World Applied Sciences Journal 3 (4): 674-683, 2008
ISSN 1818-4952

A New Method in Determining Random Wave-Induced Inline Force

[1]*Mohammad Ali Lotfollahi-Yaghin* and [2]*Bizhan Sanaaty*

[1]Faculty of Civil Engineering, University of Tabriz, Tabriz, Iran
[2]Department of Civil Engineering, Islamic Azad University – Boukan Branch, Iran

Abstract: Random waves at sea exert great forces on offshore structures. Jacket platform is a special type of offshore structures which is chiefly composed of tubular elements. Considering the several applications of these structures in marine industry, they should be designed under large forces caused by random waves. The forces on these elements, due to waves, can mainly be divided into two components namely inline force and transverse or lift force. Since inline force is usually greater and less complicated than lift force, it is mostly considered in analysis and design procedures. In this case, Morison equation is usually used as a computational method which needs two different coefficients, named drag and inertia, to determine the inline force. Choosing an appropriate method for determining these coefficients for a specified data is hard and their computing is a time-consuming affair. Usually, using these methods do not lead to suitable results because they do not involve all real environmental conditions. In this paper, to be free of determining the coefficients and other computations, we propose neural network method as a very good substitute for Morison equation as a simpler, easier and faster solution.

Key words: Neural network . random waves . Morison equation . hydrodynamics . inline force

INTRODUCTION

Offshore platforms have many applications, such as oil exploitation, military purposes and research stations and so on. An offshore platform, as well as dead and service loads, is under other loads that vary with time. These loads are mostly due to waves and currents.

The computation of the water wave forces on an offshore structure is one of the primary tasks in the design of the structure. It is also one of the most difficult tasks since it involves the complexity of the interaction of waves with the structure. Moreover, the random nature of sea waves and the inadequacy of even some of the highly nonlinear wave theories to describe it, its effect on the offshore structure is obviously even more difficult. Nonetheless, some of the theories available today coupled with our understanding of the interaction phenomenon through analytical studies, laboratory experiments and at sea measurements are reasonably accurate in predicting wave loads on a variety of offshore structures but their computing is a time-consuming affair [1].

The nature of wave forces on offshore structures, as the same as random sea, is random. It is necessary to say that it will be a more sophisticated trend when we deal with large forces caused by oscillatory loads resulted from random waves than regular waves.

Since tubular elements are usual elements in constructing offshore structure, they are mostly used in research approaches. The forces on these elements, due to sea waves, can mainly be divided into two components, namely, inline force and transverse or lift force. The inline force acts along with wave propagation direction but lift force acts in transverse direction. Random nature of inline force is like waves and nearly resembles sea surface oscillations. Hence, it has a linear relationship with waves whereas the nature of lift force is different from sea surface oscillations and has a nonlinear relationship with waves (Fig. 1).

Wave forces on offshore structures are calculated in three different ways: Morison equation, Froude-Krylov theory and diffraction theory. In order to determine where these three methods are applicable a simple dimensional analysis is performed first [1].

The Morison equation assumes the force to be composed of inertia and drag forces linearly added together. The components involve an inertia (or mass) coefficient and a drag coefficient which should be determined experimentally for different environmental conditions. Many researches have been done and good predictions resulted in determining these coefficients but still errors are considerable. Usual engineering practice is to assume them constant. This equation is applicable when the drag force is significant. This is usually the case when a structure is small compared to the water wavelength. Since the experimental data been

Corresponding Author: Bizhan Sanaaty, Department of Civil Engineering, Islamic Azad University – Boukan Branch, Iran

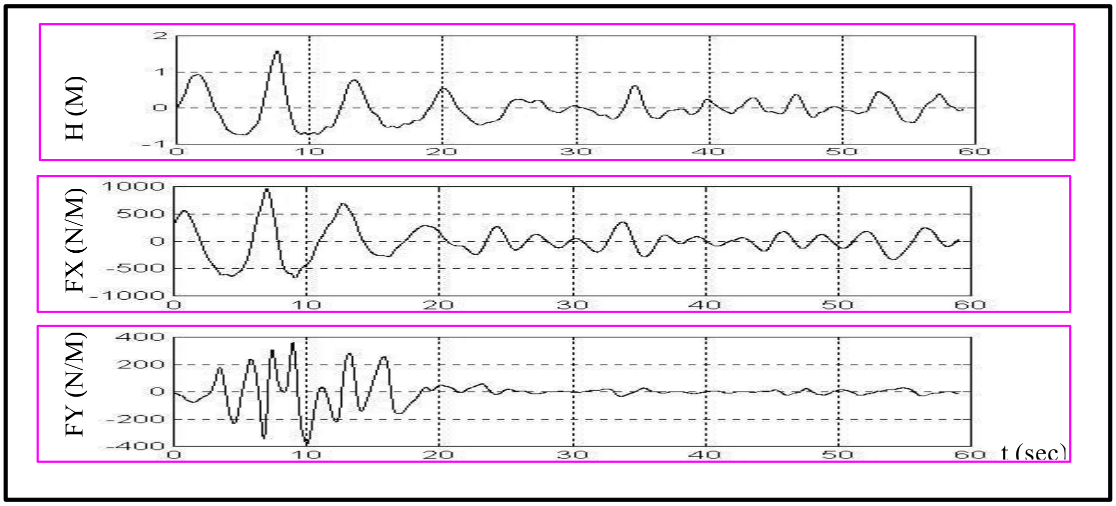

Fig. 1: Wave surface profile, inline and transverse forces at a period of time

used in this research, has the case which Morison equation applies for, this method is used among the other methods.

