SoftwareX 19 (2022) 101103

Contents lists available at ScienceDirect

SoftwareX

journal homepage: www.elsevier.com/locate/softx

Original software publication

VEqMon2D—Equations of motion generation tool of 2D vehicles with Matlab

Daniel Cantero

Department of Structural Engineering, Norwegian University of Science & Technology NTNU, Trondheim, Norway

ARTICLE INFO

Article history:
Received 10 February 2022
Received in revised form 21 April 2022
Accepted 3 May 2022

Keywords:
Vehicle
Equations of motion
Dynamics
Multibody

ABSTRACT

The tool VEqMon2D provides a framework to generate analytical expressions of equations of motion of 2D vehicle models in Matlab. Such vehicle models are mostly used in studies that focus on the infrastructure (pavement, railway tracks, bridges) and are loaded by traversing traffic. The main outcome of the tool is a Matlab function that provides the analytical expressions of the equations of motion in matrix form. The user has the flexibility to define a model with multiple bodies, articulations, and axle configurations. Subsequently, the user can use the generated function to calculate the system matrices of the vehicle model for particular values of geometric and mechanical properties. This document describes the various components and modelling possibilities available in VEqMon2D and describes the relevant model parameters and degrees of freedom associated to the vehicle model. It also includes illustrative examples of the use of the tool together with the validation of the tool.

© 2022 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Code metadata

Current code version	v1
Permanent link to code/repository used for this code version	https://github.com/ElsevierSoftwareX/SOFTX-D-22-00037
Permanent link to reproducible capsule	https://codeocean.com/capsule/2440468/tree
Legal code license	GNU General Public License v3.0
Code versioning system used	none
Software code languages, tools and services used	Matlab
Compilation requirements, operating environments and dependencies	
If available, link to developer documentation/manual	
Support email for questions	daniel.cantero@ntnu.no

1. Motivation and significance

The numerical modelling of road or rail vehicles can be extremely detailed, which requires rather complex representations of each of the mechanical components of the vehicle. However, for certain applications the model complexity can be reduced to planar 2D representations while still providing sufficient level of accuracy of the vehicle responses. This is the case for studies that focus on the infrastructure and are loaded by traversing traffic. Then simpler models are made of a combination of lumped masses, rigid bodies and elastic connections using multibody dynamics. Examples for the use of these type of models can be found in road infrastructure studies focusing on pavement deterioration [1], vehicle–bridge interaction [2], or vehicle assisted bridge monitoring [3,4]. In the field of railway engineering, these simplified vehicle representations have been used, for instance, to study the vehicle–bridge interaction [5], to determine the railway track longitudinal profile from vehicle measurements [6] and to evaluate the track condition [7].

The development of 2D vehicle models could be considered as straightforward. The massive elements of a vehicle are represented as rigid bodies with concentrated masses and inertia properties, while the elastic parts (suspensions and tyres) are modelled with springs and dashpots. The derivation of the equations of motion follow from the application of Newton's second law and conservation of angular momentum [8]. However, the difficulty appears when such simple components are combined to represent vehicles with more elaborate configurations, that include multiple trailers, articulations or axles grouped together. The analytical derivation of the system matrices become laborious and prone to error. There exists commercially available multibody software or general-purpose simulation packages, such as Abaqus [9], that facilitate the development of aforementioned vehicle models. However, these solutions do not provide the

E-mail address: daniel.cantero@ntnu.no.

https://doi.org/10.1016/j.softx.2022.101103
2352-7110/© 2022 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

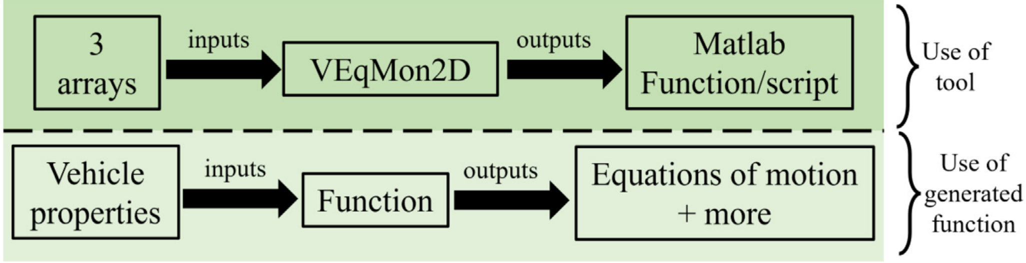

Fig. 1. Overview of VEqMon2D use and application.

