

## **Chapter 7**

# **Moving Force Identification using a 2d Model**

## 7.1 Introduction

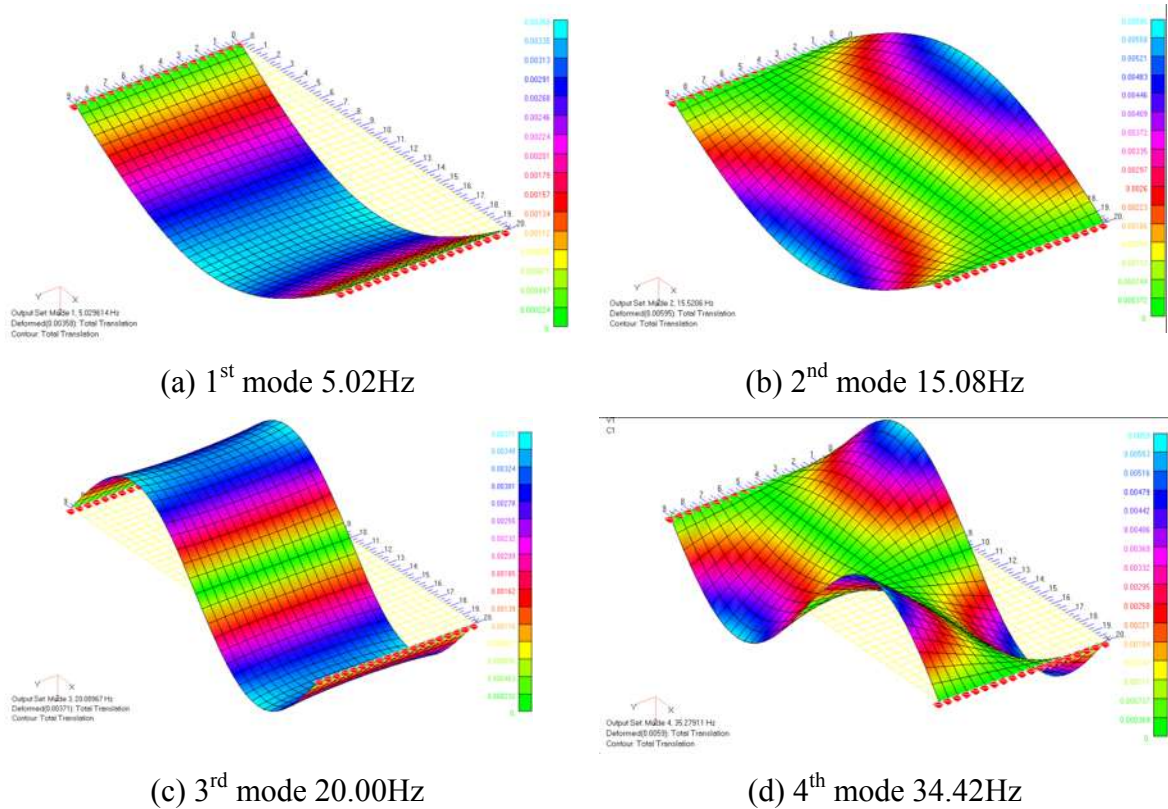
The method of moving force identification developed in Chapters 4 and 5 is extended to a 2 dimensional bridge model, to allow for the transverse behaviour of the bridge and the effect of the transverse location of the truck. A finite element model of a simply supported orthotropic bridge is modelled using a  $C_1$  plate element. The finite element model is programmed in Matlab, and a modal analysis is performed for the implementation of the eigenvalue reduction technique. Further to this, the MFI algorithm is modified, such that the optimal initial conditions of the state vector and the vector of forces can be calculated from the measurements. This means that the bridge need not be at rest before the passage of the force, and the initialisation of the forcing vector results in a significant reduction in the errors at the start and end of the force history for a particular axle. In addition to these changes, the point of maximum curvature of the L-curve is used to calculate the optimal regularisation parameter.

The algorithm is tested using the simulated strain from an independently built 3-D vehicle-bridge-road profile interaction MSc/NASTRAN finite element model that is further contaminated with 2% Gaussian noise. Analysis is carried out on the accuracy of the algorithm for various velocities and road profiles. Finally an error analysis is carried out to assess the effects of various properties on the accuracy of the solution.

## 7.2 Bridge Model and Dynamic Simulation

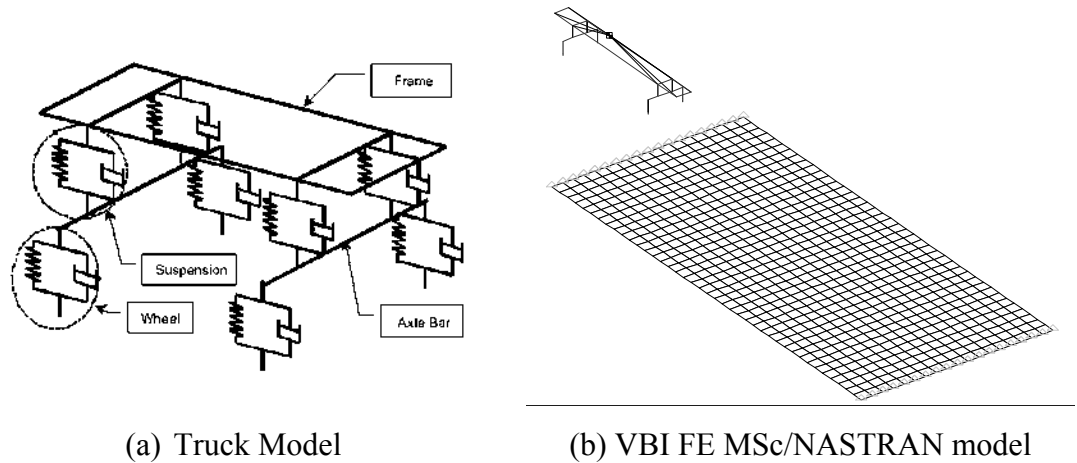
This section presents the simulation technique employed to obtain the bridge strain data that will be used for testing the moving force identification (MFI) algorithm. The truck and bridge have been modelled with the general-purpose finite element analysis package MSc/NASTRAN for Windows. The method uses an approach based on a Lagrange technique that allows for the representation of the compatibility condition at the bridge/vehicle interface through a set of auxiliary functions (Cifuentes 1989). Gonzalez (2001) developed and implemented software in C++ to generate an entry into the assembled stiffness matrix of the vehicle-bridge system provided by MSc/NASTRAN. This entry allows for the definition of the forces acting on the bridge due to the moving wheels. A compatibility condition between the vertical displacement of the wheel, the road profile irregularities and the bridge at the contact point has also been established. The simulated strain from the independently built 3-D vehicle-bridge interaction (VBI) MSc/NASTRAN FE model is used to test the 2d MFI algorithm. A detailed explanation of the (VBI) algorithm and its implementation can be found in Gonzalez (2001).

The bridge deck is idealized as a 20 m long single span orthotropic slab made of plate elements with uniform thickness and density properties. The deck has a voided cross-section of 0.8 m depth with voids of approximately .6m, the bridge properties are thought to be typical of a bridge of this type and have been taken from O'Brien and Keogh (1999). The bridge is 9 m wide and typical properties of  $2500\text{kg/m}^3$ ,  $3.5 \times 10^{10} \text{ N/m}^2$  modulus of elasticity in the longitudinal direction,  $3.22 \times 10^{10} \text{ N/m}^2$  in the transverse direction, shear modulus of  $1.4 \times 10^{10} \text{ N/m}^2$  and .2 Poisson's ratio. Bridge damping is considered to be zero. The first four natural frequencies of the bridge are 5.02 (longitudinal), 15.08 (torsional), 20.00 (longitudinal) and 34.42 (torsional) Hz, and are represented in figure 7.1.



**Figure 7.1** – Modes of vibration of the bridge

A rigid two-axle truck model is used for the simulations. The truck model is shown in figure 7.2(a) and the bridge truck-interaction model is represented in figure 7.2(b). The axle spacing for the two-axle truck is 5.5 m and the distance between wheels of an axle is 2 m. The total front and rear static axle weights are 59.5 and 108.6 kN respectively. The inner wheels of the two-axle truck are driven at .5m from the bridge centreline. This 3-D truck model is composed of bar, mass, damping, spring and rigid elements making up the tire, suspension, frame and body mass of the vehicle. The data and model for the dynamic two-axle truck have been taken from Gonzalez (2001): tyre and suspension damping are 3,000 and 5,000 N·s/m respectively, tyre and suspension stiffness are taken as 1,000,000 N/m, the body mass is 16,500 kg and the corresponding pitching moment of inertia is 425,000 kg·m<sup>2</sup>. The main modes of vibration of the truck, frame twist (0.15 Hz), body pitch (1.07 Hz), body bounce (2.08 Hz) and axle hop out of phase (16.36 Hz). Three levels of speed (20, 25 and 30 m/s) have been employed in the simulations.

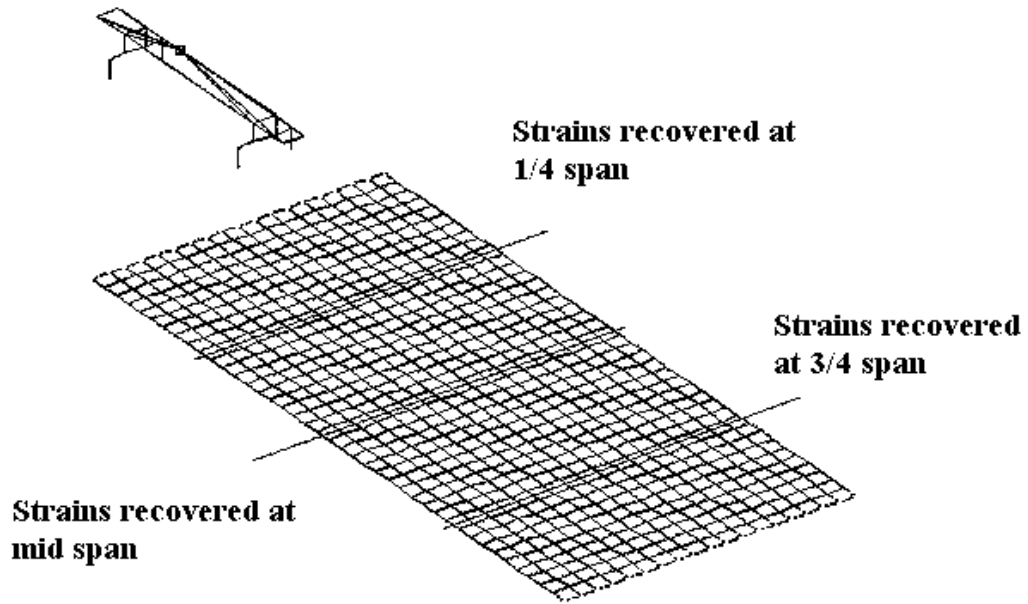


Gonzalez (2001)

**Figure 7.2** – Vehicle model and vehicle bridge interaction model

The road profile is generated stochastically from power spectral density functions following guidelines by ISO standards. Two types of road profile have been considered: ‘smooth’ with a geometric spatial mean of  $4 \times 10^{-6}$ , and ‘rough’ with a geometric spatial mean of  $32 \times 10^{-6}$ .

The results obtained are bridge strains, which are easily measurable in the field. They are obtained at 21 bridge locations and contaminated. The strains are located at 3 longitudinal sections (quarter, mid and three quarter span) and 7 transverse measurements equally spaced across each section width, see figure 7.3 and table 7.1 for the sensor numbering and location. The sensors are located in the centre of the plate elements in the finite element model, such that the strain recovery in the inverse model is from one element and therefore, there cannot be multiple numerical in the entries in the  $[Q]$  matrix.

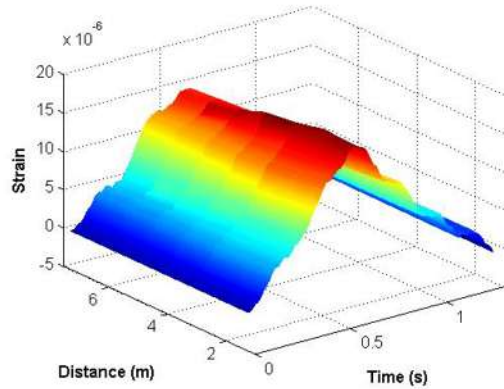


**Figure 7.3** – VBI model and location of strain sensors

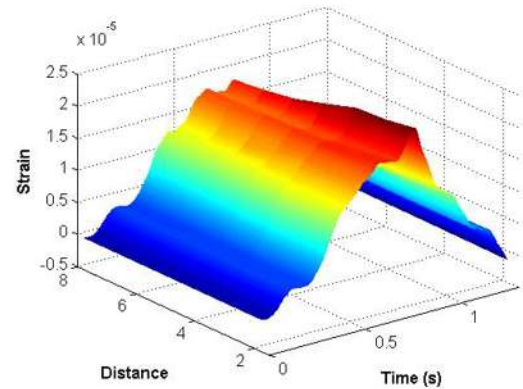
1/4 Span			1/2 Span			3/4 Span		
Sensor	x	y	Sensor	x	y	Sensor	x	y
1	5.25	1.25	8	10.25	1.75	15	15.25	0.75
2	5.25	2.25	9	10.25	2.75	16	15.25	1.75
3	5.25	3.25	10	10.25	3.75	17	15.25	2.75
4	5.25	4.25	11	10.25	4.75	18	15.25	3.75
5	5.25	5.25	12	10.25	5.75	19	15.25	4.75
6	5.25	6.25	13	15.25	6.75	20	15.25	5.75
7	5.25	7.25	14	15.25	7.75	21	15.25	6.75

**Table 7.1** – Sensor numbering scheme and coordinates of the sensors

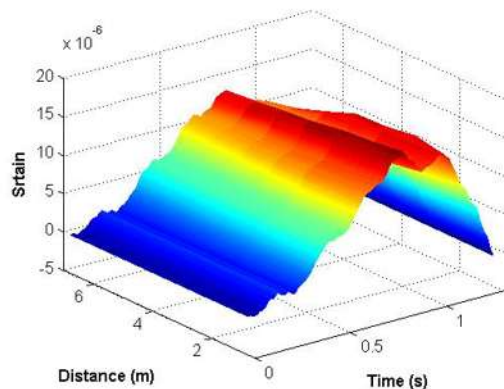
The 2% noise is added to the theoretical strains as a normally distributed variable of zero mean and standard deviation 2% of the maximum strain induced in sensor 11. Figure 7.4 shows the simulated strain at quarter mid and three quarter span. Figure 7.4 (d) shows midspan strain, the noise-free and with 2% added noise, as the truck passes the bridge at 20 m/s on a smooth profile. It is the noisy data that will be used for testing the MFI algorithm presented in the following sections. Then, forces will be estimated using MFI and compared to the ‘true’ applied forces.



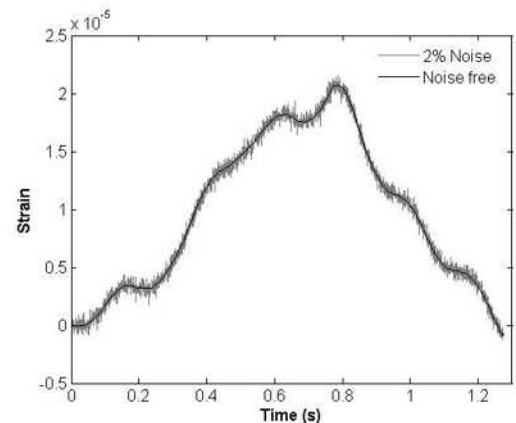
(a) - Simulated strain at quarter span



(b) - Simulated Strain at mid span



(c) - Simulated strain at three quarter span



(d) - Strain at sensor 11 with and without noise

**Figure 7.4** - Simulated strain due to two-axle vehicle moving at 20m/s left lane for the smooth profile

### 7.3 Inverse Model

The MFI algorithm described in detail in chapters four and five requires an accurate FE model of the Bridge. There are currently a multitude of FE software packages that can be used to build both simple and complex FE models of bridges. However extracting the numerical information required to make the models compatible with the MFI algorithm can be very difficult as there is generally little or no access to the globally generated stiffness and mass matrices. Therefore an orthotropic plate element is programmed in Matlab. This section describes the development of an orthotropic plate bending element based on thin plate theory, see Mansfield 1989, Reddy 1993 and appenedix G

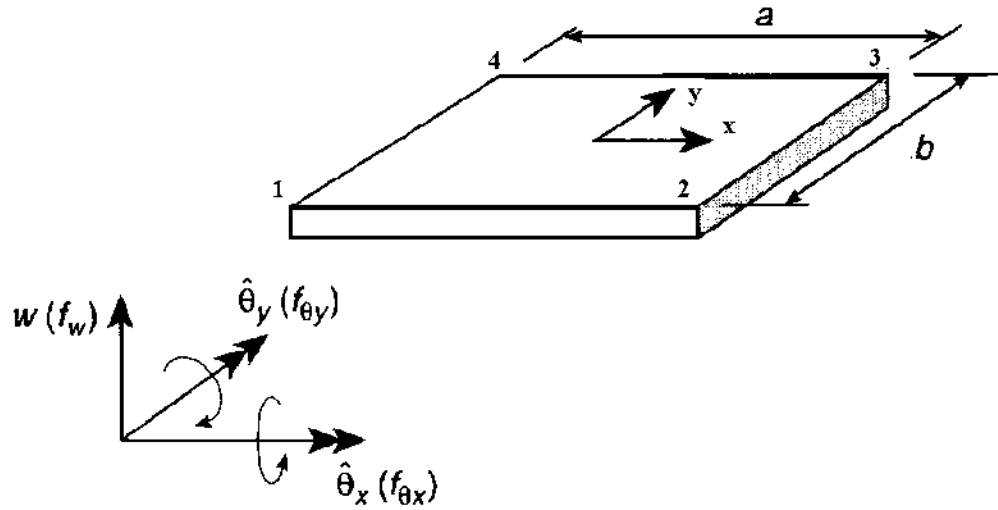
### 7.3.1 Development of a $C_1$ Conforming Orthotropic Rectangular Plate Element

The bending behaviour of thin plates can be completely described by the lateral displacement  $w$ . The standard Kirchhoff plate element (Reddy 1993, Zienkiewicz 1991) contains four nodes and three degrees of freedom per node,  $w$  the vertical displacement and the slopes or rotations about the  $x$  and  $y$  axes. The discretization of the displacement field in this manner will result in a third-order polynomial containing 12 terms in the approximation of the displacement field (Reddy 1993, Zienkiewicz 1991). Such a discretization will violate the continuity conditions of the slopes across inter-element boundaries. Thus kinking can occur which is a discontinuity of the slope across any edge of the element. This was first noticed by Irons and Draper (1965), detailed explanation can also be found in Reddy (1993) or Zienkiewicz (1991). One solution to the problem of the non-conforming plate element is to impose continuity conditions as a degree of freedom, such that the fourth degree of freedom at each node is the second derivative of  $w$  with respect to  $x$  and  $y$ . Bogner et al (1965) first proposed this method for isotropic plates, where the interpolation functions for the degrees of freedom were defined as the products of the Hermite shape functions for an Euler-Bernoulli beam (Bonger et al 1965, Zienkiewicz 1991, Smith & Griffiths 2004, Reddy 1993). However this author could not obtain satisfactory results using the Hermite interpolation functions as basis functions for the conforming orthotropic plate element.

### 7.3.2 Derivation of a 16 Degree of Freedom Orthotropic Plate Element

The solution begins by defining a four node rectangular plate with dimensions  $a$  in the  $x$ -direction and  $b$  in the  $y$ -direction, with a nodal numbering scheme numbered anti-clockwise as shown in figure 7.5.





**Figure 7.5** – Rectangular Plate element

The degree of freedom vector at a node  $i$ , is defined by,

$$\{d_i\} = \begin{Bmatrix} w_i \\ \theta_{xi} \\ \theta_{yi} \\ \theta_{xyi} \end{Bmatrix} = \begin{Bmatrix} w_i \\ -\left(\frac{\partial w}{\partial y}\right)_i \\ \left(\frac{\partial w}{\partial x}\right)_i \\ \left(\frac{\partial^2 w}{\partial x \partial y}\right)_i \end{Bmatrix} \quad (7.1)$$

where the fourth entry in the degree of freedom vector at a node is referred to as the nodal twist (Smith & Griffiths 2004). The nodal forces corresponding to the nodal displacements of equation (7.1) can be defined by a direct force, two moments and the fourth entry can be interpreted as a twisting force defined by,

$$\{F_i\} = \begin{Bmatrix} F_{zi} \\ T_{xi} \\ T_{yi} \\ T_{xyi} \end{Bmatrix} \quad (7.2)$$

The element displacement field is the vector of displacements at all nodes 1 to 4 defined by,

$$\{d^e\} = \{w_1, \theta_{x1}, \theta_{y1}, \theta_{xy1}, \dots, w_4, \theta_{x4}, \theta_{y4}, \theta_{xy4}\}^T \quad (7.3)$$

The elemental stiffness matrix will be a [16 x 16] square symmetric matrix where,

$$\{F^e\} = [K^e] \{d^e\} \quad (7.4)$$

Since the element has 16 degrees of freedom, 16 undetermined constants must be employed in the polynomial expansion. In a similar manner to that of using Pascal's triangle for the polynomial order of triangular elements, a rectangular array can be used to define the complete fourth order polynomial for a rectangular plate element (Reddy 1993). This array is defined in matrix form as,

$$\begin{array}{cccccc} 1 & - & x & - & x^2 & - & x^3 \\ | & & | & & | & & | \\ y & - & xy & - & x^2y & - & x^3y \\ | & & | & & | & & | \\ y^2 & - & xy^2 & - & x^2y^2 & - & x^3y^2 \\ | & & | & & | & & | \\ y^3 & - & xy^3 & - & x^2y^3 & - & x^3y^3 \end{array} \quad (7.5)$$

With the above definition the displacement  $w$  at an arbitrary point in the element can be defined by a fourth order polynomial with 16 coefficients;

$$\begin{aligned} w(x, y) = & \alpha_1 + \alpha_2x + \alpha_3y + \alpha_4xy + \alpha_5x^2 + \alpha_6y^2 + \alpha_7x^2y + \alpha_8xy^2 + \dots \\ & \alpha_9x^2y^2 + \alpha_{10}x^3 + \alpha_{11}y^3 + \alpha_{12}x^3y + \alpha_{13}xy^3 + \alpha_{14}x^3y^2 + \alpha_{15}x^2y^3 + \alpha_{16}x^3y^3 \end{aligned} \quad (7.6)$$

The displacement function defined by equation (7.6) gives the following expression for the nodal displacements as:

$$d_i = \begin{Bmatrix} w_i \\ \theta_{xi} \\ \theta_{yi} \\ \theta_{xyi} \end{Bmatrix} = \begin{Bmatrix} w_i \\ -\left(\frac{\partial w}{\partial y}\right)_i \\ \left(\frac{\partial w}{\partial x}\right)_i \\ \left(\frac{\partial^2 w}{\partial x \partial y}\right)_i \end{Bmatrix} = [P]_i \{\alpha\} \quad (7.7)$$

where  $\{\alpha\}$  is the vector of coefficients of the displacement field defined in equation (7.6) given by,

$$\{\alpha\} = \{\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_{14}, \alpha_{15}, \alpha_{16}\}^T \quad (7.8)$$

and the matrix  $[P]$  defined in equation (7.7) arises from the substitution of equation (7.6) into equation (7.7). By defining the coordinates at nodes 1 to 4 as  $(0,0)$   $(a, 0)$   $(a, b)$  and  $(0,b)$  and substituting these co-ordinates into equation (7.7) the elemental displacement field can be defined by,

$$\{d^e\} = \begin{Bmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \end{Bmatrix} = \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{bmatrix} \{\alpha\} = [A] \{\alpha\} \quad (7.9)$$

Where  $[P]$  and  $[A]$  are defined as follow,

$$[P] = \begin{bmatrix} 1 & x & y & xy & x^2 & y^2 & x^2y & xy^2 & x^2y^2 & x^3 & y^3 & x^3y & xy^3 & x^3y^2 & x^2y^3 & x^3y^3 \\ 0 & 0 & -1 & -x & 0 & -2y & -x^2 & -2xy & -2x^2y & 0 & -3y^2 & -x^3 & -3xy^2 & -2x^3y^2 & -3x^2y^2 & -3x^3y^3 \\ 0 & 1 & 0 & y & 2x & 0 & 2xy & y^2 & 2xy^2 & 3x^2 & 0 & 3x^2y & y^3 & 3x^2y^2 & 2xy^3 & 3x^2y^3 \\ 0 & 0 & 0 & 1 & 0 & 0 & 2x & 2y & 4xy & 0 & 0 & 3x^2 & 3y^2 & 6x^2y & 6xy^2 & 9x^2y^2 \end{bmatrix} \quad (7.10)$$

$$[A] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & a & 0 & 0 & a^2 & 0 & 0 & 0 & 0 & a^3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & -a & 0 & 0 & -a^2 & 0 & 0 & 0 & 0 & -a^3 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 2a & 0 & 0 & 0 & 0 & 3a^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 2a & 0 & 0 & 0 & 0 & 3a^2 & 0 & 0 & 0 & 0 \\ 1 & a & b & ab & a^2 & b^2 & a^2b & ab^2 & a^2b^2 & a^3 & b^3 & a^3b & ab^3 & a^3b^2 & a^2b^3 & a^3b^3 \\ 0 & 0 & -1 & -a & 0 & -2b & -a^2 & -2ab & -2a^2b & 0 & -3b^2 & -a^3 & -3ab^2 & -2a^3b & -3a^2b^2 & -3a^3b^2 \\ 0 & 1 & 0 & b & 2a & 0 & 2ab & b^2 & 2ab^2 & 3a^2 & 0 & 3a^2b & b^3 & 3a^2b & 2ab^3 & 3a^2b^3 \\ 0 & 0 & 0 & 1 & 0 & 0 & 2a & 2b & 4ab & 0 & 0 & 3a^2 & 3b^2 & 6a^2b & 6ab^2 & 9a^2b^2 \\ 1 & 0 & b & 0 & 0 & b^2 & 0 & 0 & 0 & 0 & b^3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & -2b & 0 & 0 & 0 & 0 & -3b^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & b & 0 & 0 & 0 & b^2 & 0 & 0 & 0 & 0 & b^3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 2b & 0 & 0 & 0 & 0 & 3b^2 & 0 & 0 & 0 \end{bmatrix} \quad (7.11)$$

With the above definitions, the vector of coefficients can be solved for by inverting the matrix  $[A]$  defined by;

$$\{\alpha\} = [A]^{-1} \{d^e\} \quad (7.12)$$

The matrix  $[A]$  defined by equation (7.11) is a  $[16 \times 16]$ , and its inverse can be found using the symbolic Math toolbox in Matlab. The displacement at a point  $(x, y)$  on the plate, defined by equation (7.6) can be rewritten in vector form as,

$$w(x, y) = \{1, x, y, \dots, x^2 y^3, x^3 y^3\} \begin{Bmatrix} \alpha_1 \\ \alpha_2 \\ \cdot \\ \cdot \\ \alpha_{15} \\ \alpha_{16} \end{Bmatrix} \quad (7.13)$$

substituting equation (7.12) into (7.13) the displacement can be defined by,

$$w(x, y) = \{1, x, y, \dots, x^2 y^3, x^3 y^3\} [A]^{-1} \{d^e\} = \{N\} \{d^e\} \quad (7.14)$$

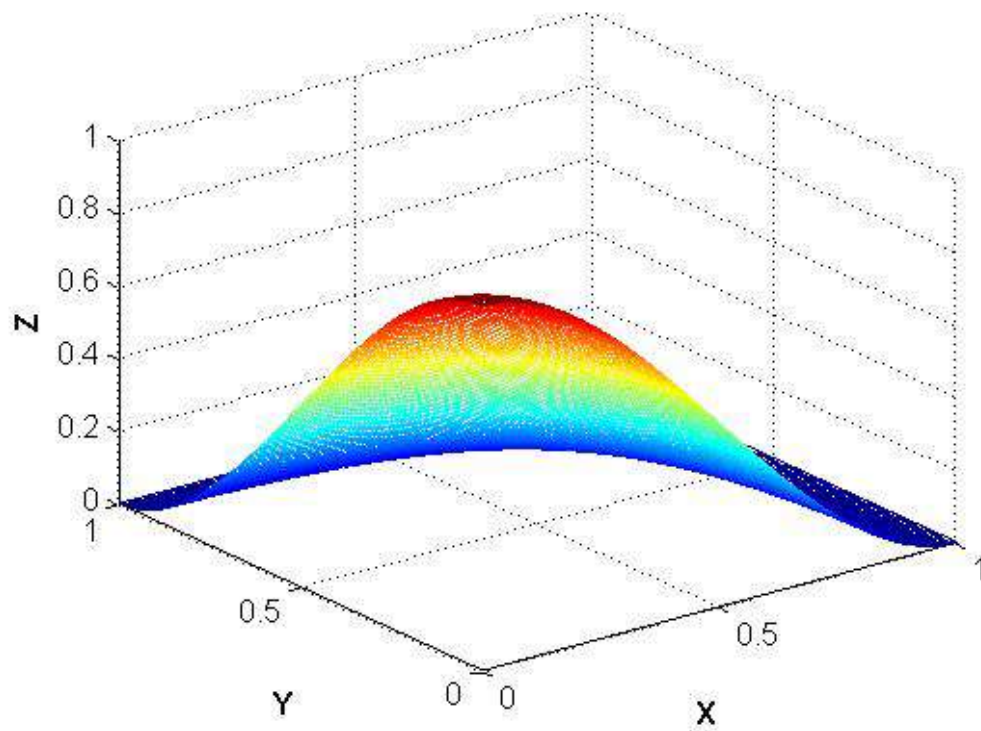
where  $\{N\}$  is the vector of shape functions for all degrees of freedom in the plate, the first 4 shape functions for node 1 are defined as, (see figures 7.7 to 7.10)

$$N_1(x, y) = (b + 2y)(b - y)^2(a + 2x)(-x + a) \frac{1}{a^3 b^3} \quad (7.15)$$

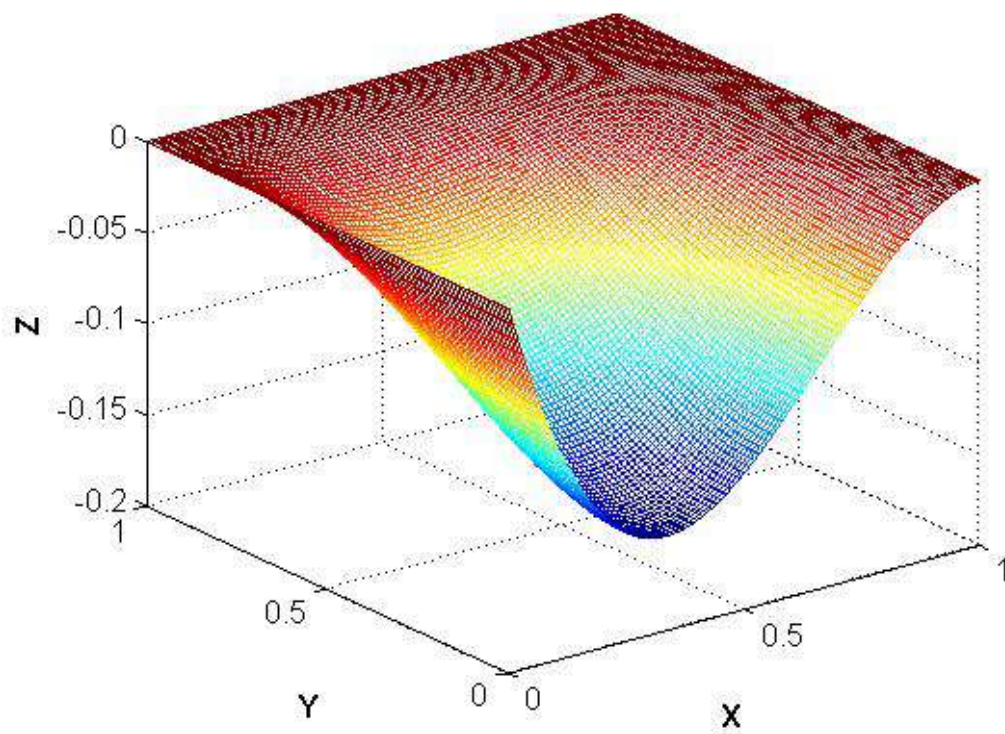
$$N_2(x, y) = -y(b - y)^2(a + 2x)(-x + a)^2 \frac{1}{a^3 b^2} \quad (7.16)$$

$$N_3(x, y) = x(b - y)(-2xab^3 + x^2 b^3 + a^2 b^2 - 2xyab^2 + x^2 yb^2 + ya^2 b^2 - 14xy^2 ab + 10x^2 y^2 b - 2y^2 a^2 b + 18xy^2 a^2 - 12x^2 y^2 a) \frac{1}{a^2 b^4} \quad (7.17)$$

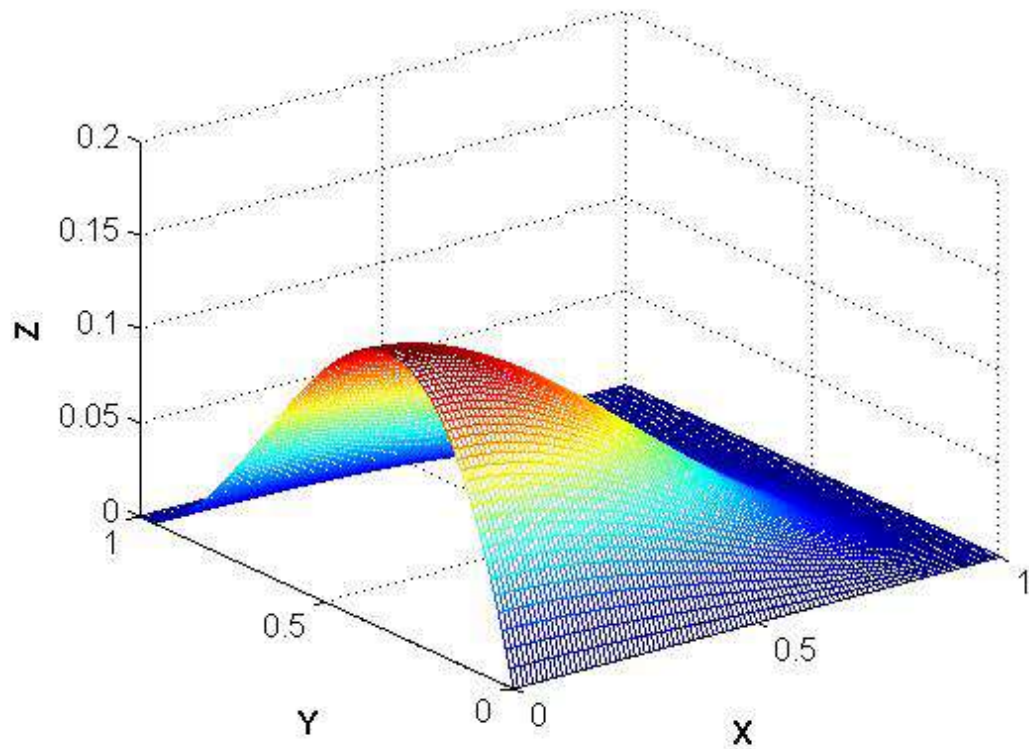
$$N_4(x, y) = xy(b - y)(-2xab^2 + x^2 b^2 + a^2 b^2 - 7xyab + 5x^2 yb - ya^2 b + 9xya^2 - 6x^2 ya) \frac{1}{a^2 b^3} \quad (7.18)$$



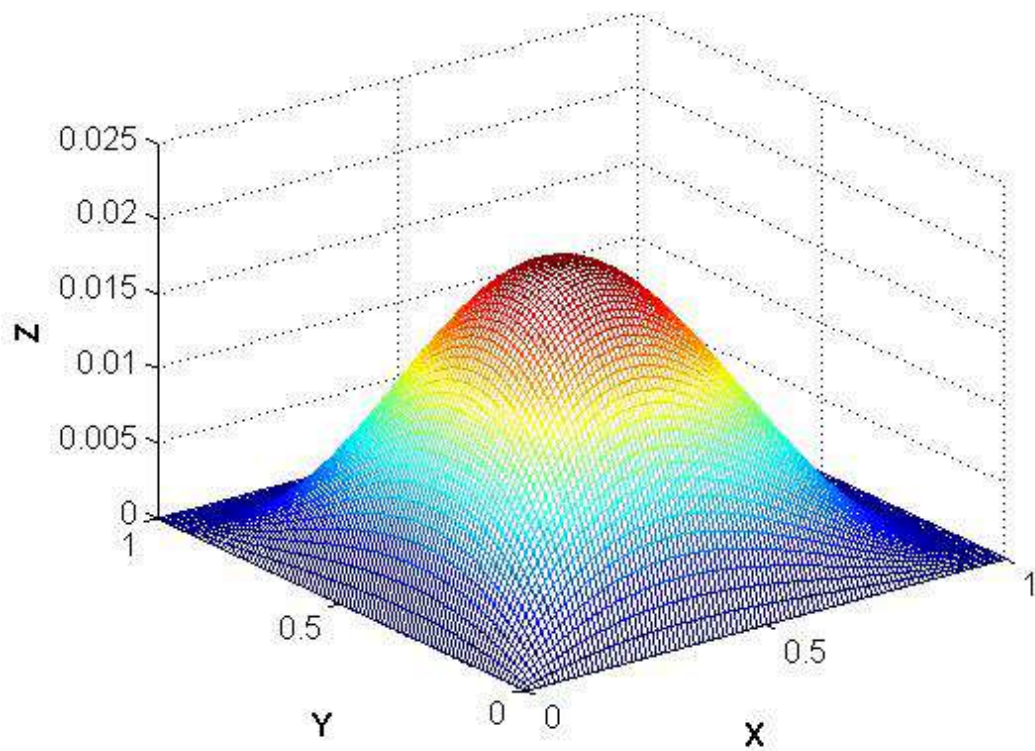
**Figure 7.7** – Shape function for the translation at node 1



**Figure 7.8** – Shape function for the rotation about the  $x$  axis at node 1



**Figure 7.9** – Shape function for the rotation about the y axis at node 1



**Figure 7.10** – Shape function for the twist at node 1

These shape functions are the interpolation formula for the displacements between nodes; the displacement at a point  $(x, y)$  in the element is defined in equation (7.14) as the product of the vector of shape functions by the elemental displacement vector. If the vertical displacement at node 1 was required the value of the first interpolation function at that point would be unity and the value of all other shape functions would be zero, for the interpolation of the vertical displacement.

The strain at a point  $(x, y)$  in the element can be defined from classical thin plate theory as,

$$\varepsilon(x, y) = \begin{Bmatrix} -\frac{\partial^2 w}{\partial x^2} \\ -\frac{\partial^2 w}{\partial^2 y} \\ 2\frac{\partial^2 w}{\partial x \partial y} \end{Bmatrix} = \begin{Bmatrix} -\frac{\partial^2 N_i}{\partial x^2} \\ -\frac{\partial^2 N_i}{\partial^2 y} \\ 2\frac{\partial^2 N_i}{\partial x \partial y} \end{Bmatrix} = B_i \quad (7.19)$$

where  $[B]$  is the strain displacement matrix. This can be written in simpler form by directly substituting equation (7.6) into (7.19) resulting in,

$$\varepsilon(x, y) = \begin{Bmatrix} -\frac{\delta^2 w}{\delta x^2} \\ -\frac{\delta^2 w}{\delta^2 y} \\ 2\frac{\delta^2 w}{\delta x \delta y} \end{Bmatrix} = \begin{Bmatrix} -2\alpha_5 - 2\alpha_7 y - 2\alpha_9 y^2 - 6\alpha_{10} x - 6\alpha_{12} xy - 6\alpha_{14} xy^2 - 2\alpha_{15} y^3 - 6\alpha_{16} xy^3 \\ -2\alpha_6 - 2\alpha_8 x - 2\alpha_{11} y - 6\alpha_{13} xy - 2\alpha_{14} x^3 - 6\alpha_{15} x^2 y - 6\alpha_{16} x^3 y \\ 2\alpha_4 + 4\alpha_7 x + 4\alpha_8 y + 8\alpha_9 xy + 6\alpha_{12} x^2 + 6\alpha_{13} y^2 + 12\alpha_{14} x^2 y + 12\alpha_{15} xy^2 + 18\alpha_{16} x^2 y^2 \end{Bmatrix} \quad (7.20)$$

This can again be written in matrix form as the product of a  $[3 \times 16]$  matrix and the vector of coefficients defined by,

$$\varepsilon(x, y) = [C]\{\alpha\} \quad (7.21)$$

The strain in the element is also defined by the  $[B]$  matrix, as defined in equation (7.19), such that,

$$\varepsilon(x, y) = [B]\{d^e\} = [C]\{\alpha\} \quad (7.22)$$



substituting equation (7.12), into (7.22) gives,

$$[B] = [C][A]^{-1} \quad (7.23)$$

The elemental stiffness matrix can now be defined by,

$$K^e = \int_{\Omega^e} [B]^T [D] [B] dx dy \quad (7.24)$$

$$[K^e] = \int_0^b \int_0^a [[C][A]^{-1}]^T [D] [[C][A]^{-1}] dx dy \quad (7.25)$$

$$[K^e] = [A^{-1}]^T \left( \int_0^b \int_0^a [C]^T [D] [C] dx dy \right) [A^{-1}] \quad (7.26)$$

The integration of equation (7.26) can be achieved using the symbolic math toolbox in Matlab.

With the above definitions of the shape functions, the consistent mass matrix can be formulated. However, as defined by Reddy (1993), the complete dynamic model for a plate contains both the consistent mass matrix, and the mass matrix for rotary inertia. The consistent mass matrix is defined in the usual manner by,

$$M_{ij}^c = \rho h \int_0^b \int_0^a N_i N_j dx dy, i = 1, 2, \dots, 16, j = 1, 2, \dots, 16 \quad (7.27)$$

The mass matrix for rotary inertia is defined by,

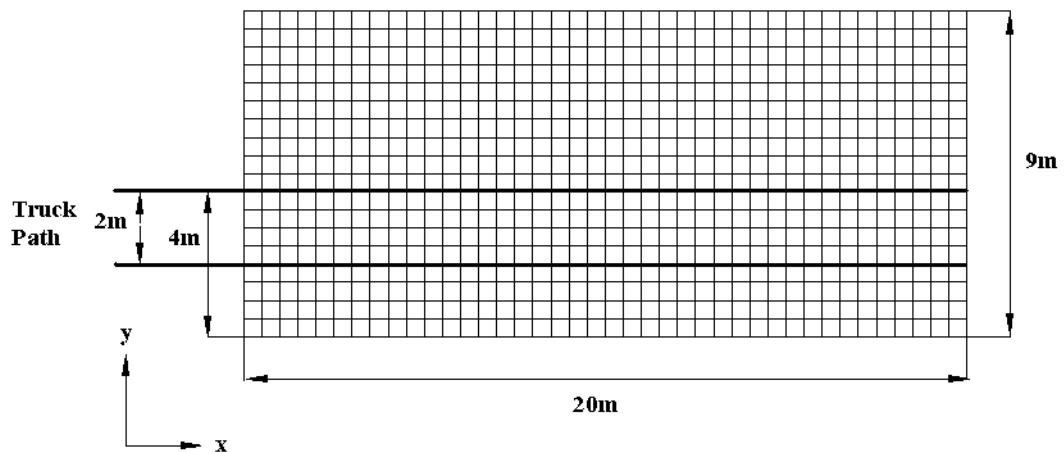
$$M_{ij}^r = \frac{1}{12} \rho h^3 \int_0^b \int_0^a \left( \frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial y} \right) dx dy, i = 1, 2, \dots, 16, j = 1, 2, \dots, 16 \quad (7.28)$$

The integration of equations (7.43) and (7.44) can be achieved using the symbolic math toolbox in Matlab, the complete elemental mass matrix is given by,

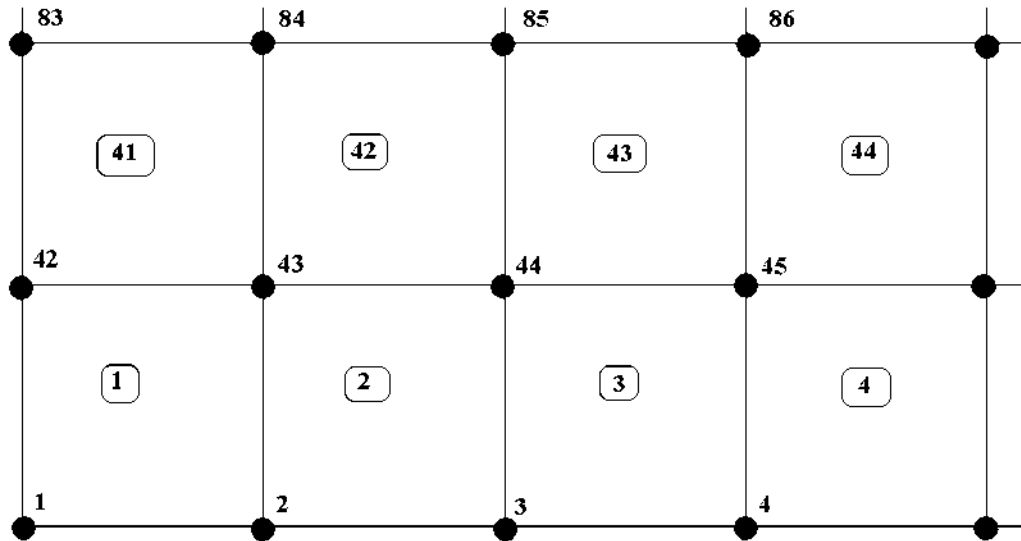
$$M_{ij}^e = M_{ij}^c + M_{ij}^r \quad (7.29)$$

### 7.3.4 Model Validation

The bridge model described in section 7.2 is meshed using the plate element described above. The bridge is discretized using square plate elements each  $.5 \times .5$  m; this gives a total of 720 elements, 779 nodes and a total of 3116 degrees of freedom. The overall bridge model and global numbering scheme used are represented in figures 7.12 and 7.13. The FE mesh is generated such that the wheels are moving along the nodal lines of the mesh. Since the moving force identification algorithm described in chapter five requires a finite number of modes to be used in the inverse analysis, a modal analysis was performed on the finite element model of the bridge and compared with the modes from the Nاستان model. The first 6 modes are represented in figure 7.14, the frequencies from the Matlab model versus those from the Msc/Nastran model are shown in table 7.1.



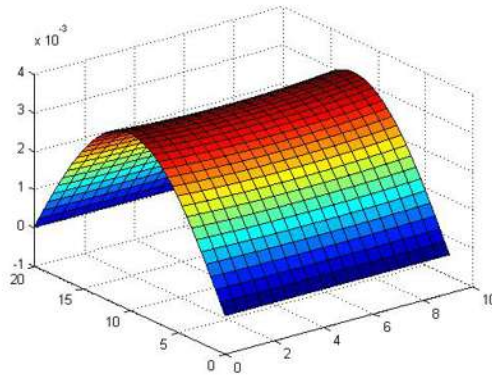
**Figure 7.11** – Finite element Mesh and Truck Path



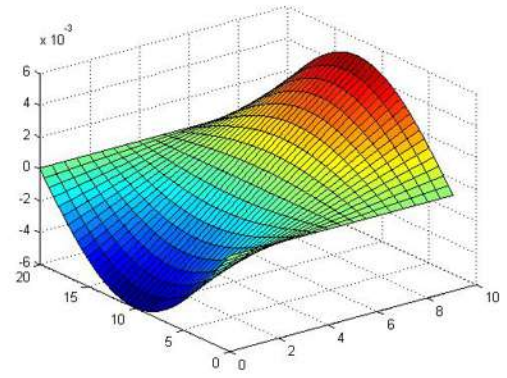
**Figure 7.13** – Global numbering scheme

<b>Msc/Nastran Frequencies</b>	<b>Matlab Frequencies</b>	<b>Percentage Difference</b>
5.0296	5.0371	0.14
15.52	16.348	5.33
20.089	20.263	0.86
35.279	37.032	4.96

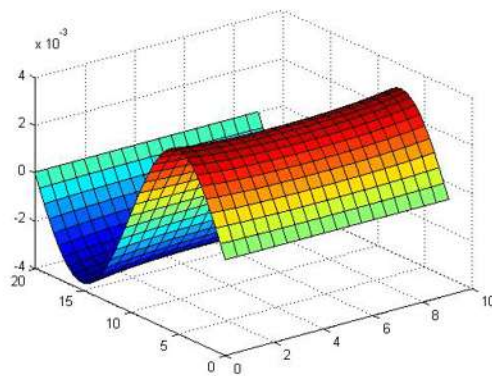
**Table 7.1** - Bridge frequencies from MSc/Nastran and Matlab Models



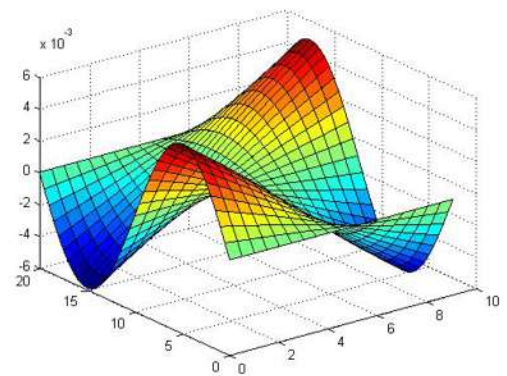
(a) 1<sup>st</sup> mode 5.0371Hz



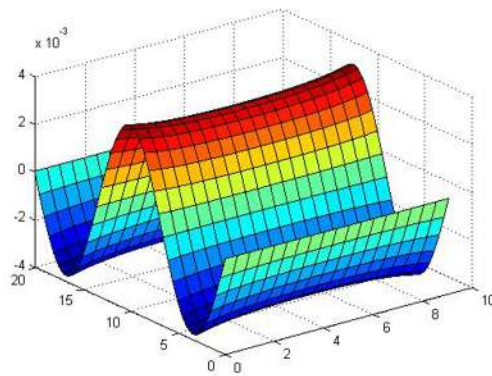
(b) 2<sup>nd</sup> mode 16.348Hz



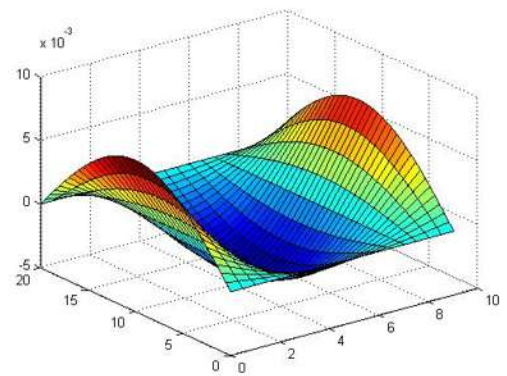
(c) 3<sup>rd</sup> mode 20.263Hz



(d) 4<sup>th</sup> mode 37.032Hz



(e) 5<sup>th</sup> mode 45.697Hz



6<sup>th</sup> mode 64.03Hz

**Figure 7.14 - Modes of vibration of the bridge**

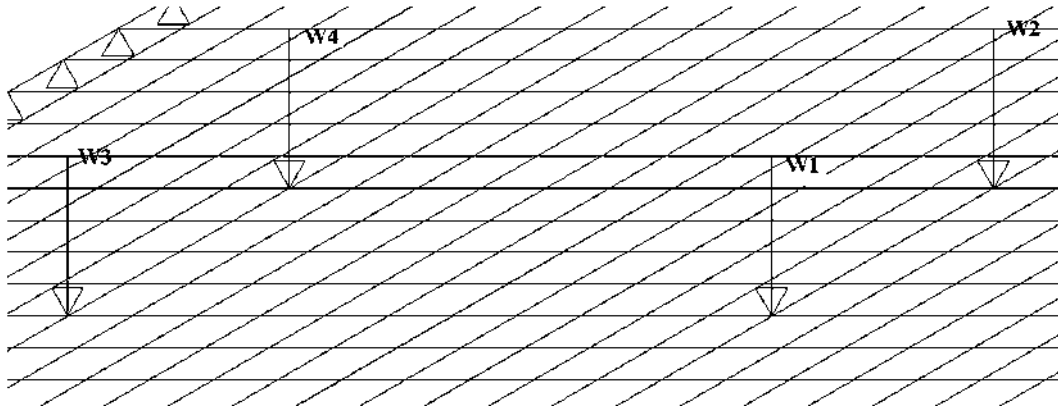
The dynamic behaviour of the MSc/Nastran bridge is compared with the simulated response of the Matlab bridge, to ensure that the difference in frequencies of the two models does not result in a significant difference in overall dynamic response. The finite element model of the bridge in Matlab is simulated for constant forces moving at constant velocity using the method of simulation outlined in section 4.2. The forces are moving along nodal lines, such that the force at a particular point in time is distributed to the dof's of the element as a product of the shape functions defined by,

$$\{F^e\} = F(t) \int_0^b \delta(y - y_0(t)) \int_0^a \delta(x - x_0(t)) \begin{Bmatrix} N_1(x_0(t), y_0(t)) \\ \vdots \\ N_{16}(x_0(t), y_0(t)) \end{Bmatrix} dx dy \quad (7.30)$$

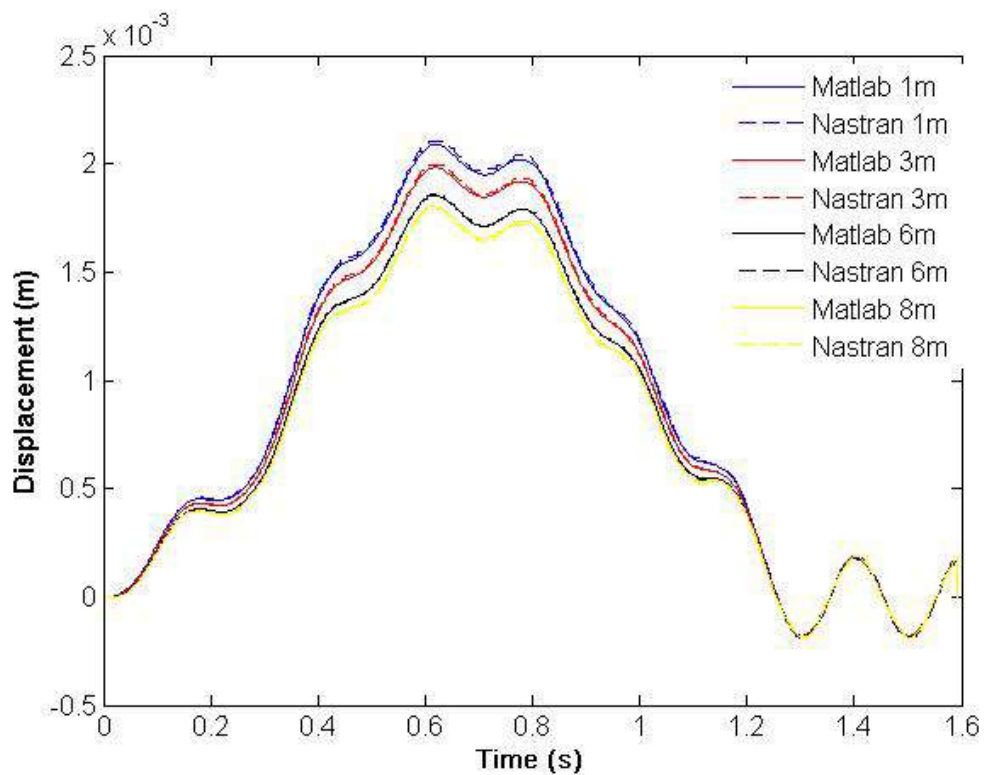
where  $y_0(t)$  and  $x_0(t)$  are the distances of the force to the first node of the element. However as it is assumed that the transverse location of the force is constant, the transverse distance  $y$  of the force from node one is either  $0$  or  $b$ , the length of the particular element that the force is acting on. Therefore the elemental load vector with respect to time can be defined by,

$$\{F^e\} = F(t) \begin{Bmatrix} N_1(x_0(t), y) \\ \vdots \\ N_{16}(x_0(t), y) \end{Bmatrix}, \quad (7.31)$$

The bridge is simulated for four constant loads equivalent to the static weight of the truck defined in section 7.2, moving at 20m/s. The longitudinal distance between the forces is 5.5m and the transverse distance is 2m. The forces are denoted  $W1$  to  $W4$ , where  $W2$  is located at distance of 4m from the first node and  $W1$  is located 2m from the first node see figures 7.16 & 7.11. The displacement with respect to time at the following coordinates (10,1) (10,3) (10,6) and (10,8) is compared with the simulated strain from the MSc/Nastran model; this comparison is represented in figure 7.17.



**Figure 7.16 – Force Location**



**Figure 7.17 - Simulated Displacement Matlab versus MSc/Nastran**

## 7.4 Moving Force Identification using a 2d Approach

The moving force identification algorithm from chapters four and five can be generalised to 2 dimensions. The problem of identifying the moving forces from the “measured” strain is formulated in the exact same manner as defined in section 4.42. The finite element model is converted into a discrete time integration scheme using a finite number of modes as described in section 6.2. As defined in section 6.5 the forces

to be predicted are distributed to the degrees of freedom as a product of the numerical value of the shape function. For the case of the two-axle truck moving at constant velocity the vector of forces to be predicted can be defined by,

$$\{g(t)\} = \begin{Bmatrix} W1(t) \\ W2(t) \\ W3(t) \\ W4(t) \end{Bmatrix} \quad (7.32)$$

These forces must then be mapped to their respective degrees of freedom via the location matrix defined by,

$$L(t) = [L_{w1}(t), L_{w2}(t), L_{w3}(t), L_{w4}(t)] \quad (7.33)$$

Each column of the location matrix  $L(t)$  maps the force to be predicted  $W(t)$  to the degrees of freedom associated with the particular element as a product of the numerical value of the shape function for that point in time. The moving force identification algorithm of chapters four and five can now be implemented. However there are some improvements that have been made to the algorithm to reduce the errors in the predicted forces, particularly at the start and end of the time history. These are as

#### 7.4.1 Curvature of the L-curve

The optimal regularisation parameter is located at the “corner” of the L-curve. In many instances, the range of regularisation parameters over which the L-curve changes from a vertical line to a horizontal line can be significant. Thus, visually selecting the optimal regularisation parameter may not be adequate. Hansen (1992, 1997) gives an operational definition of the corner of the L-curve as the point of maximum curvature. As the L-curve is the plot of the smoothing norm versus the solution norm on log log scales, the parametric sequence of points over which the curvature is to be calculated is defined by,

$$\eta(\lambda) = \log(F_{norm}) = \log\left(\sum_{j=1}^N (r_j, r_j)\right)^{1/2} \quad (7.34)$$

$$\varsigma(\lambda) = \log(E_{norm}) = \log\left(\sum_{j=1}^N ((QX_j - \varepsilon_j), W(QX_j - \varepsilon_j))^{1/2}\right) \quad (7.35)$$

The curvature is purely a geometric consideration and is by (Thomas 2001, WolframMathworld 2007),

$$\kappa(\lambda) = \frac{\varsigma'(\lambda)\eta''(\lambda) + \eta'(\lambda)\varsigma'''(\lambda)}{((\varsigma'(\lambda))^2 + \eta'(\lambda)^2)^{3/2}} \quad (7.36)$$

where differentiation is with respect to the regularisation parameter  $\lambda$ , defined by,

$$\varsigma'(\lambda) = \frac{d\varsigma(\lambda)}{d\lambda} \quad (7.37)$$

$$\eta'(\lambda) = \frac{d\eta(\lambda)}{d\lambda} \quad (7.38)$$

#### 7.4.2 Optimal Initial Conditions

Once the optimal regularisation parameter had been obtained, the solution to the MFI problem can be further improved by applying optimality conditions on the state vector at certain time instances. There are two stages involved in the dynamic programming routine: a backward and forward sweep as defined in Chapter 4. Once all of the terms in the backward sweep have been determined, the forward sweep is defined by,

$$\{r\}_{j-1} = -[D]_j[T]^T \{S\}_j - [D]_j[H]_j[K]_{j-1} \begin{Bmatrix} X_{j-1} \\ g_{j-1} \end{Bmatrix} \quad (7.39)$$

$$\begin{Bmatrix} X_j \\ g_j \end{Bmatrix} = [K]_{j-1} \begin{Bmatrix} X_{j-1} \\ g_{j-1} \end{Bmatrix} + [T] \{r\}_{j-1} \quad (7.40)$$

where  $\{r\}$  is the derivative of the force  $\{g\}$  is the vector of forces to be determined, and  $\{X\}$  is the state vector containing the displacements and velocities of all degrees of freedom in the model. The other terms in the recurrence equations of the forward sweep are as defined in Chapter 4.



The optimal policy at the first time step can be defined by,

$$f_1 \begin{Bmatrix} X_1 \\ g_1 \end{Bmatrix} = \begin{Bmatrix} X_1 \\ g_1 \end{Bmatrix}, [R_1] \begin{Bmatrix} X_1 \\ g_1 \end{Bmatrix} + \begin{Bmatrix} X_1 \\ g_1 \end{Bmatrix}, S_1 + q_1 \quad (7.41)$$

where  $(x,y)$  denotes the vector product. The optimal initial condition of the state vector can be found by minimising  $f$  with respect to the state vector, which results in,

$$\frac{\partial f_1 \begin{Bmatrix} X_1 \\ g_1 \end{Bmatrix}}{\partial \begin{Bmatrix} X_1 \\ g_1 \end{Bmatrix}} = 2 \begin{Bmatrix} X_1 \\ g_1 \end{Bmatrix}, [R_1] + S_1 = 0 \quad (7.42)$$

$$2 \begin{Bmatrix} X_1 \\ g_1 \end{Bmatrix}, [R_1] = -S_1 \quad (7.43)$$

$$\begin{Bmatrix} X_1 \\ g_1 \end{Bmatrix}^* = -[R_1]^{-1} \{S_1\} / 2 \quad (7.44)$$

where  $*$  denotes the optimal. This gives optimal initial conditions for the vector of wheel forces  $\{g^*\}$  and also the displacements and velocities of the dof's if the bridge is not at rest. Furthermore the solution can be greatly improved by applying known boundary conditions on the vector of forces  $g$  in the state vector. The forward sweep of the dynamic programming algorithm can now be divided into three distinct cases. These are,

- (a) Front axle only on the bridge
- (b) Both axles on the bridge
- (c) Rear axle only on the bridge

*Case a*

The initial condition is defined by equation (7.44); this essentially gives an initial guess for the individual wheel loads  $W1$  to  $W4$ . At the first time step in the forward sweep  $W1$  and  $W2$  are initialised by,

$$W1(0) = W2(0) = (W1^* + W2^*) / 2 \quad (7.45)$$

where  $W1^*$  and  $W2^*$  are the optimal initial wheel loads as defined in equation (7.44). As the rear axle is not on the bridge,  $W3$  and  $W4$  must be zero for every time step in case (a). This condition can be implemented in the forward sweep of the dynamic programming routine by zeroing the  $W3(t)$  and  $W4(t)$  entries in the vector  $g(t)$  after each time step of equation (7.40). It should also be noted that the imposition of these boundary conditions does not contradict the principle of optimality that the dynamic programming solution has been derived from. The solution will still constitute an optimal policy with respect to the state resulting from the first decision as the initial state as been optimised for.

*Case b*

At the instant the rear axle enters the bridge,  $W3$  and  $W4$  for that time step can be initialised by,

$$W3(t) = W4(t) = (W3^* + W4^*) / 2 \quad (7.46)$$

where  $W3^*$  and  $W4^*$  are the optimals from the first state defined in equation (7.44). Once  $W3$  and  $W4$  are initialised for the first time step of case (b) the system can be simulated for as long as that case applies.

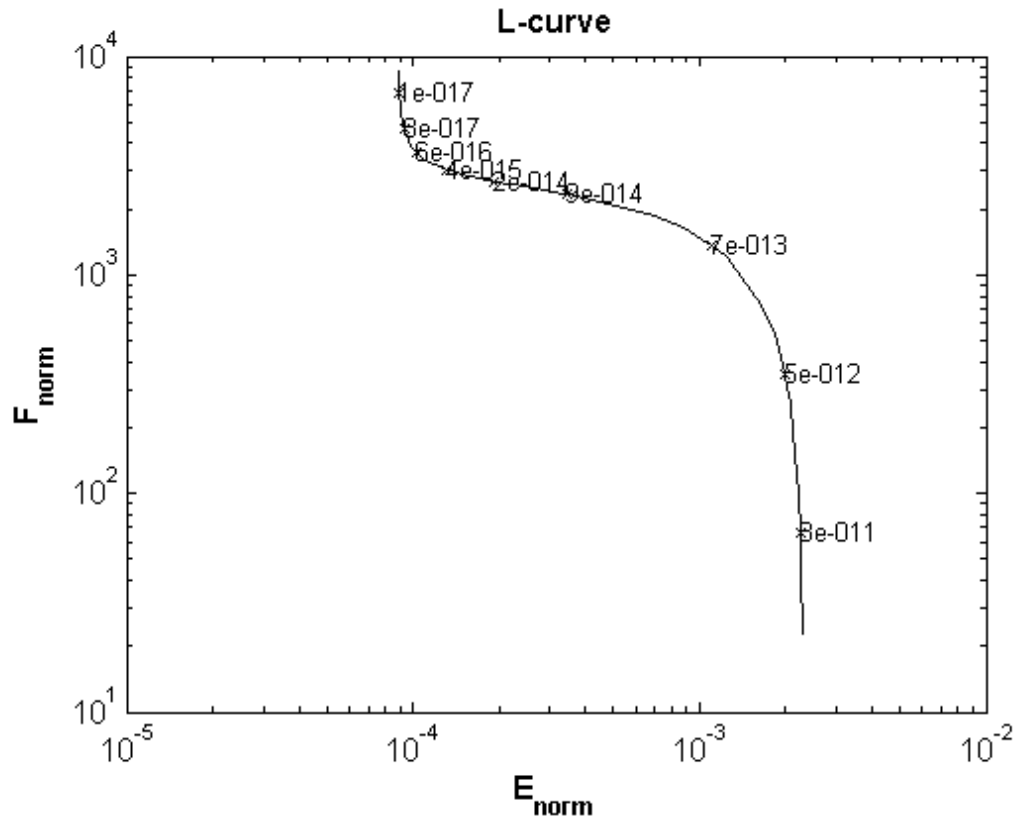
*Case c*

As the system has been simulated already to this point, this gives the optimal for the start of case (c). As in case (a) where there was only one axle on the bridge, the front axle is no longer on the bridge so  $W1$  and  $W2$  must be zero for every time step in case (c). This condition is implemented as in case (a). After each time step of the forward sweep,  $W1(t)$  and  $W2(t)$  are set to zero. It should also be stressed that the above boundary conditions are only implemented once the optimal regularisation parameter

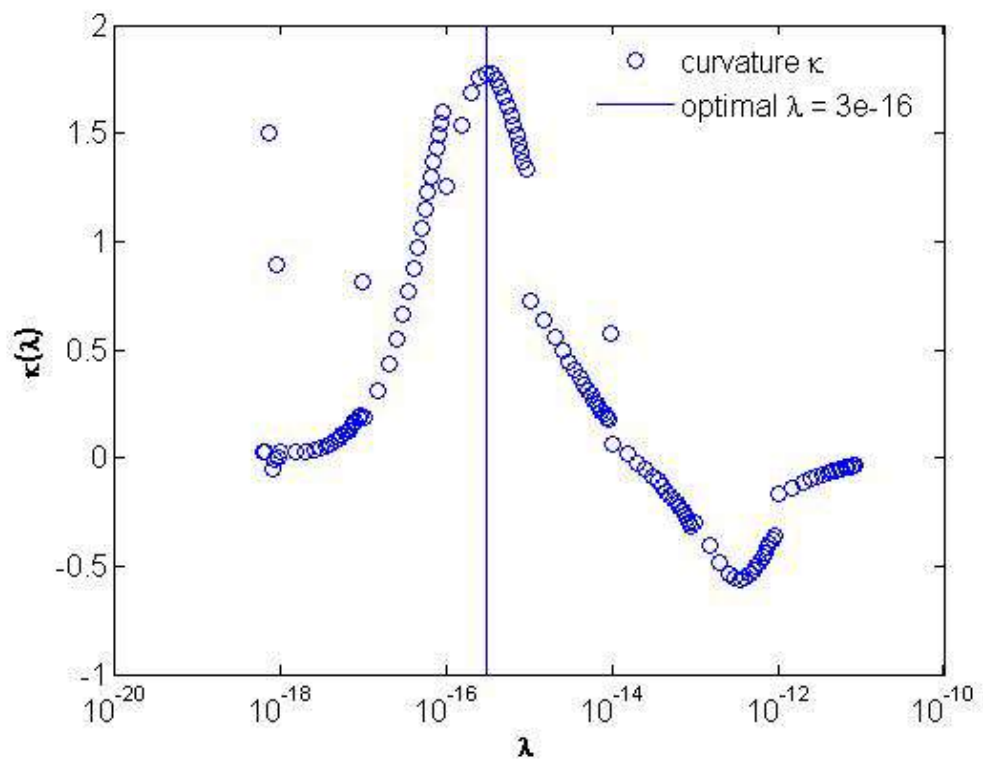
has been found. The numerical procedure for the first order regularisation of the moving forces using an eigenvalue reduction technique is essentially the same as that outlined in chapter four.

## 7.5 Testing with Simulated Measurements

The 2d moving force identification algorithm is tested using the simulated measurements from the 3d VBI MSc/NASTRAN FE model. Initially all 21 “measurement” points are used in the inverse analysis. The simulation of the rigid two axle vehicle travelling at 70km/hr on the smooth profile is used as an example to illustrate the improvements that have been made to the MFI algorithm using an eigenvalue reduction technique. The optimal regularisation parameter is calculated initially from the L-curve; 25 modes of vibration have been used in the inverse analysis. Figure 7.19 shows the L-curve for the vehicle travelling at 20m/s. it is clear from figure 7.18 that the optimal regularisation parameter lies approximately between  $8 \times 10^{-17}$  and  $1 \times 10^{-15}$ . The curvature of the L-curve is then calculated as defined in equation (7.36). Figure 7.20 shows the plot on a semi log scale of the curvature versus the regularisation parameter. Ignoring the local discontinuities due to the numerical differentiation the optimal regularisation parameter is  $3 \times 10^{-16}$ .

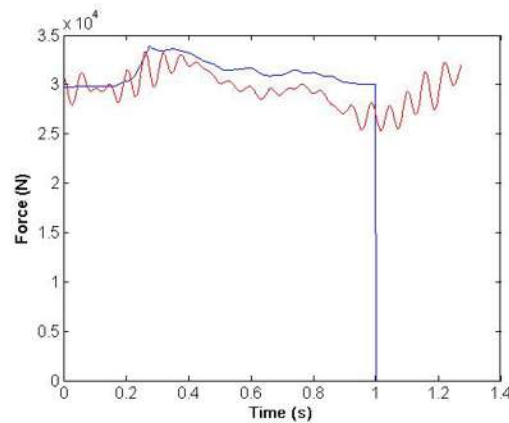


**Figure 7.19** – L-curve due to the vehicle travelling on a smooth profile at 20m/s

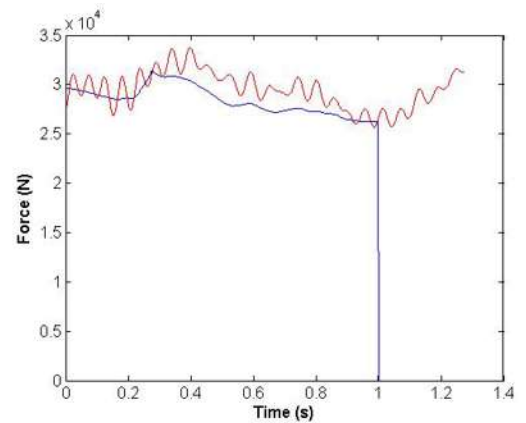


**Figure 7.20** – Variation of curvature with optimal regularisation parameter

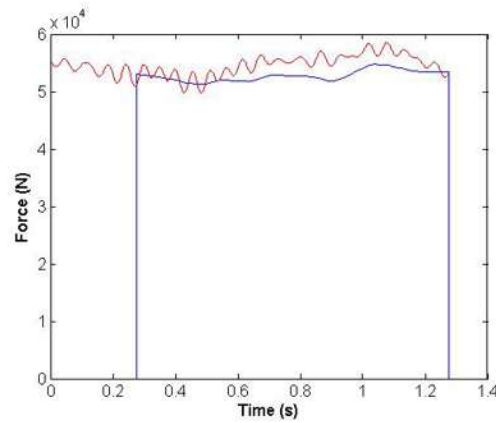
The backward sweep of the dynamic programming routine is then repeated for the optimal regularisation parameter. In the forward sweep, the optimal initial conditions are then used as defined in section (7.4.2). The dynamic axle loads are calculated as the sum of the individual wheel loads. Figures 7.21 shows the variation in predicted wheel loads with respect to time. The true and predicted axle force histories are represented in figure 7.22. The road profile is very smooth and the vehicle dynamics are very small. There is only one main low frequency that the MFI algorithm is able to model for both axles accurately. The algorithm appears to be unable to accurately predict the individual wheel loads; it is thought that this could be due to the difference in the torsional modes of vibration between the two models as shown in table 7.1. For the scenario of the vehicle moving at 25m/s on a rough profile, the L-curve provides an optimal regularisation parameter of  $4 \times 10^{-17}$  see figure 7.22 and the axle force history is illustrated in Figure 7.23. In the latter, there are two dominant axle frequencies and a clear correspondence between the predicted and simulated instantaneous axle forces. Although some peaks are missed, the average values about which dynamics oscillate, are very similar for both the true and predicted. As expected the values are slightly inaccurate at the start and end of the histories. However the improvement of the predicted forces using the optimal initial conditions of section 7.4.2 is evident when compared with the results from chapters four and five.



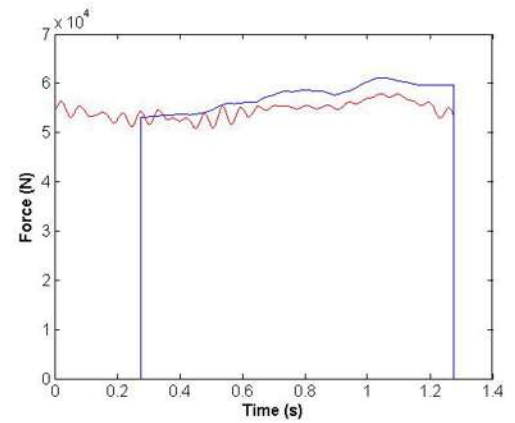
(a) W1(t)



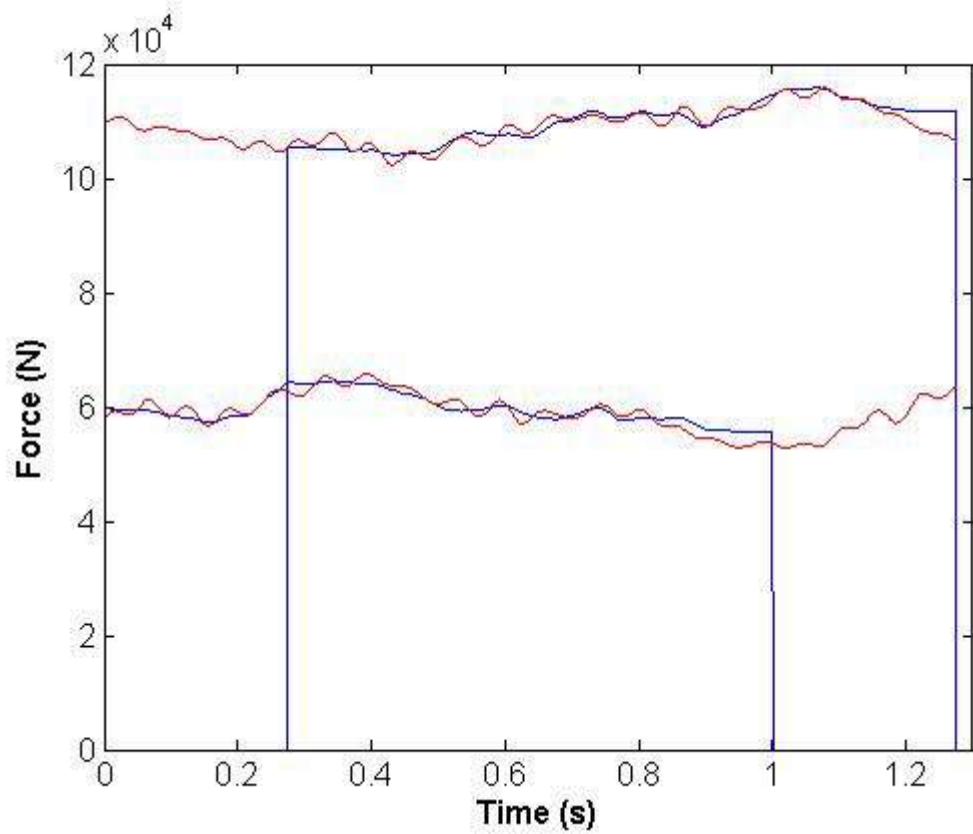
(b) W2(t)

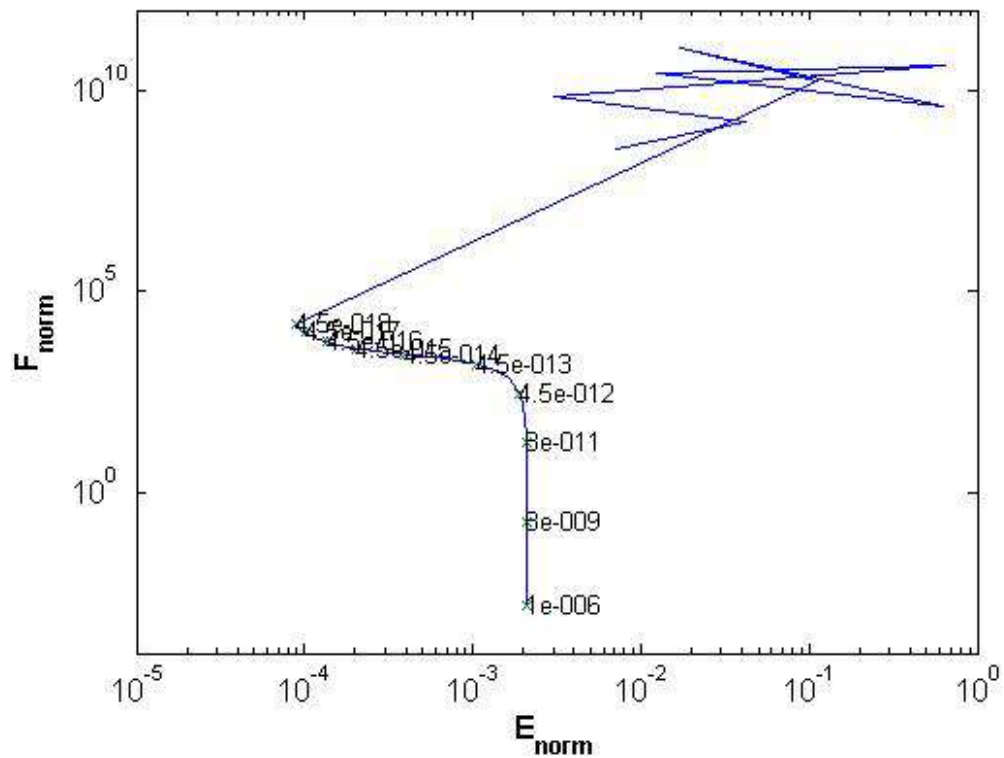


(c) W3(t)

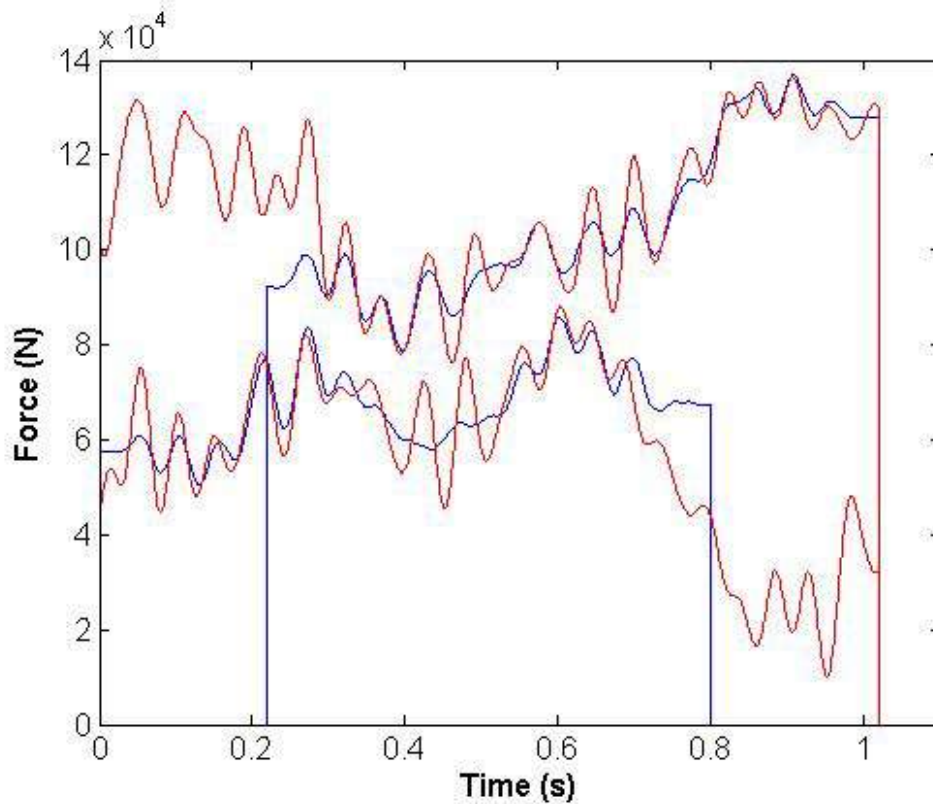


(d) W4(t)

**Figure 7.21** – Predicted wheel forces (– theoretical force, – predicted force)**Figure 7.22-** First order regularisation for the front and rear axles of a vehicle moving at 20m/s on a smooth profile, (– theoretical force, – predicted force)



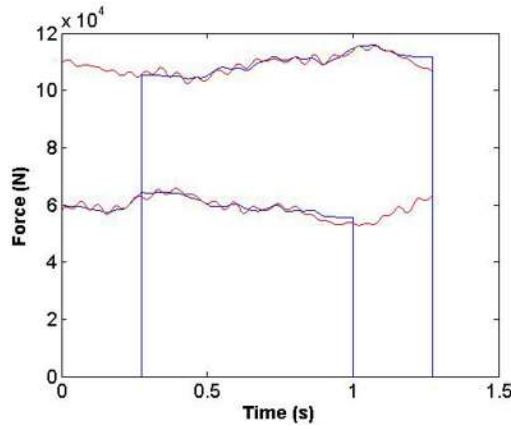
**Figure 7.22** –L-curve for the first order regularisation of a vehicle moving at 25m/s on a rough road profile, 50 modes of vibration used



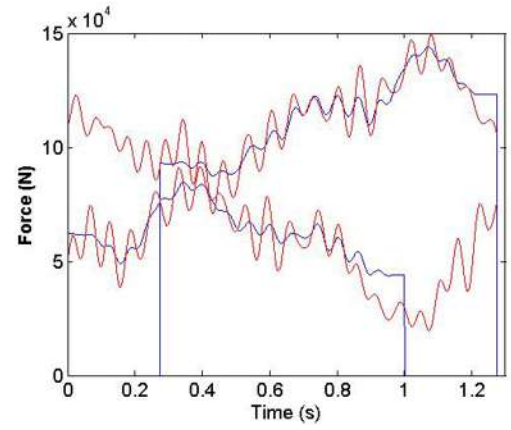
**Figure 7.23** - First order regularisation for the front and rear axles of a vehicle moving at 25m/s on a rough profile, 50 modes of vibration used (— theoretical force, — predicted force)

Previous studies have indicated that in the area of moving force identification, the number of measurements should be greater than or equal to the number of modes used in the inverse analysis (Zhu and Law 2003, Zhu and Law 2001). Therefore initially a high number of “measurement” positions were anticipated for the MFI algorithm. However it can be seen from figures 7.22 and 7.23, that the number of measurement points need not be equal to the number of modes used in the inverse analysis, as in this case 50 modes of vibration have been used and 21 sensor locations. The number of “measurement” positions was then reduced to 8 (2 at  $\frac{1}{4}$  span (sensors 3 & 6), 4 at mid span (sensors 8, 10, 12 & 14) and 2 at  $\frac{3}{4}$  span (sensors 17 & 21)). In total 6 different scenarios were used in the inverse analysis, three speeds (20, 25 and 30m/s) and two types of road profile for each of the speed, smooth and rough with geometric spatial means of  $4 \times 10^{-6} \text{m}^3/\text{cycle}$  and  $32 \times 10^{-6} \text{m}^3/\text{cycle}$  respectively. For all cases the algorithm gives excellent results; the MFI algorithm is able to accurately predict the axle forces for both the smooth and rough profiles. Although in the cases where the road is smooth, the MFI is only able to predict the first main frequency of the truck, in the cases of the road being rough the algorithm is able to catch the two main dominant frequencies of the truck. However it fails in certain instances to predict the amplitude of the axle dynamics. The regularised solutions for all cases, using 8 sensors, are represented in figures 7.24.

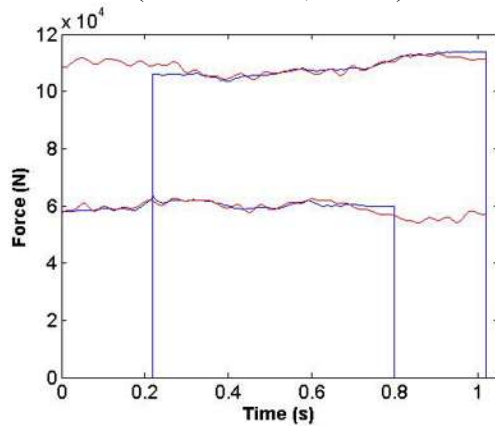




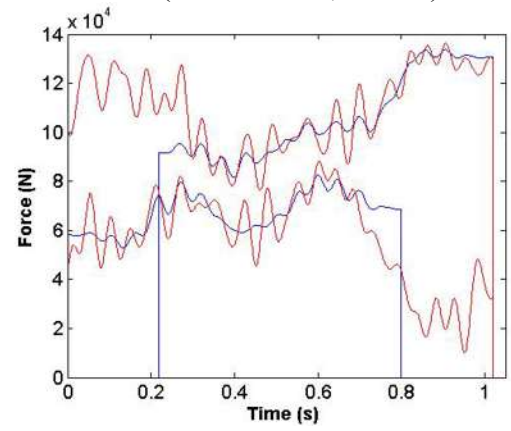
(a) 20m/s, smooth profile,  
( $\lambda = 3 \times 10^{-16}$ ,  $n_z=25$ )



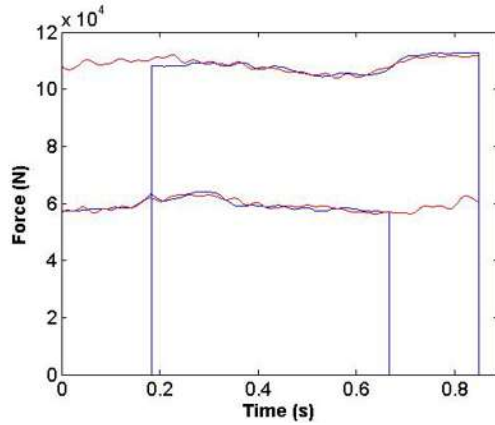
(b) 20m/s, rough profile  
( $\lambda = 9 \times 10^{-17}$ ,  $n_z=25$ )



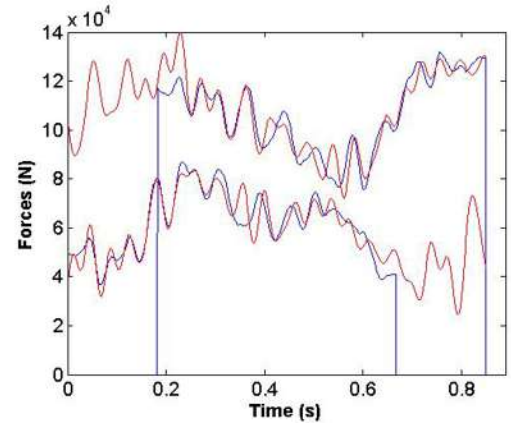
(c) 25m/s, smooth profile,  
( $\lambda = 3 \times 10^{-16}$ ,  $n_z=50$ )



(d) 25m/s, rough profile  
( $\lambda = 5 \times 10^{-17}$ ,  $n_z=50$ )



(e) 30m/s, smooth profile  
( $\lambda = 1 \times 10^{-16}$ ,  $n_z=25$ )



(f) 30m/s rough profile  
( $\lambda = 1 \times 10^{-18}$ ,  $n_z=25$ )

**Figure 7.24** – Regularised solutions for front and rear axles of the vehicle, (– theoretical force, – predicted force) ( $\lambda$  = optimal regularisation parameter,  $n_z$  = numer modes used)

## 7.6 Error Analysis

The model outlined in section 7.2 is considered here again but subject to four moving forces defined by,

$$\begin{aligned} W_1(t) &= 35,000(1 + .1\sin(3\pi t) + .1\sin(5\pi t) + .05\sin(25\pi t) + .05\sin(30\pi t)) \\ W_2(t) &= 35,000(1 + .1\sin(3\pi t) + .1\sin(5\pi t) + .05\sin(25\pi t) - .05\sin(30\pi t)) \\ W_3(t) &= 50,000(1 - .1\sin(3\pi t) + .1\sin(5\pi t) - .05\sin(25\pi t) - .05\sin(30\pi t)) \\ W_4(t) &= 50,000(1 - .1\sin(3\pi t) + .1\sin(5\pi t) - .05\sin(25\pi t) + .05\sin(30\pi t)) \end{aligned} \quad (7.47)$$

The strains at 21 locations on the bridge are simulated using the full stiffness and mass matrices and the method of simulation outlined in chapter 4 is employed. The model is remeshed, using 20 elements along the x-axis and 9 along the y-axis giving a total of 180 elements, 210 nodes and 840 degrees of freedom. The model is remeshed for the purpose of simulation only; the inverse model employed is the same as defined in section 7.3.4. Noise is added to the simulated strain in all cases; the noise added to all “measurements”, unless otherwise specified, it is 2% of the maximum strain of sensor 11. Several parameters are analysed for their effect on the accuracy of the MFI algorithm, as outlined in the following subsections.

### 7.6.1 Effect of Modal Truncation

In all of the cases analysed, the model is simulated using the full stiffness and mass matrices. However in the inverse problem the dynamic model is reduced to a finite number of modes. It is thus necessary to assess the number of modes required for an acceptable solution to be obtained. The effect of modal truncation is analysed for the forcing functions defined in equation (7.47), traversing the bridge at a constant velocity of 20m/s; the axle spacing is 5m and the transverse distance between forces is 2m. The vehicle is travelling .5m from the centreline of the bridge, the sampling time is .0005s and 2% noise is added to all sensors as a product of the maximum strain induced in sensor 11. The number of modes is increased from 10 to 45 in increments of 5, and for each analysis, 132 regularisation parameters are used in the L-curve analysis. The errors in both the predicted wheel forces and the axle forces are calculated using equation

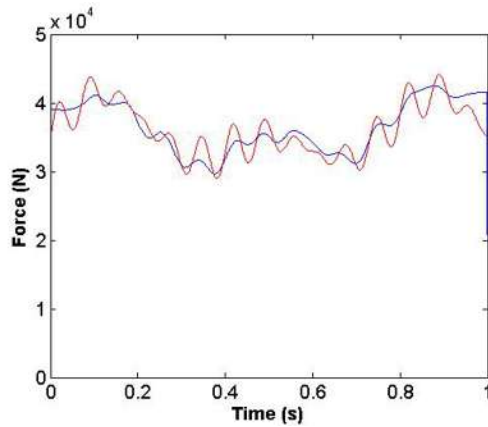
(6.31). The errors in the predicted forces versus the number of modes used in the inverse analysis are given in tables 7.2 and 7.3 and represented graphically for the case of 40 modes in figure 7.24. It can be seen from the, that once the number of modes is greater than or equal to 30, the percentage errors in the individual wheel loads is approximately 6% and the percentage errors in the axle loads is approximately 5%. Once the number of modes is increased to 40, the percentage error in individual wheel loads is approximately 5.5% and 4% in the axle loads. The predicted wheel loads and the predicted axle loads versus the theoretical applied loads with their respective errors are represented in figures.

Number of Modes	W1 % Error	W2 % Error	W3 % Error	W4 % Error	Axle 1 % Error	Axle 2 % Error
10	12.7	11.1	13.6	9.9	10.8	10.6
15	15.5	11.0	10.5	10.5	12.3	11.2
20	6.5	5.9	6.4	6.3	5.2	5.3
25	9.7	9.7	8.8	8.8	4.9	4.5
30	5.6	6.4	6.2	6.0	4.8	4.55
35	5.6	6.3	6.1	5.8	4.4	4.1
40	5.5	5.5	5.6	5.5	4.5	4.2
45	5.3	5.7	5.7	5.7	4.4	4.1

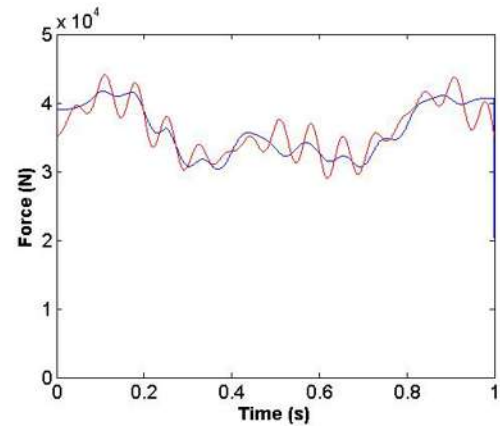
**Table 7.2** – Percentage errors in identified wheel and axle forces using 21 “measurements” in the inverse analysis.

Number of Modes	W1 % Error	W2 % Error	W3 % Error	W4 % Error	Axle 1 % Error	Axle 2 % Error
10	13.6	9.2	9.6	8.3	9.2	8.6
15	16.4	8.4	8.5	9.6	10.6	9.4
20	8.8	6.4	6.9	6.4	5.0	5.3
25	6.1	6.0	6.2	6.1	4.9	4.6
30	5.9	6.60	6.5	6.2	4.8	4.5
35	6.6	7.1	6.8	6.4	4.1	3.9
40	4.9	6.3	6.1	5.8	4.2	4.0
45	5.6	6.0	6.0	5.8	4.9	4.6

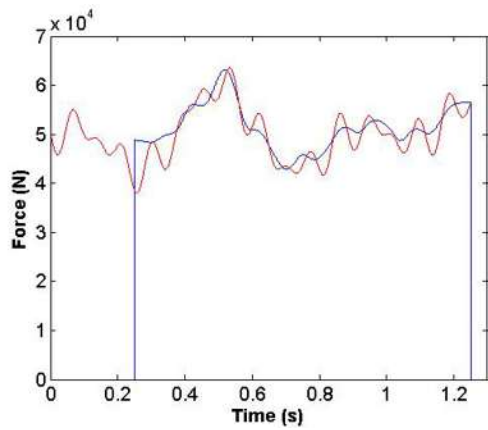
**Table 7.3** – Percentage errors in identified wheel and axle forces using 8 “measurements” in the inverse analysis.



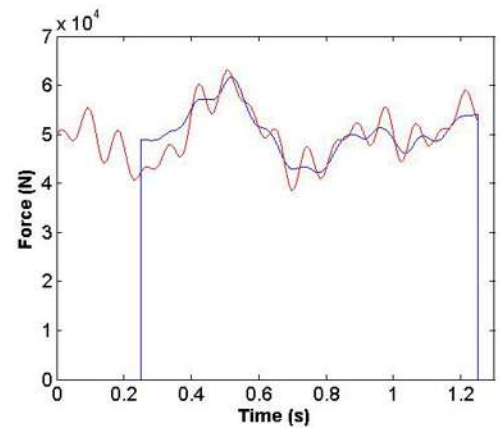
(a) W1(t), 5.5% error



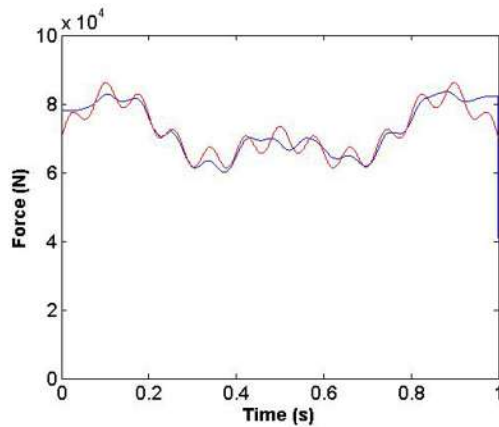
(b) W2(t), 5.5% error



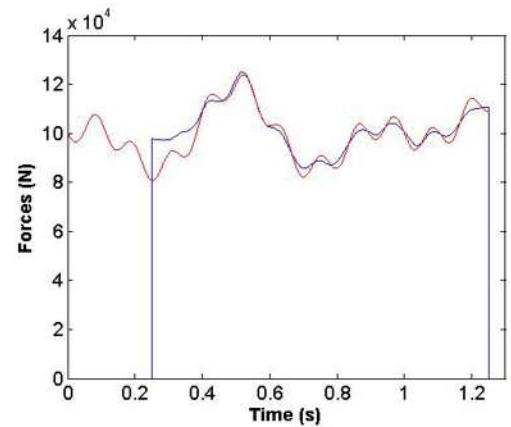
(c) W3(t), 5.6% error



(d) W4(t), 5.5% error



(e) Axle 1, 4.5% error



(f) Axle 2, 4.2% error

**Figure 7.25** – Predicted versus theoretical applied forces, using 21 “measurements” and 40 modes of vibration, (– theoretical force, – predicted force).

### 7.6.2 Effect of Noise

The effect of noise on the accuracy of the algorithm is analysed for the scenario detailed in sections 7.6 and 7.6.1. The number of modes used in all cases is 40, as the minimum error achieved in the modal truncation analysis was 40. The noise is added to the simulated strain at each “measurement” location as a product of the maximum strain of sensor 11. The noise is added in this manner as it is assumed that the nature of the noise measured in the field would be of a similar magnitude across all sensors. The impact of the noise on a particular signal can be defined by the signal to noise ratio (SNR),

$$SNR = \frac{\sigma_s}{\sigma_n} \quad (7.48)$$

where  $\sigma_s$  and  $\sigma_n$  are the variances of the true signal and the noise, and the relative signal to noise ratio is defined by,

$$RSNR = \frac{1}{SNR} \times 100 \quad (7.49)$$

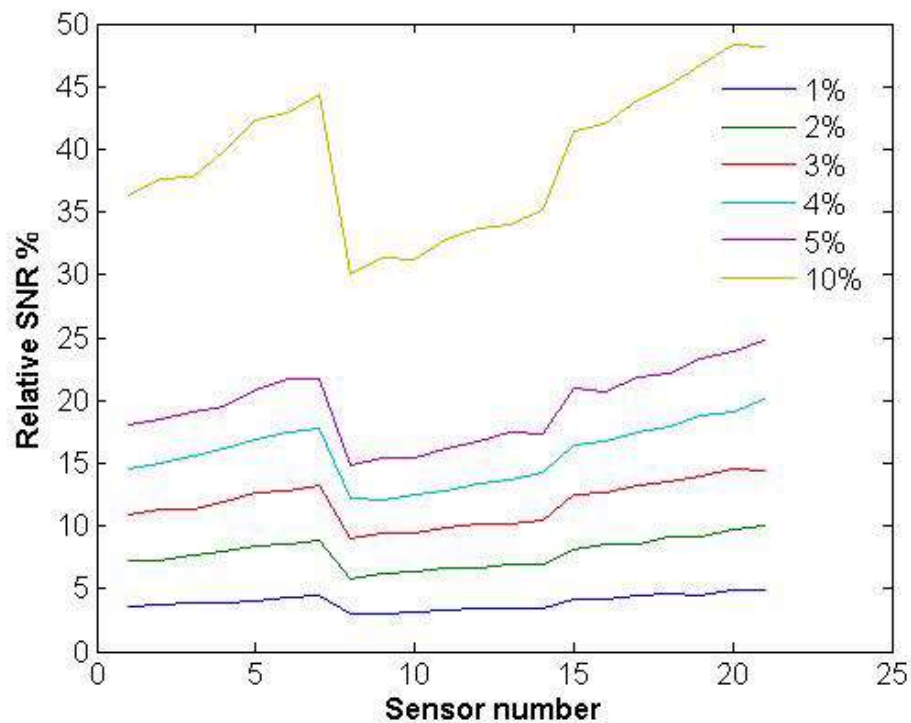
In total six levels of noise are analysed for their effect on the accuracy of the MFI algorithm. Figure 7.26 shows the relative signal to noise ratios for all 21 sensors, see table 7.1, for the six levels of noise used. The error in identified forces are calculated as defined in equation (6.31), and are shown in tables 7.4 and 7.5 and represented graphically in figures 7.27. It can be seen that, in general, the errors in the identified forces increase in proportion to the percentage of noise. However it should be noted that even in the case of 10% noise, where the RSNR is upwards of 30%, the percentage error in the identified axle forces is only 8.1% and 8.3% for axles 1 and 2, using 21 measurement locations and 9.3% and 8.7% using 8 measurement locations.

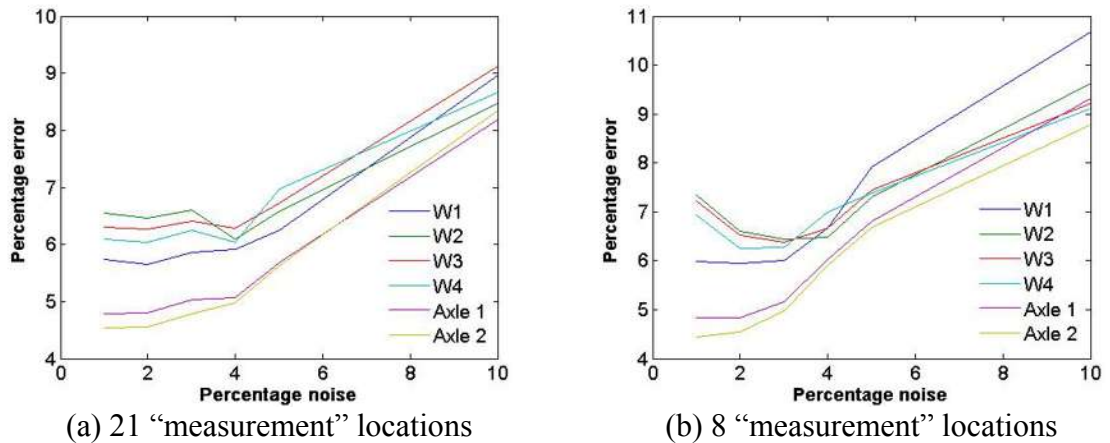
Percentage Noise	W1 % Error	W2 % Error	W3 % Error	W4 % Error	Axle 1 % Error	Axle 2 % Error
1	5.7	6.5	6.3	6.0	4.7	4.5
2	5.6	6.4	6.2	6.0	4.8	4.5
3	5.8	6.5	6.4	6.2	5.0	4.7
4	5.9	6.0	6.2	6.0	5.0	4.9
5	6.2	6.5	6.7	6.9	5.6	5.6
10	8.9	8.4	9.1	8.6	8.1	8.3

**Table 7.4** – Percentage error in the predicted forces for different noise levels, 21 measurements used in the inverse analysis

Percentage Noise	W1 % Error	W2 % Error	W3 % Error	W4 % Error	Axle 1 % Error	Axle 2 % Error
1	5.9	7.3	7.2	6.9	4.8	4.4
2	5.9	6.6	6.5	6.2	4.8	4.5
3	6.0	6.4	6.3	6.2	5.1	4.9
4	6.6	6.4	6.6	7.0	6.0	5.9
5	7.9	7.3	7.4	7.3	6.8	6.6
10	10.6	9.6	9.2	9.1	9.3	8.7

**Table 7.5** - Percentage error in the predicted forces for different noise levels, 8 measurements used in the inverse analysis



**Figure 7.26** – Relative signal to noise ratio for the 6 levels of noise used**Figure 7.27** – Error in the identified forces versus the percentage of noise added

### 7.6.3 Effect of Vehicle Velocity

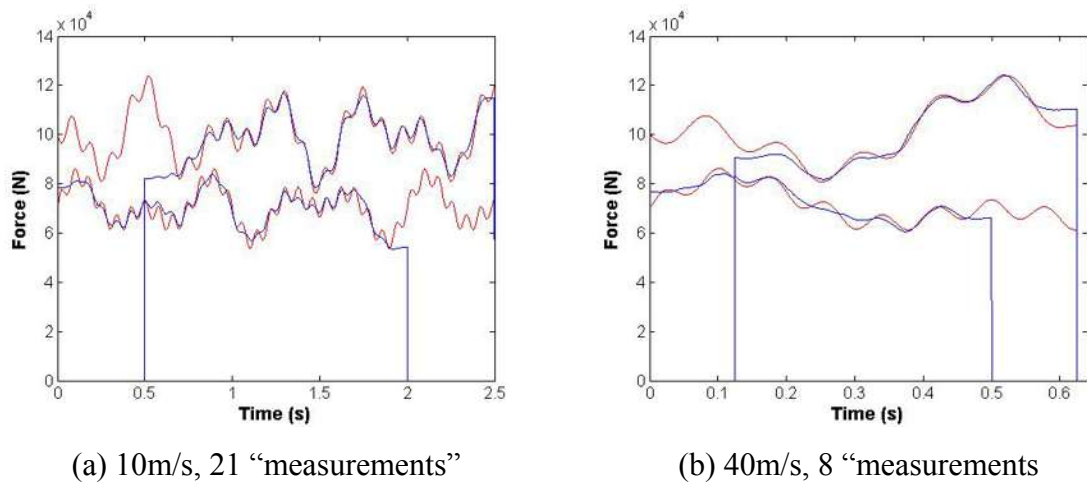
The effect of vehicle velocity on the accuracy of the moving force identification algorithm is analysed for the scenario detailed in sections 7.6 and 7.6.1. The velocity is varied from 10m/s to 40m/s in increments of 5. The errors in identified forces with respect to vehicle velocity are shown in tables 7.6 and 7.7. In general it was found that the MFI algorithm is predominately independent of the velocity with maximum and minimum errors of approximately 8 and 4% respectively. Figure 7.28 shows the predicted versus the theoretical forces for the set of forces travelling at 10 and 40m/s respectively.

Velocity m/s	W1 % Error	W2 % Error	W3 % Error	W4 % Error	Axle 1 % Error	Axle 2 % Error
10	5.7	6.0	6.0	6.3	5.0	4.8
15	7.4	6.5	7.3	7.5	6.3	6.6
20	5.6	6.3	6.1	5.8	4.4	4.1
25	6.8	6.3	6.3	6.0	5.5	5.0
30	3.7	4.9	5.6	5.9	3.5	4.1
35	4.1	4.9	6.0	6.6	4.2	5.0
40	6.1	5.4	6.0	5.4	4.9	4.9

**Table 7.6** – Errors in the identified forces versus vehicle velocity using 21 “measurement” locations

Velocity m/s	W1 % Error	W2 % Error	W3 % Error	W4 % Error	Axle 1 % Error	Axle 2 % Error
10	6.5	8.3	7.9	8.3	6.6	6.4
15	6.1	6.9	7.9	8.6	6.2	6.6
20	6.6	7.1	6.8	6.4	4.1	3.9
25	6.2	6.3	6.8	7.0	5.7	4.8
30	4.3	5.7	6.1	6.4	4.1	4.4
35	4.6	6.5	7.8	7.2	4.9	5.8
40	6.1	5.9	7.0	5.7	4.8	4.8

**Table 7.7** – Errors in the identified forces versus vehicle velocity using 8 “measurement” locations



**Figure 7.28** –Identified and theoretical forces for different velocities,  
(— theoretical force, — predicted force).

#### 7.6.4 Effect of Axle Spacing

The effect of the axle spacing on the accuracy of the moving force identification algorithm is analysed for the scenario described in sections 7.6 and 7.6.1 however in this case the vehicle is travelling at 30m/s and 40 modes of vibration are used in all cases. The axle spacing is varied from 1m to 10m in increments of 1m. Tables 7.8 and 7.9 show the percentage errors in the predicted axle forces for a range of distances between the forces. It can be seen from these tables that once the axle spacing is greater than 3m, the error in the predicated forces has stabilised to approximately 4%. Notwithstanding this, the errors for axle spacings of 3m or less are only 6 or 7%. Figure



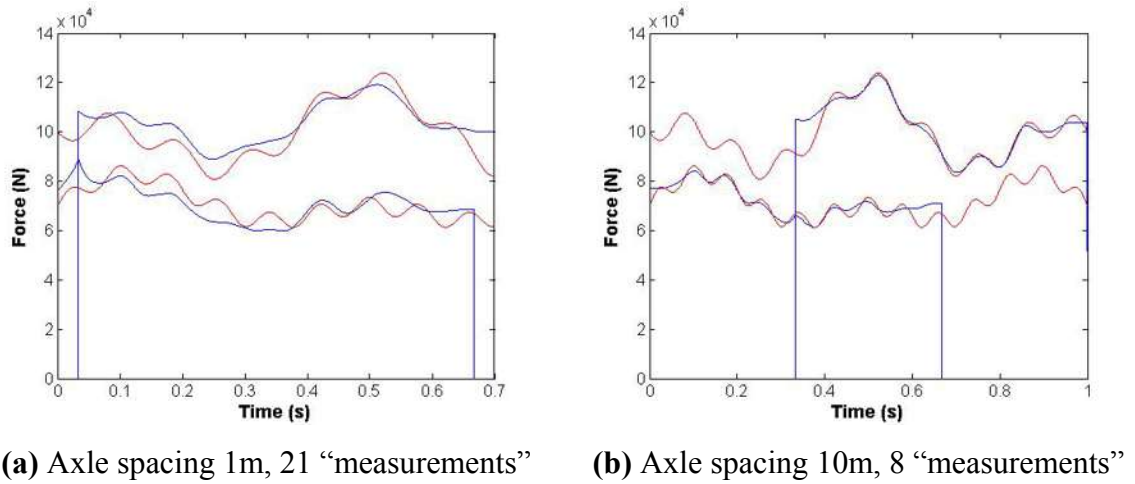
7.29 show the predicted versus the theoretical forces for the vehicle travelling at 30m/s with axle spacings of 1 and 10m respectively. Figure 7.29 shows that for an axle spacing of 10m, there is an excellent match between the theoretical and identified forces, however for the case of an axle spacing of 1m, the MFI algorithm appears to be unable to accurately distinguish between the force and hence the first axle is underestimated and the second axle is overestimated.

Axle Space (m)	W1 % Error	W2 % Error	W3 % Error	W4 % Error	Axle 1 % Error	Axle 2 % Error
1	8.0	8.3	9.9	9.6	7.4	8.7
2	5.9	7.6	8.3	8.9	4.8	5.9
3	7.2	6.8	7.3	7.5	5.9	6.1
4	4.6	5.1	5.5	6.1	4.0	4.3
5	3.7	4.9	5.6	5.9	3.5	4.1
6	4.9	5.3	6.5	6.2	4.4	5.2
7	4.4	5.1	5.9	5.6	3.8	4.2
8	3.8	4.5	5.4	5.3	3.3	3.9
9	3.9	5.0	5.7	5.4	3.5	3.9
10	4.2	5.5	5.5	5.6	3.9	3.8

**Table 7.8** – Percentage error in identified forces versus axle spacing using 21 “measurements”

Axle Space (m)	W1 % Error	W2 % Error	W3 % Error	W4 % Error	Axle 1 % Error	Axle 2 % Error
1	7.2	7.9	9.9	9.3	7.2	8.7
2	5.6	8.4	9.5	9.0	4.8	6.1
3	5.3	6.9	7.5	7.8	4.5	5.0
4	5.4	7.0	7.0	7.4	4.6	4.8
5	4.3	5.7	6.1	6.4	4.1	4.4
6	5.1	5.6	6.9	6.8	4.9	5.6
7	4.6	5.8	6.3	6.1	4.1	4.4
8	3.5	4.7	5.6	5.6	3.4	4.0
9	3.4	4.6	5.5	5.3	3.3	3.8
10	4.3	5.9	5.8	5.9	4.3	4.0

**Table 7.9** – Percentage error in identified forces versus axle spacing using 8 measurements



**Figure 7.29** - Effect of axle spacing on the accuracy of the identified forces, (– theoretical force, – predicted force)

### 7.6.5 Effect of the Transverse Location of the Forces

The effect of the transverse location of the forces on the accuracy of the MFI algorithm is analysed for the scenario of the forces moving at 20m/s with and axle spacing of 5m and transverse distance between the forces of 2m. The transverse location is varied from a point where W1 starts at (0,0), W1 is then shifted transversely in increments of 1m to 4m after this point it is assumed that the response of the bridge is symmetrical and further iteration is not necessary. Tables 7.10 and 7.11 show the errors in the identified forces versus the transverse position of the forces for 21 and 8 “measurement” locations. In all scenarios where 21 measurements are used the % errors in identified forces are approximately equal, however in the case of 8 sensors when the transverse location of W1 is located at (0,0) there is a sharp increase in the errors in the identified forces. It is thought that this is due to the number of sensors crossed by the moving forces, in all cases where 21 sensors are used, the forces cross at least 3 sensors, but in the case described above only one sensor is crossed, this is sensor 8 at midspan. Analysis was then carried out for the scenario of W1 starting at 2m, and only 3 sensors being used in the inverse analysis. The sensors were chosen such that the distance between the sensors and the moving forces is minimised. The sensors chosen are 4, 10 and 18 located at (4.25,3.25), (10.25,3.75) and (15.25,3.75). The forces are moving at 10m/s and the sampling time is .001s, again the axle spacing is 5m and the transverse distance between the forces is 2m.

Transverse Position W1	W1 % Error	W2 % Error	W3 % Error	W4 % Error	Axle 1 % Error	Axle 2 % Error
0	5.4	6.0	6.2	5.8	4.5	4.4
1	5.4	6.6	6.3	6.1	4.3	4.1
2	5.6	6.3	6.1	5.8	4.4	4.1
3	5.8	5.5	5.6	5.6	4.7	4.4
4	5.8	5.5	5.8	5.6	4.7	4.5

**Table 7.10** – Percentage errors in the identified forces versus the transverse position of the truck, 21 “measurements” used in the inverse analysis

Transverse Position W1	W1 % Error	W2 % Error	W3 % Error	W4 % Error	Axle 1 % Error	Axle 2 % Error
0	11.6	13.1	11.3	11.2	4.7	4.5
1	6.8	7.8	7.3	7.2	5.0	4.8
2	6.6	7.1	6.8	6.4	4.1	3.9
3	6.2	6.4	6.4	6.7	5.3	5.2
4	5.6	5.3	5.8	5.5	4.7	4.3

**Table 7.11** – Percentage errors in the identified forces versus the transverse position of the truck, 8 “measurements” used in the inverse analysis

Initially for the inverse analysis of the scenario described above 30 modes of vibration were used, however it was found that in using only three sensors 30 modes was not a sufficient number to achieve acceptable results. The number of modes was increased in increments of five to 60; table 7.12 shows the percentage errors in the identified forces versus the number of modes used. It can be seen from this table that once the number of modes is greater than or equal to the 50 the percentage errors in the front and back axles have stabilised to approximately 7 and 5% respectively, it should also be noted that acceptable errors could not be achieved in the individual wheel forces see table 7.10. The predicted wheel loads, and the predicted axle loads using 50 modes of vibration and 3 sensors are shown in figures 7.30 (a) to (d) and figure 7.31.

Number Modes	W1 % Error	W2 % Error	W3 % Error	W4 % Error	Axle 1 % Error	Axle 2 % Error
35	134.8	133.9	114.4	116.0	23.7	9.2
40	93.4	92.3	78.7	79.6	16.0	7.2
45	85.2	86.3	73.5	74.4	14.9	6.4
50	55.9	57.9	48.2	48.2	7.2	5.2
55	55.8	57.3	47.3	47.5	7.1	5.4

60	53.8	55.8	46.4	46.6	7.7	5.4
----	------	------	------	------	-----	-----

Table 7.12 – Percentage errors in the identified forces using 3 sensors

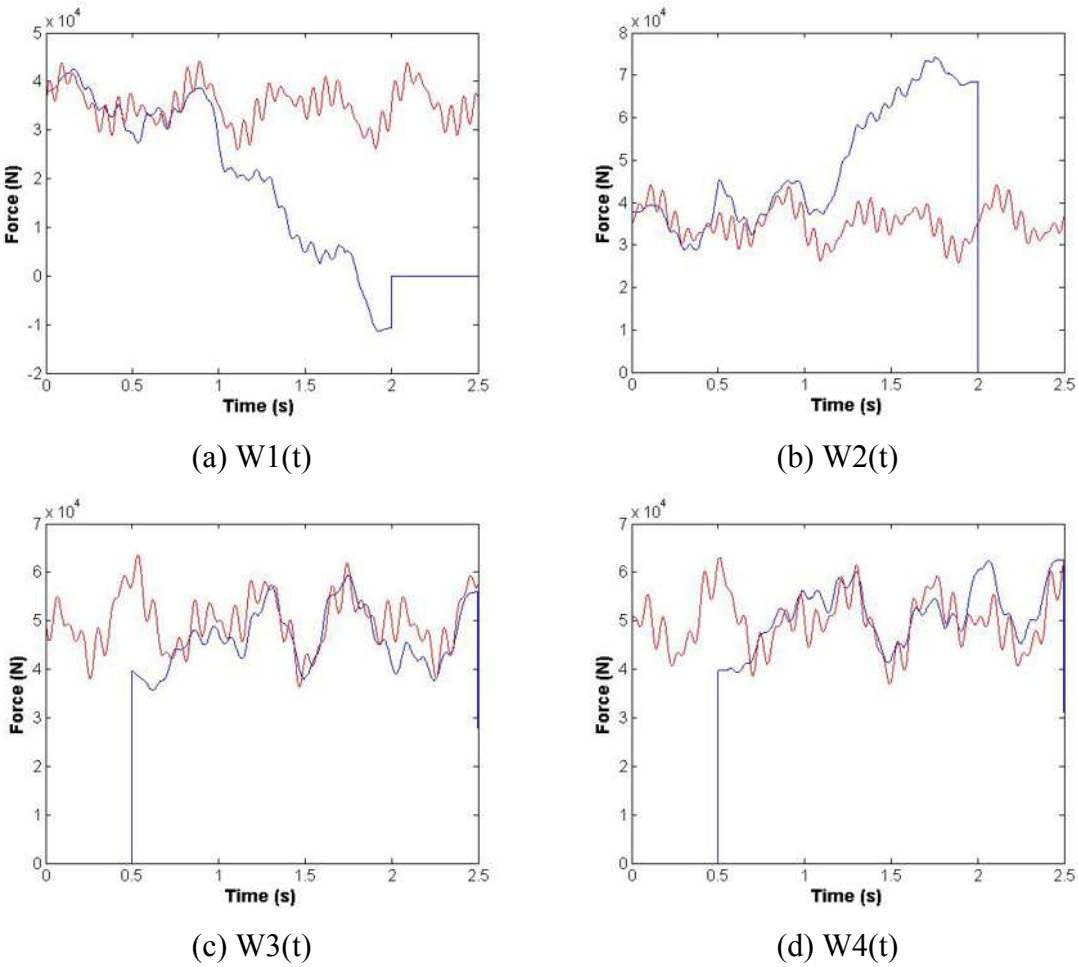
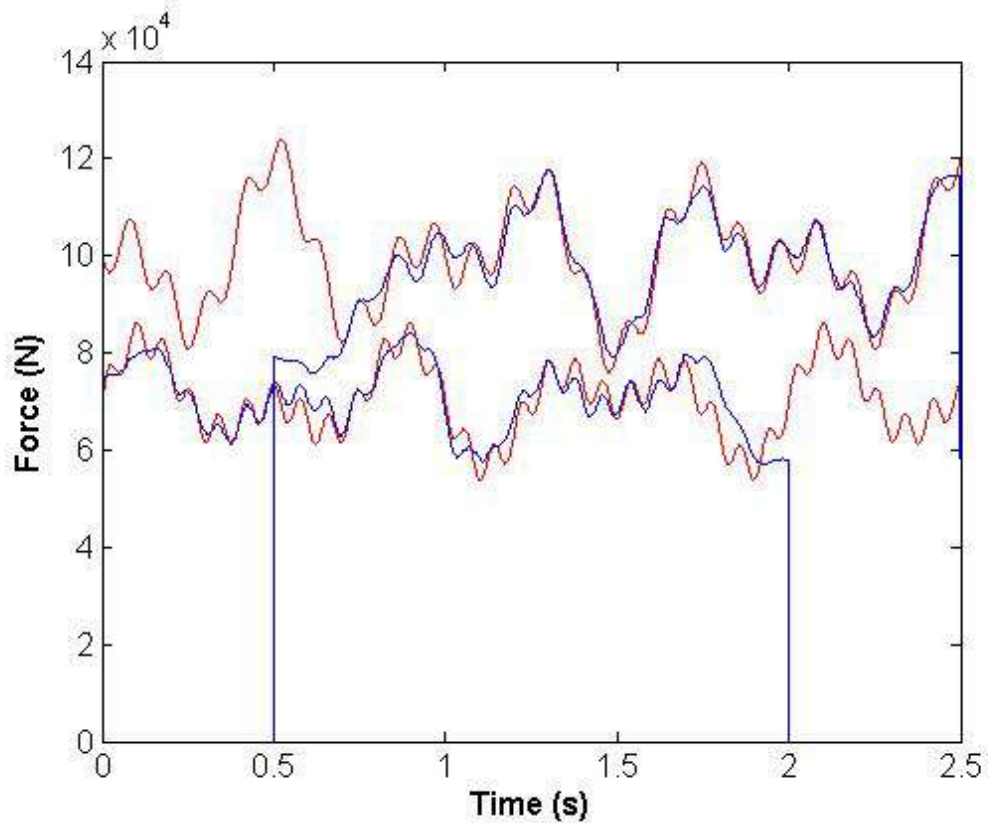


Figure 7.30 – Identified wheel loads using 3 sensors and 50 modes of vibration, (— theoretical force, — predicted force)



**Figure 7.31** – Identified axle loads using 3 sensors and 50 modes of vibration, (– theoretical force, – predicted force)

### 7.6.6 Effect of Sampling Frequency

The effect of the sampling frequency on the accuracy of the moving force identification algorithm is analysed for the scenario of a vehicle moving at 20m/s with a constant axle spacing of 5m subject to the forces defined in equation 6.62. The sampling frequency is varied from 500Hz to 2500Hz in increments of 500, 30 modes of vibration are used in all inverse analysis. Both 8 and 21 ‘measurement’ locations have been used in all inverse analysis, and 2% Gaussian noise is added to the theoretical strain signals as a product of the max strain induced. Tables 7.13 and 7.14 show the percentage errors in identified forces with respect to the sampling frequency, tables 7.13 and 7.14 show that once the sampling frequency has reached 1000Hz the percentage errors have to stabilised to approximately 5% in the front and back axles however it should be noted

that with a sampling frequency of 500Hz acceptable results are still achieved in the identified force histories.

Sampling Frequency (Hz)	$\lambda$	W1 % Error	W2 % Error	W3 % Error	W4 % Error	Axle 1 % Error	Axle 2 % Error
500	$1 \times 10^{-17}$	6.9	7.2	8.3	7.6	6.8	6.8
1000	$6 \times 10^{-17}$	6.4	6.3	6.7	6.5	5.7	5.7
1500	$5 \times 10^{-17}$	5.2	5.8	6.3	6.1	4.4	4.8
2000	$6 \times 10^{-17}$	5.6	6.4	6.2	6.0	4.8	4.5
2500	$8 \times 10^{-17}$	5.8	6.8	6.4	6.2	4.6	4.4

**Table 7.13** - Percentage errors in identified axle forces versus the sampling frequency using 21 ‘measurement’ locations

Sampling Frequency (Hz)	$\lambda$	W1 % Error	W2 % Error	W3 % Error	W4 % Error	Axle 1 % Error	Axle 2 % Error
500	$4 \times 10^{-18}$	7.5	6.9	8.4	7.7	6.9	6.8
1000	$1 \times 10^{-17}$	6.6	6.8	6.9	6.6	5.8	5.6
1500	$2 \times 10^{-17}$	5.5	6.1	6.6	6.4	4.5	4.8
2000	$3 \times 10^{-17}$	5.9	6.6	6.5	6.2	4.8	4.5
2500	$8 \times 10^{-17}$	6.2	6.9	6.9	6.9	5.1	4.7

**Table 7.14** - Percentage errors in identified axle forces versus the sampling frequency using 8 ‘measurement’ locations

## 7.7 Conclusions

The method of moving force identification outlined in chapters 4 and 5 has been extended to 2 dimensions to allow for both the transverse behaviour of the bridge and the transverse location of the truck. A  $C_1$  orthotropic plate element has been programmed in Matlab to model the bridge; the model is validated using the commercial FE package MSC/Nastran. The moving force identification algorithm of chapters 4 and 5 is improved to reduce the errors in the identified forces at the start and end of the time histories. The extended 2d algorithm has been rigorously tested using the simulated strain from a 3D VBI finite element model. An error analysis has been carried out to assess the effects of various properties on the accuracy of the algorithm. Finally it has been shown that the number of sensors need not be very large, in fact three sensors is adequate to predict the individual axle loads, provided the sensors are in close proximity to the vehicle path. However using only three sensors is not acceptable for the prediction of individual wheel loads, and the optimum number and location of sensors for the prediction of wheel loads requires further research.