PREDICTION OF THRUST AND TORQUE IN DRILLING USING CONVENTIONAL AND FEEDFORWARD NEURAL NETWORKS

Vishy Karri, Tossapol Kiatcharoenpol
School of Engineering, PO Box. 252-65, University of Tasmania, Australia – 7001

Abstract

Drilling performance prediction, using traditional mechanics of cutting approach, is based on the extension of three-dimensional oblique cutting theory. The quantitative reliability of such conventional models depend on a numerous number process variables and quantitative accuracy of the data bank for a given work material. The complexity of such models is increased when inevitable eccentricity and drill deflections are incorporated into the analysis. In this paper, using a novel neural network architecture that optimises the output layer, the thrust and torque in drilling operation are carried out. A set of comprehensive drilling tests is carried out to train and test the architecture. It has been shown that the percentage deviations of drilling predictions using the neural network architecture is -0.56% and 1.03% for thrust and torque compared to 4.20% and -10.25% using traditional mechanics of cutting approach.

Keywords : Drilling Performance Prediction, Neural Networks

1. Introduction

Traditional mechanics of cutting approach and empirical approaches have been used for drilling performance prediction, in the past. Artificial neural networks are used in the recent year for cutting tool wear estimation, tool condition monitoring, vibration control and surface finish detection [12]. While these predictions are comparable with conventional models, 'simultaneous' estimation of more than one performance feature is quite often necessary for 'on-line' control of a machining process. The conventional mechanics of cutting models are efficient to the extent of predicting individual performance but cannot estimate different performance features 'simultaneously'. A brief description of unified mechanics of cutting and empirical approaches to drilling performance prediction are carried out in this work before the capabilities of neural network models are presented.
drill. Due to the high negative rake angles and low cutting velocities encountered at the chisel edge, a discontinuous orthogonal cutting model was applied [14,20].

Figure 1. Geometry and shape of general purpose drill

The elemental chisel edge length $dL_c$, the mean radius $r$, dynamic angles and cut thickness at each element for the selected number of elements could be obtained, hence the elemental thrust and torque on the chisel edge could be determined by summation of the elemental thrust and torque values. The total thrust and torque on the drill as a whole were found by summing the corresponding values in the lip and chisel edge regions [4,13,14,17,21].

The thrust and torque predictions using the mechanics of cutting models were ±15% to the experimental values while machining S1214 free machining steel [14]. While the approach above seems most promising for thrust and torque prediction, it should be noted the commonly specified drill point features $2p$, $2W$, $\delta$, and $D$ and $r$, together with cutting conditions $N$ and $f$ should be given to predict the thrust and torque in drilling. The accuracy of the mechanics of cutting approach depended on these features along with the orthogonal cutting data bank. Therefore the accuracy of this traditional approach to predictions, was found to be dependent on the reliable orthogonal cutting data bank, the accuracy of the edge forces and the reliable estimation of drilling geometrical features. Neural network modelling involved a fewer parameters for the simultaneous prediction of thrust and torque as discussed below.

3. Brief Description of Back Propagation and Optimised Layer by Layer (OLL) Neural Network Architectures

While the specified literature provides adequate theory on the neural network models studied in this paper, it is useful here to consider the basic theory associated with each of these neural networks with an understanding of the industrial application applied. It is important to note that while the objective of each neural network is to predict the values of thrust and torque in drilling, the architecture and algorithms used by each network to achieve this are significantly different. A brief note on the BP and OLL networks is discussed below.

The standard backpropagation network [5,6,8,15,23,24] comprises 3 layers of processing elements, fully feedforward connected. With the sigmoid on the hidden layer as shown in the figure 3 below, only the basic equations are:

$$y_k = \sum_{j=1}^{H} \mu_{kj} z_j$$

$$z_j = \frac{1}{1 + \exp(-\theta_j)}$$
The Least Mean Square error

\[
E = \frac{1}{2} \sum_{k=1}^{N} (y_k - t_k)^2
\]

All the data is scaled between 0..1 but it can be scaled between -1..1 to standardise all the inputs with various dimensions. The weights \( w_{ij} \) and \( u_{ik} \) are assigned random numbers in the range -1..1, and a random pair of input / output vectors are picked from the training set. The input vector is fed through the network to get an output vector (feed forward process), this is then compared with the output vector and an error is found.

This error is then passed back through the neural network (back propagation process) to modify the weights using the following equations

\[
\begin{align*}
\Delta u_{kj} &= u_{kj}^{old} + \Delta u_{kj}^{new} \\
\Delta w_{ji} &= w_{ji}^{old} + \Delta w_{ji}^{new}
\end{align*}
\]

The gradient descent optimisation technique is used to calculate the change in each weight. This is then repeated by picking another random pair of input / output vectors and continuing until the error is at a minimum.

