A fast mode superposition algorithm and its application to the analysis of bridges under moving loads

A. Camara

Department of Civil Engineering. City, University of London. Northampton Square, London, EC1V 0HB, United Kingdom

Abstract

Modal superposition (MS) techniques are widely used in linear dynamic analyses but the study of large structures subject to actions such as long vehicle convoys is not currently feasible. This work presents an algorithm called MS5 embedded in the Python library MDyn that combines an efficient indexing strategy to deactivate specific structural nodes and movements with a novel modal truncation based on a new dynamic participation factor and the vectorisation of the MS algorithm. The MS5 is compared with the conventional MS method, the analytical solution and the commercial finite element software ABAQUS in the dynamic analysis of a short-span beam bridge and of a large cable-stayed bridge under different load scenarios. The results obtained with MS5 are almost identical to other methods, but it is on average 9 times faster than the standard MS method. The proposed algorithm is applied to the study of critical traffic actions in both structures, observing important dynamic amplification effects for certain convoy arrangements that are able to trigger resonant responses in the deck. MS5 is also applicable to other line-like structures such as towers, masts, etc.

Keywords:
modal superposition; structural dynamics; vectorised programming; bridges; vehicle convoys; resonance.

Email address: alfredo.camara@city.ac.uk (A. Camara)

1. Introduction

The mode superposition (MS) method is widely used in the dynamic analysis of structures in elastic range because of its reduced computational cost and accuracy compared to the approaches based on the direct integration (DI) of the system of dynamics such as the HHT algorithm [1]. Considering a structure with N degrees of freedom (DOF), the MS method decomposes the N coupled differential equations of motion into a reduced set of J uncoupled single-DOF differential equations that are related to the response of each vibration mode, which are later superimposed to obtain the total structural motion [2]. The reduced computational time of the MS method is due to (1) the separate time-history solution of each SDOF equation of motion, and (2) the reduction in the size of the problem by including in the analysis only the first J vibration modes that are important for the structural response, with $J < N$. This work presents a fast general-purpose MS algorithm developed in the high-level programming language Python [3] that exploits the benefits of array-based vectorisation, modal truncation, deactivation of structural nodes and degrees of freedom.

Within the field of structural engineering, the MS method is applied to solve a large number of dynamic problems involving seismic actions, live loads and wind, among others. Camara and Astiz [4] demonstrated that the MS method is more accurate than the HHT algorithm in the analysis of the elastic seismic response of cable-stayed bridges. This is because the MS method avoids introducing artificial period elongations, which can be significant in the HHT solution of high-frequency vibrations [1]. Integration errors can be reduced or eliminated with DI algorithms based on state-space formulations [5, 6], but their computational time can be much higher than the one required by MS decoupling techniques in the dynamic analysis of structures with a large number of DOF. However, the standard MS method is limited by the Caughey-O'Kelly condition, which usually implies stiffness- and mass-proportional damping and makes it only applicable to classically damped structures. Foss and others [7, 8, 9] extended the standard MS approach to include vibration modes with complex values that appear in structures with non-classical damping due to the presence of seismic base isolation, supplemental energy dissipation devices, etc. [10]. Nevertheless, the added complexity of extended MS methods may not be justified in the dynamic analysis of conventional structures under service conditions. This is particularly the case in the study of traffic-induced vibrations in bridges. In such studies the vehicles or the trains can be defined as moving loads (see e.g. [11, 12, 13]), masses (e.g. [14]) or multi-DOF sub-systems (e.g. [15, 16, 17]). In most cases, describing the vehicles as moving loads is appropriate if the goal is to obtain the global response of the structure and not the assessment of the vehicle vibrations.

None of the above references address the implementation and optimisation of the MS analysis in a programming language. This work presents different strategies to accelerate the standard MS method, and develops a new algorithm referred to as MS5 that exploits the computational efficiency of array-based operations in Python [3]. The main novelties of MS5 are the vectorisation of the MS algorithm and a selective modal truncation that filters the modal matrix with the indices of the relevant vibration modes below a certain threshold. These two techniques are combined with an efficient way of selecting the relevant parts of the structure in the analysis and the deactivation of structural motions that are not of interest. The result is a fast and accurate MS algorithm implemented in a Python library called MDyn that is applied to the analysis of the traffic-induced vibrations in a simply supported beam and in a large cable-stayed bridge. The results obtained with MS5 are almost identical to those given by the commercial FE software ABAQUS [18], and in the case of the simply supported beam they are also very close to the existing analytical solution. However, MS5 is on average 9 times faster than the standard MS solution in the dynamic analysis of both structures, and it is mainly thanks to the proposed vectorisation and the modal truncation. Finally, MS5 is used to analyse a large number of vehicle arrangements, beyond the limits of existing commercial software, and it is observed that certain spacing between consecutive trucks can lead to significant resonant effects.

2. Accelerated mode superposition method

Let's consider a three-dimensional (3D) structure discretised with beam-type elements interconnected at N_n nodes with 6 different types of structural movements (SM, three translations and three rotations) per node, giving a total of $N = 6N_n$ degrees of freedom (DOF). The equations of motion can be obtained by applying the D'Alembert principle in terms of the generalised nodal displacements ($\mathbf{r}_{s|N\times 1}$, with the sub-index product denoting the array dimensions: rows $\times$ columns) and their time-derivatives ($\dot{\mathbf{r}}_{s|N\times 1}$ for the velocities and $\ddot{\mathbf{r}}_{s|N\times 1}$ for the accelerations)

$$\mathbf{M}_s\ddot{\mathbf{r}}_s(t) + \mathbf{C}_s\dot{\mathbf{r}}_s(t) + \mathbf{K}_s\mathbf{r}_s(t) = \mathbf{P}_s(t), \tag{1}$$

where $\mathbf{M}_{s|N\times N}$, $\mathbf{C}_{s|N\times N}$ and $\mathbf{K}_{s|N\times N}$ are the mass, damping and stiffness matrices of the structure; $\mathbf{P}_{s|N\times 1}(t)$ is the nodal forcing vector, which includes the generalised forces in each structural DOF due to e.g. moving vehicles, wind, ground motions, etc.; t represents the time. The $N\times N$ coupled system of equations of motion can be decomposed when it is expressed in the space of its N orthogonal vibration mode shapes $\mathbf{\Phi}_{|N\times N} = \{\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \cdots, \boldsymbol{\phi}_N\}$, with $\boldsymbol{\phi}_{j|N\times 1}$ being the j-th mode shape of the structure. For reasons that will become clear in Section 2.3, the rows of the mode shape matrix $\mathbf{\Phi}$ are conveniently arranged by grouping together the modal coordinates corresponding to each SM, as it is illustrated in Fig. 1(a). $\mathbf{\Phi}$ is obtained from the generalised eigenvalue problem

$$\mathbf{K}_s\mathbf{\Phi} = \left(\mathbf{\Omega}.^2\right)^{\mathrm{T}} .* \left(\mathbf{M}_s\mathbf{\Phi}\right), \tag{2}$$

where $\mathbf{\Omega}_{|N\times 1} = \{\omega_1, \omega_2, \cdots, \omega_N\}^{\mathrm{T}}$ and ω_j is the circular frequency of the j-th mode. In this work the matrix-level operators are omitted for convenience and the element-level operators are preceded by a dot symbol: consequently, in Eq. (2) the operators $(.^2)$ and $(.*)$ represent the element-wise square and multiplication of each scalar frequency contained in $\mathbf{\Omega}$, respectively. Thanks to the orthogonality of $\mathbf{\Phi}$ (and assuming classical damping) Eq. (1) can be decoupled into a system of N independent SDOF differential equations. The time-history contribution of the j-th vibration mode to the structural response is represented with the modal coordinate $q_j(t)$, which can be obtained from

$$\ddot{q}_j(t) + 2\xi_j\omega_j\dot{q}_j(t) + \omega_j^2 q_j(t) = \frac{\boldsymbol{\phi}_j^{\mathrm{T}}\mathbf{P}_s(t)}{m_j}, \tag{3}$$

in which ξ_j and m_j are the modal damping ratio and the modal mass of the j-th mode, respectively. If the mode shapes are normalised with respect to their modal mass: $m_j = \boldsymbol{\phi}_j^{\mathrm{T}}\mathbf{M}_s\boldsymbol{\phi}_j = 1$. In that case the right hand side of this equation, which is referred to as the modal forcing, is simplified as $P_j(t)_{|1\times 1} = \boldsymbol{\phi}_j^{\mathrm{T}}\mathbf{P}_s(t)$. Finally, the response of the structure can be obtained by superposition of the modal responses and their time-derivatives

$$\mathbf{r}_s(t) = \sum_{j=1}^{N}\boldsymbol{\phi}_j q_j(t);\ \dot{\mathbf{r}}_s(t) = \sum_{j=1}^{N}\boldsymbol{\phi}_j \dot{q}_j(t);\ \ddot{\mathbf{r}}_s(t) = \sum_{j=1}^{N}\boldsymbol{\phi}_j \ddot{q}_j(t). \tag{4}$$

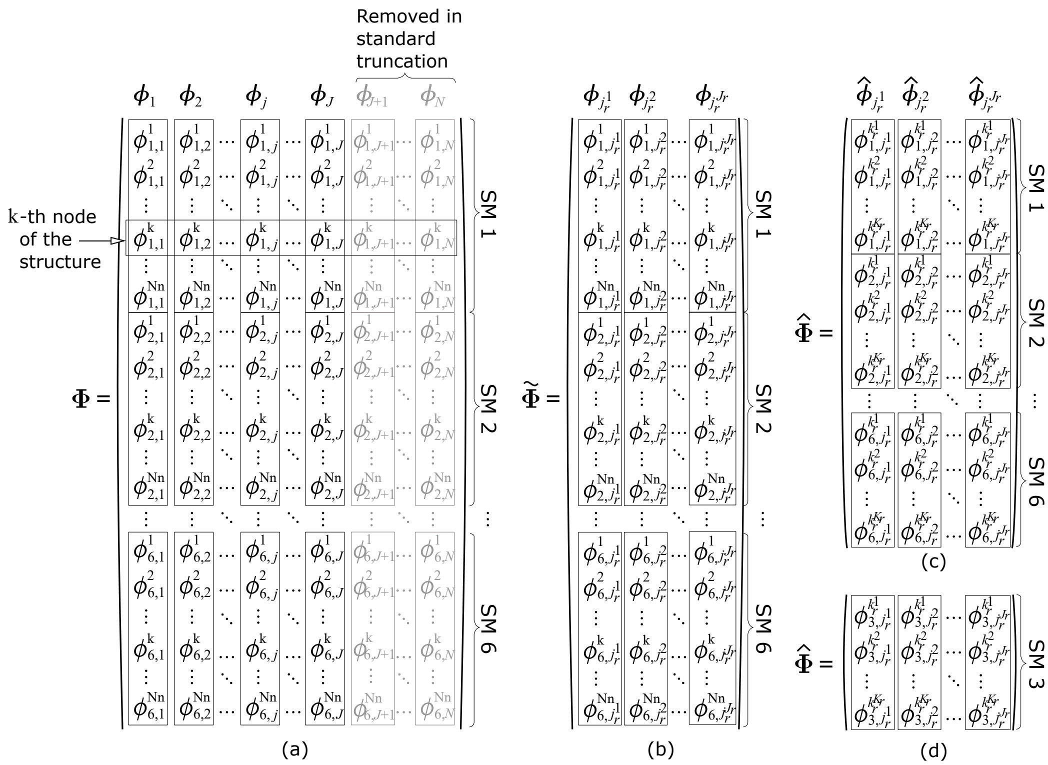

Figure 1: Different forms of the mode shape matrix $\mathbf{\Phi}$; (a) full matrix and standard modal truncation; (b) reduced matrix with indexed modal truncation; (c) reduced matrix from (b) with nodal deactivation; (d) reduced matrix from (c) with deactivation of all the SM except the third one (example).

This work presents four ways to accelerate the standard MS method described above, namely the indexed modal truncation, the deactivation of nodes and SM, and the array-based vectorised programming of the MS algorithm. These strategies are implemented as sub-routines in a Python library called MDyn, and they are used to analyse the dynamic responses of two different bridges subject to traffic actions in Sections 3, 4 and 5.

2.1. Indexed modal truncation

One of the advantages of the classical MS method is the ability to describe the structural response using a reduced set of J vibration modes, with $J < N$. The value of J depends on the structure and on the response of interest, and it is generally considered as an upper bound limit below which all the vibration modes are included in the analysis ($j = 1, 2, \cdots, J$), with the modes above being removed from the superposition ($j = J + 1, \cdots, N$), see e.g. [19, 20]. This standard modal truncation approach is illustrated in Fig. 1(a). While it can reduce the computational time significantly, it is inadequate in structures with a significant number of vibration modes of order lower than J that have a negligible contribution to the dynamic response of interest. This is the case in many structures, for example in those with interconnected members of significantly different stiffnesses (e.g. the deck and the piers of long-span cable-supported bridges) or in structures that are much stiffer in one direction than in the others (e.g. the vertical response of laterally-flexible buildings, wind turbine towers, chimneys or masts).

An additonal problem of the MS method is related to the selection criteria for the vibration modes to be included in the dynamic analysis. This is usually done in terms of modal properties that are inherent to the structure but independent of the dynamic loads to which it is subject, such as the modal participation factors or the modal activated mass. However, these parameters can fail to describe the importance of vibration modes in which the mass of different parts of the structure moves in opposite direction. That is for example the case when considering the influence of antisymmetric or vertical modes in the vehicle-induced response of continuous decks. This is illustrated in Section 4.1 for a long cable-stayed bridge, where it is also demonstrated that moving vehicle actions can excite different vibration modes in a way that cannot be captured by the structure-related modal parameters.

In order to account for dynamic loading effects in the contribution of different modes it is proposed here to define a dynamic contribution ratio of the vibration mode j in the direction SM at node k as

$$\eta^k_{\mathrm{SM},j} = \frac{\phi^k_{\mathrm{SM},j} \max\limits_t (q_j(t))}{\max\limits_t \left(r^k_{\mathrm{SM},J}(t)\right)}, \tag{5}$$

where the numerator represents the peak structural movement at node k in the direction SM due to the contribution of the vibration mode k exclusively (i.e. only mode k is activated in the MS analysis); the denominator of Eq. (5) refers to the peak response at the same node and directon when all the vibration modes below a certain threshold J are activated in the MS analysis ($r^k_{\mathrm{SM},J}$).

A selective modal truncation based on the dynamic modal contribution factor η is proposed. The idea is to create an index vector $\mathbf{j}_{r|J_r\times 1}$ that contains the order (position) of all the J_r modes below J that are important for the response (with $J_r \leq J$): $\mathbf{j}_r = \{j^1_r, j^2_r, \cdots, j^{J_r}_r\}^{\mathrm{T}}$, where j^j_r is the order of the j-th mode that is relevant for the structural response of interest. The index vector $\mathbf{j}_r$ can be defined from the dynamic modal factor η as

$$j^j_r \in \mathbf{j}_r \text{ if } \eta^k_{\mathrm{SM},j} \geq \eta_{\min} \text{ and if } j \leq J \tag{6}$$

which means that only the vibration modes below J and for which the dynamic modal contribution is above a certain threshold ($\eta_{\min}$) are considered in the MS analysis. This can be implemented in a programming language with a `for`-loop running over all the vibration modes below J, and an inner `if`-statement that checks the values of η in directions (SM) and nodes (k) that are of interest to the analyst, grouping them in the vector of indices $\mathbf{j}_r$ as shown in the schematic Python code of Fig. 2(a). It is remarked that $\mathbf{j}_r$ does not need to be sorted in ascending order and its elements are generally not consecutive; for example $\mathbf{j}_r = \{1, 3, 8\}^{\mathrm{T}}$ indicates that only the first, third and eighth modes are to be included in the MS analysis. After defining $\mathbf{j}_r$ the full vibration mode matrix ($\mathbf{\Phi}_{|N\times N}$) can be reduced by isolating the columns that correspond to the indices contained in $\mathbf{j}_r$, and removing the rest. Analogously, the column vectors that include the modal frequencies and their time-history coordinates ($\mathbf{\Omega}$ and $\mathbf{q}$, respectively) are reduced by isolating the rows that correspond to the $\mathbf{j}_r$ indices. In the Python programming language these operations are automatic and can be expressed in a compact way as

$$\widetilde{\mathbf{\Phi}} = \mathbf{\Phi}[:, \mathbf{j}_r];\ \widetilde{\mathbf{\Omega}} = \mathbf{\Omega}[\mathbf{j}_r];\ \widetilde{\mathbf{q}} = \mathbf{q}[\mathbf{j}_r], \tag{7}$$

```
jr=[]
for j in range(J): # Loop in all modes
    if η^k_SM,j ≥ η_min:
        jr.append(j)
```

(a)

```
kr=[]
for k in range(Nn):  # Loop in all nodes
    if Node[k] in SelectedNodes:
        kr.append(k)
```

(b)

Figure 2: Selection of (a) relevant vibration modes and (b) relevant structural nodes for the MS analysis in the Python environment. The symbol # indicates that the following text is a comment in the code.

where $\widetilde{\mathbf{\Phi}}_{|N\times J_r}$, $\widetilde{\mathbf{\Omega}}_{|J_r\times 1}$ and $\widetilde{\mathbf{q}}_{|J_r\times 1}$ are the indexed mode-truncated arrays (matrices or vectors) with the modal shapes, frequencies and time-history coordinates, respectively; '[]' represents the array indexing and ':' calls *all* the elements in the corresponding direction (rows or columns). The proposed indexed modal truncation is illustrated in Fig. 1(b), where the potential reduction of the size of the problem compared with the classical truncation in Fig. 1(a) can be appreciated.

2.2. Deactivation of nodes in the FE model

Previous attempts to reduce the total number of DOF (N) involved in the MS analysis focused on reducing the size of the structural model from which $\mathbf{\Phi}$ is extracted. This has been done by reducing the number of DOF per element in the FE model from which the modal properties of the structure are extracted, or by increasing its mesh size. However, simplifying the FE model may affect the accuracy of the modal properties extracted from it, and therefore the MS results. In other cases there may be elements that can affect significantly the global structural response but that do not need to be included in the MS analysis because (1) they are not subject to any direct dynamic action, and (2) their response is not required or it can be calculated indirectly from the responses of other structural nodes included in the analysis. This is for example the case of cable elements or the non-structural masses in buildings or in bridges.

This problem can be alleviated by removing certain groups of nodes from the dynamic analysis after conducting the modal study, where all the necessary detail and mesh refinement is included in the FE model to extract the modal information ($\mathbf{\Phi}$ and $\mathbf{\Omega}$) accurately. Following this modal study, the time-history analysis concentrates on the group of K_r nodes of the structure where the dynamic actions are applied and/or output is required. Because $K_r < N_n$, the number of rows in $\mathbf{\Phi}$ for the MS analysis is reduced, and the MS solver and the post-processing of the results are faster. This is implemented in the proposed MS framework by creating an index vector $\mathbf{k}_{r|K_r\times 1} = \{k_r^1, k_r^2, \cdots, k_r^{K_r}\}^{\mathrm{T}}$ in which k_r^k is the order of the k-th structural node to be included in the MS analysis. Fig. 2(b) shows a Python code that creates the vector $\mathbf{k}_r$, where `Node` and `SelectedNodes` are lists that contain the numbers of all the nodes in the FE model and the numbers of those that will remain active in the subsequent MS dynamic analysis, respectively. Other conditions can be included to obtain $\mathbf{k}_r$, for example including only nodes that satisfy a particular geometric condition. If there are 6 SM per node, the reduced mode shape matrix after the proposed nodal deactivation (without modal truncation) is given automatically as $\mathbf{\Phi}_{|6K_r\times N}[\mathbf{k}_r,:]$. Fig. 1(c) shows the reduced matrix after both the indexed modal truncation and the nodal deactivation are applied, which can be expressed as

$$\hat{\mathbf{\Phi}} = \widetilde{\mathbf{\Phi}}[\mathbf{k}_r,:] = \mathbf{\Phi}[\mathbf{k}_r,\mathbf{j}_r], \tag{8}$$

where $\hat{\mathbf{\Phi}}_{|6K_r\times J_r}$ is the reduced mode shape matrix that results from both operations.

One of the benefits of the proposed nodal deactivation at the MS level is that the modal analysis can be obtained from a very refined FE model without affecting significantly the computational time. This is because the modal analysis only needs to be conducted once and then it can be used in the subsequent MS dynamic analysis with different loading parameters (e.g. different wind speeds, vehicle velocities or earthquake records), without the need to perform the modal analysis again.

2.3. Deactivation of structural movements (SM)

The total number of DOF involved in the MS analysis and also in its postprocessing can be reduced further if certain SM that are not of interest and that are not directly involved in the forcing vector can be ignored (or deactivated) in the dynamic analysis. For example, this is beneficial in the study of a straight bridge deck subject to vehicles moving without eccentricity, where only the vertical displacement of the deck is of interest and other SM can be ignored. The SM deactivation is also applicable in buildings under wind excitation if only the along- and across-wind responses are needed, in which case the vertical movement of the structural nodes and their rotations around the two horizontal axes could be ignored.

To this end, it is proposed to conduct the modal analysis from a fully 3D FE model (with three translational and three rotational SM per node in beam-like models), perform the indexed truncation and the nodal deactivation to obtain the reduced mode matrix $\hat{\mathbf{\Phi}}$, and finally include in the MS analysis only the rows of the mode matrix that correspond to the relevant SM. This is facilitated by arranging the mode shape vectors $\boldsymbol{\phi}_j$ so that the DOF related to the same SM appear as a horizontal block of consequtive rows in $\hat{\mathbf{\Phi}}$, as shown in Fig. 1. If the number of interesting structural movements is SM_a, the reduced mode matrix is $\hat{\mathbf{\Phi}}_{|\mathrm{SM}_a K_r \times J_r}$, which can be significantly smaller than its original size $\hat{\mathbf{\Phi}}_{|N \times N}$, with $N = 6N_n$ ($6 \geq \mathrm{SM}_a$, $N_n \geq K_r$ and $N \geq J_r$). Fig. 1(d) shows as an example how the mode shape matrix is reduced if only the vertical response (SM 3) of the sub-set of K_r nodes is of interest (i.e. $\mathrm{SM}_a = 1$ and $\hat{\mathbf{\Phi}}_{|K_r \times J_r}$). It should be mentioned that even if certain SM are deactivated in the MS analysis, they are considered in the preceding modal study and therefore the active modal components included in the reduced matrix $\hat{\mathbf{\Phi}}$ indirectly account for those SM.

2.4. Vectorisation with array-based programming

The conventional implementation of the MS method in a computer program has an outer `for`-loop that ranges over time (t) and an inner loop that ranges over the vibration modes (j), as it is conceptually described in Fig. 3(a). However, codes with nested loops are inefficient in the study of large problems and they can be significantly accelerated if they are vectorised. The idea is to substitute the j-loop in the standard MS method by array-based operations as shown in Fig 3(b), which adopts the Python environment without losing generality.

Comparing the two codes in Fig. 3 it can be seen that both start by loading the `numpy` library (which contains a large number of built-in numerical operations), and then creating the analysis time vector ($\mathbf{t}_v$) and zero-valued variables that will contain the modal properties and the structural responses (memory allocation). The first difference between the two codes appears at the calculation of the modal properties for each vibration mode. The non-vectorised code loops over all the J_r modes to be considered in the MS method (*Lines 3-6* in Fig. 3(a)) to obtain each component of the reduced modal mass ($\widetilde{\mathbf{M}}$), damping ($\widetilde{\mathbf{C}}$) and stiffness ($\widetilde{\mathbf{K}}$) vectors by means of scalar operations: $\widetilde{\mathbf{M}} = \{m_1, \cdots, m_{J_r}\}^{\mathrm{T}}$ (in which $\widetilde{\mathbf{M}} = \mathbf{1}$ if the modes are mass-normalised), $\widetilde{\mathbf{C}} = \{2\xi_1\omega_1, \cdots, 2\xi_{J_r}\omega_{J_r}\}^{\mathrm{T}}$, and $\widetilde{\mathbf{K}} = \{\omega_1^2, \cdots, \omega_{J_r}^2\}^{\mathrm{T}}$. On the

```
import numpy as np
t_v=np.arange(0.,t_max,Δt) # Time vector
# Memory allocation goes here ...
# Modal properties:
for j in j_r:          # Loop in mode
    M̃[j]=1                  # Modal mass
    C̃[j]=2*ξ_j*ω_j          # Modal damping
    K̃[j]=ω_j*ω_j            # Modal stiffness
for t in range(len(t_v)): # Loop in time
    P̂_s=P̂_v=P̂_w=np.zeros(K_r*SM_a)# Initialise forcing
    r̂_s[t]=np.zeros((K_r,1))   # Initialise response
    r̂̇_s[t]=np.zeros((K_r,1))     at time t_v[t]
    r̂̈_s[t]=np.zeros((K_r,1))
    # Nodal force calculation goes here ...
    P̂_s=P̂_v+P̂_w+P̂_e+···   # Total nodal fording
    for j in j_r:       # Loop in mode
        # Modal forcing:
        # P_j=np.dot(np.transpose(φ̂_j),P̂_s/M̃[j])
        P_j=np.dot(np.transpose(φ̂_j),P̂_s) # If M̃=1
        q_j,q̇_j,q̈_j=... #Scalar solution Eq. (3)
        r̂_s[t]=r̂_s[t]+φ̂_j*q_j   # Structural response
        r̂̇_s[t]=r̂̇_s[t]+φ̂_j*q̇_j     at time t_v[t]
        r̂̈_s[t]=r̂̈_s[t]+φ̂_j*q̈_j     after loop j is exhausted
```

(a)

```
import numpy as np
t_v=np.arange(0.,t_max,Δt) # Time vector
# Memory allocation goes here ...
# Modal properties:
M̃=np.ones(J_r,1)          # Modal mass
C̃=2*np.multiply(ξ̃,Ω̃)# Modal damping
K̃=np.multiply(Ω̃,Ω̃)  # Modal stiffness
P̃_0=np.zeros((J_r,1))  # Initialise modal force
for t in range(len(t_v)): # Loop in time
    # Initialise Newmark:
    K̄,a,b,q̈̃_0=InitVNewmark(P̃_0,other args.)
    P̂_s=P̂_v=P̂_w=np.zeros(K_r*SM_a) # Initialise forcing
    # Nodal force calculation goes here ...
    P̂_s=P̂_v+P̂_w+P̂_e+···  # Total nodal forcing
    # Modal forcing:
    # P̃=np.dot(np.transpose(Φ̂),P̂_s/M̃)
    P̃=np.dot(np.transpose(Φ̂),P̂_s) #If M̃=1
    #Newmark solution to Eq. (3) for all the modes
    q̃,q̃̇,q̃̈=VNewmark(P̃,P̃_0,K̄,a,b,q̈̃_0,other args.)
    r̂_s[t]=np.dot(Φ̂,q̃)   # Structural response
    r̂̇_s[t]=np.dot(Φ̂,q̃̇)     at time t_v[t]
    r̂̈_s[t]=np.dot(Φ̂,q̃̈)
    P̃_0=P̃
```

(b)

Figure 3: Conceptual implementation of the MS method in Python programming: (a) without vectorisation; (b) proposed vectorisation. The symbol # indicates that the following text is a comment in the code.

other hand, the vectorised code calculates the same modal properties in single array-based operations (*Lines 4-6* in Fig. 3(b)).