analytical expressions of the equations of motion but rather the final numerical expression of the vehicle's system matrices. Nevertheless, it is advantageous to obtain the analytical expressions in terms of the geometrical and mechanical properties that define a vehicle model. This provides better insight into the vehicle model and facilitates, for instance, Monte Carlo simulations of random traffic with similar vehicle models but different numerical values for the model parameters.

Therefore, the tool VEqMon2D provides a platform to generate the analytical expressions of the equations of motion of 2D vehicle models in Matlab [10]. To use it, the user needs to define 3 input arrays that depend on the desired vehicle configuration, which specify the number of axles per body, the grouping of axles and the existence of articulations. The tool VEqMon2D generates a Matlab function with the implementation of the equations of motion of the desired vehicle, together with additional useful information. The annotations within that generated function clearly describe the vehicle parameters and degrees of freedom (DOF) that define the numerical model. As indicated in Fig. 1, the user can subsequently generate the equations of motion of that vehicle for the particular numerical values of the model parameters. The scripts provided in the VEqMon2D repository contain the functions needed to run it together with illustrative examples. The repository also includes an input file to run in Abaqus environment [9] that defines one example and is used as validation of the tool.

2. Vehicle modelling

This section presents the different configuration possibilities, vehicle components, mechanical and geometrical properties, together with the defining DOFs and relations that describe the vehicle behaviour.

The range of possibilities considered in VEqMon2D to define the vehicle model is extensive, depending on the desired type and number of components. Fig. 2 provides an overview of all the components and possibilities. First, there are the main bodies of the vehicle, that represent the tractor and subsequent trailers. Each body i is a rigid body, with mass m_{Bi} and pitching moment of inertia I_{Bi}, characterized by the location of its centre of gravity. Two neighbouring bodies can either be connected by an articulation or not. The bodies are resting on top of suspensions. Each suspension j consists of a system a spring and a dashpot, with stiffness k_{Sj} and viscous damping c_{Sj} respectively. The suspensions are then connected to either a single axle or an axle group. Whereas the single axle is represented as a lumped mass of value m_{Gj}, the axle group is a rigid body with mass m_{Gj} and pitching moment of inertia I_{Gj}. The mass and inertia properties of the axle/group represent what is generally referred to as the 'unsprung' part of the vehicle. Finally, each vehicle tyre k is represented by additional spring and dashpot systems, with properties k_{Tk} for stiffness and c_{Tk} for viscous damping.

Fig. 3 shows the variables that define the geometry of body i, axle/group j, and tyre k. The values a_i and b_i define the front and back distance from the centre of gravity of body i to the location of the neighbouring articulations. For the first body ($i = 1$) the distance a_1 can be set to zero, and for the last body ($i = n_{bodies}$) it is not necessary to define $b_{n_{bodies}}$ and is not requested as an input. If there is no articulation between bodies i and i+1 then the total $b_i + a_{i+1}$ is simply the distance between their centres of gravity. The location of the suspensions is defined as the horizontal coordinate with respect to the centre of gravity of the body they belong to. As shown in Fig. 3, the suspension j is located at d_j from the body's centre of gravity, and so negative values of d_j define suspensions located to its left. The value d_j also defines the location of a single axle or the centre of gravity of an axle group. Finally, the location of tyre k is specified with the horizontal coordinate e_k with respect to the centre of gravity of the axle/group, where negative values represent tyres located to its left. In case the axle/group is only a single axle, then the value of e_k should be set to zero.

The equations of motion of such a vehicle establish the relationships between certain DOFs. In the formulation used in VEqMon2D, all rigid bodies have two DOFs and lumped masses have one. The movement of the bodies can be defined by the vertical displacement of their centre of gravity y_{Bi} and the rotation around the out of plane perpendicular axis θ_{Bi}. Similarly, the rigid body for axle group j is defined by corresponding vertical displacements y_{Gj} and rotations θ_{Gj} with respect to its centre of gravity. In the case of individual axles, then only the vertical displacements y_{Gj} are included in the formulation. The sign convention adopted in the derivation is positive for upwards vertical displacements and anticlockwise rotations.

The implementation in VEqMon2D derives the equations of motion by ensuring equilibrium of forces and equilibrium of moments based on Newton's laws and conservation of angular momentum. The derivation assumes small rotations and no deformation of the bodies and axle groups. No additional explicit constraints to the DOFs have been included. These assumptions are generally valid for the fields of study where such planar vehicle models are used. The vehicle response can be obtained by solving the set of second order differential equations. Eq. (1) shows the generic expression of the equations of motion in terms of the system matrices (namely, mass M, damping C and stiffness K matrices), where x is the vector of DOFs to solve and F contains the values of the external forces applied to the vehicle.

$$M\ddot{x} + C\dot{x} + Kx = F \tag{1}$$

The existence of an articulation between two bodies influences the definition of DOFs. The displacements and rotations of the connected bodies by the articulation are linked together by a geometric relation [8], which reduces the number of independent DOF of the system, and thus the number of equations of motion. For two bodies i and $i+1$ connected by an articulation, the tool VEqMon2D removes the DOF corresponding to the vertical displacement of the body $i+1$, namely y_{Bi+1}. This DOF is denoted as a dependent DOF and can always be calculated from the values of the independent DOFs using the geometric relationship. For completeness, this relationship is also provided by the final function/script together with the equations of motion.

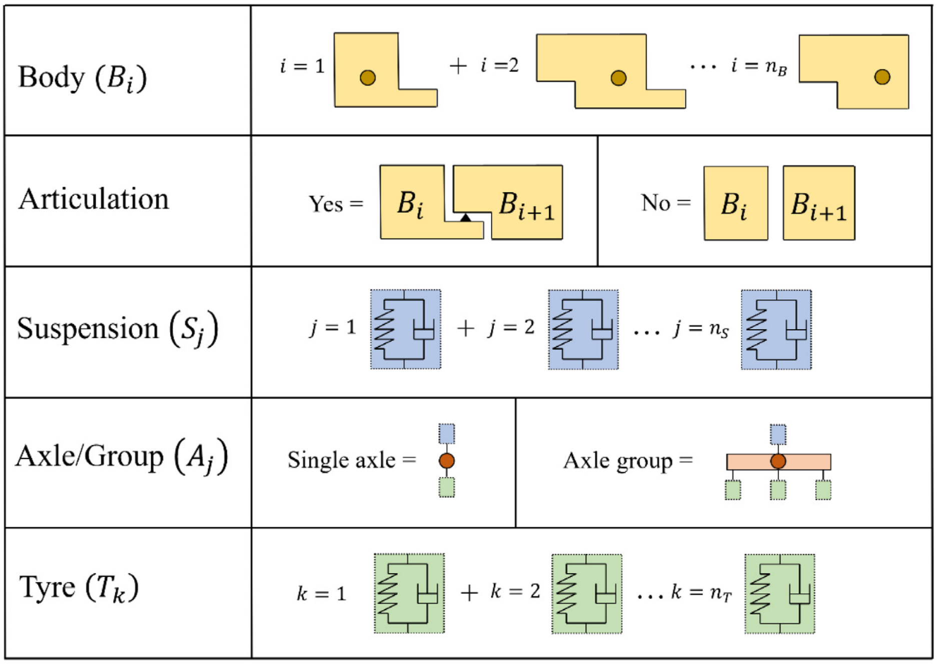

Fig. 2. Overview of vehicle model components and possibilities.

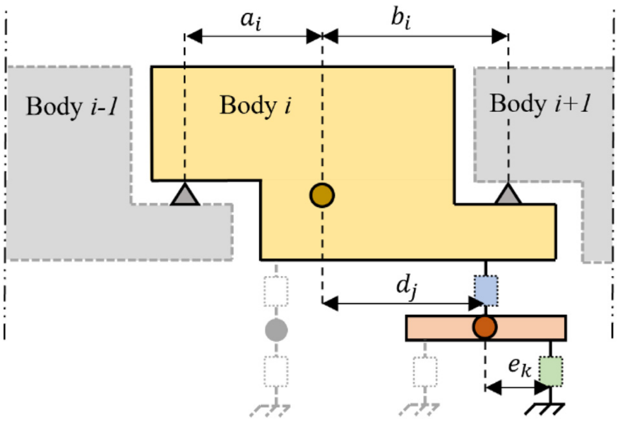

Fig. 3. Parameters for the geometry definition.

3. Software description

3.1. Software architecture

The tool VEqMon2D generates analytical expressions of planar vehicle models based on the inputs by the user. The output of the tool is a new function with helpful annotations that can be used to generate the model's system matrices for particular numerical values of the model parameters (See Fig. 1). The tool is implemented for Matlab programming language and uses the functions available in the Symbolic Math Toolbox [10].

Considering all the modelling possibilities that VEqMon2D offers it is important to define a systematic notation to uniquely label vehicle models. The adopted notation starts indicating the number of axles that each body has:

Vehicle _ (Num. of axles in body 1) _ (Num. of axles in body 2) ...

When two consecutive bodies are connected by an articulation the notation includes the letter *A* between the number of axles of the corresponding bodies. Also, if there exist groups of axles in the model, then the previous notation is extended to indicate how many axles each axle/group has as follows:

(Previous notation) _ G _ (Num. axles in group 1) _ (Num. axles in group 2) ...

As an example, the notation of a typical 5-axle articulated truck with tandem axle groups in the rear of the trailer and in the semitrailer is *Vehicle_3A2_G_1_2_2*.

To use the VEqMon2D tool the user needs to define the appropriate values of the input parameters in *A01_main.m* script and run it. This script calls the necessary functions to process the inputs, perform the operations and report the results. The input parameters are grouped in the following 3 arrays:

num_axles_per_body = Array with number of axles per body. The length of this array corresponds to the total number of bodies.

num_axles_per_group = Array with number of axles per group. The length of this array corresponds to the total number of axle/groups.

with_articulation = Array with 1/0 flags to indicate the presence (or not) of articulations. The entry *i* describes the existence of articulation between body *i* and *i+1*.

The core computations in VEqMon2D are performed in the function *B01_calculations.m*. For completeness, the most important operations performed there are:

1. Input checking, initialization, and auxiliary variables definition.
2. Symbolic definition of all possible equations of motion associated to each mass and rigid body.
3. Modification of the system of equations of motion to account for possible geometric relationships due to the presence of articulations.
4. Generation of Matlab function by printing the results into the adequate format.

3.2. Software functionalities

The main outcome of the VEqMon2D tool is the generation of a Matlab function that contains the equations of motion in matrix form (see Eq. (1)) of the desired vehicle model. The user has the flexibility to define a model with multiple bodies, articulations, and axle configurations. The result is saved in a separate m-file

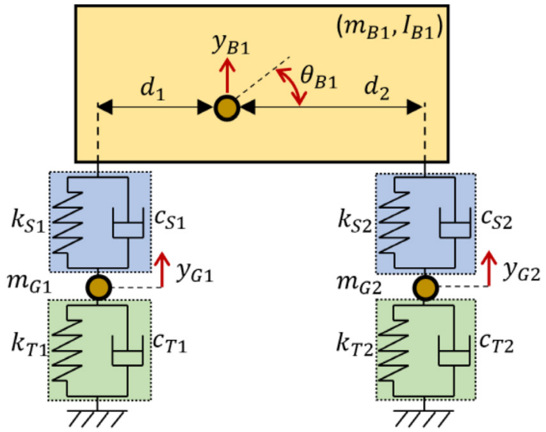

Fig. 4. Sketch of 2-axle vehicle model, indicating DOFs, mechanical properties and dimensions names.

with the name following the mentioned vehicle model notation in the destination folder specified by the user. The generated function provides clear annotations of the name and order of the DOFs for the chosen vehicle model, together with the list of vehicle parameters. The user can then define the numerical values of these parameters as inputs on the generated function to obtain the numerical expressions of the equations of motion.

