International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:2, No:12, 2008

Estimation of the Bit Side Force by Using Artificial Neural Network

Mohammad Heidari

Abstract**—**Horizontal wells are proven to be better producers because they can be extended for a long distance in the pay zone. Engineers have the technical means to forecast the well productivity for a given horizontal length. However, experiences have shown that the actual production rate is often significantly less than that of forecasted. It is a difficult task, if not impossible to identify the real reason why a horizontal well is not producing what was forecasted. Often the source of problem lies in the drilling of horizontal section such as permeability reduction in the pay zone due to mud invasion or snaky well patterns created during drilling. Although drillers aim to drill a constant inclination hole in the pay zone, the more frequent outcome is a sinusoidal wellbore trajectory. The two factors, which play an important role in wellbore tortuosity, are the inclination and side force at bit. A constant inclination horizontal well can only be drilled if the bit face is maintained perpendicular to longitudinal axis of bottom hole assembly (BHA) while keeping the side force nil at the bit. This approach assumes that there exists no formation force at bit. Hence, an appropriate BHA can be designed if bit side force and bit tilt are determined accurately. The Artificial Neural Network (ANN) is superior to existing analytical techniques. In this study, the neural networks have been employed as a general approximation tool for estimation of the bit side forces. A number of samples are analyzed with ANN for parameters of bit side force and the results are compared with exact analysis. Back Propagation Neural network (BPN) is used to approximation of bit side forces. Resultant low relative error value of the test indicates the usability of the BPN in this area.

Keywords**—** Artificial Neural Network, BHA, Horizontal Well, Stabilizer.

I. INTRODUCTION

BOTTOM hole assembly (BHA) is the part of drill string that affects the trajectory of borehole by bit side force and tilt. The bit side force is the controlling factor for hard formations (drilling rates 1 to 10 ft/h). For formations that are soft to medium hard, the side force is not the only component that will influence the inclination and direction of the bit; the bit tilt becomes influential as well. Because of the curvature of BHA near the bit, the bit is canted or tilted in some resultant direction and inclination, some what like the bent housing and bent sub side force becomes a controlling factor once again for very soft formations (drilling rates excess of 100 ft/h) [1]. For a given BHA, the bit side force and tilt depends on a number of parameters. These parameters are hole, drill collar and stabilizer sizes, BHA material properties, mud weight, stabilizer locations, borehole inclination, direction, contact length, location between pipe and hole, and finally weight on bit. Unfortunately, analytical techniques derived to estimate bit side force ignores some of these parameters for the sake of simplicity. For example the analytical method developed based on three moment equation ignores the effect of pipe to wall contact as well as hole curvature [2], [3]. Other techniques have been developed to handle the well bore curvature, variable gauge holds and combination BHA components, and situations in which pipe/ wall contact occurs between the bit and stabilizers, as well as the cases in which increases in weight on bit force the creation of additional points of contact [4], [5], [6]. Nowadays Artificial Neural Network (ANN) is used in many of engineering problem. The back propagation network (BPN) is trained to estimate the bit side force in three cases: slick BHA, single stabilizer BHA and two stabilizers BHA. The results show that, an ANN can be estimated the bit side force with high accuracy. By using this method the time of analyzing will be very short.

Mohammad Heidari is the Member of Scientific Committee of Islamic Azad University- Aligudarz Branch, (098-381-3343096; fax: 098-664-2234631; e-mail: moh104337@ yahoo.com).

II. ESTIMATION OF THE BIT SIDE FORCE BASED ON EXISTING ANALYTICAL TECHNIQUES

A set of equations has been derived and published [3] to determine the bit side force based on three moment equation. These equations are derived depending on how many stabilizers are attached such as slick BHA, single–stabilizer BHA and two-stabilizer BHA. In all three cases, the essence of this technique is to determine the point of contact between the pipe and wall of the hole called tangency point.

A. Slick BHA

Slick BHA has no stabilizers attached to it. The tangency point in this case is the first point where pipe departs from the borehole wall above the bit. F_B is the bit side force. Equation (1) is determined the bit side force.

$$F_B = -0.5 W_C L_T B_C \sin\phi + \frac{(WOB - 0.5 W_C B_C L_T \cos\phi) l_3}{L_T} \quad (1)$$

Where L_T is the first point tangency and is measured from the bit. It has to be estimated before substituting in equation (1). L_T is determined by trial and error as follows:

Assume an initial guess for L_T and calculate new L'_T from equation (2). If L_T is not equal to L'_T, then set new $L_T = L'_T$ and repeat the procedure until L_T agrees with L'_T.

$$L'_T = [\frac{24EI.l}{W_C B_C X \sin\phi}]^{\frac{1}{4}} \tag{2}$$

$$X = \frac{3(\tan u - u)}{u^3} \tag{3}$$

$$u = \frac{L_T}{2}\sqrt{\frac{WOB - 0.5W_C B_C L_T \cos\phi}{EI}} \tag{4}$$

in these equations, B_C is buoyancy factor, E is young's modulus, l_3 is hole clearance around drill collars, W_C is unit weight of drill collars, WOB weight on bit, I is axial moment of inertia and ϕ is hole inclination at bit.

B. Single – stabilizer BHA

Equation (5), as in the previous case, side force can be determined as a function of tangency point:

$$F_B = -\frac{B_C W_{C1} L_T \sin\phi}{2} + \frac{(WOB - 0.5W_{C1} B_C L_T \cos\phi) l_1 - m}{L_T} \tag{5}$$

The tangency point L_T is the first point where pipe departs from the borehole wall above the stabilizer.
However, L_T stands for the length from stabilizer up to the tangency point. The same analysis technique can be applied to determine L_T by assuming an initial guess for L_2 and calculating L_T from equation (6). If L_T is not equal to L_2, then set new $L_2 = L_T$ and repeat the procedure until L_T agrees with L_2.

$$L_T = [\frac{24EI(l_3 - l_1) - 4mL_T^2 W_2}{q_2 x_2}]^{\frac{1}{4}} \tag{6}$$

$$m = \frac{-\frac{q_1 L_1^2}{4} X_1 - \frac{q_2 L_2^3 I_1}{4 L_1 L_2} X_2 + \frac{6EI_1 l_1}{L_1^2} + \frac{6EI_1(l_1 - l_3)}{L_1 L_2}}{2(V_1 + \frac{L_2 I_1}{L_1 I_2} V_2)} \tag{7}$$

$$q_1 = W_{C1} B_C \sin\phi \tag{8}$$

$$q_2 = W_{C2} B_C \sin\phi \tag{9}$$

$$W_2 = \frac{3}{u_2}[\frac{1}{\sin(2u_2)} - \frac{1}{2u_2}] \tag{10}$$

$$V_i = \frac{3}{2u_i}[\frac{1}{2u_i} - \frac{1}{\tan(2u_i)}] \tag{11}$$

Where i=1 and 2

$$X_i = \frac{3(\tan u_i - u_i)}{u_i^3} \tag{12}$$

$$u_1 = \frac{L_1}{2}\sqrt{\frac{WOB - 0.5W_{C1} B_C L_1 \cos\phi}{EI_1}} \tag{13}$$

$$u_2 = \frac{L_2}{2}\sqrt{\frac{WOB - 0.5W_{C2} B_C L_2 \cos\phi - W_{C1} B_C L_1}{EI_2}} \tag{14}$$

$$l_1 = 0.5(d_b - d_{SI}) \tag{15}$$

$$l_3 = 0.5(d_b - d_2) \tag{16}$$

in these equations, m is bending moment, d_b is bit diameter, d_2 is drill collar diameter, d_{SI} is 1st stabilizer diameter , l_1 and l_3 are respectively hole clearance around 1st stabilizer and hole clearance around drill collars.

C. Two-stabilizer BHA

The two stabilizer BHA can also be solved with the same technique. The distance between the second stabilizer and the point of tangency (L_3) is unknown, and, as in the previous two cases, L_3 must be guesses initially. This solution technique accommodates three different, the two moments, m_1 and m_2 can be determined and the bit side force can be calculated from the following equations:

$$F_B = -\frac{W_C B_C L_T}{2}\sin\phi + \frac{(WOB - \frac{W_{C1} B_C L_T}{2}\cos\phi) l_1 - m_1}{L_T} \tag{17}$$

$$l_1 = 0.5(\frac{d_b - d_{SI}}{12}) \tag{18}$$

$$l_2 = 0.5(\frac{d_b - d_{S2}}{12}) \tag{19}$$

$$l_3 = 0.5(\frac{d_b - d_2}{12}) \tag{20}$$

$$L_T = [\frac{24EI_3(l_3 - l_2) - 4m_2 L_3^2 W_3}{q_3 X_3}]^{\frac{1}{4}} \tag{21}$$

m_1 and m_2 are solved simultaneously from the following two equations:

$$2m_1(V_1 + \frac{L_2 I_1}{L_1 I_2} V_2) + m_2 \frac{L_2 I_1}{L_1 I_2} W_2 = -\frac{q_1 L_1^2}{4} X_1 - \frac{q_2 L_2^3 I_1}{4 L_1 I_2} X_2 + \frac{6EI_1 l_1}{L_1^2} + \frac{6EI_1(l_1 - l_2)}{L_1^2} \tag{22}$$

$$m_1 W_2 + 2m_2(V_2 + \frac{L_3 I_2}{L_2 I_3} V_3) = -\frac{q_2 L_2^2}{4} X_2 - \frac{q_3 L_3^2 I_2}{4 L_2 I_3} + \frac{6EI_2(l_1 - l_2)}{L_2^2} - \frac{6EI_2(l_3 - l_2)}{L_3^2} \tag{23}$$

Where

$$q_i = W_C B_C \sin\phi \tag{24}$$

$$X_i = \frac{3[\tan(\frac{L_i}{2}\sqrt{\frac{p_{ci}}{EI_i}}) - \frac{L_i}{2}\frac{p_{ci}}{EI_i}]}{(\frac{L}{2}\sqrt{\frac{p_{ci}}{EI_i}})^3} \tag{25}$$

$$W_i = \frac{3}{(\frac{L_i}{2}\sqrt{\frac{p_{ci}}{EI_i}})}[\frac{1}{\sin(L_i\sqrt{\frac{p_{ci}}{EI_i}})} - \frac{1}{(L_i\sqrt{\frac{p_{ci}}{EI_i}})}] \quad (26)$$

$$V_i = \frac{3}{(\frac{L_i}{2}\sqrt{\frac{p_{ci}}{EI_i}})}[\frac{1}{(L_i\sqrt{\frac{p_{ci}}{EI_i}})} - \frac{1}{\tan(L_i\sqrt{\frac{p_{ci}}{EI_i}})}] \quad (27)$$

Where i=1, 2 or 3

$$p_{C1} = WOB - \frac{W_{C1}B_C L_1}{2}\cos\phi \quad (28)$$

$$p_{C2} = WOB - (W_{C1}B_1L_1 + \frac{W_{C2}B_C L_2}{2})\cos\phi \quad (29)$$

$$p_{C3} = WOB - (W_{C1}B_1L_1 + W_{C2}B_C L_2 + \frac{W_{C3}B_C L_3}{2})\cos\phi \quad (30)$$

in these equations, l_2 is hole clearance around 2nd stabilizer, p_{ci} are compressive load on the collars (i=1,2 and 3), L_1 and L_2 are respectively distance between the bit and first stabilizer and distance between the 1st and 2nd stabilizer, W_{C1} and W_{C2} are respectively unit weight of 1st and 2nd stabilizer.

III. Artificial neural network

The human beings brain anatomy, considering the thinking process, has always been one of the extreme mysteries to scientists. Researchers have exerted efforts aiming at mechanically and electronically imitating the reactions of human beings. The invention of computers and the affordability of personal computers with significant processing speeds and huge capacities have encouraged researchers worldwide to tackle problems that previously been out of the scope of their imagination. ANNs are one of these tools that can be considered as problem solving programs modelled on the structure of the human brain where the neural network technology mimics the brains own problem-solving process. The neural network can suit pattern recognition problems, while other problems are best solved with conventional methods. Tracing human's behavior, a neural network takes previously solved examples to build a system of neurons that makes new decisions, classifications, and forecasts [7]. ANNs look for patterns in training sets of data, learn these patterns, and develop an ability to correctly classify new patterns, or to make forecasts and predictions. ANNs excel at problem diagnosis, decision making, prediction, and other classifying problems where pattern recognition is important while precise computational answers are not required.

In a supervised network, the network is shown how to make predictions, classifications, or decisions by giving it a large number of correct classifications or predictions from which it can learn. Back propagation networks (BPNs), general regression neural networks (GRNNs), and probabilistic neural networks (PNNs) are examples of supervised network types. On the other hand, unsupervised networks can classify a set of training patterns into a specified number of categories through clustering patterns rather than being shown in advance how to categorize. Kohonen networks are unsupervised ones. Three basic entities specify ANNs models: namely, models of neurons themselves; models of the synaptic interconnections and structures; and the training or learning rules for updating the connecting weights. A group of neurons is called a slab. Neurons are also grouped into layers according to their connection to the outside world. Thus, a neuron receiving data from outside the network is in the input layer while that containing the networks prediction is in the output layer. Neurons in between the input and output layers are in the hidden layers. A layer may contain one or more slabs of neurons. Neural network "learns" by adjusting the interconnection weights between layers. Iterations take place until reaching an acceptable tolerance between the output results obtained by the network and the actual, correct output initially fed to the system. Eventually, if the problem can be learned, a stable set of weights adaptively evolves that will produce good answers for all sample decisions or predictions. The real power of ANNs is evident when the trained network is able to produce good results for data that the network has never seen before. Unlike statistical methods, ANNs "discover" relationships in the input data sets through the iterative presentation of the data and the intrinsic mapping characteristics of neural topologies "learning" [7].

Two main phases operate ANNs. First, the training or learning phases which is very time consuming since the data is repeatedly presented to the network, while weights are updated to obtain a desired response. The second phase is the recall or the retrieval phase, where the trained network with frozen weights is applied to data that it has never seen before. To the contrary of the training phase, the retrieval phase can be very fast.

It is worth mentioning that a professional experience is the time to stop training. In other words, training may be insufficient and consequently the network will not learn the patterns, while the training may also be excessive which results in the network learning the noise or memorizing the training patterns rather than generalizing well with new patterns. A practical guide to overcome such problems is to randomly extract about 20% of the patterns in the training set to be used for cross validation. The error should then be monitored in the training and validation set. When the error in the validation set increase, this is a signal to stop training where the point of best generalization has then been reached. Cross validation is amongst the most powerful methods to stop the training.

Generally, neural networks offer viable solutions when there are large volumes of data available for training. Moreover, ANNs are considered appropriate solution when field or experimental data is available and a problem is difficult, or impossible, to formulate analytically.

In this paper, back propagation neural network (BPN) has been used as a tool for the analysis

IV. Back propagation of neural network

ANN is a mathematical system which can model the ability of biological neural networks by interconnecting many simple neurons. The neuron accepts inputs from a single or multiple sources and produces outputs by a simple calculating process with a predetermined non-linear function. A typical three-layered network with an input layer (I), a hidden layer (H) and an output layer (O) is shown in Fig.1. Each layer consists of several neurons and the layers are interconnected by sets of correlation weight. The neurons receive inputs from the initial inputs or the interconnections and produce outputs by transformation using an adequate non-linear transfer function. A common transfer function is the sigmoid function expressed by $f(x)=(1+e^{-x})^{-1}$ which a characteristic of has $df/dx = f(x)[1-f(x)]$. The training process of the neural network is essentially executed through a series of patterns. In the learning process, the interconnection weights are adjusted within the input and output values. Therefore, the primary characteristics of ANN can be presented as follows:

(1) The ability to learn; (2) distributed memory; (3) fault tolerance and (4) parallel operation. With these characteristics, ANN has been widely applied to various areas. Since the principal of ANN has been well documented in the literature, only a brief is given in this section.