Momentum can be used to decrease times in training and the chance of the network getting stuck in a shallow minimum. This is done by accelerating the convergence of the error but is not applied in this situation.

The architecture of an OLL network [9], shown in Figure 4, consists of an input layer, one or more hidden layers and an output layer. All input nodes are connected to all hidden nodes through weighted connections, \( w_{ji} \), and all hidden nodes are connected to all output nodes through weighted connections, \( V_{kj} \).

**Training Algorithm of OLL with one hidden layer (fig. 4.)**

**Step 1** Initialize weights: set all weights to small random value range (-1,1)

\( W_{ji} \): weight connecting input node \( i \) to hidden layer node \( j \)

\( V_{kj} \): weight connecting hidden node \( j \) to output layer node \( k \)

Set weight factor \( \mu = 0.0001 \), set Bias value = 1

{ Optimization of output-hidden layer weight Step 2 to 3 }

**Step 2.1** Calculate the response of each hidden layer node by the activation function, the Sigmoidal function, until the end of training patterns by following equation.

For \( j = 1, \ldots, H \)

\[
w_{neti} = \sum_{i=0}^{M} (x_i \cdot w_{ji})
\]

For \( j = 1, \ldots, H \)

\[
f(n_{net}) = \frac{1}{1 + \exp(-n_{net})}
\]

where \( n_{net} \) : target output value of hidden layer node \( j \)

**Step 2.2** Calculate weights \( A_{(j)} = A^{j \times B} \)

\( A_{(j)} \) : matrix \( (a_{ij}) \)

\( a_{ij} = \sum_{o_{kj}} [o_{kj} \cdot o_{0j}] ; j, i = 0, \ldots, H \)

\( B_{(k)} \) : matrix \( (b_{kj}) \)

\( b_{kj} = \sum_{o_{kj}} [o_{kj} \cdot o_{0j}] ; k = 1, \ldots, N \) (number of output nodes)

where \( o_{kj} \) : target output for node \( k \)

\( P \) : number of training patterns

\( \sum_{q=1}^{P} \) : summation from \( q = 1 \) to \( P \)

**Step 3.1** Calculate the response of each output layer node by the activation function, the linear function:

For \( k = 1, \ldots, N \)

\[
y_k = f(n_{net})
\]

where \( n_{net} \) : weighted summed input to output layer node \( k \)

\( y_k = \) the output value of output layer node \( k \)

\( \sum_{j=0}^{H} \) : summation from \( j = 0 \) to \( H \) (fig. 4.)

**Step 3.2** Calculate Root Mean Square error (RMS)

\[
RMS = \sqrt{\frac{1}{P} \sum_{k=1}^{N} (y_k - y_{0k})^2}
\]
Step 4.1 Calculate linearized weights in each output layer node:

For $k=1$ to the last node at output layer

$$V_{linj} = \sum_{h=1}^{H} f'(net_h) V_{ij}$$

Where $V_{linj}$ = linearized weight connecting hidden node $j$ to output layer node $k$

$f'(x) = \text{derivative of the sigmoidal function} = f(x)[1-f(x)]$

Step 4.2 Calculate weight correction term:

$$\Delta W_{opt} = A_{u} x b_{u}$$

$$A_{u} = \text{matrix} \{ a_{ij}(h) \};$$

$$a_{ij}(h) = \text{for} (j \neq h) = \sum_{k=1}^{S} \sum_{l=1}^{S} ((V_{linj} x_k)(V_{linh} x_m))$$

$$= \text{for} (j = h) = \sum_{k=1}^{S} \sum_{l=1}^{S} ((V_{linj} x_k)(V_{linj} x_m)) + (\mu/H) \text{abs}(V_{ij}) f'(net) x_i x_m$$

$$b_{u} = \text{vector} \{ b_{ij} \};$$

$$b_{j} = \sum_{i=1}^{S} \sum_{l=1}^{S} (t_i - y) V_{linj} x_i$$

where $\sum_{j=1}^{S} = \text{summation from k=1 to N (fig.4.)}$

abs(x) = absolute value of x

Remark: matrix $A_{u}$ is $(S \times S)$ dimension square matrix.
vector $b_{u}$, $\Delta W_{opt}$ is $(S)$ dimension vector.