However, the calculations performed within the time-loop are more important from the point of view of the computational time. In the t-th time-step of the analysis, corresponding to the same component of the time vector ($\mathbf{t}_v$), the non-vectorised code first initialises the structural response vectors for the selected K_r nodes ($\hat{\mathbf{r}}_s(t)$ and its time-derivatives), as shown in *Lines 9-11* of Fig. 3(a). Then it calculates the forcing in each node of the selected sub-set of structural members ($\hat{\mathbf{P}}_s$), and it repeats for each of the selected J_r vibration modes (1) the calculation of the modal forcing (P_j in *Line 14* of Fig. 3(a)), (2) the solution of the SDOF differential Eq. (3) to obtain the corresponding modal coordinates (q_j), and (3) the superposition of the contribution of each mode to the structural response (*Lines 16-18* of Fig. 3(a)). On the other hand, the proposed vectorisation avoids iterating over the selected set of J_r vibration modes in each step of the time-loop. To this end, after $\hat{\mathbf{P}}_s$ is computed the modal forcing for all the selected vibration modes is obtained directly as the vector $\widetilde{\mathbf{P}}_{|J_r\times 1} = \{P_1, \cdots, P_{J_r}\}^{\mathrm{T}}$ with

$$\widetilde{\mathbf{P}} = \widetilde{\mathbf{\Phi}}^{\mathrm{T}}\hat{\mathbf{P}}_s./\widetilde{\mathbf{M}}, \tag{9}$$

in which the element-wise division (./) does not need to be performed if the mode shapes are mass-normalised, as shown in *Line 12* of Fig. 3(b). With the modal forcing for the previous and the current time-steps in vector format ($\widetilde{\mathbf{P}}_0$ and $\widetilde{\mathbf{P}}$, respectively), the array with the modal coordinates of all the relevant vibration modes ($\widetilde{\mathbf{q}}$) can be obtained by solving simultaneously the set of Eq. (3) using a vectorised version of the non-iterative Newmark-β method. The calculation of $\widetilde{\mathbf{q}}$ at any time-step with the vectorised Newmark-β method requires the computation of the following $J_r \times 1$-arrays

$$\ddot{\widetilde{\mathbf{q}}}_{\mathbf{0}} = \left(\widetilde{\mathbf{P}}_0 - \widetilde{\mathbf{C}}.*\dot{\widetilde{\mathbf{q}}}_{\mathbf{0}} - \widetilde{\mathbf{K}}.*\widetilde{\mathbf{q}}_{\mathbf{0}}\right)./\widetilde{\mathbf{M}}, \tag{10a}$$

$$\bar{\mathbf{K}} = \widetilde{\mathbf{K}} + \frac{\gamma}{\beta\Delta t}\widetilde{\mathbf{C}} + \frac{1}{\beta\Delta t^2}\widetilde{\mathbf{M}}, \tag{10b}$$

$$\mathbf{a} = \frac{1}{\beta\Delta t}\widetilde{\mathbf{M}} + \frac{\gamma}{\beta}\widetilde{\mathbf{C}}, \tag{10c}$$

$$\mathbf{b} = \frac{1}{2\beta}\widetilde{\mathbf{M}} + \Delta t\left(\frac{\gamma}{2\beta} - 1\right)\widetilde{\mathbf{C}}, \tag{10d}$$

in which $\ddot{\tilde{\mathbf{q}}}_{\mathbf{0}}$ is the array with the modal accelerations at the end of the previous analysis step; Δt is the analysis time-step; γ and β are scalars that define the Newmark integration (taken in this work as 0.25 and 0.5, respectively). In the vectorised MS algorithm the arrays $\ddot{\tilde{\mathbf{q}}}_{\mathbf{0}}$, $\bar{\mathbf{K}}$, $\mathbf{a}$ and $\mathbf{b}$ are obtained at the start of each time-step by calling the function `InitVNewmark` (*Line 9* of Fig. 3(b)), and they are used as inputs for the non-iterative Newmark-β array-based calculation of the modal coordinates $\tilde{\mathbf{q}}$ and their time-derivatives in incremental form

$$\mathbf{\Delta P} = \left(\widetilde{\mathbf{P}} - \widetilde{\mathbf{P}}_0\right) + \mathbf{a}.*\dot{\tilde{\mathbf{q}}}_{\mathbf{0}} + \mathbf{b}.*\ddot{\tilde{\mathbf{q}}}_{\mathbf{0}}, \tag{11a}$$

$$\mathbf{\Delta}\tilde{\mathbf{q}} = \mathbf{\Delta P}./\bar{\mathbf{K}}, \tag{11b}$$

$$\mathbf{\Delta}\dot{\tilde{\mathbf{q}}} = \frac{\gamma}{\beta\Delta t}\mathbf{\Delta}\tilde{\mathbf{q}} - \frac{\gamma}{\beta}\dot{\tilde{\mathbf{q}}}_{\mathbf{0}} + \Delta t\left(1 - \frac{\gamma}{2\beta}\right)\ddot{\tilde{\mathbf{q}}}_{\mathbf{0}}, \tag{11c}$$

$$\mathbf{\Delta}\ddot{\tilde{\mathbf{q}}} = \frac{1}{\beta\Delta t^2}\mathbf{\Delta}\tilde{\mathbf{q}} - \frac{1}{\beta\Delta t}\dot{\tilde{\mathbf{q}}}_{\mathbf{0}} - \frac{1}{2\beta}\ddot{\tilde{\mathbf{q}}}_{\mathbf{0}}, \tag{11d}$$

$$\tilde{\mathbf{q}} = \tilde{\mathbf{q}}_{\mathbf{0}} + \mathbf{\Delta}\tilde{\mathbf{q}};\ \dot{\tilde{\mathbf{q}}} = \dot{\tilde{\mathbf{q}}}_{\mathbf{0}} + \mathbf{\Delta}\dot{\tilde{\mathbf{q}}};\ \ddot{\tilde{\mathbf{q}}} = \ddot{\tilde{\mathbf{q}}}_{\mathbf{0}} + \mathbf{\Delta}\ddot{\tilde{\mathbf{q}}}, \tag{11e}$$

where $\tilde{\mathbf{q}}_{\mathbf{0}}$ and $\dot{\tilde{\mathbf{q}}}_{\mathbf{0}}$ are the arrays with the modal displacements and velocities at the end of the previous analysis step. The solution of Eq. (11) is obtained in the vectorised MS algorithm by calling the function `VNewmark` in *Line 13* of Fig. 3(b), whose Python code is included in Appendix A. Compared with the classical Newmark-β algorithm (see e.g. [21]), its vectorised form proposed in Eqs. (10) and (11) handles arrays with element-wise operations to give the modal coordinates of all the selected modes, instead of a single scalar value. Finally, in the vectorised MS method the sum of the contributions of the J_r relevant vibration modes is obtained by expressing Eq. (4) in matrix form in *Lines 14-16*

$$\hat{\mathbf{r}}_s(t) = \hat{\mathbf{\Phi}}\tilde{\mathbf{q}}(t);\ \dot{\hat{\mathbf{r}}}_s(t) = \hat{\mathbf{\Phi}}\dot{\tilde{\mathbf{q}}}(t);\ \ddot{\hat{\mathbf{r}}}_s(t) = \hat{\mathbf{\Phi}}\ddot{\tilde{\mathbf{q}}}(t). \tag{12}$$

It is noted that the vector with the loads and moments in each of the selected nodes ($\hat{\mathbf{P}}_s$) is represented with generality as the sum of the external actions induced by moving vehicles ($\hat{\mathbf{P}}_v$), wind ($\hat{\mathbf{P}}_w$), earthquakes ($\hat{\mathbf{P}}_e$), etc. These actions may also involve interaction with the structural movement and feedback. However, this work focuses on the optimisation of the MS solver, and the detailed treatment of the nodal forcing is out of its scope.

2.5. Summary of the accelerating MS techniques and analysis flow of MDyn

The standard MS solution without any of the proposed acceleration techniques is referred to as MS0, and the algorithm that combines all of them is called MS5. In addition, the modal truncation, nodal deactivation, SM deactivation and vectorisation techniques have been implemented separately in order to explore their influence on the computational efficiency and accuracy of the MS method. All the MS approaches have been implemented as subroutines in a Python library called MDyn and they are summarised in Table 1. The general analysis flow of MDyn is as follows: first the solution of the generalised eigenvalue problem is obtained from a commercial FE package like ABAQUS [18]. After this, an in-house Python script converts the FE results into plain text files that include the J mode shape vectors normalised with respect to their modal masses ($\boldsymbol{\phi}_j$) and their frequencies ω_j. These files are used as the only structure-related input for MDyn, which then performs the modal filtering (Fig. 2(a)), nodal deactivation (Fig. 2(b)) and SM deactivation to build the reduced mode matrix $\hat{\boldsymbol{\Phi}}$. In addition, information about the dynamic loading is included in MDyn before the time-history analysis described in Fig. 3 starts.

	MS0	MS1	MS2	MS3	MS4	MS5
Modal truncation	-	✓	-	-	-	✓
Nodal deactivation	-	-	✓	-	-	✓
SM deactivation	-	-	-	✓	-	✓
Vectorisation	-	-	-	-	✓	✓

Table 1: Different MS algorithms implemented in MDyn.

3. Case study 1: Simply supported bridge

3.1. Analytical solution

This section considers a moving load P travelling with constant speed V along the centreline of a straight simply supported bridge (SSB) of span L. Ignoring the shear deformation and assuming that the load enters the bridge at $t = 0$ s when it is at rest, the dynamic response at midspan is given by MS [13]

$$\frac{r_d^Z(t)}{r_s^Z} = \sum_{j=1}^{N} \left\{ \frac{1}{j^4\sqrt{(1-\bar{V}_j^2)^2+(2\xi_j\bar{V}_j)^2}} \left[\sin(\bar{V}_j\omega_j t) - \frac{\bar{V}_j}{\sqrt{1-\xi_j^2}} \mathrm{e}^{\xi_j\omega_j t} \sin\left(\omega_j t\sqrt{1-\xi_j^2}\right) \right] \right\}, \tag{13}$$

where $r_s^Z = PL^3/48EI$ is the maximum static displacement at midspan; EI is the vertical flexural stiffness of the deck; $\bar{V}_j = \Omega_j/\omega_j$ is the non-dimensional speed and $\Omega_j = j\pi VL$ is the driving excitation frequency with respect to the j-th vibration mode of the structure. The circular frequencies of the vertical vibration modes in a SSB are defined as

$$\omega_j = j^2\pi^2\sqrt{\frac{EI}{\mu L^4}}, \tag{14}$$

in which μ is the mass of the deck per unit length.

Eq. (13) is only valid when the load is on the SSB, i.e. if $t \le L/V$. When $t > L/V$, the free-vibration response at midspan is given in [13] as:

$$r_d^Z(t) = \sum_{j=1}^{N} X_j \mathrm{e}^{-\xi_j\omega_j t} \sin(j\pi/2)\sin(\omega_{d,j}t - \varphi_j), \tag{15}$$

where $\omega_{d,j} = \omega_j\sqrt{1-\xi_j^2}$ is the j-th damped frequency and X_j is the amplitude of its contribution to the free-vibration of the structure, which is

$$X_j = \frac{-2P\bar{V}_j\sqrt{1+\mathrm{e}^{2c}-2\mathrm{e}^{c}\cos(j\pi)\cos\left(\frac{j\pi}{\bar{V}_j}\sqrt{1-\xi_j^2}\right)}}{\mu L\omega_j^2\sqrt{(1-\xi_j^2)[(1-\bar{V}_j^2)^2+(2\xi_j\bar{V}_j)^2]}}, \tag{16}$$

and φ_j is the phase angle of the free-vibration response corresponding to the j-th mode

$$\varphi_j = \tan^{-1}\left[\frac{-\mathrm{e}^{c}\sin\left(\frac{j\pi}{\bar{V}_j}\sqrt{1-\xi_j^2}\right)}{\cos(j\pi)-\mathrm{e}^{c}\cos\left(\frac{j\pi}{\bar{V}_j}\sqrt{1-\xi_j^2}\right)}\right], \tag{17}$$

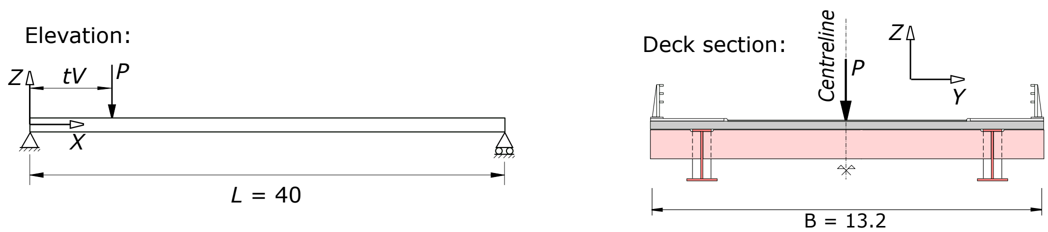

Figure 4: Elevation and deck cross-section of the proposed simply supported bridge (SSB). Dimensions in meters.

with $c = -\xi_j j\pi/\bar{V}_j$.

3.2. Numerical model of the SSB, loading and analysis characteristics

The proposed SSB is a typical 40-m span composite bridge with vertical flexural stiffness $EI = 9.97$ GNm^2 and distributed mass $\mu = 18455$ kg/m. Fig. 4 shows the elevation and the cross-section of the bridge with the load at its centreline. A detailed description of the geometry of this bridge is included in [22].

The numerical model of the SSB is built in the commercial FE software ABAQUS [18] using 3D beam elements with linear interpolation of the curvature and without shear deformation. The model includes a total of 100 elements to discretise the deck. This relatively fine mesh is adequate to represent the first 9 vibration modes with vertical flexure of the deck. The vibration of the bridge in the other directions is prevented by fixing the longitudinal and the transverse movements in all the nodes of the FE model, as well as their torsional rotation. The fundamental mode has a frequency of $f_1 = 1.91$ Hz and it involves a single vertical wave of the deck. The last mode of interest is the 9-th, which represents a high-order vertical flexure of the deck with a frequency of 139.17 Hz. This mode requires a time-step of $\Delta t = 0.7$ ms in the dynamic analysis to obtain at least 10 results in one of its full oscillation cycles. The damping ratio is $\xi_j = 0.5\%$ in all the vibration modes, which is in agreement with EN1991-2 [23] and with previous research works (e.g. [17, 16]). The calculations are conducted with the analytical expressions (13) and (15), and also numerically using the full-FE MS method implemented in ABAQUS and the proposed MS algorithms in Python (MDyn).

The load scenarios considered in the study of the SSB include a moving load of $P = 182.5$ kN travelling with a constant speed of $V = 100$ km/h (case A1), or with a constant speed that ranges from $V = 5$ to 250 km/h, each 1 km/h (i.e. 246 analysis with different speeds, case A2), as described in Table 2. The analysis stops after the vehicle leaves the SSB and travels further a distance equals to its span in order to give enough time for the free vibrations to develop; in the analysis case A1 this means that the simulation time is $t_{\max} = 2.88$ s (giving 4115 analysis steps), whereas in the case A2 $t_{\max}$ varies from 57.6 s to 1.15 s for the lower and the upper values of the vehicle speed range considered, respectively (resulting 82286 and 1646 time-steps in the analysis of vehicles moving with these two extreme velocity values). The definition of the nodal forcing due to moving loads ($\mathbf{P}_v$) in ABAQUS and in MDyn is based on the time-dependent nodal amplitude factors described in [24]. However, ABAQUS requires the loads to be created before the dynamic analysis in all the nodes of the deck, and then to be associated with their corresponding amplitude factors to interpolate the position of the moving forces at each time-step. This leads to load files that can be significantly large (particularly in long structures like the one presented in Section 4) and it increases the pre-calculation time in ABAQUS. On the other hand, MDyn considers that $\mathbf{P}_v = \mathbf{0}$ in all the nodes except from those belonging to the beam element of the deck that is under the moving load at each time-step.

Case study	Label	Traffic scenario
SSB	A1	Single load P moving at $V = 100$ km/h
	A2	Single load P moving at 246 different speeds
CSB	B1	Single H20-44 truck moving at $V = 100$ km/h
	B2	Single H20-44 truck moving at 10 different speeds
	B3	Convoy of 10 H20-44 trucks moving at 10 speeds

Table 2: Traffic scenarios considered in the two case studies of this work.

3.3. Numerical versus analytical solution

Considering the load case A1, Fig. 5 compares the time-history of the vertical displacement of the bridge at midspan (r_d^Z) obtained analytically, with ABAQUS (full-FE solution) and with the standard MS0 algorithm in MDyn. The static deflection at midspan (r_s^Z) is also included in this figure. It is observed that the two numerical solutions (ABAQUS and MS0) are almost identical and they are very close to the analytical result, particularly when only the fundamental mode of the bridge is included in the analysis (i.e. $J = 1$), as shown in Fig. 5(a). Considering the effect of the vibration

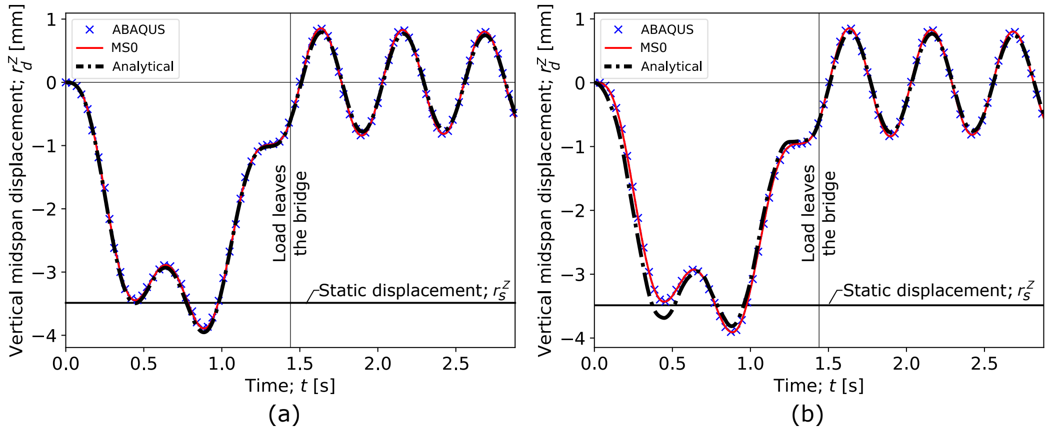

Figure 5: Time-history of the vertical displacement of the SSB at midspan including; (a) only the fundamental mode ($J = 1$), (b) the first 9 modes ($J = 9$). Load case A1.

modes up to the 9-th ($J = 9$) the maximum difference between the analytical and the numerical results increases to 8% (Fig. 5(b)). This is attributed to the mesh-sensitivity of the modal shapes of high-order vibration modes in the FE model, and it can be reduced with a finer mesh discretisation.

Considering load case A2, the ratio between the peak dynamic displacement and the maximum static deflection at midspan $R = r^Z_{d,\max}/r^Z_s$ is given in Fig. 6 for a wide range of vehicle speeds. The results include the contribution of high-order modes ($J = 9$) and distinguish the factor R obtained during the forced response (i.e. when the load is on the bridge) or in the free-vibration stage. The solutions obtained with ABAQUS and with all the MS algorithms in MDyn are superimposed, and they are generally very close to the analytical result. The largest differences between the analytical and the numerical solutions are observed in the forced response factors close to the velocities for which the response is minimum (cancellation velocities). The highest cancellation speed of the forced response occurs at $V \approx 100$ km/h with the analytical solution, whilst this velocity is $V \approx 110$ km/h in ABAQUS and in MDyn. However, the local maxima of the forced and the free responses that occur between consecutive cancellation speeds is very close in the numerical and in the analytical methods.

Table 3 presents the process (CPU) time required to complete the analysis with different MS methods in a Workstation with 32GB of installed RAM and a processor Intel Xeon with 2.4 GHz. In load cases A1 and A2 the standard MS0 algorithm implemented in Python (without considering any of

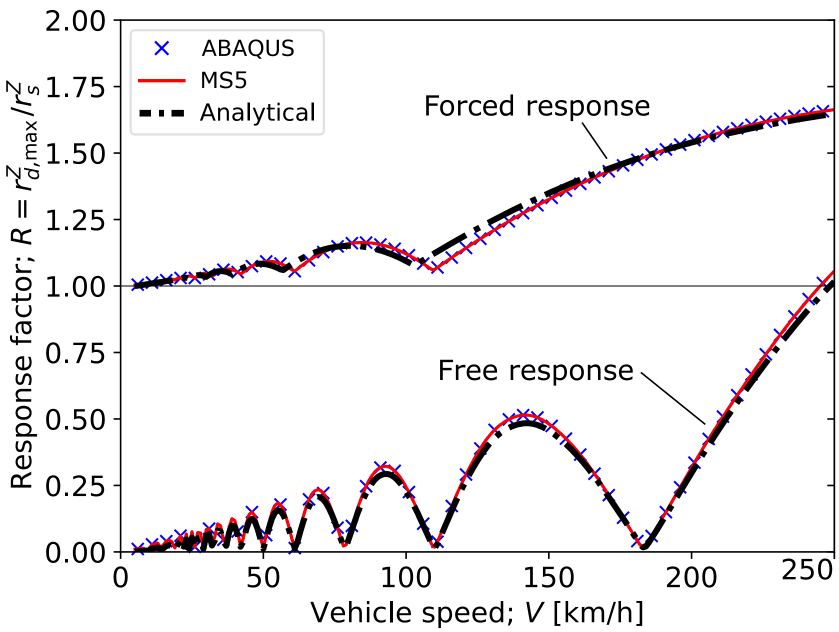

Figure 6: Response factors at the midspan section of the SSB in load case A3 ($J = 9$).

the four accelerating strategies) is approximately 13 times faster than the full-FE analysis. However, the comparisons between the standard MS algorithm in Python and the MS analysis in ABAQUS are given in this paper only for illustration purposes, because the latter includes the time required to perform internal checks that are associated with commercial software but are not present in MDyn. In addition, the comparison with the ABAQUS CPU time includes the time required to generate the nodal load time-histories that represent the vehicle motion, which is not needed in MDyn. For this reason the reference CPU time considered for comparison in this work is the one associated with the standard MS0 in Python. This can be further reduced with the indexed modal truncation (MS1) following the analysis of the dynamic contribution of each mode. However, in this simply supported structure it is clear that odd-numbered symmetric modes are the only ones with contribution to the response at midspan, and therefore the symmetric modes below the 9-th can be the only ones considered in the analysis: $\mathbf{j}_r = \{1, 3, 5, 7, 9\}^{\mathrm{T}}$. It is verified that the results at the midspan section obtained with the MS1 indexed modal selection ($J_r = 5$) are identical to those with the MS0 classical truncation ($J = 9$). Nevertheless, the number of modes considered in MS1 is 44.4% smaller than in MS0 for this case study, which results in a reduction of the CPU time of 35% and 40% in load cases A1 and A2, respectively. Note that the MS2 method is not considered in this case study because all the nodes in the SSB model are directly affected by the moving load and therefore it is not possible to apply the nodal deactivation.

Case			MDyn					
study	Label	Full-FE	MS0	MS1	MS2	MS3	MS4	MS5
SSB	A1	0.3 (0.03)	0.026	0.017	-	0.022	0.008	0.005
	A2	165.9 (15.0)	12.35	7.40	-	9.01	3.76	1.98

Table 3: CPU time in minutes for all the analysis cases in the SSB solved with different methods. The values between brackets represent the additional time required to generate the nodal load time-histories for the full-FE analysis in ABAQUS. Results obtained using a Workstation with 32GB of installed RAM and a processor Intel Xeon with 2.4 GHz.

The SM deactivation (MS3) is implemented in this case study by reducing the DOF of the $N_n = 101$ nodes of the beam model from SM $= 6$ (3 displacements and 3 rotations) to $\mathrm{SM}_a = 1$ (the vertical displacement) in the dynamic analysis. Therefore, with the proposed SM deactivation in MS3 the size of the mode matrix $\hat{\mathbf{\Phi}}$ is reduced to $\mathrm{SM}_a N_n \times J = 909$, which is 83% smaller than the size of the original $\mathbf{\Phi}$ in MS0 ($\mathrm{SM}N_n \times J = 5454$). However, the results in Table 3 indicate that the SM deactivation makes the MS3 algorithm only 15% and 27% faster than the standard MS0 in load cases A1 and A2, respectively. It is observed that the benefit of the SM deactivation is below the one obtained by reducing the size of the inner j-loop in Fig. 3(a) with the indexed modal truncation in MS1. The influence of the inner loop in the CPU time is more evident when MS0 (non-vectorised) is compared with MS4 (its vectorised counterpart described in Fig. 3(b)), which is 70% faster in both load cases. Finally, the algorithm MS5 combines all the possible accelerating techniques and it reduces the computational effort by 81% and 84% compared with MS0 in load cases A1 and A2, respectively. If the CPU time in MS5 is compared with the full-FE solution the reduction of the calculation time is above 98% in both load cases. This allows to analyse the response of the bridge under a large number of traffic cases, as it is explored in Section 5.

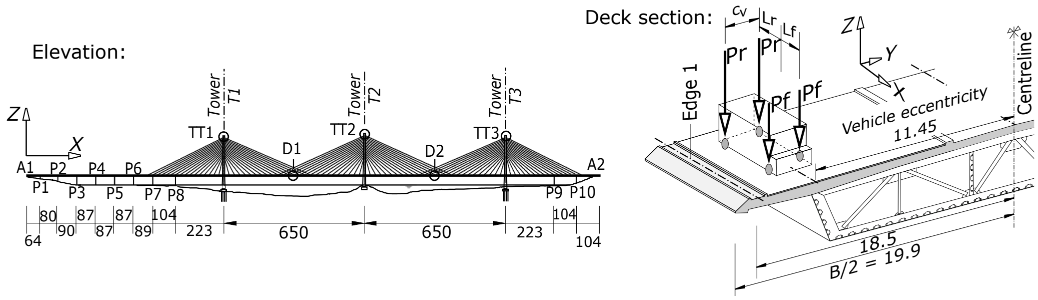

Figure 7: Elevation and deck cross-section of the Queensferry cable-stayed bridge (CSB). Dimensions in meters.

4. Case study 2: cable-stayed bridge

The second case study considers the Queensferry Bridge in Scotland, illustrated in Fig. 7. The structure is a continuous cable-stayed bridge (CSB) with two main spans of 650 m each supported by a central plane of cables. The 4.9-m deep cross-section of the cable-stayed deck is a metallic box closed by a 39.8-m wide concrete slab. The deck is 2643-m long and it has two carriageways, each one with two road lanes. The abutments, the piers (P1 - P10) and the towers restrain the vertical, the transverse and the torsional movements of the deck, with the exception of the side towers (T1 and T3 in Fig. 7) that allow the free vertical movement of the deck. The central tower (T2) is fully fixed to the girder at their connection.