The main objective of this paper is the evaluation of neural network ability in predicting the inline force in comparison with Morison equation as a conventional method.

This paper, firstly contains a review on Morison equation and the methods of determining its coefficients then neural network method and our proposed networks used in this research for inline force prediction are introduced.

EXPERIMENTS

The data used in this research are from the experiments done in delta wave flume of Delft Hydraulics Laboratory (DHL) in the Netherlands. The flume is approximately 250m long, 5m wide and is normally filled with water to a depth of 5m. The waves are generated by a hydraulically driven plane waveboard and their energy is dissipated at the other end of the flume through the use of a sloping concrete beach. The facility is capable of generating both regular and random wave trains. Various random wave spectra can be generated with the *JONSWAP* spectrum being most commonly used. The random waves used in these tests were based on the *JONSWAP* spectrum of H=1.5m significant wave height and T = 5.9 s mean period.

All physical positions of the cylinders and measuring instruments are referred to a standard Cartesian axis system in the flume, where the origin is at the mean water level on the centre line of the flume and the positive x direction is from the wave generator paddle and towards the beach (Fig. 2a and 2b).

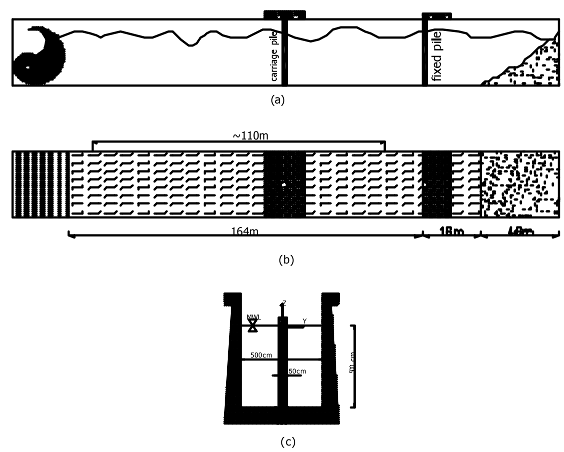

Fig. 2: Delta wave flume

Figure 2c shows the general elevation of cylinders when mounted in the flume. The large fixed cylinder used in this experiments has two force measuring sleeves at elevations of 1.5m and 2.5m below the mean water level. The length of the sleeves is 0.5 diameters (0.5D) and both are capable of measuring both the in-line (x) and transverse (y) forces. Wave height gauges were positioned at the same downstream position as the cylinder. The flow velocity was always measured at the same elevation and downstream position as the force sleeves [6].

The data of the fixed cylinder with 0.5m diameter and the roughness of k/d= 0.038 at the elevation of 1.5m below the mean water level, under the effect of random waves, were used in this research.

These data has been contaminated with unwanted noises during the experiments. The Matlab program command "*filtfilt*" was used in order to remove these noises. Sampling frequency of 40 HZ has been used in these experiments.

MORISON EQUATION

The Morison equation was developed in describing the horizontal wave forces acting on a vertical pile which extends from the bottom through the free surface. The equation says that the force exerted by unbroken surface waves on a vertical cylindrical pile is linearly composed of two terms, inertia and drag. The principle involved in the concept of the inertia force is that a water particle moving in a wave carries a momentum with itself. As the water particle passes around the circular cylinder it accelerates and then decelerates. This requires that work be done through the application of a force on the cylinder to increase this momentum. The incremental force on a small segment of the cylinder needed to accomplish this is proportional to the water particle acceleration at the center of the cylinder.

The principle cause of the drag force term is the presence of a wake region on the downstream side of the cylinder. The wake is a region of low pressure compared to the pressure on the upstream side and thus a pressure differential is created by the wake between the upstream and downstream of the cylinder at a given instant of time. The pressure differential causes a force to be exerted in the direction of the instantaneous water particle velocity. In an oscillatory flow, the absolute value of the water particle velocity is inserted to insure that the drag force is in the same direction as the velocity.