$S = H \times (M+1)$ dimensions (fig.4)

Step 4.3 Calculate $W_{test}$ by adding weight correction term:

For $j=1$ to the last node at hidden layer

For $i=0$ to the last node at input layer

$$W_{testi} (new) = W_{ii} (old) + \Delta W_{opt}$$

Step 5 Calculate Root Mean Square error (RMStest) by using $W_{test}$

$$\text{RMStest} = \sum_{i=1}^{S} \sum_{j=1}^{S} 1/2 (t_i - y)^2$$

Step 6 Compare between RMS and RMStest:

If (RMStest > RMS) then

$$\mu = \mu*1.2 \text{ (increase } \mu)$$

Go back to Step 4

Step 7 If (RMStest < RMS) then

$$W_{ii} = W_{testi}$$

$$\text{RMS = RMStest}$$

Step 8 Decrease weight factor

$$\mu = \mu*0.9 \text{ (decrease } \mu)$$

Step 9 Do Step 2 to Step 8 until the end of iterations

4. Development of Training Data Results and Discussion

In order to train the network on a comprehensive range of cutting conditions and process variables, drilling experiments were carried out. ANCA automatic drilling machine was used to carry out the experiments. The thrust and torque were measured using three component dynamometer and associated data acquisition system. Taking the handbook recommendations and associated feasible drill geometrical features a total of 72 experiments were carried out. The training of the network is carried out for the 57 cutting conditions above. All the input variables were scaled between 0-1 and the training was carried out over 57 combinations of cutting conditions.

The training was found to be excellent accuracy with a small error at training stage indicating that the network is well trained with only 10 inputs and meets the target thrust and torque accurately. The ten inputs were $P_1$, $P_2$, $dP$, and $D$ and $W/R$, together with cutting conditions $eR$ and $e6$ and $f$. It can be seen that the BP and OLL neural networks have trained well with great quantitative accuracy highlighting the predictive capability of the networks.

The error was calculated using the deviation formula $(\text{Predicted-Exp.}/\text{Exp.})*100$ and the percentage deviations at the training stage were excellent as shown in Table I for the thrust and for the torque respectively. It can be seen from Table I that at the training stage for both thrust and torque there is no significant bias either for over prediction or under prediction for both BP and OLL neural network models.

The neural network architecture was tested over 15 various conditions. The multi-layer perceptron with back propagation program was run to check the predictability of the neural network model for the testing stage.
Table I. Average percentage deviations for training predictions for BP and OLL compared to conventional methods

<table>
<thead>
<tr>
<th>Method</th>
<th>% Deviation</th>
<th>Thrust</th>
<th>Torque</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conv</td>
<td>3.91</td>
<td>-8.79</td>
<td></td>
</tr>
<tr>
<td>BP</td>
<td>-1.25</td>
<td>0.19</td>
<td></td>
</tr>
<tr>
<td>OLL</td>
<td>0.07</td>
<td>0.18</td>
<td></td>
</tr>
</tbody>
</table>

From testing point of view, the histograms in the Fig.6a-f highlight the comparison of the quantitative accuracy of three different approaches to thrust and torque prediction. The average percentage deviations of predicted thrust and torque by conventional methods are 4.20% and 10.25% respectively (Fig.6a and d). For neural network based predictive models, BP had an average percentage deviations of -4.68 and -2.00% (Fig.6b and 6e) and OLL had average percentage deviations of -0.56% and 1.03% for thrust and torque respectively (Fig.6c and 6f). Comparing with conventional method, neural network approaches yield higher quantitative accuracy. While the dispersions of histograms of all three models are similar, it is evident that neural network approaches can offer high accuracy prediction.
5. Conclusion

The need for reliable and simultaneous prediction of thrust and torque in drilling operations is highlighted. Mechanics of cutting approach to thrust and torque predictions has been extensively used in the past. The mechanics of cutting approach and its complexity is highlighted in this paper from predictive point of view. It has been shown that a number of process parameters is required together with for the drilling performance prediction. The accuracy is further tested when reliable geometrical features are not established. In this work the well-established multi-layer back propagation neural network (BP) with 2 outputs has been chosen as a first architecture together with optimised layer by layer forward network (OLL) for performance prediction as a second architecture. Experiments were carried out over a range of cutting conditions to gather thrust and torque components in drilling operations. A range of drilling conditions covering 57 different cutting conditions and tool geometrical features were selected as a training set. The neural network algorithm has trained well with excellent quantitative accuracy with less that 2% average percentage deviation to the experimental values using both BP and OLL networks. The network is tested with 15 different cutting conditions and showed excellent predictive capability. The average percentage deviation of predicted thrust and torque by conventional methods are 4.20% and 10.25% respectively. For neural network based predictive models, BP had an average percentage deviation of -4.68 and -2.00% and OLL had average percentage deviations of -0.56% and 1.03% for thrust and torque respectively.

References: