabazAfraj.pdf