Combining the inertia and drag terms of force, the Morison equation is written as:

$$\begin{aligned} df &= df_{inertia} + df_{drag} \\ &= C_M . A_I .(\partial u / \partial t).ds + C_D . A_D |u|.u.ds \\ A_I &= \rho.\pi.D^2/4 \qquad A_D = 1/2.\rho.D \end{aligned} \tag{1}$$

In which df = inline force on the segment ds of the vertical cylinder, D = cylinder diameter, $\partial u/\partial t$ = local water particle acceleration at the centerline of the cylinder, u = instantaneous water particle velocity, ρ = mass density of water, C_M = inertia coefficient and C_D = drag coefficient.

Since the drag term is proportional to the square of water particle velocity, it is non-linear. The inertia term is linear if the partial derivative of sinusoidal water particle velocity (linear theory) is used for the acceleration. On the other hand, the inertia term is non-linear if the local horizontal acceleration including convective terms.

$$\dot{u} = \frac{du}{dt} = \frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + w\frac{\partial u}{\partial z} \tag{2}$$

In which u, v *and* w are the components of the water particle velocity vector in a rectangular Cartesian coordinate system [1].

Least Mean Square method (LSM): For considering environmental conditions in Morison equation, two inertia and drag coefficients are used which their determinations needs having experimental data. In the data we used here, horizontal and vertical velocities, wave profile and exerted forces on the cylinder are existed. There are many methods for determining these coefficients like Bearman (BM), Klopman (KM), least squared method (LSM), Fourier averaging method (FAM) and least square fit of the force spectrum (LS-FS) [7]

Since, in this research for existent data, it was found that LSM method compared to the other mentioned methods gives less error in determining coefficients, it will be explained here.

In this method the coefficients are computed in a way which the sum square error of measured and computed forces becomes minimum. If f is the measured data and f_p the calculated data, the difference between these two forces for each sampling point is:

$$e_f = f - f_p = f - C_D.A_D.u|u| - C_M.A_I.\dot{u} \tag{3}$$

The sum of squares will become:

$$L_s = \sum_{i=1}^{N}(e_f)^2 \tag{4}$$

In the above formulas N can be either the number of points in a cycle or all of the points existed in the time-series. For determining force coefficients the LS value should be minimum and hence:

$$\begin{aligned} \frac{\partial L_S}{\partial C_M} &= A_D.C_D.\sum(u|u|.\dot{u}) + A_I.C_M.\sum(\dot{u})^2 - \sum(f.\dot{u}) = 0 \\ \frac{\partial L_S}{\partial C_D} &= A_D.C_D.\sum(u)^4 + A_I C_M.\sum(u|u|\dot{u}) - \sum f.u|u| = 0 \end{aligned} \tag{5}$$

A simultaneous solution of two above equations gives the necessary equations for coefficients determination as follows:

$$\begin{aligned} C_M &= \frac{\sum(f.\dot{u}).\sum(u)^4 - \sum(f.u.|u|).\sum(u|u|\dot{u})}{A_I.\left(\sum(\dot{u})^2.\sum(u)^4 - \left(\sum(u|u|\dot{u})\right)^2\right)} \\ C_D &= \frac{\sum(f.u|u|).\sum(\dot{u})^2 - \sum(f.\dot{u}).\sum(u|u|\dot{u})}{A_D.\left(\sum(u)^4.\sum(\dot{u})^2 - \left(\sum(u|u|\dot{u})\right)^2\right)} \end{aligned} \tag{6}$$

NEURAL NETWORK METHOD

Artificial neural networks are simplified models of human central nervous system. They are networks of highly interconnected neural computing elements (neurons) that have the ability to respond to input stimuli and to learn and adapt to the environment. Neural network is a method, which does not involve the use of complicated computational solutions. Being a relatively recent technique of computation, its applicability in different areas in yet to be proved. The network is fed with input-output patterns from which it learns and thereby determines the connection weights and biases. After undergoing such training it becomes ready to predict the output of new input data. Built-in dynamism, good generalization, data error tolerance make the network suitable for the present application. Occurrence of waves is basically random and hence neural networks seem suitable as they provide a non-deterministic and model-free mapping between a given set of input and output values. In the present study, we simulated the relationship between random wave particle kinematics with their corresponding inline force using artificial neural network to see whether it is capable of learning the relation or not. We used Matlab neural network toolbox4 for simulation of the network [8].

Linear network: Linear network can only solve linearly separable problems since its transfer function is linear. This allows their outputs to take on any value (Fig. 3). For each input vector we can calculate the network's output vector. The difference between an output vector and its target vector is the error. We would like to find values for the network weights and biases such that the sum of the squares of the errors is minimized or below a specific value. This problem is manageable because linear systems have a single error minimum. In most cases, we can calculate a linear network directly, such that its error is a minimum for the given input vectors and targets vectors. In other cases, numerical problems prohibit direct calculation. Fortunately, we can always train the network to have a minimum error by using the Least Mean Squares (Widrow-Hoff) algorithm. For every multilayer linear network, there is an equivalent single-layer linear network hence we use a single layer linear network which consists of three weights and one bias for training our data [2].

Unlike most other network architectures, linear networks can be designed directly if input/target vector pairs are known. Specific network values for weights and biases can be obtained to minimize the mean square error by using the function *newlind* in Matlab toolbox as follows:

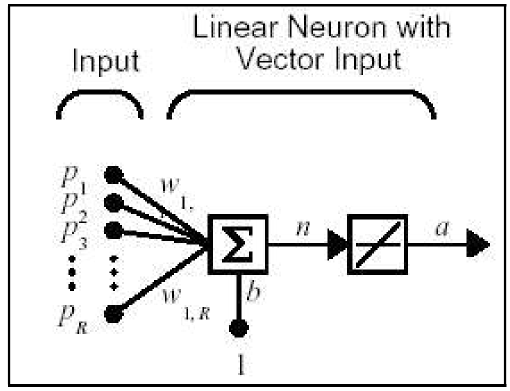

Fig. 3: Structure of a single-layer linear network

$$\text{Net} = \text{newlind (P, T)} \qquad (7)$$

This returns a linear layer designed to output T (with minimum sum square error) given input P.

Mean square error: The least mean square error (LMS) algorithm is an example of supervised training, in which the learning rule is provided with a set of examples of desired network behavior:

$$\{p_1,t_1\}, \{p_2,t_2\}, \ldots,\{p_Q, t_Q\}$$

Here p_q is an input to the network and t_q is the corresponding target output. As each input is applied to the network, the network output is compared to the target. The error is calculated as the difference between the target output and the network output. We want to minimize the average of the sum of these errors.

$$mse = \frac{1}{Q}\sum_{K=1}^{Q} e(k)^2 = \frac{1}{Q}\sum_{K=1}^{Q}(t(k) - a(k))^2 \qquad (8)$$

The LMS algorithm adjusts the weights and biases of the linear network so as to minimize this mean square error. Fortunately, the mean square error performance index for the linear network is a quadratic function. Thus, the performance index will either have one global minimum, a weak minimum or no minimum, depending on the characteristics of the input vectors. Specifically, the characteristics of the input vectors determine whether or not a unique solution exists [2].

LMS or Widrow-Hoff learning algorithm: The LMS algorithm or Widrow-Hoff Delta Rule, introduced by Widrow and Hoff 1960, is based on an approximate steepest descent procedure which is a form of supervised learning. Here again, linear networks are trained on examples of correct behavior. Widrow and Hoff had the insight that they could estimate the mean square error by using the squared error at each iteration. If we take the partial derivative of the squared error with respect to the weights and biases at the *kth* iteration we have

$$\frac{\partial e^2(k)}{\partial w_{1,j}} = 2e(k)\frac{\partial e(k)}{\partial w_{1,j}} \quad (9)$$

for j = 1, 2,…, R and

$$\frac{\partial e^2(k)}{\partial b} = 2e(k)\frac{\partial e(k)}{\partial b} \quad (10)$$

next look at the partial derivative with respect to the error.

$$\frac{\partial e(k)}{\partial w_{1,j}} = \frac{\partial[t(k)-a(k)]}{\partial w_{1,j}} = \frac{\partial}{\partial w_{1,j}}\left[t(k)-(w_p(k)+b\right] \text{ or}$$
$$\frac{\partial e(k)}{\partial w_{1,j}} = \frac{\partial}{\partial w_{1,j}}\left[t(k)-\left(\sum_{t=1}^{R} w_{1,t}p_i(k)+b\right)\right] \quad (11)$$

Here p_i (k) is the *ith* element of the input vector at the *k*th iteration.

Similarly,

$$\frac{\partial e(k)}{\partial w_{1,j}} = -p_j(k) \quad (12)$$

This can be simplified to:

$$\frac{\partial e(k)}{\partial w_{1,j}} = -p_j(k) \quad (13)$$

and

$$\frac{\partial e(k)}{\partial b} = -1 \quad (14)$$

Finally, the change to the weight matrix and bias will be:

2? ∂ (*k*). P (k) and 2? ∂(*k*).

These two equations form the basis of the Widrow-Hoff (LMS) learning algorithm.

These results can be extended to the case of multiple neurons and written in matrix form as:

$$W(k+1) = W(k) + 2\alpha e(k).p^T(k)$$
$$b(k+1) = b(k) + 2\alpha e(k) \quad (15)$$

Here the error e and the bias b are vectors and ? is a *learning rate*. If ? is large, learning occurs quickly, but if it is too large it may lead to instability and errors may even increase. To ensure stable learning, the learning rate must be less than the reciprocal of the largest eigenvalue of the correlation matrix p*T*p of the input vectors. In Matlab toolbox we have a MAXLINLR function which returns the maximum learning rate for a linear layer with a bias [2].

Backpropagation neural network: Backpropagation is a systematic method for training multilayer networks. It has a mathematical foundation that is strong if not highly practical. Despite its limitations, back propagation has dramatically expanded the range of problems to which ANNs can be applied and it has generated many successful demonstrations of its power (Wassermann). This model is used to provide a mapping between some input and output quantities by forming a continuous function. It can be applied to any multilayer network that uses differentiable activation functions and supervised training. It is an optimization procedure based on gradient descent that adjusts weights to reduce the system error. During the learning phase, input patterns are presented to the network in some sequence. Each training patterns set is propagated forward layer by layer until an output pattern is computed. The computed output is then compared to a desired or target output and an error value is determined. The errors are used as inputs to feedback connections from which adjustments are made to the synaptic weights layer by layer in a backward direction. The backward linkages are used only for the learning phase, whereas the forward connections are used for both the learning and the operational phases. The process is repeated a number of times for each pattern in the training set until the total output error converges to a minimum or until some limit is reached in the number of training iterations (epochs) completed. For more details on backpropagation algorithm refer to references [3, 5, 8, 9, 11].

PROBLEM DEFINITION

We want to find out that on what wave parameters or wave particle kinematics, namely, wave height, horizontal velocity and vertical velocity, does the neural network depends to learn the relationship between inputs and output appropriately which is called sensitivity analysis. Various training pattern cases were formed to obtain the desired output in the form of predicted inline force. Each case involves a different combination of input-output parameters (Table 1). Inputs to the network are random wave particle

Table 1: Different cases of network training

Input case	Inputs	Output	No. of training patterns	No. of testing patterns
1	Wave height	Inline force	24000	5000
2	Horizontal velocity	Inline force	24000	5000
3	Vertical velocity	Inline force	24000	5000
4	Wave height, Horizontal velocity, Vertical velocity	Inline force	24000	5000
5	Wave height, Horizontal velocity	Inline force	24000	5000
6	Wave height, Vertical velocity	Inline force	24000	5000

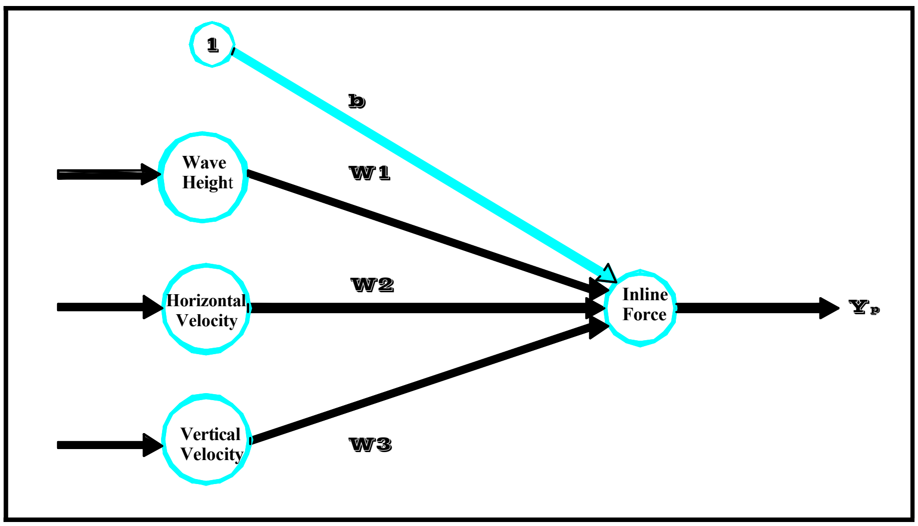

Fig. 4: A schematic of the proposed single-layer network

kinematics while the output of the network is inline force. The networks we used here are of feedforward type. The first one is a single-layer network with Widrow-Hoff learning algorithm. The other one is a multilayer network with backpropagation algorithm.

The data have been divided into two training and testing sets (Table 1). For the Morison equation method we use the training set for calculating inertia and drag force and the testing set for predicting the inline force using the coefficients computed from training set with LSM method.

For neural network method we use the training set for training the network to determine the connection weights and biases then the testing set for evaluating the network's performance.

In Fig. 4, a schematic of the proposed single-layer linear network has been shown. The inputs to the network are wave height, horizontal and vertical water particle velocity and the output is inline force. The connection weights are *w1*, *w2* and *w3* with one bias *b*.

The relationship between the inputs and the output is as follows:

$$Y_{predicted} = w_1 H + w_2 . V_X + w_3 . V_y + b \qquad (16)$$

Our proposed values of *w1*,*w2*, *w3* and *b* will come in the results section.

For comparing the performance, the root-mean-square (RMS) error between the measured and predicted values is used as the agreement index which is defined as:

$$RMS = \sqrt{\frac{\sum_{i=1}^{N}\left(y_{mi} - y_{pi}\right)^2}{\sum_{i=1}^{N}\left(y_{mi}\right)^2}} \qquad (17)$$

In which y_{mi} is the value of measured force and y_{pi} denotes the value of predicted force and N is the

Table 2: A comparison of RMS error, correlation coefficient and time of training of different methods for four cases

	RMS error of testing set			Correlation coefficient (R)			Time of training (sec)	
Input cases	Linear network	Nonlinear network Epochs = 1000	Morison equation	Linear network	Nonlinear network	Morison equation	Linear network	Nonlinear network Epochs = 1000
1	0.82732	0.82211	0.35754	0.548	0.560	0.97	0.08	510.50
2	0.82608	0.81686		0.549	0.563		0.08	549.35
3	0.63838	0.64395		0.765	0.760		0.08	558.00
4	0.35033	0.33373		0.936	0.936		0.13	576.80
5	0.82048	0.80270		0.558	0.584		0.10	563.00
6	0.40322	0.38274		0.914	0.922		0.11	568.00

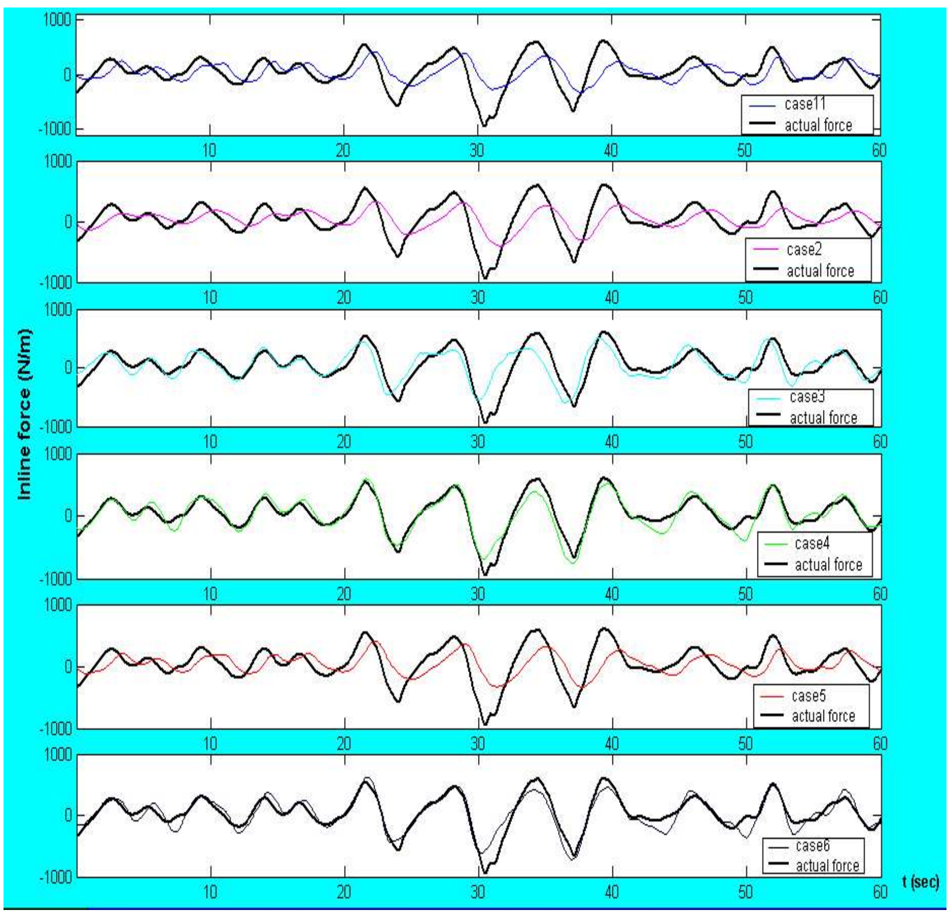

Fig. 5: Inline force prediction in four different cases with linear network

number of data in testing set. The accuracy improves if RMS approaches zero. The other agreement index used in this paper is the coefficient of correlation between the measured and predicted forces [10].

SIMULATION RESULTS

Simulations of different methods, on our data, for all six cases were done. Table 2 shows results of these simulations consisting RMS error and correlation coefficient of each method. The case four gives the least RMS error for neural networks and that implies that vertical velocity parameter has a key role in network performance. Neural network (both type) showed less error compared to the Morison equation but not too far differences. The nonlinear network with BP algorithm showed less error than linear network but many much more time for its training than linear

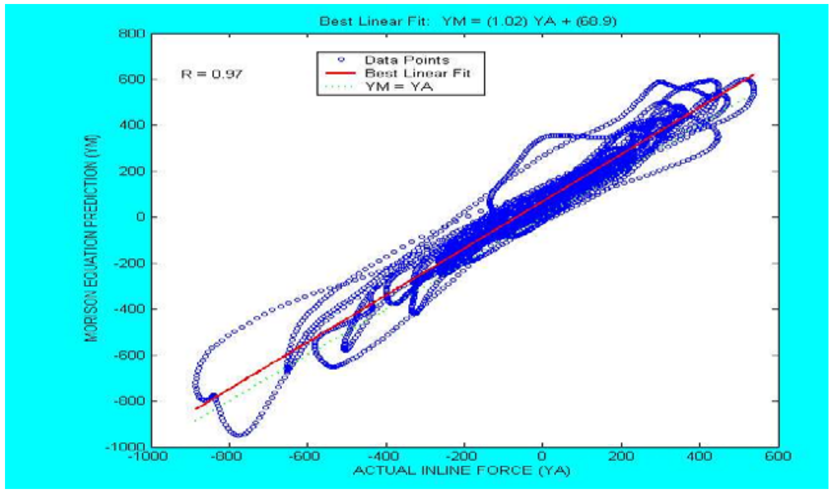

Fig. 6: A regression between Morison equation prediction and the actual force

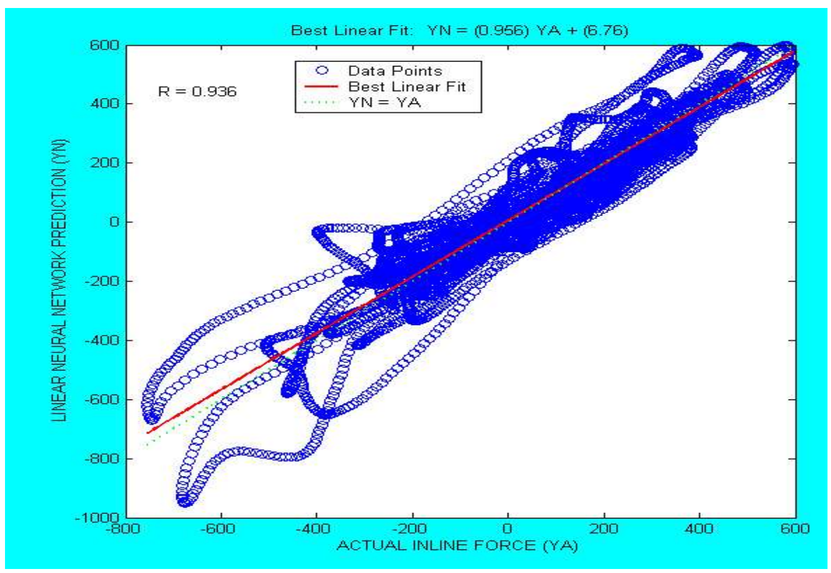

Fig. 7: A regression between linear neural network prediction (case4) and the actual force

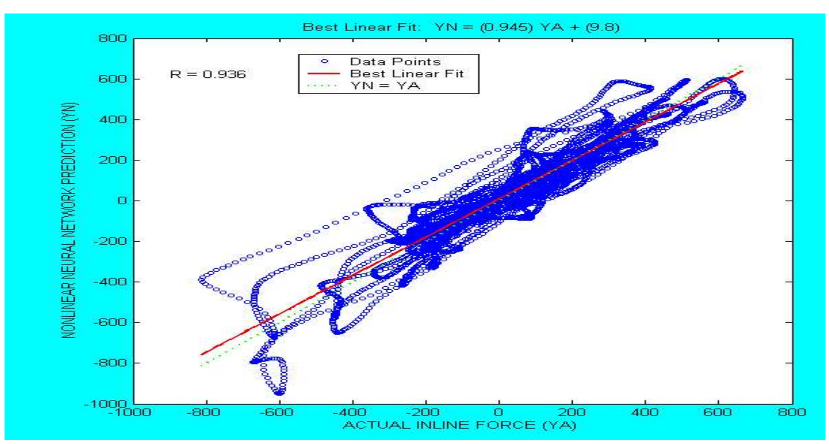

Fig. 8: A regression between nonlinear network (case4) prediction and the actual force

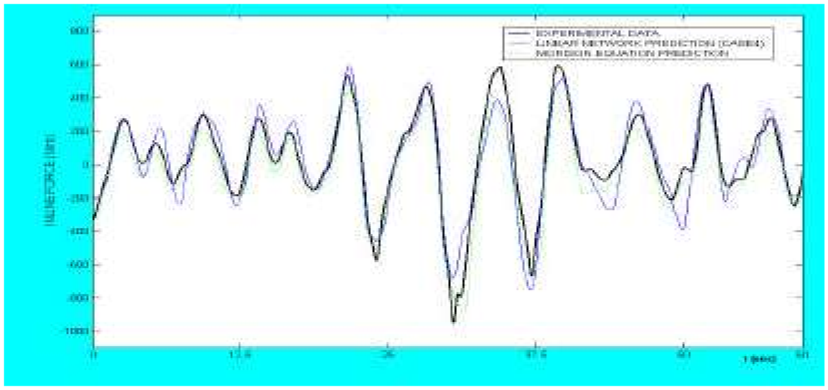

Fig. 9: A comparison between inline force prediction of Morison equation and linear neural network with actual data