BPN, developed by Rumelhart, Hinton and Williams [8], is the most prevalent of the supervised learning models of ANN. BPN used the gradient steepest descent method to correct the weight of the interconnectivity neuron. BPN easily solved the interaction of processing elements by adding hidden layers. In the learning process of BPN, the interconnection weights are adjusted using an error convergence technique to obtain a desired output for a given input.

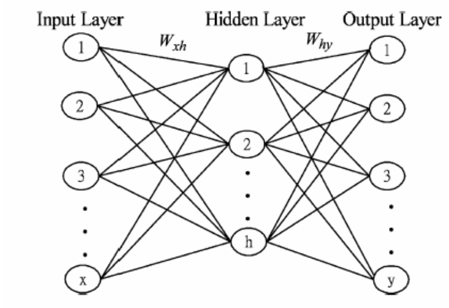

Fig. 1 Structure of artificial neural network

In general, the error at the output layer in the BPN model propagates backward to the input layer through the hidden layer in the network to obtain the final desired output. The gradient descent method is utilized to calculate the weight of the network and adjusts the weight of interconnections to minimize the output error. The error function at the output neuron is defined as:

$$E=\frac{1}{2}\sum_k (T_k - A_k)^2 \tag{31}$$

In which T_k and A_k represent the actual and predicted values of output neuron, respectively, and k is the output neuron.

The gradient descent algorithm adapts the weights according to the gradient error, which is given by:

$$\Delta W_{ij} = -\eta \times \frac{\partial E}{\partial W_{ij}} \tag{32}$$

Where η is the learning rate and the general from of the $\frac{\partial E}{\partial W_{ij}}$ term is expressed by the following from:

$$\frac{\partial E}{\partial W_{ij}} = -\delta_j^n . A_i^{n-1} \tag{33}$$

Substituting equation (33) into equation (32), we have the gradient error as:

$$\Delta W_{ij} = \eta . \delta_j^n . A_j^{n-1} \tag{34}$$

In which A_j^{n-1} is the output value of sub-layer related to the connective weight (W_{ij}). δ_j^n is the error signal, which is computed based on whether or not neuron j is in the output layer.

If neuron j is one of the output neurons, then:

$$\delta_j = (T_j - Y_j).Y_j.(1-Y_j) \tag{35}$$

If neuron j is a neuron of hidden layer:

$$\delta_j = [\sum_j \delta_j .(W_{hy})_{hj}].H_h.(1-H_h) \tag{36}$$

Where H_h is the value of hidden layer. Finally, the value of weight of the interconnectivity neuron can be expressed as follows:

$$W_{ij}^m = W_{ij}^{m-1} + \Delta W_{ij}^m = W_{ij}^{m-1} + \eta . \delta_j^n . A_i^{n-1} \tag{37}$$

To accelerate the convergence of the error in the learning procedure, Jacobs [9] proposed the momentum term with momentum gain, α, in equation (37).

$$W_{ij}^m = W_{ij}^{m-1} + \alpha . \Delta W_{ij}^{m-1} + \eta . \delta_j^n . A_i^{n-1} \tag{38}$$

in which the value for α is between 0 and 1[10], [11].

V. Analysis

ANNs have been used by many researchers in the field of mechanical engineering. In the current study, several neural networks have been developed by using the MATLAB [12] software. This software implements different neural network algoritthms such as BPN, PNN (Probabilistic Neuaral Network), GRNN (General Regression Neural Network) and many other network. To use the MATLAB, a set of inputs must be defiend, and a suitable trianing set has to be developed.

The develop networks in this paper are shown in table 1, that consists of many input parameres and 3 different output parametes, each one depending on the BHA materail properties, hole, weight of bit, stabilizer loction, borehole inclination, direction and contact length.

Each net is composed of four slabs; an input slab, two hidden layer and an output layer. The nodes at input and output layer are determined by the number of predictor and predicted variable.

In this resech, there are 8, 12 and 25 nodes in input layers due to the number of input variable, and 1 node in output layer, for similar reasons. There are no rules given to determine the exact number of hidden layers and the number of nodes in hidden layers.

A large number of hidden–layer nodes will lead to an over-fit intermediate points, which can slow down the operation of NN. On the other hand an accurate output may not be achived if too few hidden layer nodes are included in the neural network. The number of nodes in the first and scond hidden layers are chosen 6-6, respectivily. Table I presents the topology of the built network.

TABLE I
TOPOLOGY OF THE USED BPN

Input	Hidden Layer No. 1	Hidden layer No.2	Output (F_B)
$W_C, E, I, \phi, B_C, WOB, l_3$	4 neurons	5 neurons	slick BHA
$W_{C1}, W_{C2}, E, I_1, I_2, \phi, B_C, WOB, l_3, l_1, m, d_b, d_{s1}, d_2, L_1, L_2$	7 neurons	8 neurons	single stabilizer BHA
$W_{C1}, W_{C2}, E, I_1, I_2, I_3, \phi, B_C, WOB, l_1, l_2, l_3, m_1, m_2, d_b, d_{s1}, d_{s2}, d_2, L_1, L_2, L_3, p_{c1}, p_{c2}, p_{c3}$	9 neurons	10 neurons	two stabilizer BHA

The activation function in the input and the hidden layers is sigmoid fuction and linear fuction in the output layer.

For a proper working of the neural network a pre-processing of the input and output data is performed. The input values are normalized between -1 and 1, since the activation fuction is a sigmoid fuction in the input layer. The output values are normalized between 0 and 1 and a linear function in the output layer. Finally the network is ready to be trained. This process is expianed in the section 7.

For the training of the network the MATLAB neural network toolbox is used. The Levenberg-Marquardt algorithm is chosen to perform the training with default values suggested in [12].

The stopping criteria is adjusted, that mean square error should be less then 10^{-9} and number of epochs (iterations) should be less then 5000.

The output values estimated by the NN are compared with the target values calculated with analyticaly method (see tables II, III and IV). In these table Tr.T and T.T are the training time and test time of the neural network, respectivily.

TABLE II
BIT SIDE FORCE IN SLICK BHA (lbf)

No.	F_B (Analytical)	F_B (NN)	Tr.T(s)	T.T(s)	Error (%)
1	-133.21	-131.42	24.74	0. 06	1.34
2	-113.94	-111.72	13.34	0. 11	1.98
3	-93.96	-90.41	28.11	0. 16	3.77
4	-73.14	-70.30	19.47	0. 21	3.88
5	-51.62	-51.1	15.73	0. 03	1
6	-29.24	-27.76	11.56	0. 16	5.06

F_B = bit side force, NN = neural network, Tr.T = training time, T.T = test time, s = second, No =number, lbf = pound force.

TABLE III
BIT SIDE FORCE IN SINGLE STABILIZER BHA (lbf)

No.	F_B (Analytical)	F_B (NN)	Tr.T(s)	T.T(s)	Error (%)
1	3031.11	2984.24	37.11	0.81	1.54
2	3044.49	2829.17	20.01	0.79	7.07
3	3058.33	3000.45	42.16	0.75	1.89
4	3073.14	3010.74	29.22	0.85	2.03
5	3087.38	3044.23	23.59	0.77	1.39
6	3101.23	3012.18	17.34	0.77	2.91

F_B = bit side force, NN = neural network, Tr.T = training time, T.T = test time, s = second, No =number, lbf= pound force.