In addition, the generated function includes a series of features that can be useful for the user, including:

- Anonymous functions and expressions that describe the relations between dependent and independent DOFs derived from the presence of articulations.
- The matrices necessary to transform DOFs' displacements to the vertical displacements of each wheel.
- Array with the vertical static forces from each wheel
- Array with axle spacing

The toolbox repository also includes an example on the usage of the generated vehicle model function. The file *A03_Simulation_Example.m* performs the numerical analysis of a vehicle travelling at a certain speed over a step-like irregularity. The vehicle behaviour is obtained by direct integration (with Newmark-β numerical scheme) of the system of equations of motion. The example also demonstrates how to calculate the vertical wheel forces for the time-history of nodal displacements including the effect of road/rail irregularities.

4. Illustrative examples

To demonstrate the usage of VEqMon2D, this section presents two examples. The first one reproduces the analytical expressions of the system matrices of a rather simple 2-axle vehicle model. The second example is for a complex vehicle model, which showcases most of the tool's capabilities.

4.1. Equations of motion of 2-axle vehicle

Fig. 4 shows the schematic representation of a 2-axle vehicle model, indicating the DOFs and the names of relevant geometrical and mechanical properties. The equations of motion of such a 2-axle vehicle can be computed using VEqMon2D. The generated function for this vehicle model is *Vehicle_2.m* and contains the implementation of the analytical expressions of the system matrices.

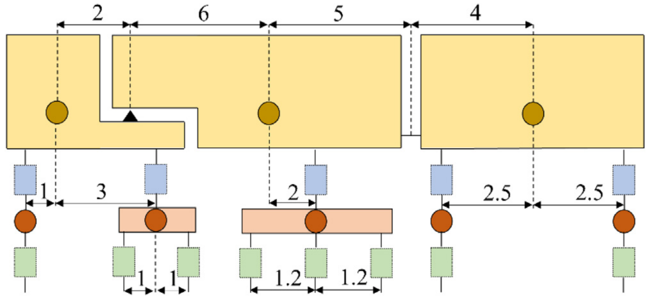

Fig. 5. Sketch of vehicle model example. Note: sketch is not to scale.

For illustrative purposes one can find below the analytical expressions of the vectors and matrices that describe the behaviour of the 2-axle vehicle in Fig. 4. Eq. (2) is the vector of DOFs. The expressions of the system matrices are reproduced for the mass (Eq. (3)), damping (Eq. (4)) and stiffness (Eq. (5)) matrices. This example shows that the analytical expressions of vehicle models with more complex configuration (and thus more DOFs) can quickly become too large to be transcribed into a document.

$$x = \left\{ y_{B1} \quad \theta_{B1} \quad y_{G1} \quad y_{G2} \right\}^T \tag{2}$$

$$M = \begin{bmatrix} m_{B1} & 0 & 0 & 0 \\ 0 & I_{B1} & 0 & 0 \\ 0 & 0 & m_{G1} & 0 \\ 0 & 0 & 0 & m_{G2} \end{bmatrix} \tag{3}$$

$$C = \begin{bmatrix} c_{S1}+c_{S2} & c_{S1}d_1+c_{S2}d_2 & -c_{S1} & -c_{S2} \\ c_{S1}d_1+c_{S2}d_2 & c_{S1}d_1^2+c_{S2}d_2^2 & -c_{S1}d_1 & -c_{S2}d_2 \\ -c_{S1} & -c_{S1}d_1 & c_{S1}+c_{T1} & 0 \\ -c_{S2} & -c_{S2}d_2 & 0 & c_{S2}+c_{T2} \end{bmatrix} \tag{4}$$

$$K = \begin{bmatrix} k_{S1}+k_{S2} & k_{S1}d_1+k_{S2}d_2 & -k_{S1} & -k_{S2} \\ k_{S1}d_1+k_{S2}d_2 & k_{S1}d_1^2+k_{S2}d_2^2 & -k_{S1}d_1 & -k_{S2}d_2 \\ -k_{S1} & -k_{S1}d_1 & k_{S1}+k_{T1} & 0 \\ -k_{S2} & -k_{S2}d_2 & 0 & k_{S2}+k_{T2} \end{bmatrix} \tag{5}$$