The vehicle considered in this case study is the 4-wheeled H20-44 truck defined by AASHTO [25], which combines both heavy vehicle weight (18.6 t) and potentially high velocities. The dimensions of the truck and its load distribution are included in Fig. 7, where $P_f = 17.8$ kN and $P_r = 71.2$ kN represent the load of each front and rear wheel, respectively, and $L_f = L_r = 2.135$ m refer to their corresponding longitudinal distances to the centroid of the vehicle. The transverse distance between wheels is $c_v = 2.05$ m and the total length of the truck is 8.53 m, with the front wheels separated 1.7 m from the vehicle front. The vehicle moves in the positive X direction following a straight path centered in the outer lane of the carriageway to maximise the torsional response of the deck, with the centroid of the truck separated 11.45 m from the midplane of the bridge.

4.1. FE model and modal analysis

The MS solutions are based on the vibration mode shapes and modal properties obtained from the FE model of the structure, which was conducted in ABAQUS. The deck and the towers were defined using a combination of linear beam elements and lumped masses that represent the cable anchorages, the barriers and other relevant dead loads. The deck is discretised with 229 beam elements that have a typical length of approximately 10 m, whilst each reinforced concrete tower is divided in 27 beam elements that represent their variable cross-section. The v-shaped reinforced concrete piers P3-P9 are discretised with the same type of beam elements. The cables are modelled using single truss elements with reduced (Ernst) elasticity moduli to account for cable-sag effects. The FE model includes a total of 1380 nodes and 1601 elements. The structural movements in the longitudinal (traffic), transverse and vertical directions are referred to as SM $= 1, 2, 3$, respectively, and the rotations in the corresponding axes are SM $= 4, 5, 6$.

From the modal analysis conducted in the FE model, the fundamental mode of the bridge has a frequency $f_1 = 0.15$ Hz and it is shown in Fig. 8(a). This mode involves the vertical flexure of the two main spans, as well as the longitudinal movement of the three towers. In this structure the vibration mode threshold is initially set as $J = 557$, which is the order of the highest vibration mode with a frequency below 20 Hz (this is a common frequency limit in Civil Engineering structures). Fig. 8 compares the relative contribution of the first 30 vibration modes of the CSB calculated with the proposed dynamic contribution factor $\eta^k_{\mathrm{SM},j}$ and with the relative effective modal mass

$$\bar{m}^{\mathrm{eff}}_{\mathrm{SM},j} = \frac{m^{\mathrm{eff}}_{\mathrm{SM},j}}{\sum\limits_{j=1}^{J} m^{\mathrm{eff}}_{\mathrm{SM},j}} = \frac{\left(\boldsymbol{\phi}_j^{\mathrm{T}} \mathbf{M}_s \boldsymbol{\iota}_{\mathrm{SM}}\right)^2}{m_j \sum\limits_{j=1}^{J} m^{\mathrm{eff}}_{\mathrm{SM},j}} \tag{18}$$

where $\boldsymbol{\iota}_{\mathrm{SM}|N\times 1}$ is the displacement vector of the structure when a unit movement is imposed at all its supports in direction SM; $m_j = \boldsymbol{\phi}_j^{\mathrm{T}} \mathbf{M}_s \boldsymbol{\phi}_j$ is the j-th modal mass. It can be observed from Eq. (18) that the relative effective modal mass is not particularised at any structural node, and it depends only on the properties of the structure and not on the dynamic actions applied, unlike the proposed dynamic contribution factor $\eta^k_{\mathrm{SM},j}$ in Eq. (5). The values of $\eta^k_{\mathrm{SM},j}$ are obtained with the algorithm MS4 under the traffic case B1 described in Table 2, and they show in Fig. 8(a) the large participation of the fundamental mode ($j = 1$) in the vertical response of the bridge (SM = 3) at midspan D1. However, the effective mass activated by the first mode in the vertical direction is null because the mass of the deck moving upwards in the first main span compensates the mass moving downwards in the second main span. Therefore, the proposed dynamic factor η can identify the contribution of vibration modes to the vehicle-induced response better than traditional modal factors like $\bar{m}^{\text{eff}}$. This is also noticed in the modal contribution to the torsional response of the deck (SM=4) presented in Fig. 8(b); η clearly identifies the first symmetric and antisymmetric torsional modes of the deck at the two main spans (Modes 11 and 12, respectively) as the dominant modes, but this information is lost by $\bar{m}^{\text{eff}}_{\text{SM}=4,j}$ due to the large contribution of the lateral movement of the towers to the rotational mass $\sum_1^J m^{\text{eff}}_{\text{SM}=4,j}$. The effect is attributed to the larger distributed mass in the towers compared with the deck, and makes it difficult to establish valid modal selection criteria based on $\bar{m}^{\text{eff}}$. Consequently, only η is used hereinafter to identify the dominant modes of the bridge for the traffic-induced vibrations, imposing a limit $\eta_{\min} = 0.1\%$ below which vibration modes are discarded for the subsequent MS analysis.

Fig. 9 shows the dynamic participation factor η of all the modes below 20 Hz. It is observed that all the vibration modes above 9.1 Hz (which is the frequency of the 281-th mode) have a contribution that is below $\eta_{\min} = 0.1\%$ of the total response ($r^k_{\text{SM},J}$). Fig. 9(a) also indicates that the fundamental mode alone contributes to more than 60% of the total vertical deck displacement at second midspan (Point D2 in Figs. 7 and 9(a)), but clusters of vibration modes between 0.15 Hz and 2.7 Hz also have a significant contribution to the response. The importance of high-order modes is stronger for the torsional response of the deck in this CSB, as shown in Fig. 9(a). However, the longitudinal movement at the top of the three towers (points TT1, TT2 and TT3) can be captured with relatively few low-order vibration modes due their large flexibility, as shown in Fig. 9(b). This figure includes the shape of the 139-th mode (3.9 Hz), which is the highest-order vibration mode with a relative contribution to the total response of the towers above $\eta_{\min} = 0.1\%$. The study of η results in the definition of the index vector $\mathbf{j}_r$ for the MS1 modal selection, as described in Section 2.1. This modal study also concludes that the order of the highest mode of interest from the point of view of the global vehicle-induced vibrations in the CSB is 281, which is set as the new threshold: $J = 281$.

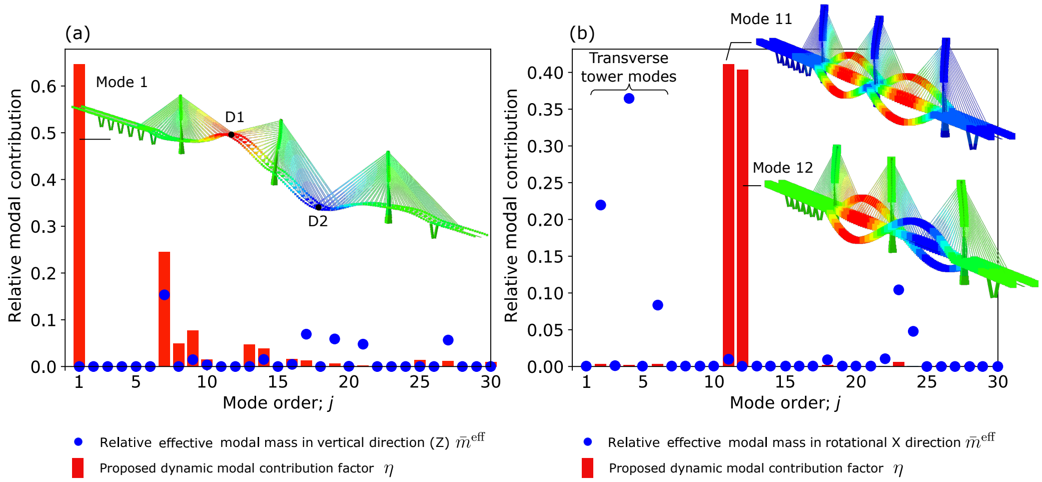

Figure 8: Relative modal contribution of the first 30 vibration modes obtained with the effective modal mass ($\bar{m}^{\text{eff}}$) and with the dynamic modal contribution (η); (a) vertical response at midspan D1, (b) torsional response at midspan D1. Relevant vibration modes shapes of the CSB are included.

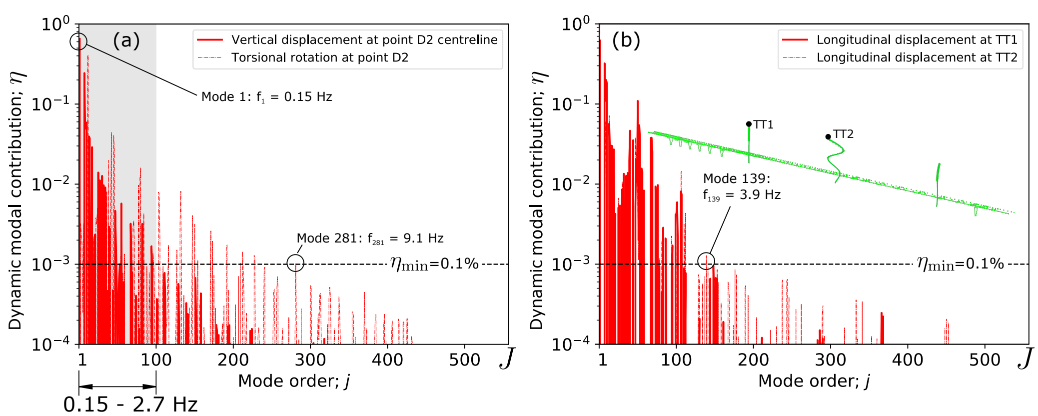

Figure 9: Dynamic modal contribution of all the modes below 20 Hz; (a) response of the deck at midspan (D2), (b) response at the top of the towers (TT1 and TT2). The shape of Mode 139 is included with the cable-system removed for clarity.

4.2. MS acceleration strategies in the study of the CSB

The full-FE solution in ABAQUS includes in the analysis the first 281 vibration modes, which are also considered in the MDyn strategies without indexed modal truncation (MS0, MS2, MS3 and MS4). The time-step of the analysis is $\Delta t = 0.01$ s to have at least 10 analysis results in each cycle of the highest mode of interest (9.1 Hz). The structural damping ratio is considered to be constant and equal to $\xi_j = 0.5\%$ for all the modes.

The full-FE MS analysis incorporates all the nodes in the model ($N_n = 1380$), with 5748 DOF and a mode matrix $\mathbf{\Phi}_{|5748\times281}$ that has $16.15{\cdot}10^5$ modal components. On the other hand, the standard MS0 algorithm in MDyn does not consider in the dynamic analysis the parts of the structure that are not object of study and that are not subject to the direct vehicle actions, namely the cables (192 truss elements), the intermediate piers P3-P9 (126 beam elements), and the masses describing the deck barriers and the anchorages in the deck and in the towers (844 lumped mass elements). This results in a reduced set of $N_n = 314$ nodes connected by beam elements. Each node has 6 active DOF (SM = 6) and therefore the dynamic problem in MS0 has 1884 DOF in total, giving a matrix $\mathbf{\Phi}_{|1884\times281}$ composed of $5.29{\cdot}10^5$ modal components. Consequently, MS0 reduces the size of $\mathbf{\Phi}$ by 67.2% compared with the full-FE solution. In addition to this, the following strategies are adopted in the CSB to improve the performance of the MS algorithms in MDyn:

- Indexed modal truncation (MS1): only the vibration modes with relative contributions to the movements at the reference points D1, D2, TT1, TT2 and TT3 that are above 0.1% are included; i.e. $\mathbf{j}_r$ contains only the modes for which $\eta > \eta_{\min} = 0.1\%$ in Fig. 9, in any direction. This strategy reduces the number of vibration modes to be included in the analysis from $J = 281$ (standard modal truncation in MS0) to $J_r = 112$ (indexed modal truncation). Therefore, the reduced mode matrix is $\widetilde{\mathbf{\Phi}}_{|1884\times112}$ and contains $2.11{\cdot}10^5$ modal components.

- Nodal deactivation (MS2): only the deck is considered (230 nodes) and the towers (84 nodes) are eliminated from the dynamic analysis, hence the reduced mode matrix $\hat{\mathbf{\Phi}}_{|1380\times281}$ has $3.88{\cdot}10^5$ modal components.

- SM deactivation (MS3): instead of considering 6 DOF per node in the beam elements (SM = 6), only the degrees of freedom that are directly excited by the traffic actions are included in the analysis, namely the vertical displacement (r^Z) and the torsional rotation (r^{XX}). Consequently, $\mathrm{SM}_a = 2$ and the number of DOF in the problem is $2 \times 314 = 628$, which results in a mode matrix $\mathbf{\Phi}_{|628\times281}$ with $1.76{\cdot}10^5$ modal components.

- Vectorisation (MS4): with this strategy the nested loop in the MS algorithm is replaced by a simple time-loop, keeping the size of the problem unchanged with respect to MS0: i.e. $\mathbf{\Phi}_{|1184\times281}$, with $5.29{\cdot}10^5$ modal components.

- Combination of the four strategies (MS5): combining MS1 to MS3 the size of the problem is reduced to $2 \times 230 = 460$ DOF and $0.5{\cdot}10^5$ modal components ($\hat{\mathbf{\Phi}}_{|460\times112}$). Note that MS5 also includes vectorisation.

4.3. Single truck moving at constant speed; load case B1

The analysis of the CSB starts considering the load case B1 in Table 2. It consists of a single H20-44 truck crossing the bridge with a constant velocity of $V = 100$ km/h.

Fig. 10 compares the time-history of the responses at the reference points of the deck and the towers obtained with the full-FE solution and with MDyn. The responses obtained with any of the MS algorithm variations included in Table 1 are almost identical, but the movement of the towers is not obtained in the cases for which deactivation of all the nodes apart from the deck is considered (MS2 and MS5). Fig. 10 shows that the solutions obtained with the full-FE analysis and with the proposed MS algorithms are superimposed. Both capture accurately the movement of the deck at point D1, which presents its maximum downward deflection (Fig. 10(a)) and torsional rotation (Fig. 10(b)) at $t = 44.78$ s, when the rear wheels of the truck (P_r) are located at this point of the deck. The deformed configurations of the bridge obtained with ABAQUS and with MS4 at this particular instant are presented in Fig. 11, showing good consistency in the entire structure. It should be noted that the cable elements and the lateral edges of the deck illustrated in the MS4 solution are introduced for visualisation purposes in this figure; as it was explained previously these elements were not included in the MDyn analysis.

The longitudinal and the transverse responses of the towers obtained with MDyn and with ABAQUS are also coincident, as shown in Figs. 10(c) and

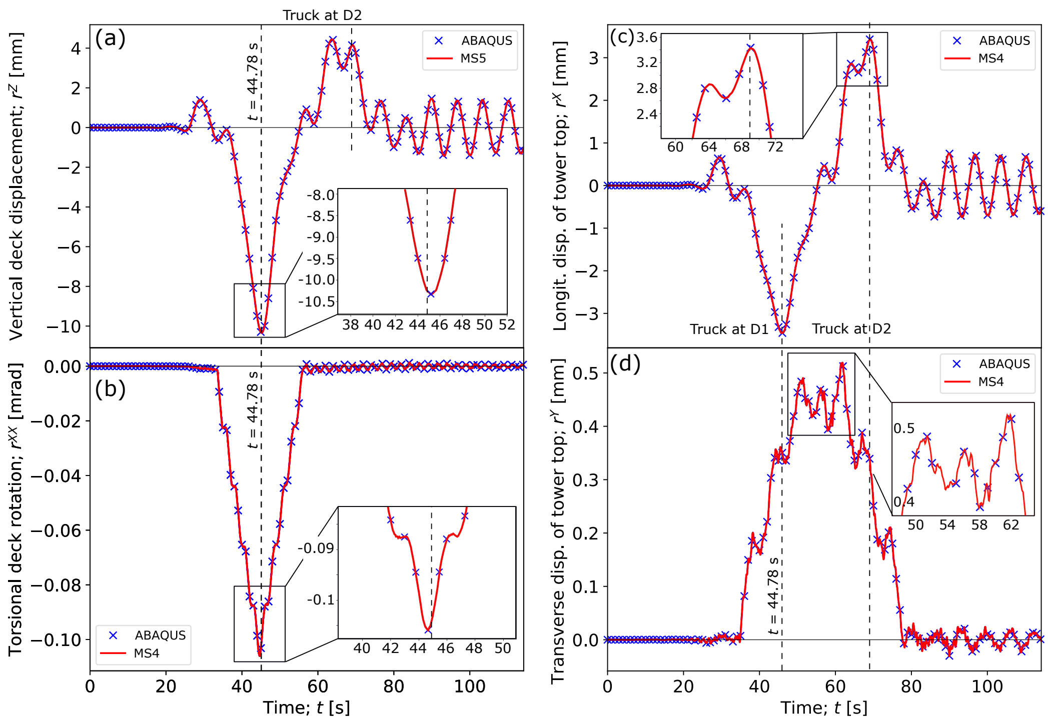

Figure 10: Time-histories of the responses at different points of the CSB with ABAQUS and with MDyn; (a) vertical displacement of the deck at the centre of the first main span (D1), (b) torsional rotation of the deck at D1; (c) longitudinal displacement of the top of the central tower (TT2), (d) transverse displacement of the tower at TT2. Load case B1.

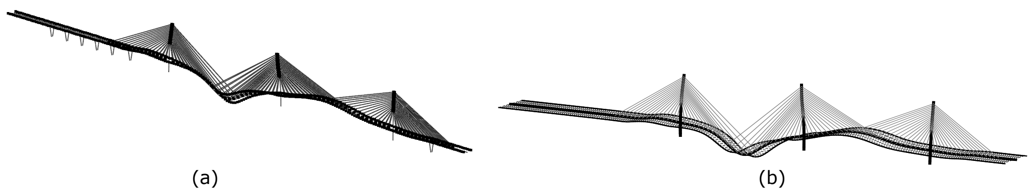

Figure 11: Deformed configuration of the CSB at time $t = 44.78$ s in load case B1: (a) ABAQUS, (b) MS4. Movement amplified 10^4 times.

(d), respectively. The same results are obtained with MS1, which indicates that the indexed modal truncation does not influence the global tower response, and that it captures accurately 3D effects like the lateral movement of the towers induced by the eccentricity of the vehicle.

Table 4 shows the CPU time required to complete the analysis of load case B1 in ABAQUS and in MDyn. In all the cases the analysis stops after the truck exits the CSB and travels 530 m further ($t_{\max} = 114.2$ s), which results in $11.42{\cdot}10^3$ time-steps per analysis. ABAQUS completes the calculation in 14.7 minutes, and the generation of the necessary load and moment time-histories before the dynamic analysis needs 0.18 minutes more. The standard MS0 algorithm requires 2.15 minutes to complete the analysis, which represents a 86-% reduction compared with the full-FE solution. This is partly attributed to the reduction of 67.2% of the size of $\mathbf{\Phi}$ by removing the cables, the anchorages and the intermediate piers from the MS dynamic analysis, and it is also due to the additional operations performed internally by commercial FE software to check the quality of the results.

Case			MDyn					
study	Label	Full-FE	MS0	MS1	MS2	MS3	MS4	MS5
CSB	B1	14.7 (0.18)	2.15	0.89	1.97	1.77	0.21	0.16
	B2	206.1 (20.9)	29.7	12.1	27.2	24.3	2.82	2.23
	B3	- (427.2)	38.7	16.3	35.9	31.1	5.59	4.70

Table 4: CPU time in minutes for all the analysis cases in the CSB solved with different methods. The values between brackets represent the additional time required to generate the nodal load time-histories for the full-FE analysis in ABAQUS. Results obtained using a Workstation with 32GB of installed RAM and a processor Intel Xeon with 2.4 GHz.

The CPU time is further reduced by adopting the selective choice of the vibration modes in MS1. Compared with MS0, reducing the number of vibration modes and the size of the mode matrix by 60.1% in MS1 decreases the CPU time in the same proportion (59%) without any appreciable influence in the results. MS0 and MS1 are not vectorised, and the efficiency of the latter stems from the reduction of the size of the modal loop (j-loop in Fig. 3(a)). The vectorised algorithm MS4 completely removes this inner loop (Fig. 3(b)) and the reduction of the CPU time increases to 93% with respect to MS0. Note that both MS0 and MS4 consider exactly the same number of modal coordinates in the problem.

However, the nodal and the SM deactivation techniques are less efficient because they reduce the number of the modal coordinates in the problem with respect to MS0, but not the range of the inner modal loop, which is repeated $J = 281$ times in both cases (one per vibration mode included in the analysis). More specifically, by dectivating all the nodes apart from those in the centreline of the deck (MS2) a 26.8%-reduction of the number of DOF in the analysis (and in the size of $\hat{\mathbf{\Phi}}$ and $\mathbf{P}(t)$) is achieved, but the CPU time is reduced only 8%. This is even more evident in MS3, where the active SM are reduced by 66.7% because only the vertical and the torsional movements are considered, but the CPU time decreases only 18%. It has also been observed that activating only the vertical displacement (SM_a=1) gives exactly the same result in terms of the movement of the deck centreline, but it is not adequate to study the response at its edges because the torsion induced by the vehicle cannot be captured.

By combining the four accelerating strategies in MS5 the analysis time is reduced down to 0.16 minutes. This represents a time-saving of 93% and 99% compared with MS0 and with ABAQUS, respectively, getting dynamic responses in the bridge that are virtually identical. The reduction of the computational effort with MS5 increases the slower the vehicle (because the time that it needs to cross the bridge is longer), and the larger the structural model or the number of time-steps in the analysis.

4.4. Single vehicle moving at different speeds; load case B2

Load case B2 considers a single H20-44 truck that crosses the CSB at 10 different speeds ranging from 25 to 250 km/h each 25 km/h. Velocities above a feasible limit of 175 km/h are considered in order to obtain a complete picture of the dynamic response of the bridge.

Fig. 12(a) shows the forced and the free response factors R in the CSB under load case B2, referred to the movement at the midspan point D1. ABAQUS and MDyn, in any of its algorithm variations, give very similar

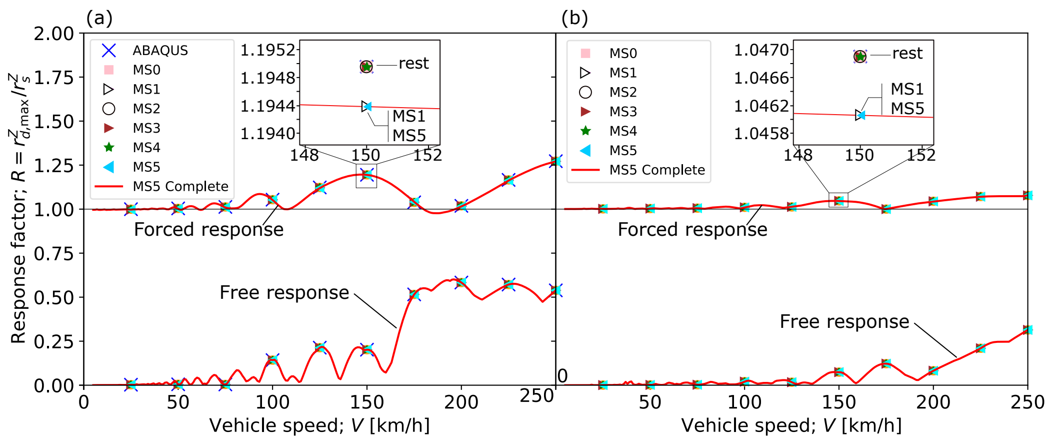

Figure 12: Response factors at midspan D1 of the CSB in; (a) load case B2, (b) in load case B3.

results for the 10 driving velocities considered. A closer look at the results with $V = 150$ km/h, for which the forced response factor is maximised, indicates that the algorithms without selective modal truncation (MS2, MS3 and MS4) give exactly the same response as ABAQUS. This demonstrates that the nodal/SM deactivation and the vectorisation of the solver do not affect the results. It is also observed that the solution considering all the modes below 20 Hz is only 0.05% higher than the result with the proposed modal selection in MS1 and MS5, which suggests that the modal filtering based on the minimum dynamic modal participation factor $\eta_{\min} = 0.1\%$ is adequate.

Table 4 presents the CPU time required by ABAQUS and by MDyn to perform the 10 different dynamic analyses that compose load case B2. The lowest vehicle velocities increase significantly the calculation time because the same time-step ($\Delta t = 0.01$ s) is maintained in dynamic analyses of longer duration. This is extremely onerous in ABAQUS, which takes 3.8 hours to complete the analyses and produces results and data files that combined occupy more than 6 GB of space. On the other hand, the basic algorithm MS0 reduces the calculation time by 87% and limits the space requirements to less than 9 MB. The performance of the MS algorithm can be further improved with the proposed acceleration techniques. The selective modal truncation in MS1 reduces the CPU time of the standard truncation in MS0 by 59%. This is significantly higher than the 8-% and 18-% reductions observed with the nodal deactivation and with the SM deactivation in MS2 and MS3, respectively. The vectorisation in MS4 decreases the computational time of its non-vectorised counterpart MS0 by 91%, which is almost the same as the 93-% reduction obtained with MS5 by combining all the acceleration strategies. This shows the efficiency of vectorising the modal operations included in the time-loop of the MS algorithm.

For completeness, the MS5 algorithm is used to calculate the bridge response to a moving H20-44 truck with velocity increments of 1 km/h, from 5 to 250 km/h. The study is not repeated in ABAQUS because of the very large computational time involved; the complete solution of the 256 dynamic analyses with different vehicle speeds required 75.5 minutes using MS5, which is three times less time that what is needed to perform 10 calculations with the same vehicle using ABAQUS. The results of the complete analysis in MS5 show that the peak forced and free dynamic factors in the CSB are limited to $R \approx 1.25$ and 0.5, respectively, which are smaller than in the SSB under the load case A3. This is attributed to the longitudinal distance between the vehicle wheels in the H20-44 truck ($L_r + L_f = 4.27$ m) and to the continuity of the girder between its vertical supports along the deck. The single load moving on the SSB induces an undisturbed oscillation in the deck that is only dissipated by the structural damping. However, the CSB is a highly hyperstatic structure in which different parts of the deck contribute to the vertical stiffness of the loaded span, reducing its dynamic amplification factors. The comparison of these factors in the SSB and in the CSB in Figs. 6 and 12(a), respectively, also indicates that the latter does not have a clear pattern of vehicle speeds that create the cancellation or amplification of dynamic effects, as it was the case in the SSB, particularly in its free response. This is explained by the significant importance of the first vibration mode in the response of the SSB at its midspan section, whereas in the CSB higher-order modes have more relevance in the response.

4.5. Vehicle convoy moving at different speeds; load case B3

In this load case the number of vehicles crossing the bridge (N_v) is increased to explore the influence of the number of moving wheels and the length of the vehicle convoy in the results. Load case B3 is composed of 10 H20-44 trucks spaced at $d_v = 71.7$-m intervals that cross the CSB at the same constant speeds as those in load case B2; d_v is the longitudinal distance between the centroids of consecutive vehicles. The total length of the convoy (650 m) is selected to load the complete length of the main spans at certain time instants of the analysis.

Fig. 12(b) compares the response factors obtained with the proposed MS algorithms at the midspan point D1 of the CSB in load case B3. The results are identical in the solutions that include all the vibration modes below 20 Hz, and the difference with the selective modal truncation approaches (MS1 and MS5) is only 0.08% for the forced response ratio at $V = 150$ km/h.

The analysis with load case B3 is not conducted in ABAQUS because of its significant computational cost. Only the calculation of the load and moment time-histories necessary for the dynamic analysis with the 10 different vehicle speeds in ABAQUS requires more than 7 hours and data files of 20 GB. This is because the analysis needs to be extended to allow for the last truck of the vehicle convoy to exit the bridge, which implies increasing the simulation in more than $55 \cdot 10^3$ and $1.1 \cdot 10^3$ additional time-steps when $V = 5$ and $V = 250$ km/h, respectively. However, in the same conditions the CPU time needed to obtain the solution with the standard algorithm MS0 is 38.7 minutes (see Table 4), which can be further reduced by 7% and 20% with the nodal and the SM deactivation strategies in MS2 and MS3, respectively. The selective modal truncation implemented in MS1 is more efficient as it allows a 58-% reduction of the CPU time with respect to MS0. Once again, the most efficient strategy is the vectorisation of the time-loop that is implemented in the MS4 code, although the time saving with this approach (86% in load case B3) is smaller than in the same structure with only one vehicle crossing the bridge at 10 different speeds (91% in load case B2). This is because the calculation of the nodal force vector $\mathbf{P}_v$ within the time-loop is repeated for each wheel, and therefore the increment in the number of moving loads in load case B3 (with 40 moving wheels) in comparison with those in load case B2 (with 4 moving wheels) reduces slightly the efficiency of the vectorisation in MS4. Further improvements may be achieved if the calculation of $\mathbf{P}_v$ is also vectorised, and this could be the scope of a separate study.

Combining all the proposed strategies in MS5 the calculation time required to obtain the results for 10 different vehicle speeds in load case B3 is 4.7 minutes. This allows to extend the analysis and calculate the responses for a wider range of velocities in Fig. 12(b). Completing the analyses for the convoy of trucks travelling at 246 different speeds takes 159.5 minutes using MS5 and it allows to observe interesting dynamic effects in the bridge. The results show that compared with the single moving truck in Fig. 12(a), the dynamic response factors are significantly smaller when 10 vehicles spaced a distance $d_v = 71.7$ m are considered in Fig. 12(b), even if this vehicle arrangement loads one of the main spans completely and therefore maximises the static response of the bridge at certain instants of the analysis. This is because after loading the first main span the convoy continues its movement and due to its length it affects both main spans in a period of time, which reduces the dynamic amplification of the response at midspan.

5. Critical vehicle arrangements

The previous section demonstrated that the dynamic amplification of the response can be reduced significantly if the vehicles keep a certain uniform spacing (d_v), but other configurations may induce resonant effects if they match important vibration modes of the structure. This section presents an extensive parametric analysis on the influence of the number of vehicles and their spacing in the response of the SSB and the CSB. The study considers constant separation between vehicles, which not only has an academic interest in the study of potential resonant effects in bridges but also a significant relevance with the advent of connected autonomous trucks (CATs) in the future [26].

As it was observed in Figs. 6 and 12 the peak response factor depends on the vehicle speed. In order to capture the maximum dynamic effects the parametric analysis considers H20-44 trucks crossing the bridges at constant speeds ranging from $V = 100$ km/h to 175 km/h (with 5-km/h increments). The number of vehicles in the convoy varies from $N_v = 1$ (single truck) to $N_v = 10$. More than 500 different values of the longitudinal spacing between consecutive vehicle centroids (d_v) were proposed in each bridge, varying from a minimum of $d_v = 10$ m (which leaves a distance of only 1.47 m between the rear of a vehicle and the front of the following one) to a maximum of $d_v = 4$ km and $d_v = 10$ km in the SSB and in the CSB, respectively. The duration of each analysis is adjusted depending on the length of the bridge, the length of the convoy and its velocity to allow the last truck to exit the deck before finishing the simulation. The only output requested in the analysis is the vertical displacement of the midspan section (point D1 in the CSB) to obtain its forced response factor $R = r^Z_{d,\max}/r^Z_s$, where the maximum static displacement at midspan (r^Z_s) is substituted for simplicity by a quasi-static value given by the convoy crossing the bridge at a very low speed ($V = 5$ km/h) for which the dynamic effects are negligible. Altogether, the proposed parametric study is composed of more than $1.6 \cdot 10^5$ dynamic analyses that were completed using the algorithm MS5. It is remarked that conducting

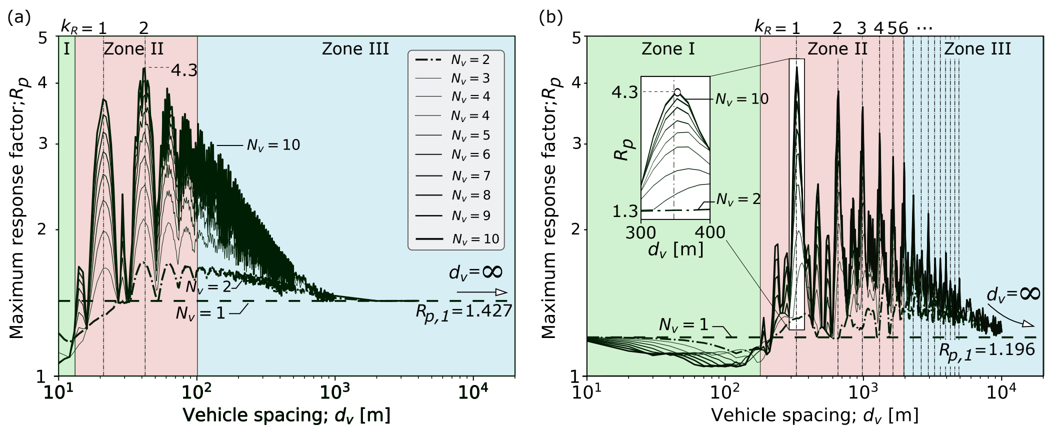

Figure 13: Peak dynamic response factor induced by convoys of trucks H20-44 with different configurations in the: (a) SSB, (b) CSB.

this study using a commercial FE software would not be possible due to the excessive calculation time.

Fig. 13 presents the peak forced response factors of the vertical midspan displacement at the centreline of both bridges for any driving speed: $R_p = \max(R)$. When $d_v \to 0$ the positions of all the vehicles in the convoy coincide in a single (heavier) truck and $R_p \to R_{p,1}$, with $R_{p,1}$ representing the peak response factor obtained with a single moving vehicle. The results confirm that $R_{p,1}$ is higher in simply-supported bridges ($R_{p,1} = 1.427$ in the SSB) than in bridges with a continuous deck ($R_{p,1} = 1.196$ in the CSB). As the vehicle spacing d_v increases its influence on R_p can be divided in three regions: (I) reduction, (II) amplification, and (III) attenuation.

The convoy spacing d_v in Zone I leads to a reduction of the dynamic response compared with the value obtained with a single moving vehicle. This is because of the interruption of the free dynamic oscillation of the deck due to the presence of several vehicles on the bridge at certain instants. The deamplification of the dynamic response in Zone I is more significant the more vehicles are included in the convoy. In the case of the SSB this zone is relatively narrow, up to $d_v \approx 13$ m, because of the short length of the bridge. However, in the CSB vehicle spacings of up to $d_v = 200$ m can lead to significant reductions of the peak dynamic response induced by a single moving truck.

Zone II covers a region of vehicle spacings that are much larger than those considered in previous works [13, 16, 22] and it shows significant resonant effects. These are induced by certain vehicle spacings and velocities that result in a cadence of loads matching important vibration periods of the structure. In both bridges the peak responses are observed when the vehicle spacing is tuned to the fundamental vertical period of the deck ($1/f_1$). Therefore, the resonant vehicle spacing ($d_{v,R}$) is given as

$$d_{v,R} = \frac{k_R V_p}{f_1}, \tag{19}$$

where $k_R = 1, 2, 3, \cdots$ represents the number of cycles of the fundamental mode of the bridge that are completed in the time that it takes between the passing of consecutive vehicles; V_p is the driving speed for which R is maximised: in the SSB the first two resonant peaks occur for $V_p = 145$ km/h, whilst in the CSB $V_p = 175$ km/h (the maximum value considered) in all the cases.

Fig. 13 represents with vertical lines the main resonant vehicle spacings obtained with Eq. (19) in both structures. The strongest resonance is observed for $k_R = 1, 2$, where the dynamic effects build up as the number of vehicles in the convoy increases. However, above $N_v = 5$ the increment in the number of vehicles is less significant for the peak dynamic displacements of the deck, particularly in the CSB. This saturation of the dynamic response when $N_v > 5$ may be attributed to the contribution of high-order modes. Other peaks of the dynamic response observed in Zone II correspond to different vehicle speeds and higher-order vibration modes, but they have smaller effects than those observed with $d_{v,R}$. It is important to remark that the maximum value of the response factor with $N_v = 10$ vehicles is $R_p = 4.3$ in both bridges, but it occurs for very different vehicle spacings: $d_{v,R} = 42$ m in the SSB and $d_{v,R} = 327$ m in the CSB. In fact, the region of Fig. 13 that is dominated by resonance (zone II) occurs for significantly shorter values of the vehicle spacing in the SSB (from $d_v = 13$ m to $d_v = 100$ m) than in the CSB (d_v from 200 to 4000 m) because the latter has a longer fundamental period. Therefore, vehicle-induced resonance is less likely in the CSB because it requires long convoys of trucks with large spacings and without any intermediate vehicle breaking the loading sequence.

Finally, there is region of long vehicle spacings that is characterised by the fast attenuation of the peak dynamic responses (zone III). In this region the resonant spacing is associated with high values of k_R, which implies that the deck completes a significant number of oscillation cycles before being excited by the next vehicle of the convoy. The amplitude is reduced for each of these cycles thanks to the structural damping, which attenuates the resonant effects. This becomes more relevant for shorter vehicle spacings in the SSB ($d_v > 100$ m) because its higher fundamental frequency allows to dissipate the vibrational energy faster than in the CSB, even if the structural damping is considered to be the same in both bridges. Finally, for very long vehicle spacings the dynamic responses of the convoys tend asymptotically to that produced by a single vehicle (i.e. $R_p \to R_{p,1}$ when $d_v \to \infty$). This is because the spacing between vehicles is so large that the bridge is completely at rest when the next truck of the convoy enters the bridge.

6. Conclusions

This work proposes a new modal superposition (MS) algorithm applicable with generality to any line-like structure. It is based on two novel acceleration techniques that involve vectorisation and selective modal truncation based on a new dynamic participation factor η, in addition it allows the efficient deactivation of specific structural nodes and types of movements from the analysis. The standard MS algorithm (referred to as MS0), along with the four MS acceleration strategies, have been implemented in isolation (algorithms MS1 to MS4) and in combination (algorithm MS5) in a Python [3] library called MDyn. These algorithms were used to obtain the traffic-induced response of a short-span simply supported bridge (SSB) and a long-span cable-stayed bridge (CSB) subject to different vehicle arrangements. The solution is compared with that given by the general-purpose finite element (FE) solver ABAQUS [18] and with the analytical solution (when it is available). The results show that:

- The MS solutions using ABAQUS (full-FE method) and MDyn are very close to the analytical one in the SSB, both in terms of the time-history displacements and the peak dynamic response factors for a wide range of moving load speeds. On the other hand, the responses given by ABAQUS and by MDyn are identical in the SSB and in the CSB when the same vibration modes are included in the analysis. However, MDyn is significantly faster than ABAQUS in the analysis of both structures.

- The proposed indexed modal truncation (MS1) allows to select only the relevant vibration modes below a certain frequency. The modal analysis has two steps in which (1) the relative dynamic contribution to the total traffic-induced response (η) of all the modes below a certain frequency is calculated, and (2) a vector with the indices of all the modes that have a relative contribution to the total dynamic response above a certain limit ($\eta_{\min}$) is created. This is used in the MS analysis as an index vector to filter out the modes that are irrelevant to the response of interest. In the CSB under traffic actions with $\eta_{\min} = 0.1\%$ the indexed modal truncation reduces the number of vibration modes considered in the conventional truncation by 60%, and it brings CPU time savings of the same magnitude with truncation errors that are below 1%.

- The deactivation of all the nodes apart from those in the centreline of the CSB deck and the consideration of only vertical and torsional movements in the structure can reduce significantly the number of DOF involved in the analysis (by approximately 27% and 67%, respectively) without affecting the results. However, the CPU time savings associated with these two strategies are below 20%, which is not proportional to their corresponding reductions in the problem size.

- In all the cases the most efficient acceleration technique is the vectorisation of the modal forcing calculation and the Newmark-β solver provided in Section 2.4. This is implemented in the algorithm MS4, which compared with the non-vectorised algorithm MS0 reduces the CPU time by approximately 70% and 90% in the SSB and in the CSB, respectively. It is observed that the efficiency of the vectorisation is reduced slightly when the number of moving loads increases, which is attributed to the inner loop established to calculate the nodal forcing of each vehicle.

- A parametric study of more than $1.6 \cdot 10^5$ combinations with different vehicle arrangements and driving speeds was conducted in MDyn to demonstrate the potential of the algorithm MS5. The results show the significant influence of the number of vehicles and their spacing in convoys like those resulting from connected autonomous trucks (CATs). Reduced vehicle spacings can improve the structural response, but larger values can induce significant resonance problems in the deck if the cadence of passing trucks matches the fundamental period of the structure, particularly in short-span bridges. Therefore, the study of critical convoy arrangements should be conducted for all the bridges of the road network to potentially optimise the spacing and the velocity of CATs in the future.

MDyn is available upon request to the author at: `acamara@ciccp.es`.

Acknowledgements

This work is derived from the project "Driving stability in the Orwell Bridge under high winds", funded by Highways England. Their support is greately appreciated.

Appendix A

Function `InitVNewmark` to initialise the non-iterative vectorised Newmark solver:

```python
def InitVNewmark(P0,q0,qdot0,beta,gamma,dt,M,C,K):
'''
P0: Modal force array in the previous analysis step. Dimension: Number of modes x 1
q0, qdot0, q2dot0: Modal displacement, velocity and acceleration arrays in the previous analysis step, respectively. Dimensions: Number of modes x 1
beta,gamma: Newmark time-integration parameters. Floating point real scalar.
dt: Time-step.
M, C, K:  Modal mass, damping and stiffness arrays, respectively. Dimensions: Number of modes x 1.
Note: M = 1 if modes are normalised with respect to mass. C = 2*xi*w, K = w**2, with xi and w being the damping ratio and the circular frequency of each mode.
'''

  tempv = P0-np.multiply(C,qdot0)-np.multiply(K,q0)
  q2dot0 = np.multiply((1./M),tempv)
  Kbar = K+(gamma/(beta*dt))*C+(1./(beta*(dt**2)))*M
  a = (1./(beta*dt))*M+(gamma/beta)*C
  b = (1./(2*beta))*M+dt*((gamma/(2*beta))-1)*C

  return q2dot0,Kbar,a,b
```

Function `VNewmark` to execute the non-iterative vectorised Newmark solver:

```python
def VNewmark(P0,P,q0,qdot0,q2dot0,beta,gamma,dt,Kbar,a,b):
'''
P0, P: Modal force array in previous and current analysis steps, respectively. Dimensions: Number of modes x 1
q0, qdot0, q2dot0: Modal displacement, velocity and acceleration arrays in the previous analysis step, respectively. Dimensions: Number of modes x 1
q, qdot, q2dot: Modal displacement, velocity and acceleration arrays in the current analysis step, respectively. Dimensions: Number of modes x 1
beta,gamma: Newmark time-integration parameters. Floating point real scalar.
dt: Time-step.
'''

  dpbar = (P-P0)+np.multiply(a,qdot0)+np.multiply(b,q2dot0)
  dq = np.multiply((1./Kbar),dpbar)
  dqdot = (gamma/(beta*dt))*dq-(gamma/beta)*qdot0+dt*(1-(gamma/(2*beta)))*q2dot0
  dq2dot = (1./(beta*(dt**2)))*dq-(1./(beta*dt))*qdot0-(1./(2*beta))*q2dot0
  q = q0 + dq
  qdot = qdot0 + dqdot
  q2dot = q2dot0 + dq2dot

  return q,qdot,q2dot
```

Nomenclature

$\bar{m}^{\mathrm{eff}}_{\mathrm{SM},j}$ Relative effective modal mass activated by mode j in direction SM

$\bar{V}_j$ Non-dimensional speed of the moving load with respect to the j-th mode

β, γ Newmark-β time-integration parameters

$\hat{\boldsymbol{\phi}}$ Modal shape of the j-th vibration mode after nodal deactivation

$\boldsymbol{\iota}_{\mathrm{SM}}$ Rigid body motion of the structure in direction SM

$\widetilde{\boldsymbol{\xi}}$ Vector with the damping ratios of all the modes after the indexed modal truncation

Δt Analysis time-step

$\eta_{\mathrm{SM},j}^{k}$ Dynamic modal contribution factor of the j-th mode at node k in direction SM

$\eta_{\min}$ Minimum dynamic modal contribution above which a vibration mode should be included in the MS analysis

$\bar{\mathbf{K}}, \mathbf{a}, \mathbf{b}$ Arrays needed to initialise the Newmark-β solver

$\hat{\mathbf{\Phi}}$ Reduced modal matrix after the indexed modal truncation and the nodal deactivation

$\hat{\mathbf{P}}_s$ Total generalised force at the nodes selected after nodal deactivation

$\hat{\mathbf{P}}_v, \hat{\mathbf{P}}_w, \hat{\mathbf{P}}_e$ Generalised forces due to vehicles, wind and earthquakes at the nodes selected after nodal deactivation

$\hat{\mathbf{r}}_s, \dot{\hat{\mathbf{r}}}_s, \ddot{\hat{\mathbf{r}}}_s$ Generalised nodal displacements in the nodes selected after nodal deactivation, and their time-derivatives

$\mathbf{\Omega}$ Frequency vector

$\mathbf{\Phi}, \boldsymbol{\phi}, \phi$ Modal matrix, vector and component, respectively

$\widetilde{\mathbf{\Omega}}$ Reduced frequency vector after the indexed modal truncation

$\widetilde{\mathbf{\Phi}}$ Reduced modal matrix after the indexed modal truncation

$\widetilde{\mathbf{M}}, \widetilde{\mathbf{C}}, \widetilde{\mathbf{K}}$ Vectors containing the modal mass, damping and stiffness of all the modes considered after the indexed modal truncation

$\widetilde{\mathbf{P}}_0, \widetilde{\mathbf{P}}$ Modal forcing vectors in the previous and in the current time-step of the analysis

$\widetilde{\mathbf{q}}_\mathbf{0}, \dot{\widetilde{\mathbf{q}}}_\mathbf{0}, \ddot{\widetilde{\mathbf{q}}}_\mathbf{0}$ Reduced vector of modal coordinates after the indexed modal truncation at the end of the previous analysis step, and its time-derivatives

$\widetilde{\mathbf{q}}, \dot{\widetilde{\mathbf{q}}}, \ddot{\widetilde{\mathbf{q}}}$ Reduced vector of modal coordinates after the indexed modal truncation, and its time-derivatives

$\mathbf{j}_r$ Vector with the indices of the relevant vibration modes

$\mathbf{k}_r$ Vector with the indices of the nodes selected after nodal deactivation

$\mathbf{M}_s, \mathbf{C}_s, \mathbf{K}_s$ Mass, damping and stiffness matrices

$\mathbf{P}_s$ Nodal forcing vector

$\mathbf{q}$ Vector of modal coordinates

$\mathbf{r}_s, \dot{\mathbf{r}}_s, \ddot{\mathbf{r}}_s$ Generalised nodal displacements and their time-derivatives

$\mathbf{t}_v$ Time vector

μ Mass of the deck per unit length

Ω_j Driving excitation frequency of the vehicle with respect to the j-th mode

ω_j Circular frequency of the j-th mode

$\omega_{d,j}$ Damped circular frequency of the j-th vibration mode

$\phi^k_{\mathrm{SM},j}$ j-th modal coordinate at node k in direction SM

φ_j Phase angle of the free vibration corresponding to the j-th mode of the SSB

ξ_j Modal damping ratio of the j-th mode

c Parameter related to the free vibration of the SSB

c_v Transverse distance between wheels in the H20-44 truck

d_v Distance between the centroids of consecutive vehicles in the convoy

$d_{v,R}$ Vehicle spacing in the convoy that induces resonance in the deck

EI Vertical flexural stiffness of the deck

f_1 Frequency of the fundamental vibration mode

J Order of the highest vibration mode of interest

J_r Number of vibration modes included after the indexed modal truncation

j^j_r Index of the j-th mode selected in the indexed modal truncation

K_r Number of nodes selected for the dynamic analysis after nodal deactivation

k_R Multiplier of the bridge fundamental period in the calculation of $d_{v,R}$

k_r^k Index of the k-th node selected after nodal deactivation

L Span of the SSB

L_f, L_r Distance between the H20-44 truck centroid and its front and rear axles

$m_{\mathrm{SM},j}^{\mathrm{eff}}$ Effective modal mass activated by mode j in direction SM

m_j Modal mass of the j-th mode

N Number of degrees of freedom (DOF)

N_n Number of nodes in the structure

N_v Number of vehicles in the convoy

P Wheel load in the SSB case study

P_f, P_r Load of each front and rear wheel in the H20-44 truck

P_j Modal forcing of the j-th mode

$q_j, \dot{q}_j, \ddot{q}_j$ Modal coordinate of the j-th mode and its time-derivatives

R Ratio between the peak dynamic displacement and the maximum static deflection at midspan

$r_{\mathrm{SM},J}^k$ Response of the structure at node k in direction SM when the first J vibration modes are included in the MS analysis

r_d^Z Vertical dynamic displacement at midspan

r_s^Z Maximum vertical static displacement at midspan

R_p Peak dynamic response factor induced by a vehicle convoy for any driving speed

$R_{p,1}$ Peak dynamic response factor induced by a single moving vehicle for any driving speed

$t_{\max}$ Time instant in which the simulation stops

V Vehicle speed

V_p Driving speed for which R is maximised

X_j Amplitude of the contribution of the j-th vibration mode to the free vibration of the SSB

SM Number of structural movements before the SM deactivation

SM_a Number of structural movements after the SM deactivation

References

[1] H. Hilber, T. Hughes, R. Taylor, Improved numerical dissipation of time integration algorithms in structural dynamics, Earthquake engineering and structural dynamics 5 (1977) 283–292.

[2] R. Clough, J. Penzien, Dynamics of structures, McGraw-Hill, 1993, second edition.

[3] Python Software Foundation, Python Language, version 3.8, Available at http://www.python.org.

[4] A. Camara, M. Astiz, Applicability of the strategies for the elastic seismic analysis of cable-stayed bridges, Revista Internacional de Metodos Numéricos para Cálculo y Diseño en Ingeniería (in Spanish) 30 (1) (2014) 42–50.

[5] L. Meirovitch, Computational methods in structural dynamics, Sijthoff & Noordhoff, Alphen aan den Rijn, The Netherlands, 1980, vol. 5 of Mechanics: Dynamical Systems.

[6] S. Krenk, State-space time integration with energy control and fourth-order accuracy for linear dynamic systems, International Journal for Numerical Methods in Engineering 65 (2006) 595–619.

[7] K. Foss, Coordinates which uncouple the equations of motion of damped linear dynamic systems, Journal of Applied Mechanics 25 (3) (1958) 361–364.

[8] A. Veletsos, C. Ventura, Modal analysis of non-classically damped linear systems, Earthquake Engineering and Structural Dynamics 14 (2) (1986) 217–243.

[9] A. Ibrahimbegovic, E.L.Wilson, Simple numerical algorithm for the mode superposition analysis of linear structural systems with nonproportional damping, Computers and Structures 33 (1989) 523–533.

[10] D. D. Domenico, G. Ricciardi, Dynamic response of non-classically damped structures via reduced-order complex modal analysis: Two novel truncation measures, Journal of Sound and Vibration 452 (2019) 169–190.

[11] N. Galdos, D. Schelling, Methodology for impact factor of horizontally curved box bridges, Journal of Structural Engineering 119 (6) (1993) 1917–1934.

[12] C. Johansson, C. Pacoste, R. Karoumi, Closed-form solution for the mode superposition analysis of the vibration in multi-span beam bridges caused by concentrated moving loads, Computers and Structures 119 (2013) 85–94.

[13] C. S. Kumar, C. Sujatha, K. Shankar, Vibration of simply supported beams under a single moving load: a detailed study of cancellation phenomenon, International Journal of Mechanical Sciences 99 (2015) 40–47.

[14] C. Bowe, T. Mullarkey, Unsprung wheel-beam interactions using modal and finite element models, Advances in Engineering Software 39 (2008) 911–922.

[15] T. Wang, D. Huang, M. Shahawy, K. Huang, Dynamic response of highway girder bridges, Computers and Structures 60 (6) (1995) 1021–1027.

[16] C. Cai, S. Chen, Framework of vehicle-bridge-wind dynamic analysis, Journal of Wind Engineering and Industrial Aerodynamics 92 (2004) 579–607.

[17] S. Chen, C. Cai, Accident assessment of vehicles on long-span bridges in windy environments, Journal of Wind Engineering and Industrial Aerodynamics 92 (2004) 991–1024.

[18] ABAQUS, Analysis User's Manual, Dassault Systèmes, Simulia Inc., Version 2018.

[19] Y. Xu, Q. Li, D. Wu, Z. Chen, Stress and acceleration analysis of coupled vehicle and long-span bridge systems using the mode superposition method, Engineering Structures 32 (2010) 1356–1368.

[20] EN1998-1, Eurocode 8: Design of structures for earthquake resistance. part 1: General rules, seismic actions and rules for buildings, european Committee for Standardization (CEN) (2004).

[21] A. Chopra, Dynamics of structures, theory and applications to earthquake engineering, Prentice Hall, University of California, Berkeley, 2007, 3rd Edition.

[22] A. Camara, A. Ruiz-Teran, Multi-mode traffic-induced vibrations in composite ladder-deck bridges under heavy moving vehicles, Journal of Sound and Vibration 355 (2015) 264–283.

[23] EN1991-2, Eurocode 1: Actions on structures - part 2: Traffic loads on bridges, EN 1991-2:2003 (2003).

[24] A. Camara, K. Nguyen, A. Ruiz-Teran, P. Stafford, Serviceability limit state of vibrations in under-deck cable-stayed bridges accounting for vehicle-structure interaction, Engineering Structures 61 (2014) 61–72.

[25] AASHTO, LRFD bridge design specifications, 2nd Edition (1998).

[26] J. R. Simpson, S. Mishra, A. Talebian, M. M. Golias, An estimation of the future adoption rate of autonomous trucks by freight organizations, Research in Transportation Economics 76 (2019) 100737.