TABLE IV
BIT SIDE FORCE IN TWO STABILIZER BHA (lbf)

No.	F_B (Analytical)	F_B (NN)	Tr.T(s)	T.T(s)	Error (%)
1	-559.20	-534.85	40.12	0.91	4.35
2	-550.51	-520.12	30.18	0.64	5.55
3	-541.48	-510.36	33.41	0.84	5.74
4	-531.81	-515.14	52.18	0.51	3.13
5	-520.23	-508.92	25.69	0.33	2.17
6	-508.39	-501.74	19.35	0.62	1.30

F_B = bit side force, NN = neural network, Tr.T = training time, T.T = test time, s = second, No = number, lbf = pound force.

VI. CONCLUSION

According to the previous studies, it is evident that ANNs perform better than, or as well as, conventional methods used as a basis for comparison in many situations, whereas, they fail to perform well in a few.

The main objective of this paper has been to develop reliable BPNs that can be used to assess bit side force susceptibility. The developed nets can be used as standalone, simple, yet reliable.

The proposed study proved that using BPNs in the problem of bit side force susceptibility produces excellent results via an accurate, yet simple tool. The results show that the BPNs can be estimated the bit side force with high accuracy (the maximum error is below 5 percent). By using this method decreases the time of analysis.

The reduction of the required time was due to the fact that only the bit side force was used in the drill has not the complex equations. The BPN method also has the advantages of computational speed, low cost and ease of use by people with little technical experience.

References

[1] T. B. Adam, K. K. Millheim, M. E. Chenevert, F. S. Young, "Applied Drilling Engineering", vol. 2, 1991, SPE Tex book Series, Dallas, TX, USA.

[2] S. G. Timoshenko, *Theory of Elastic Stability*, MCGraw-Hill, New York, 1936.

[3] B. Jiazhi, Bottom Hole Assembly Problems Solved by Beam- Column Theory, SPE Int. Meeting of Petroleum Engineering, Beijing SPE10561, 1986.

[4] H. B. Walker, Down hole assembly design increases ROP., World Oil, 1977, pp.59-65.

[5] K. K. Millheim, M. C. Apostal, The effect of bottom hole assembly dynamics on the trajectory of a bit, J. Pet. Technol., 1981, pp. 2323-2338.

[6] M. Agawani, S. S. Rahman, E. E. Maidla, "BHA Design Algorithm for Extended Reach Wells", SPE Petroleum Computer Conference, Dallas, TX, USA, 1996.

[7] S. Hayken, "Neural Networks: A Comprehensive Foundation", Macmillan College Publishing Co., New York, 1994.

[8] D. E. Rumelhart, G. E. Hinton, R. J. Williams, "Learning Representations by Back Propagating Error", Nature 323, 1986, pp. 533-536.

[9] R. A. Jacobs, Increased rates of convergence through learning rate adaptation, neural networks, vol. 1, 1988, pp. 295-307.

[10] P. D. Wasserman, *Neural Computing: Theory and Practice*, Van Nostrand Reinhold, New York, 1989.

[11] J. A. Freeman, *Simulating Neural Networks*, Addison-Wesley Publishing Company, Inc., New York, 1994.

[12] H. Demuth, M. Beale, M. Hagan, *Neural Network Toolbox for Use with MATLAB*, The Math Works, Inc, 2006

Mohammad Heidari is was born in khoramshar, Iran in 7th of October in 1977. He got his B.S degree in mechanic engineering from Shiraz University in 2001 in Iran. He received his M.S degree in mechanic engineering in 2003 from Shahid Chamran University of Ahvaz in Iran. He is specialist in Artificial Neural Networks and Shakedown. He became a member of scientific committee in Aligudarz Azad University in 2003 and has been there for five years. He has written two books named Residual Stresses and their Effects and Engineering Elasticity.