Prediction of bead geometry in pulsed GMA welding using back propagation neural network

K. Manikya Kanti, P. Srinivasa Rao*

Department of Mechanical Engineering, