

Article

An Effective Tangent Stiffness of Train–Track–Bridge Systems Based on Artificial Neural Network

Quan Gu ¹, Jinghao Pan ¹, Yongdou Liu ², Minhong Fu ^{1,*} and Jianguo Zhang ¹

¹ School of Architecture and Civil Engineering, Xiamen University, Xiamen 361005, China; quangu@xmu.edu.cn (Q.G.); panjinghao@stu.xmu.edu.cn (J.P.); zjianguo@xmu.edu.cn (J.Z.)

² School of Civil Engineering, Qinghai University, Xining 810016, China; liuyongdou@stu.xmu.edu.cn

* Correspondence: fuminhong@sina.cn; Tel.: +86-183-7060-0142

Abstract: Dynamic response analysis of a train–track–bridge (TTB) system is a challenging task for researchers and engineers, partially due to the complicated nature of the wheel–rail interaction (WRI). When Newton’s method is used to solve implicit nonlinear finite element equations of a TTB system, consistent tangent stiffness (CTS) is essential to guarantee the quadratic convergence rate. However, the derivation and software implementation of CTS for the WRI element require significant efforts. Artificial neural network (ANN) can directly obtain a potentially good tangent stiffness by a trained relationship between input nodal displacement/velocity and output tangent stiffness. In this paper, the backpropagation neural-network-based tangent stiffness (BPTS) of the WRI element is derived and implemented into a general finite element software, OpenSees, and verified by dynamic response analysis of a high-speed train running on a seven span simply supported beam bridge. The accuracy and efficiency are compared between the use of BPTS and CTS. The results demonstrate that BPTS can not only save the significant efforts of deriving and software implementing CTS but also improve computational efficiency while ensuring good accuracy.

Keywords: wheel–rail interaction; BP neural network; tangent stiffness; train–track–bridge system



Citation: Gu, Q.; Pan, J.; Liu, Y.; Fu, M.; Zhang, J. An Effective Tangent Stiffness of Train–Track–Bridge Systems Based on Artificial Neural Network. *Appl. Sci.* **2022**, *12*, 2735. <https://doi.org/10.3390/app12052735>

Received: 31 December 2021

Accepted: 1 March 2022

Published: 7 March 2022

Publisher’s Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Dynamic response analysis of a train–track–bridge (TTB) system is a challenging task for researchers and engineers, partially due to the strong nonlinearity and complex contact conditions of the wheel–rail interaction (WRI). In recent years, many researchers have proposed different WRI models based on various methods. Zhai et al. [1–4] developed a novel explicit two-step method and a new family of predictor–corrector integration algorithms used in the solving of dynamic problems, based on which they developed a computer program named TTISIM to predict the vertical and lateral dynamic responses of the vehicle–track coupled system. Montenegro et al. [5] proposed a wheel–rail contact formulation for analyzing the nonlinear train–structure interaction that takes into account wheel and rail geometry and used a WRI element to model the behavior of the contact interface based on Hertz’s theory and Kalker’s laws. Pombo [6] presented a general wheel–rail contact detection formulation and calculated the contact forces using Kalker linear theory, a heuristic nonlinear model, and the Polach formulation. Liu et al. [7] developed a 3D WRI element by wrapping the WRI into an element where the contact force is obtained by using Kalker linear theory, nonlinear Hertz theory, and the Polach formulation. The WRI element has been demonstrated to be efficient and accurate, able to simulate complicated TTB systems, and is therefore used in the dynamic responses analysis of the TTB systems in this paper.

When Newton’s method is used to solve implicit nonlinear finite element equations, e.g., the TTB system, the consistent tangent stiffness (CTS) is essential to guarantee the quadratic convergence rate [8–12]. For the WRI element mentioned above, the computation of the CTS requires the differentiation of the internal forces with respect to the nodal

displacement, following the complicated calculation process of the internal forces. Although the CTS is accurate and efficient and is able to guarantee the quadratic rate of convergence of Newton's method [13], its derivation and software implementation require significant efforts. Therefore, it is necessary to find an alternative method for computing the tangent stiffness that can reduce the efforts of derivation and software implementation while maintaining high computational efficiency.

Recently, artificial neural network (ANN) has become increasingly popular in solving a wide range of engineering problems. The ANN can directly obtain a potentially good tangent stiffness by a trained relationship between input nodal displacement/velocity and output tangent stiffness. There are relatively less formula derivation and programming work. The ANN model has been used to solve complex nonlinear problems. Jung et al. [14] presented a rate-dependent ANN constitutive formulation to analyze the time-dependent behaviors of concrete. Luo et al. [15] proposed a novel ANN-based model for fast predicting the backbone curves of nonlinear flexure- and shear-critical columns. Huang [16] proposed an ANN-based method for the identification of a discrete-time nonlinear hysteretic system under strong earthquakes. Jung et al. and Kang et al. [17] proposed an ANN-based health monitoring model for predicting the displacement responses of gravity dams. The ANN has been applied to various fields as an accuracy and efficiency method; however, it has not yet been applied to solving the tangent stiffness of complex nonlinear TTB systems.

In this paper, a BP neural-network-based tangent stiffness (BPTS) method is presented for the WRI element proposed by Liu and co-workers [8]. The input of the BP neural network is the displacements/velocities of the wheel node and rail virtual node, while the output is the tangent stiffness matrix of these nodes. The element tangent stiffness is obtained straightforwardly based on the output tangent stiffness. The trained BPTS model is implemented in a general finite element software, OpenSees [18], to substitute the calculation of the consistent tangent stiffness (CTS) [13] and verified by dynamic response analysis of a high-speed train running on a seven-span simply supported beam bridge. The accuracy and efficiency are compared between the use of the BPTS and the CTS. The BPTS is demonstrated to not only be able to save significant efforts of deriving and software implementing the CTS but also to greatly improve computational efficiency while ensuring acceptable accuracy.

2. The WRI Element and the BP Neural Network

2.1. The Internal Resisting Force and Consistent Tangent Stiffness of the WRI Element

The WRI element consists of a wheel node and all the rail nodes ($b_1, b_2 \dots b_i \dots$) that the wheel node may pass through, and the crucial part of the WRI element [8] is the so-called active portion, as can be seen in Figure 1.

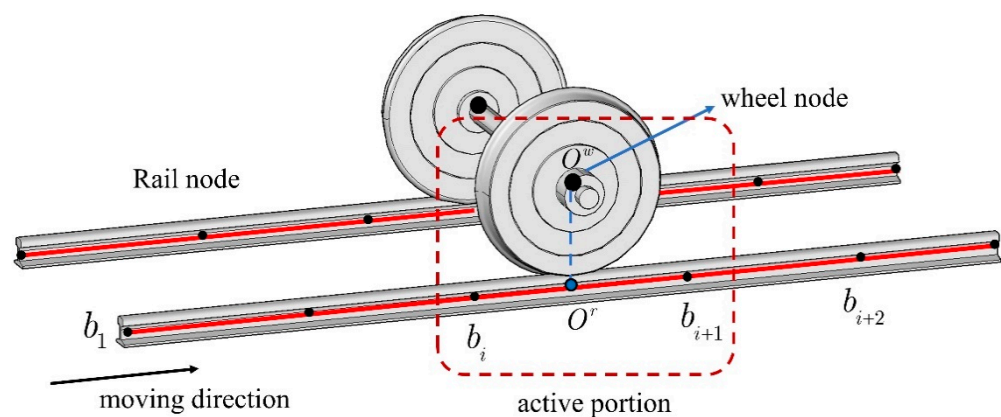


Figure 1. The WRI element and its active portion.

The active portion contains one wheel node and two rail nodes, marked as $O^w, b_i,$ and b_{i+1} , respectively. Then, by projecting O^w vertically onto the rail element, a so-called

‘virtual’ rail node O^r can be obtained. The coordinates of O^r can be obtained by linear interpolation from the coordinates of b_i and b_{i+1} , which can be expressed as

$$\begin{cases} X_1^{O^r} = N_1^L X_1^{b_i} + N_2^L X_1^{b_{i+1}} \\ X_2^{O^r} = N_1^H X_2^{b_i} + N_2^H X_2^{b_i} + N_3^H X_2^{b_{i+1}} + N_4^H X_2^{b_{i+1}} \\ X_3^{O^r} = N_1^H X_3^{b_i} + N_2^H X_3^{b_i} + N_3^H X_3^{b_{i+1}} + N_4^H X_3^{b_{i+1}} \\ X_4^{O^r} = N_1^L X_4^{b_i} + N_1^L X_4^{b_{i+1}} \\ X_5^{O^r} = N_{1,1}^H X_3^{b_i} + N_{2,1}^H X_5^{b_i} + N_{3,1}^H X_3^{b_{i+1}} + N_{4,1}^H X_5^{b_{i+1}} \\ X_6^{O^r} = N_{1,1}^H X_2^{b_i} + N_{2,1}^H X_6^{b_i} + N_{3,1}^H X_2^{b_{i+1}} + N_{4,1}^H X_6^{b_{i+1}} \end{cases} \quad (1)$$

where X_j^i denotes the j th coordinate component of the point i , N_I^L ($I = 1, 2$) denotes the shape function of the linear interpolation, N_I^H ($I = 1, 2, 3, 4$) denotes the shape function of the Hermite interpolation, and $N_{I,1}^H$ ($I = 1, 2, 3, 4$) denotes the derivative of the Hermite interpolation shape functions with respect to x . In each time step, the wheel node moves forward, and the shape functions change. After the wheel node O^w crosses b_{i+1} , the active position is updated immediately and consists of O^w , b_{i+1} , and b_{i+2} .

The element internal resisting force \mathbf{R} consists of the forces on the nodes O^w , b_i , and b_{i+1} , i.e., $\mathbf{R} = [\mathbf{R}_w, \mathbf{R}_{b_i}, \mathbf{R}_{b_{i+1}}]$ and can be obtained based on the nodal displacement \mathbf{u} and velocity \mathbf{v} , i.e., $\mathbf{R} = \mathbf{R}(\mathbf{u}, \mathbf{v})$, where the nodal displacement \mathbf{u} consists of the displacements of the wheel node and two rail nodes, i.e., $\mathbf{u} = [\mathbf{u}_w, \mathbf{u}_{b_i}, \mathbf{u}_{b_{i+1}}]$, while the velocity \mathbf{v} is a function of \mathbf{u} , i.e., $\mathbf{v} = \mathbf{v}(\mathbf{u})$. It is worth mentioning that the function can be explicitly obtained based on the approximation of velocity using displacement when a time-stepping discretization (e.g., Newmark method) is employed. The element internal resisting force \mathbf{R} can be calculated based on the contact force \mathbf{F} between wheel node O^w and virtual rail node O^r , which consists of the normal contact force F_n and the tangential contact force F_t , i.e.,

$$\mathbf{R} = f(\mathbf{F}) = f(F_n(X(\mathbf{u})), F_t(\mathbf{u}, \mathbf{v})) \quad (2)$$

where X denotes the coordinates obtained by element nodal displacement \mathbf{u} .

Suppose there is a virtual interpenetration between the rail and wheel profile, as can be found in Figure 2. The normal contact force F_n^c and tangential contact forces F_{t1}^c and F_{t2}^c are calculated using nonlinear Hertz theory, Kalker linear theory, and the Polach formulation [19].

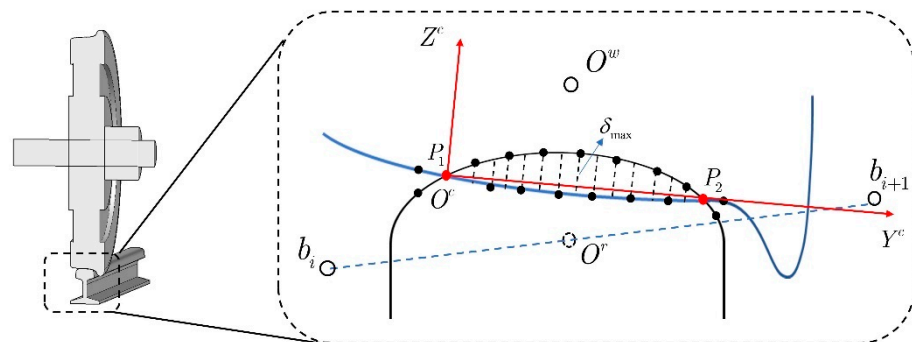


Figure 2. Virtual interpenetration of wheel-rail profile.

After the contact force \mathbf{F} is obtained, the resisting force on the wheel node, \mathbf{R}_w and the resisting force on the virtual rail node \mathbf{R}_r (in Figure 3) can be calculated respectively by

$$\begin{aligned} \mathbf{R}_w &= \mathbf{T}_w^T \mathbf{A}^T \mathbf{F} \mathbf{A} \mathbf{T}_w \\ \mathbf{R}_r &= \mathbf{T}_r^T \mathbf{A}^T \mathbf{F} \mathbf{A} \mathbf{T}_r \end{aligned} \quad (3)$$

where \mathbf{A} denotes the transformation matrix of the coordinate system, \mathbf{T}_w and \mathbf{T}_r denote the transformation matrix that converts \mathbf{F} to the wheel node and the virtual rail node, which is related to the vector from the wheel–rail contact point to the wheel node and that to the virtual rail node [7]. The superscript T denotes the transpose of the matrix. The resisting force on the two rail nodes, \mathbf{R}_{b_i} and $\mathbf{R}_{b_{i+1}}$, are calculated using the force on the virtual node \mathbf{R}_r based on their displacement relation in Equation (1), i.e., using the virtual work principle, as shown in Figure 3.

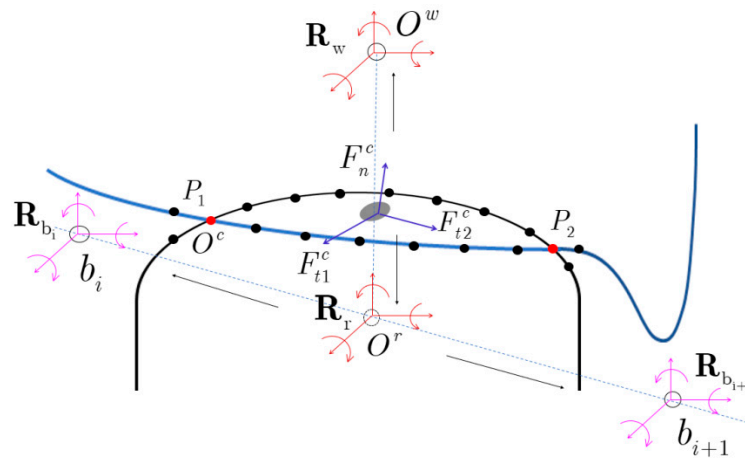


Figure 3. The transform of \mathbf{F} and the resisting force \mathbf{R} .

The consistent tangent stiffness (CTS) is important in solving implicit nonlinear finite element equations, which can guarantee the quadratic rate of convergence when the Newton–Raphson algorithm is used. The CTS is calculated by differentiating the element resisting force with nodal displacement using the chain rule of derivation, i.e.,

$$\frac{\partial \mathbf{R}}{\partial \mathbf{u}} = \left\{ \begin{array}{l} \frac{\partial \mathbf{R}_w}{\partial \mathbf{u}} \\ \frac{\partial \mathbf{R}_{b_i}}{\partial \mathbf{R}_r} \cdot \frac{\partial \mathbf{R}_r}{\partial \mathbf{u}} \\ \frac{\partial \mathbf{R}_{b_{i+1}}}{\partial \mathbf{R}_r} \cdot \frac{\partial \mathbf{R}_r}{\partial \mathbf{u}} \end{array} \right\} \quad (4)$$

where $\frac{\partial \mathbf{R}_w}{\partial \mathbf{u}}$ and $\frac{\partial \mathbf{R}_r}{\partial \mathbf{u}}$ are calculated as

$$\frac{\partial \mathbf{R}_i}{\partial \mathbf{u}} = \frac{\partial \mathbf{R}_i}{\partial F_n} \frac{\partial F_n}{\partial X} \frac{\partial X}{\partial \mathbf{u}} + \frac{\partial \mathbf{R}_i}{\partial F_t} \left(\frac{\partial \mathbf{v}}{\partial \mathbf{u}} \frac{\partial F_t}{\partial \mathbf{v}} + \frac{\partial F_t}{\partial \mathbf{u}} \right) \quad (5)$$

where the subscript “ i ” can be w or r , denoting the force of wheel and rail, respectively. The derivation and implementation of Equation (5) require significant efforts, and the detailed derivation formula can be found in [13].

2.2. The BP-Based Tangent Stiffness (BPTS)

A backpropagation (BP) neural network is used for obtaining an alternative tangent stiffness. The BP neural network consists of an input layer, a hidden layer, and an output layer, as can be seen in Figure 4. Each layer is made up of a few neurons. The number of neurons in the input and output layers is usually determined by the number of input parameters and output parameters, while the number of neurons in the hidden layer is uncertain. Too few neurons may affect the accuracy of the predicted results, but too many may lead to overfitting. Therefore, multiple tests are often needed to select the best number of hidden layer neurons.

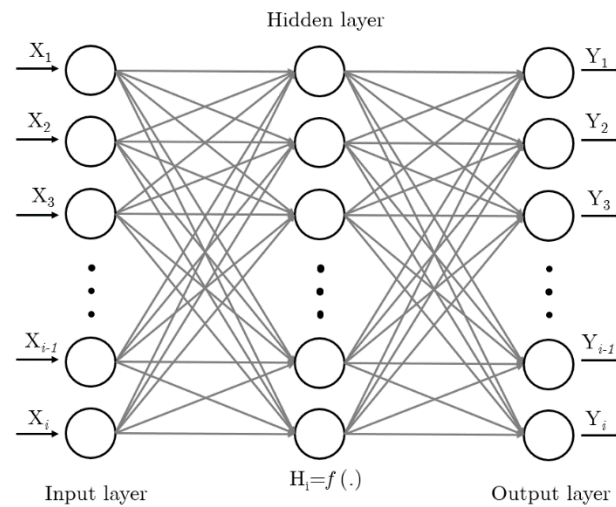


Figure 4. A representation of the feedforward BP neural network.

In the hidden layer, each neuron receives parameters from the input layer and then calculates as follows:

$$h_i = f \left(\sum_{j=1}^m w_{ij} X_j + b_i \right), \quad i = 1, \dots, n \tag{6}$$

where n indexes the neurons in the hidden layer, and m denotes the number of inputs. h_i denotes the i th neuron of the hidden layer, where i ranges from 1 to n . X_j is the j th input, where j ranges from 1 to m . w_{ij} is a weighting coefficient for X_j 's contribution to h_i , while b_i is a bias for the i th neuron in the hidden layer. w_{ij} can be assembled into an $n \times m$ matrix. The f represents a nonlinear activation function, which is the crucial part of a neural network that builds the nonlinear mappings. There are many forms for f , but in this study, the tanh function was used. It can be expressed as

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{7}$$

where e denotes the basis of the natural logarithms. After each neuron in the hidden layer has been calculated, the neurons in the output layer can be expressed as

$$Y_j = \sum_{k=1}^N v_{jk} h_k + d_j, \quad j = 1, \dots, M \tag{8}$$

where M denotes the number of outputs, and N is the number of neurons in the hidden layer. Y_j represents the j th output, where j ranges from 1 to M . h_k is the k th neuron of the hidden layer, where k ranges from 1 to N . v_{jk} is a weighting coefficient for the contribution of h_k to Y_j , while d_j denotes the bias for the j th output. v_{jk} can be assembled into an $M \times N$ matrix. The weighting coefficients and biases (w_{ij} , b_i , v_{jk} , and d_j) in Equations (6) and (8) must be obtained by training with a large number of samples.

The BP neural network's performance is evaluated by comparing its predicted results, and the training samples are performed by using the mean squared error (MSE) of the discrepancies for quantification. The MSE is

$$MSE = \sum_{j=1}^R (Y_j - \bar{Y}_j)^2 / R \tag{9}$$

where Y_j denotes the output calculated by Equation (8), \bar{Y}_j denotes the sample value, and R is the number of the output sample. The MSE is usually guaranteed within a narrow

range (0.001 in this study) to get weighting coefficients and biases that ensure the network’s performance.

In the ANN model for the tangent stiffness of WRI, the inputs X are nodal displacement \mathbf{u} and velocity \mathbf{v} , while the outputs Y are $\frac{\partial \mathbf{R}_w}{\partial \mathbf{u}}$ and $\frac{\partial \mathbf{R}_r}{\partial \mathbf{u}}$ in Equation (4). The BPTS of the WRI element can be obtained by the following steps: (1) Selecting an appropriate example, running enough time steps, recording input X and output Y at each iteration of each time step. (2) Selecting an appropriate activation function in Equation (7) and the learning algorithm, determining the number of hidden layers and number of neurons in each layer. (3) Training the BP neural network model to obtain the weight coefficients and deviations in Equations (6) and (8) (i.e., w_{ij} , b_i , v_{jk} , and d_j). (4) Substituting the CTS with the BPTS, i.e., the mapping relationship in Equations (6) and (8), by software implementing the BPTS algorithm in the WRI element. In practice, the calculation to obtain input X and output Y in step 1 and the training process in step 3 can be performed simultaneously, i.e., at the beginning of the calculation, the CTS (or the tangent by the perturbation method that obtains the tangent by the perturbation of displacement [7]) is used, and the input and output obtained are used to train the ANN model. After the ANN model has been well trained, the BPTS is used instead of the CTS to continue the calculation process so as to realize the intelligent choice of the tangent stiffness.

3. Application Examples

3.1. The Model of Train–Rail–Bridge System

A seven-span simply supported beam bridge [20] is analyzed in this study, the model of which is shown in Figure 5a. The single-span length of the bridge is 32.6 m, and the total length is 228.2 m. Both ends are supported on the foundation, and the middle spans are supported by 8-m-high piers. The model in OpenSees is shown in Figure 5b. The track, girder, and piers are simulated by the elasticBeamColumn elements in OpenSees, the parameters of which can be found in Table 1. The simulation of the cross-section of the bridge in OpenSees can be referred to in Figure 6. The tolerance of the convergence is selected as 0.1 N. The analysis type in this paper is algorithm Newton.

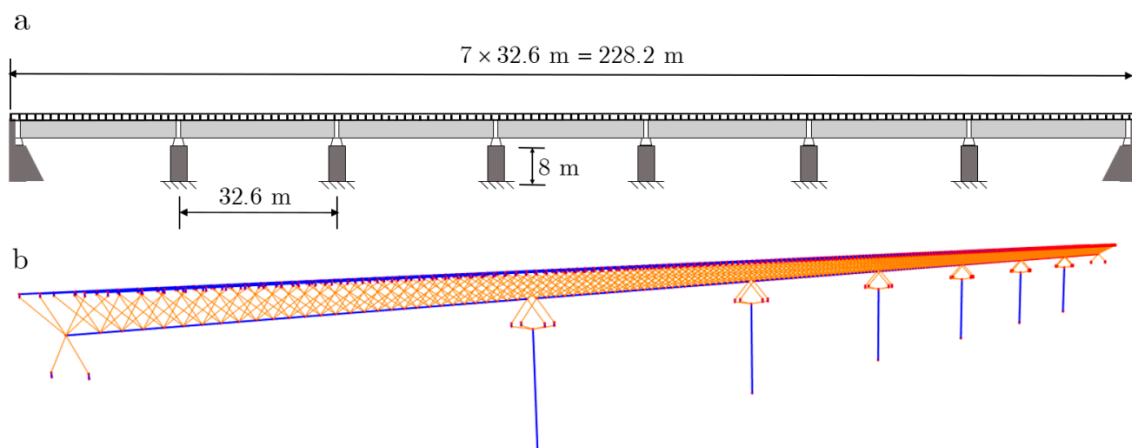


Figure 5. (a) A seven-span simply supported beam bridge. (b) The model of the bridge in OpenSees.

Table 1. The parameters of the elasticBeamColumn beams in OpenSees.

	$A \text{ (m}^2\text{)}$	$I_x \text{ (m}^4\text{)}$	$I_y \text{ (m}^4\text{)}$	$I_z \text{ (m}^4\text{)}$	$E \text{ (N/m}^2\text{)}$	$\rho \text{ (kg/m)}$	μ_r
rail beam	7.67×10^{-3}	2.104×10^{-6}	3.038×10^{-5}	5.123×10^{-6}	2.059×10^{11}	60.21	0.3
girder beam	8.343	16.815	9.0	69.282	4.4×10^{10}	20,857.5	0.2
pier beam	15.86	97.85	29.24	68.61	3.45×10^{10}	39,650	0.2

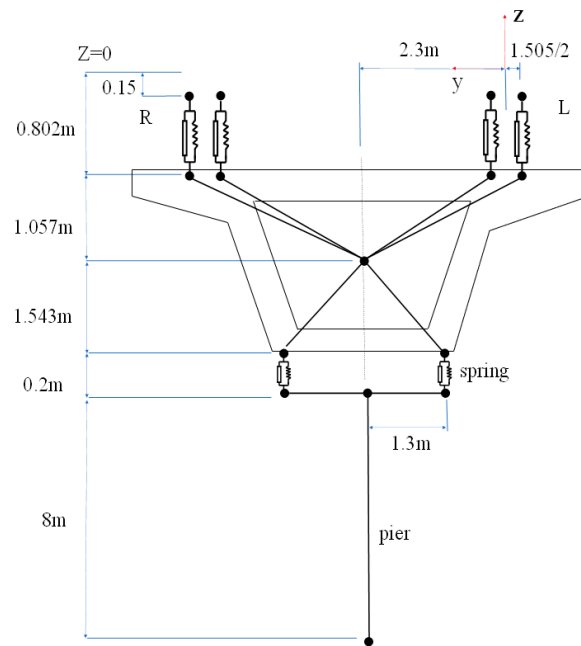


Figure 6. Simulation of bridge cross-section in OpenSees.

The connection between the track and the track plate is simplified by the spring-dampers, and the rail nodes are connected with the track plate nodes through the twoNodeLink element in OpenSees. The length of the twoNodeLink element is 0.787 m, and the spring and damping are three-dimensional. The bridge bearing is also simulated with the twoNodeLink element, and the element length is 0.2 m. However, since the bridge is simply supported, the bearings of each span are different, as can be seen in Figure 7, where k_f, c_f represents the stiffness and damping of the fixed bearing, k_m, c_m represents the stiffness and damping of the movable bearing. The parameters of the twoNodeLink element in OpenSees can be seen in Table 2. All the mass of the girder is concentrated on the girder nodes, and the girder node is connected with the track slab node and the upper node of the bearing through the rigidLink beam in OpenSees. The lower node of the bearing is also connected with the upper node of the pier through the rigidLink beam. The length of the rigidLink beam mentioned above is determined by the actual bridge size. The nodes at the bottom of each pier are fixed.

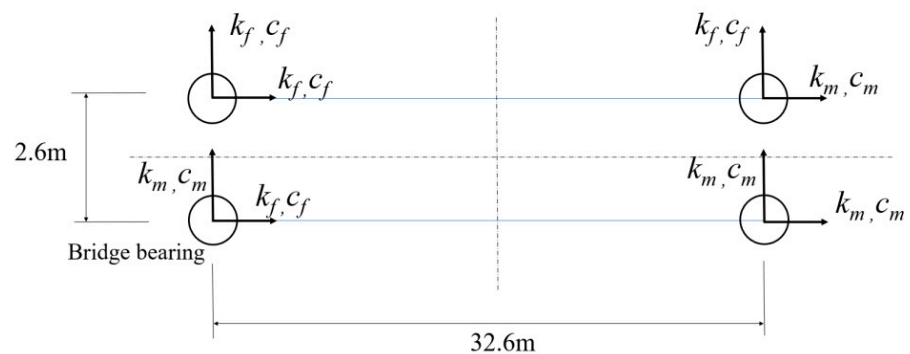


Figure 7. The model of bridge bearing (vertical view).

Table 2. The parameters of the twoNodeLink beams in OpenSees.

	k (N/m)	c (Ns/m)
x-direction	2.0×10^7	5.0×10^4
y-direction	2.0×10^7	5.0×10^4
z-direction	5.0×10^8	2.0×10^5
fixed bearing	1.0×10^{12}	2.0×10^7
movable bearing	7.65×10^5	5.0×10^4
vertical bearing	2.78×10^9	2.0×10^7

3.2. The Data for the BP Neural Network

As mentioned above, the WRI is the most critical part of the TTB system, while the rest, such as the train, is relatively simple and has a limited effect on the element reaction of the WRI element in terms of efficiency and convergence rate. Therefore, only the CTS of the WRI element is replaced by the BPTS. This will greatly reduce the time spent training the neural network. In this paper, the single wheelset is running on the bridge mentioned in Section 3.1 at a constant speed of 20 m/s, as can be seen in Figure 8. Before the dynamic analysis of the wheelset, a horizontal force is applied in the lateral direction such that an original misalignment of the wheelset is 1 mm. The wheelset herein used is the ML95 [6]; the parameters of the wheelset can be found in Table 3, where M_w represents the mass of the wheelset. J_{xx} , J_{yy} , J_{zz} represents inertia about the x , y , and z -axes, respectively. R_0 represents the rolling radius of the wheel. W_r and W_w represent the distance of the right and left rail and wheel nodes, respectively. E is Young's modulus of the rail and wheel and μ is Poisson's ratio. The analysis type in this paper of the OpenSees model is algorithm Newton. The Newmark-Beta method [21] is used for the time integration, with $\gamma = 0.5$ and $\beta = 0.25$ with a time step of 0.001 s. The sign for judging convergence in OpenSees is that the maximum unbalanced force at each time step is less than 0.1 N.

As mentioned in Section 2.2, for the ANN model for the tangent stiffness of WRI, the inputs X are the nodal displacement \mathbf{u} and velocity \mathbf{v} , while the outputs Y are the approximate of $\frac{\partial \mathbf{R}_w}{\partial \mathbf{u}}$ and $\frac{\partial \mathbf{R}_r}{\partial \mathbf{u}}$ obtained by perturbation [7] because its derivation and programming are quite simple. There are 36 inputs and 216 outputs (decomposed as 12 groups with 18 outputs in each group for fast training purposes). A hidden layer neural network is composed of 20 neurons. For training purposes, the above wheelset example runs 3000 time steps, with an average convergence of 6–7 iterations per time step, so as to obtain a total of 19,927 sample data sets.

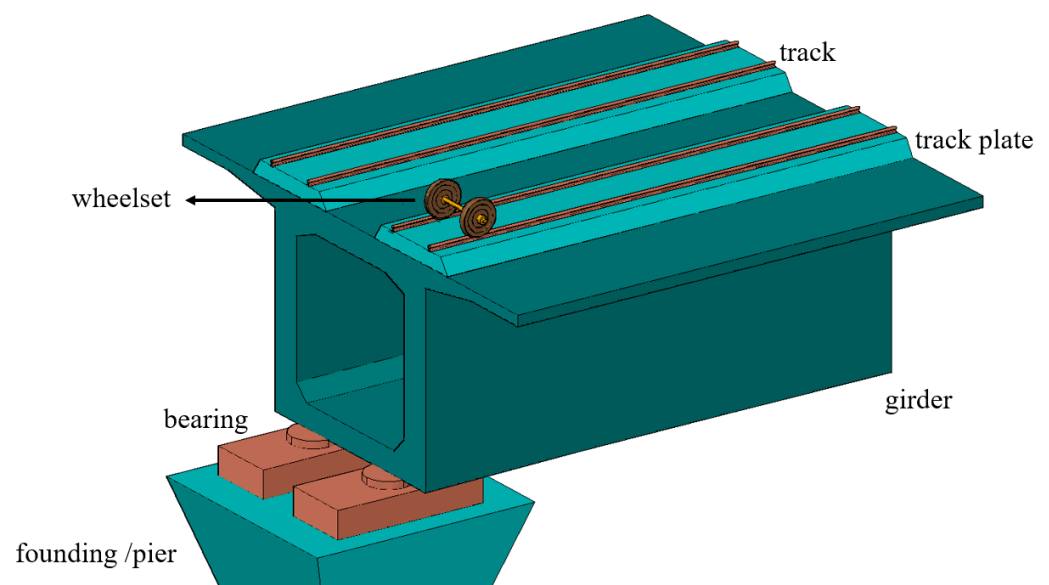
**Figure 8.** A single wheelset running on the bridge.

Table 3. The parameters of the wheelset.

Symbol	Value	Symbol	Value
M_w	933 kg	W_r	1.505 m
J_{xx}	461.4 m ²	W_w	1.5046 m
J_{yy}	61.6 m ²	E_w	2.06×10^{11} N/m ²
J_{zz}	461.4 m ²	μ_w	0.296
R_0	0.43 m		

3.3. Verification of BPTS in the Wheelset Example

The input of the above wheelset example is used to test the performance of the obtained BP neural network, and the predicted results are compared with the actual results, i.e., by CTS. In order to make the results more visible, the data are sampled and plotted every 50 points. Additionally, only some data comparisons are listed to save space, as can be seen in Figures 9–11, respectively. The data in Figure 9 represent the derivative of the resisting force in the x-direction of the wheel node with respect to the translational displacement in the x-direction of the wheel node, marked as $\partial R_{w1}/\partial u_1$. The data in Figure 10 represent the derivative of the resisting force in the y-direction of the rail virtual node to the translational displacement in the y-direction of the wheel node, marked as $\partial R_{r2}/\partial u_2$. The data in Figure 11 represent the derivative of the moment of the rail virtual node around the x-direction to the rotational displacement of the rail node b_{i+1} around the x-direction, marked as $\partial R_{r4}/\partial u_{16}$. It can be observed that the predicted components of tangent stiffness agree well with the actual ones by CTS, although there are irregular jumps of the CTS with iterations. A mean absolute percentage error (MAPE) is used to measure the error between the predicted result and the actual result herein, i.e.,

$$MAPE = \left(\frac{100}{n} \right) \sum_{i=1}^n \left| \frac{Y_i - \bar{Y}_i}{Y_i} \right| \tag{10}$$

where Y represents the actual result and \bar{Y} represents the result obtained by the BP neural network. The MAPEs of the two curves are 1.14%, 0.34%, and 0.0042%, respectively, which shows that the neural network has a good training effect.

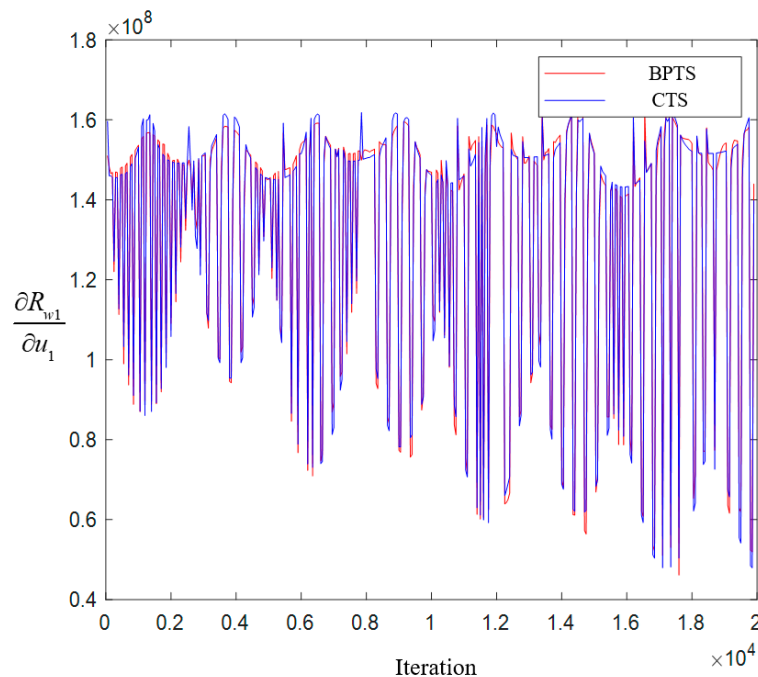


Figure 9. Comparison between the predicted value of the BP neural network and the real value ($\partial R_{w1}/\partial u_1$).

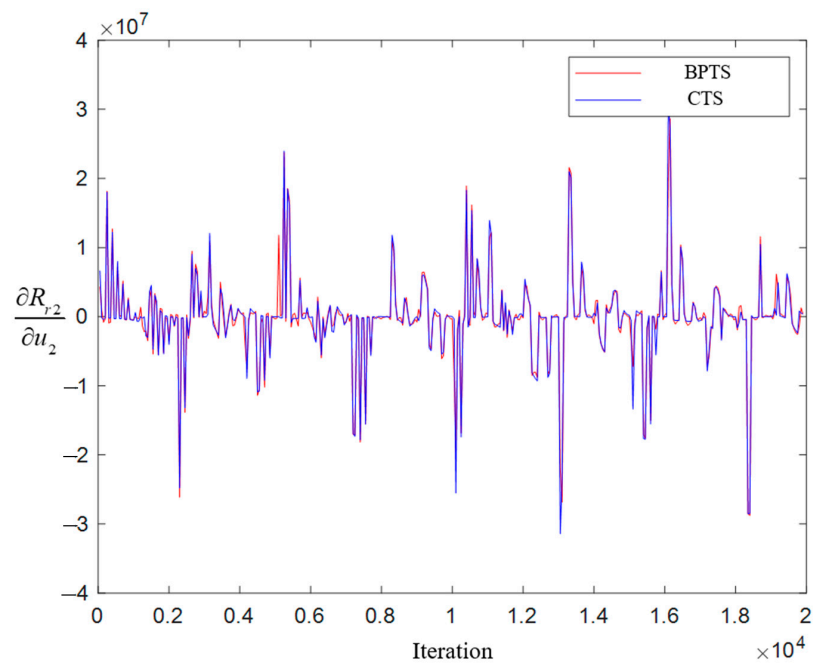


Figure 10. Comparison between the predicted value of the BP neural network and the real value ($\partial R_{r2}/\partial u_2$).

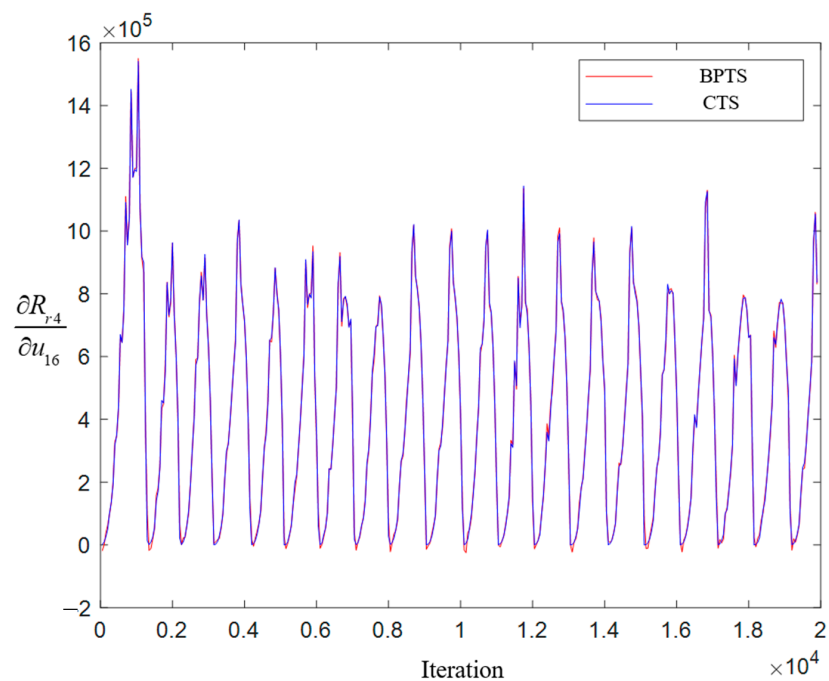


Figure 11. Comparison between the predicted value of the BP neural network and the real value ($\partial R_{r4}/\partial u_{16}$).

After the training is performed, the CTS is replaced by the tangent stiffness predicted by the BP neural network (i.e., BPTS) and implanted into the above wheelset to verify its responses. Figure 12 shows the comparison of the lateral displacement of the left wheel obtained by two tangent stiffness. Figure 13 shows the comparison of the acceleration of the left wheel obtained by two tangent stiffness. The wheelset response calculated by BPTS agrees well with that calculated by CTS.

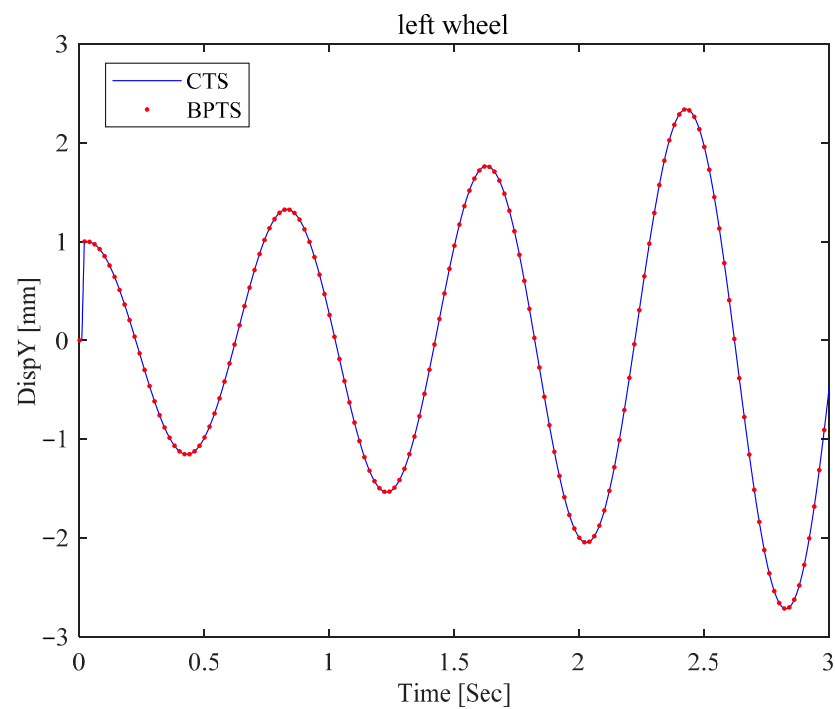


Figure 12. The lateral displacement of the left wheel.

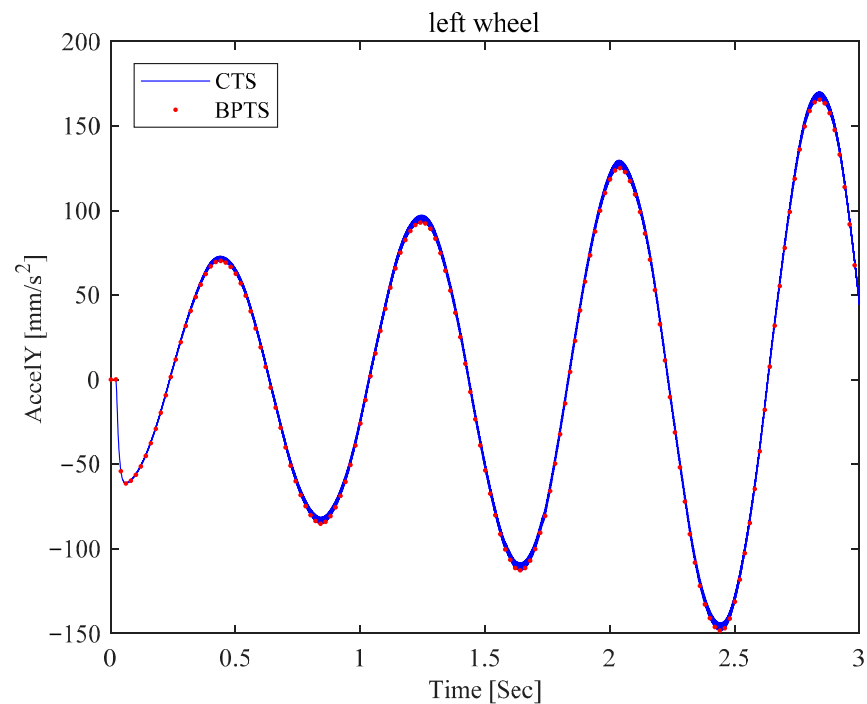


Figure 13. The acceleration of the left wheel.

To compare the convergence rate by using the two tangent stiffness, the convergence data, i.e., the norm of the unbalance force at each iteration in a few representative time steps, are shown in Table 4. It can be observed that the convergence of the Newton method by use of BPTS is slightly slower than the use of CTS because CTS can guarantee quadratic convergence of the Newton method, while BPTS is obtained by the ANN model that approximates CTS but cannot achieve strict quadratic convergence. However, because the calculation of BPTS is much simpler in each iteration step, it has great advantages in overall computational efficiency. In this example, the calculation time using CTS is 7618 s,

while the calculation time using BPTS is 4959 s, which saves 35.0% of the calculation time compared with the former.

Table 4. The norm of the unbalance force in a few representative time steps (wheelset).

Time [s]	Model	1	2	3	4	5	6
0.500	CTS	4.99×10^3	3.04×10^3	6.83×10^1	8.63×10^{-1}	2.38×10^{-4}	
	BPTS	8.37×10^3	6.58×10^3	4.65×10^2	0.36×10^1	9.07×10^{-2}	
1.500	CTS	2.28×10^4	1.13×10^4	5.37×10^1	2.65×10^{-1}	7.54×10^{-5}	
	BPTS	1.25×10^4	8.45×10^3	8.51×10^2	1.88×10^1	0.83×10^{-1}	
2.500	CTS	1.58×10^5	8.79×10^4	2.06×10^2	0.39×10^1	1.67×10^{-3}	
	BPTS	5.45×10^5	8.64×10^3	9.64×10^2	2.60×10^1	1.26×10^{-1}	4.31×10^{-2}

3.4. A Light Rail Vehicle Running on the Seven-Span Bridge

Using the same trained BPTS above (i.e., that obtained by a wheelset running with a constant speed), an ML95 trailer vehicle running on the bridge is studied to verify the BPTS derived herein, as shown in Figure 14. The detail of the model of the car body and bogie can be found in [22–24]. The vehicle consists of the car body, bogies, and wheelsets, respectively. The vehicle body and bogies are modeled using lumped mass points and rigid beams, while the wheelsets are simulated by the wheel nodes mentioned in Section 2.1 and the elastic beam (axis of the wheelsets). The primary suspension of the vehicle is simulated by spring-dampers I between the wheelsets and the bogies, which appear in black. The secondary suspension of the vehicle is simulated by spring-dampers II between the bogies and the car body, which appear in pink. The parameters of spring-dampers I and II can be found in Table 5, where k_I denotes the stiffness of spring-dampers I, c_I represents the damping of spring-dampers I, k_{II} denotes the stiffness of spring-dampers II, and c_{II} represents the damping of spring-dampers II. Both k and c are three-dimensional herein and correspond to x, y, z in Table 5. The rail profile used herein is the UIC50. The element of rail is simulated by an elastic beam, the parameters of which can be found in Table 6, where A_r is the cross-sectional area of the rail, I_{rx}, I_{ry}, I_{rz} represent the area moment of inertia about the $x, y,$ and z -axes, respectively. During the static analysis, the lateral forces applied to each wheelset to make the lateral displacement of the vehicle is 1 mm, and then the forces are removed at the beginning of the dynamic analysis.

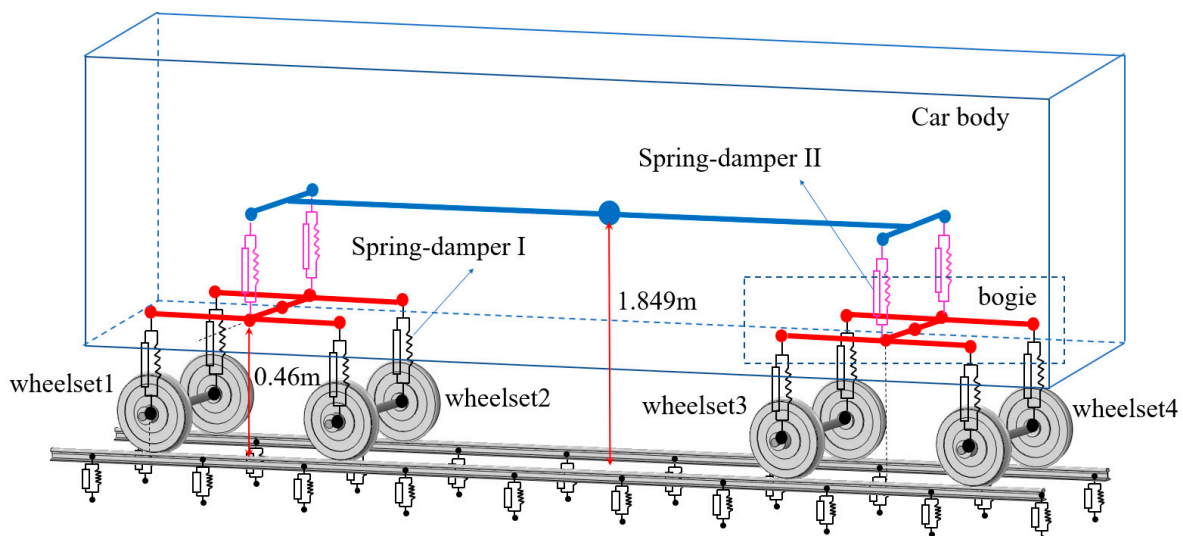


Figure 14. The model of the train and the track.

Table 5. The parameters of spring-dampers I and II.

	<i>x</i>	<i>y</i>	<i>z</i>
k_I (N/m)	1.24×10^7	2.06×10^6	1.28×10^6
k_{II} (N/m)	7.50×10^4	7.50×10^4	2.50×10^5
c_I (Ns/m)	1.73×10^5	7.03×10^4	5.54×10^4
c_{II} (Ns/m)	2.60×10^4	2.60×10^4	4.75×10^4

Table 6. The parameters of the rail element.

Symbol	Value	Symbol	Value
A_r	$7.67 \times 10^{-3} \text{ m}^2$	I_{rz}	$5.123 \times 10^{-5} \text{ m}^4$
I_{rx}	$2.104 \times 10^{-6} \text{ m}^4$	E_r	$20.59 \times 10^{11} \text{ N/m}^2$
I_{ry}	$3.038 \times 10^{-5} \text{ m}^4$	μ_r	0.3

The BPTS trained previously using a wheelset is re-used in the TTB system; the lateral displacement of the left wheel of wheelset 1 is shown in Figure 15 for cases of running speed at 30 and 50 m/s, respectively. Additionally, the lateral displacement of the car body is shown in Figure 16. The responses obtained by the two tangent stiffness are identical.

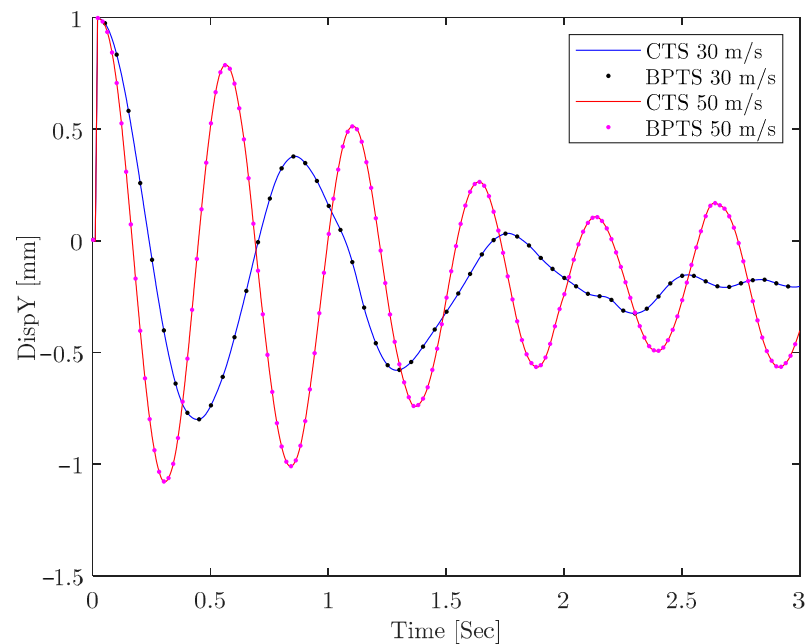


Figure 15. The lateral displacement of the left wheel of wheelset 1.

The convergence data are also recorded. The convergence rates are compared between the use of the two tangent stiffness, as shown in Tables 7 and 8. It can be seen that the convergence rate of BPTS is slightly less than that of CTS. However, the total calculation time is faster when using BPTS over CTS. The total calculation time using CTS is 11,508 s when the speed of the train is 30 m/s, while the calculation time using BPTS is 8264 s, which saves 28% of time. When the train speed is 50 m/s, the calculation times are 16,528 and 13,004 s for the use of CTS and BPTS, respectively, and the use of BPTS saves 21.3% of time. The efficiency improvement by BPTS reduces slightly with the incensement of running speed. The above results show that the BPTS has good applicability in the TTB system. Compared with the CTS, the BPTS can improve computational efficiency and maintain good accuracy.

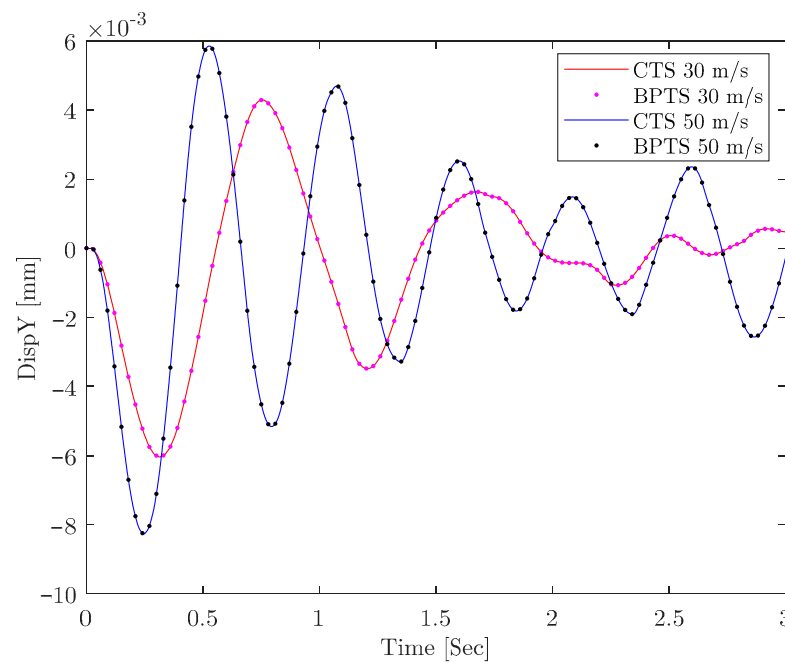


Figure 16. The lateral displacement of the car body.

Table 7. The norm of the unbalance force in a few representative time steps (speed is 30 m/s).

Time [s]	Model	1	2	3	4	5	6	7
0.500	CTS	7.02×10^4	3.26×10^3	1.04×10^1	1.76×10^{-3}			
	BPTS	1.05×10^5	4.76×10^3	5.30×10^2	5.55×10^{-1}	3.21×10^{-2}		
1.500	CTS	6.52×10^4	4.31×10^4	2.34×10^1	3.82×10^{-1}	2.97×10^{-3}		
	BPTS	5.47×10^4	4.70×10^4	6.41×10^2	4.35×10^1	0.17×10^{-1}	1.01×10^{-2}	
2.500	CTS	4.58×10^5	7.73×10^4	1.44×10^3	0.27×10^1	2.43×10^{-2}		
	BPTS	3.31×10^5	1.21×10^5	3.05×10^4	8.61×10^3	0.45×10^1	2.41×10^{-1}	1.95×10^{-2}

Table 8. The norm of the unbalance force in a few representative time steps (speed is 50 m/s).