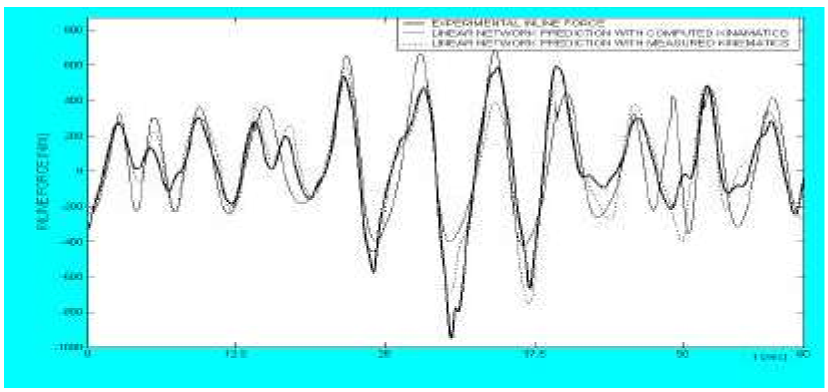

Fig. 10: Neural Network prediction of inline force when input velocities to network are experimental or computed from stokes wave theory

network (Table 2). The simulations were done by a pentium4 PC with 1.8 GB CPU and 256 MB DDRAM.

Since time is very important in design procedures and the RMS error of two neural network types does not differ much, a single layer linear network with one neuron and Widrow-Hoff learning algorithm can be appropriately employed for inline force prediction.

Figure 5 shows the results of testing the single-layer linear network for six cases of Table 1. As you see without vertical velocity parameter as one of the inputs to the network, the prediction of the network for inline force is not desirable.

In Fig. 6-8, regressions between simulation results and the actual force have been done. The value of the correlation coefficient (R) for each regression is shown on its figure. As you see, the Morison equation has the best R-value (0.97).

Figure 9 shows a comparison between the predictions of Morison equation and linear network. The values of inertia and drag coefficients obtained from LSM method for the training data are $C_M = 2.1408$ and $C_D = 1.454$. These coefficients were used for predicting the inline force of the testing set.

Table 3: Values of weights, bias and learning rate of designed linear network

Network	W1	W2	W3	b	Lr (learning rate)
Linear	31.588	285.88	-1089.6	42.919	0.00013608

The designed linear network with case4 inputs gives values of weights and bias as in Table 3.

Because usually we have just wave height parameter measured at sea, we also investigated the case which we have only the wave height measured along with velocities computed with Stokes (Fenton) wave theory, as testing inputs to the network which had been trained with measured parameters (Fig. 10). For more details on Stokes (Fenton) wave theory refer to reference [4]. The RMS of this case of testing inputs which consists of measured wave height, computed horizontal velocity and computed vertical velocity is 0.57146.

CONCLUSION

The main results of this research are:

- Since linear neural network method is not restricted to the type of data and calculation of some coefficients, it can be used for inline force prediction.
- Neural network (both type) showed less error compared to the Morison equation but not too far differences.
- Since time is very important in design procedures and the RMS error of two neural network types does not differ much, a single layer linear network with one neuron and Widrow-Hoff learning algorithm can be appropriately employed for inline force prediction.
- Linear network is preferable to the other networks since it needs the least time to train.
- It can be seen that neural network can give satisfactory results, very close to the real ones.
- A comparison between outputs of Morison equation and Neural network method showed that NN can be a simpler, easier and faster solution.
- The case four gives the least RMS error for neural networks and that implies that vertical velocity parameter has a key role in network performance.
- It was observed that the trained single-layer linear network with just a hidden neuron can appropriately predict any new data which the network had not experienced before.

REFRENCES

1. Chakrabarti, S.K., 1987. Hydrodynamics of Offshore Structures. Computational Mechanics Publications. New York.
2. Mackwood, P.R., 1993. Wave and Current Flows around Circular Cylinders at Large Scale. LIP Project 10D.
3. Naghipour, M., 1996. The Accuracy of Hydrodynamic Force Prediction for Offshore Structures and Morison's Equation. Ph.D Thesis, Heriot Watt University.
4. Patterson, D.W., 1996. Artificial Neural Network, Theory and Applications. Prentice Hall. Singapore.
5. Demuth, H. and M. Beale, 2001. Neural Network Toolbox for Use with Matlab, User's Guide (version 4). The Mathworks, Inc.
6. Fausett, L., 1994. Fundamentals of Neural Networks Architectures, Algorithms and Applications. Prentice Hall. United States.
7. Haykin, S., 1994. Neural Networks, Macmillan College Publishing Company, Inc. United States.
8. Rumelhart, D.E., G.E. Hinton and R.J. Williams, 1986. Learning Representations by Back propagation Errors. Nature, 323: 533-536.
9. Wasserman, P.D., 1989. Neural Computing, Theory and Practice.Van Nostrand Reinhold. United States.
10. Tsai, C.P., C. Lin and J. Shen, 2002. Neural Network for Wave Forecasting among Multi-Stations. Ocean Engineering, 29: 1683-1695.
11. Fenton, J.D., 1985. A Fifth-order Stokes Theory for Steady Waves. Journal of Waterway, Port, Coastal and Ocean Engineering, 111 (2): 216-234.