4.2. Use of VEqMon2D for a complex configuration

Fig. 5 shows schematically the vehicle model of a complex configuration that can only be described by a rather large system of equations. It consists of a tractor with a single axle at the front and a tandem axle group at the back, followed by a semitrailer resting on an articulation on the tractor and a tridem axle group, completed with a towed trailer with two separate single axles. This example has a total of 8 tyres and 12 independent DOFs (plus one dependent DOF because of the articulation between bodies 1 and 2). The notation of such a vehicle model is *Vehicle_3A3_2_G_1_2_3_1_1* and its corresponding input arrays are:

```
Inputs.num_axles_per_body = [3,3,2];
Inputs.num_axles_per_group = [1,2,3,1,1];
Inputs.with_articulation = [1,0];
```

The script *A01_Main.m* with these input arrays generates a Matlab function with the name *Vehicle_3A3_2_G_1_2_3_1_1.m* that contains the formulation of the vehicle's equations of motion. Now this function can be used to find the numerical representations of the system matrices for particular vehicle properties.

The inputs necessary to run the function are explained within the function and an example of their definition is shown below.

```
Veh.Prop.mBi = [5e3, 30e3, 20e3];           % [kg]
Veh.Prop.IBi = [4e3, 20e3, 10e3];           % [kg*m2]
Veh.Prop.kSj = [0.5, 2, 3, 1, 1]*1e6;       % [N/m]
Veh.Prop.cSj = [1, 2, 3, 1, 1]*1e3;         % [N*s/m]
Veh.Prop.mGj = [1, 2, 3, 1, 1]*750;         % [kg]
Veh.Prop.IGj = [0, 2, 3, 0, 0]*1e2;         % [kg*m2]
Veh.Prop.kTk = [1.75, 3.5, 3.5, 3.5, 3.5, ...
    3.5, 3.5, 3.5]*1e6;                     % [N/m]
Veh.Prop.cTk = [1, 1, 1, 1, 1, 1, 1, 1]*1e4;% [N*s/m]
Veh.Prop.ai = [0, 6, 4];                    % [m]
Veh.Prop.bi = [2, 5];                       % [m]
Veh.Prop.dj = [-1, 3, 2, -2.5, 2.5];        % [m]
Veh.Prop.ek = [0, -1, 1, -1.2, 0, 1.2, 0,0];% [m]
```

For this example, the mechanical properties are chosen to be comparable to those found in literature [8] for similar vehicle models. Fig. 5 also shows the geometric properties of the example, namely, distances between bodies and articulations (a_i, b_i), locations of the suspensions with respect to the bodies (d_j), and location of the tyres with respect to the axle/group (e_k).

At the same time, this example is also used for the validation of VEqMon2D. The vehicle model is built independently as an Abaqus model via the software's own input file structure. A copy of this file (*Abq_Vehicle_3A3_2_G_1_2_3_1_1.inp*) is also include in the repository. The modal analysis of the model from VEqMon2D (using *A02_Validation_Vehicle_3A3_2_G_1_2_3_1_1.m*) is compared to the modal analysis of the Abaqus model. The list of natural frequencies from both models match exactly, thus validating the presented tool.

5. Impact

The tool VEqMon2D has been already used in recent publications, see [11,12]. These numerical studies required the use of more elaborated vehicle models with different body configurations and axle arrangements. The functions obtained from VEqMon2D facilitated the generation of many vehicle passages with different parameter values, emulating the loading produced by random traffic.