Time [s]	Model	1	2	3	4	5	6	7
0.500	CTS	4.19×10^4	2.66×10^4	1.84×10^2	2.81×10^{-1}	3.29×10^{-2}		
	BPTS	2.17×10^4	1.37×10^4	2.97×10^3	1.14×10^1	1.16×10^{-1}	7.31×10^{-3}	
1.500	CTS	3.30×10^4	1.42×10^4	2.76×10^3	0.74×10^1	9.71×10^{-2}		
	BPTS	4.21×10^5	3.27×10^4	2.64×10^4	2.64×10^3	1.06×10^1	1.02×10^{-1}	9.39×10^{-3}
2.500	CTS	2.89×10^5	1.58×10^5	2.52×10^4	0.87×10^3	8.50×10^{-1}	2.14×10^{-3}	
	BPTS	6.67×10^5	2.43×10^4	1.87×10^4	4.76×10^3	4.42×10^2	2.41×10^1	3.92×10^{-2}

4. Conclusions

This paper presents a tangent stiffness calculation method based on a backpropagation neural network (i.e., BPTS) for a wheel–rail interaction (WRI) element and implements it into a general finite element software, OpenSees. The BPTS obtained using a wheelset running on a bridge at a constant speed is used for the dynamic response analysis of a realistic train–track–bridge (TTB) system with different vehicle speeds. The accuracy and efficiency of BPTS are studied by comparing the results with those using consistent tangent stiffness (CTS). In the example of a single wheelset, the lateral displacement and acceleration obtained by the two methods are compared and proven to be close. The convergence rate of BPTS is also verified by comparing the norm of the unbalance force and the total computation time. The results show that BPTS needs more convergence numbers but can significantly save the computation time, e.g., when BPTS is used, 35% of time is

saved in the example of a single wheelset. In the example of a light rail vehicle, when BPTS is used, 28% of time is saved. In summary, compared with the CTS, the BPTS can save the significant efforts of derivation and software implementation as well as improve computational efficiency and maintain good accuracy.

This paper discusses a simple working condition with a TTB system, and the load on the train is relatively simple, that is, a small lateral disturbance. The application of BPTS under complex working conditions will be studied in the future, such as the effect of track irregularity and wind load, in order to improve the generality of this method.

Author Contributions: Conceptualization, Q.G.; methodology, Q.G. and Y.L.; software and validation, J.P.; writing—review and editing, J.P. and M.F.; funding acquisition, Q.G. and J.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (No. 51978591) and the Natural Science Foundation of Fujian Province (No. 2018J01085).

Conflicts of Interest: The authors declare no conflict of interest.

References

- Chen, G.; Zhai, W. A new wheel/rail spatially dynamic coupling model and its verification. *Veh. Syst. Dyn.* **2004**, *41*, 301–322. [[CrossRef](#)]
- Zhai, W.; Wang, K.; Cai, C. Fundamentals of vehicle–track coupled dynamics. *Veh. Syst. Dyn.* **2009**, *47*, 1349–1376. [[CrossRef](#)]
- Zhai, W. Two simple fast integration methods for large-scale dynamic problems in engineering. *Int. J. Nu-Merical Methods Eng.* **1996**, *39*, 4199–4214. [[CrossRef](#)]
- Zhai, W.; Cai, C.; Guo, S. Coupling model of vertical and lateral vehicle/track interactions. *Veh. Syst. Dyn.* **1996**, *26*, 61–79. [[CrossRef](#)]
- Montenegro, P.; Neves, S.; Calçada, R. Wheel-rail contact formulation for analyzing the lateral train-structure dynamic interaction. *Comput. Struct.* **2015**, *152*, 200–214. [[CrossRef](#)]
- Pombo, J.; Ambrósio, J.; Silva, M. A new wheel–rail contact model for railway dynamics. *Veh. Syst. Dyn.* **2007**, *45*, 165–189. [[CrossRef](#)]
- Liu, Y.; Montenegro, P.A.; Gu, Q.; Guo, W.; Calçada, R.; Pombo, J. A practical three-dimensional wheel-rail interaction element for dynamic response analysis of vehicle-track systems. *Comput. Struct.* **2021**, *254*, 106581. [[CrossRef](#)]
- Miehe, C. Numerical computation of algorithmic (consistent) tangent moduli in large-strain computational inelasticity. *Comput. Methods Appl. Mech. Eng.* **1996**, *134*, 223–240. [[CrossRef](#)]
- Doghri, I. Fully implicit integration and consistent tangent modulus in elasto-plasticity. *Int. J. Numer. Methods Eng.* **1993**, *36*, 3915–3932. [[CrossRef](#)]
- Jirasek, M.; Patzak, B. Consistent tangent stiffness for nonlocal damage models. *Comput. Struct.* **2002**, *80*, 1279–1293. [[CrossRef](#)]
- Gu, Q.; Conte, J.P.; Yang, Z.; Elgamal, A. Consistent tangent moduli for multi-yield-surface J2 plasticity model. *Comput. Mech.* **2011**, *48*, 97–120. [[CrossRef](#)]
- Simo, J.C.; Taylor, R.L. Consistent tangent operators for rate-independent elastoplasticity. *Comput. Methods Appl. Mech. Eng.* **1985**, *48*, 101–118. [[CrossRef](#)]
- Gu, Q.; Pan, J.; Liu, Y. Consistent tangent stiffness of a three-dimensional wheel–rail interaction element. *Int. J. Struct. Stab. Dyn.* **2021**, *21*, 2140010. [[CrossRef](#)]
- Jung, S.; Ghaboussi, J. Neural network constitutive model for rate-dependent materials. *Comput. Struct.* **2006**, *84*, 955–963. [[CrossRef](#)]
- Luo, H.; Paal, S. Machine learning-based backbone curve model of reinforced concrete columns subjected to cyclic loading reversals. *J. Comput. Civ. Eng.* **2018**, *32*, 04018042. [[CrossRef](#)]
- Huang, C.-C.; Loh, C.-H. Nonlinear identification of dynamic systems using neural networks. *Comput. Civ. Infrastruct. Eng.* **2001**, *16*, 28–41. [[CrossRef](#)]
- Kang, F.; Liu, J.; Li, J.; Li, S. Concrete dam deformation prediction model for health monitoring based on extreme learning machine. *Struct. Control Health Monit.* **2017**, *24*, e1997. [[CrossRef](#)]
- McKenna, F. OpenSees: A Framework for earthquake engineering simulation. *Comput. Sci. Eng.* **2011**, *13*, 58–66. [[CrossRef](#)]
- Polach, O. A fast wheel-rail forces calculation computer code. *Veh. Syst. Dyn.* **1999**, *33*, 728–739. [[CrossRef](#)]
- Guo, W.; Zeng, C.; Gou, H.; Gu, Q.; Wang, T.; Zhou, H.; Zhang, B.; Wu, J. Real-time hybrid simulation of high-speed train-track-bridge interactions using the moving load convolution integral method. *Eng. Struct.* **2020**, *228*, 111537. [[CrossRef](#)]
- Chopra, A.K.; Chopra, A.K. *Dynamics of Structures: Theory and Applications to Earthquake Engineering*; Prentice Hall: Hoboken, NJ, USA, 1995.
- Pombo, J.; Ambrósio, J. *A Multibody Methodology for Railway Dynamics Applications*; Technical Report IDMEC/CPM-2004/003; IDMEC-Institute of Mechanical Engineering, Instituto Superior Técnico: Lisbon, Portugal, 2004.

23. Pombo, J.; Ambrósio, J. Modelling tracks for roller coaster dynamics. *Int. J. Veh. Des.* **2007**, *45*, 470–500. [[CrossRef](#)]
24. Pombo, J.; Ambrosio, J. Dynamic analysis of a railway vehicle in real operation conditions using a new wheel-rail contact detection model. *Int. J. Veh. Syst. Model. Test.* **2005**, *1*, 79–105. [[CrossRef](#)]