In addition, there is a need of more studies that address the effect of new vehicle types on the infrastructure. In recent years, there has been an increase of the types of vehicle configurations allowed to circulate over roads worldwide, including conventional trucks with multiple trailers. But there has been also a surge in numbers of special transport vehicles with multiple closely spaced axles and moving cranes with unconventional axle spacing [2]. These vehicle configurations can be easily modelled with VEqMon2D.

Furthermore, the tool also opens the door for future developments and improvements on vehicle modelling. It provides a systematic framework to work with symbolic variables and derive analytical expressions. The intention of the clear notation and extended annotations is to provide an intuitively easy to follow implementation. That allows future users to easily adapt the provided scripts and the resulting functions to their own needs.

6. Conclusions

The tool VEqMon2D generates analytical expressions of planar vehicle models based on the inputs by the user. The output of the tool is a new function with helpful annotations that can be used to generate the model's system matrices for particular numerical values of the model parameters The tool is implemented for Matlab programming language and uses the functions available in the Symbolic Math Toolbox. The document describes a list of possible modelling components and configurations, the relevant geometrical and mechanical properties and the degrees of freedom that describe the numerical models. The tool description is completed with illustrative examples and validation case. The tool has been used previously in published studies and has the potential to be used frequently in new studies, while at the same time it allows for further developments if necessary.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Taheri A, OBrien EJ, Collop AC. Pavement damage model incorporating vehicle dynamics and a 3D pavement surface. Int J Pavement Eng 2012;13(4):374–83. http://dx.doi.org/10.1080/10298436.2012.655741.

[2] OBrien EJ, Cantero D, Enright B, González A. Characteristic dynamic increment for extreme traffic loading events on short and medium span highway bridges. Eng Struct 2010;32(12):3827–35. http://dx.doi.org/10.1016/j.engstruct.2010.08.018.

[3] Nguyen T, Swolana P, Lechner B, Wong YD. An experimental comparison of mathematical heavy-duty city bus models to evaluate passenger ride comfort induced by road roughness. Math Comp Mod Dyn Sys 2021;27:203–21. http://dx.doi.org/10.1080/13873954.2021.1887276.

[4] Malekjafarian A, McGetrick P, OBrien EJ. A review of indirect bridge monitoring using passing vehicles. Shock Vib 2015;286139. http://dx.doi.org/10.1155/2015/286139.

[5] Cantero D, Arvidsson T, OBrien EJ, Karoumi R. Train–track–bridge modelling and review of parameters. Struct Infrastruct Eng 2016;12(9):1051–64. http://dx.doi.org/10.1080/15732479.2015.1076854.

[6] OBrien EJ, Quirke P, Bowe C, Cantero D. Determination of railway track longitudinal profile using measured inertial response of an in-service railway vehicle. Struct Health Monit 2018;17(6):1425–40. http://dx.doi.org/10.1177/1475921717744479.

[7] Malekjafarian A, OBrien EJ, Quirke P, Cantero D, Golpayegani F. Railway track loss-of-stiffness detection using bogie filtered displacement data measured on a passing train. Infrastructures 2021;6(6):93. http://dx.doi.org/10.3390/infrastructures6060093.

[8] Cantero D, O'Brien EJ, González A. Modelling the vehicle in vehicle–infrastructure dynamic interaction studies. Proc IMechE Part K: J Multi-Body Dyn 2010;224(2):243–8. http://dx.doi.org/10.1243/14644193JMBD228.

[9] Smith M. ABAQUS/Standard User's Manual, Version 2019. Providence, RI: Dassault Systèmes Simulia Corp; 2019.

[10] The Math Works, Inc. MATLAB, Version 2020a. The Math Works, Inc; 2020.

[11] Cantero D. Moving point load approximation from bridge response signals and its application to bridge weigh-in-motion. Eng Struct 2021;233:111931. http://dx.doi.org/10.1016/j.engstruct.2021.111931.

[12] Sarwar MZ, Cantero D. Deep autoencoder architecture for bridge damage assessment using responses from several vehicles. Eng Struct 2021;246:113064. http://dx.doi.org/10.1016/j.engstruct.2021.113064.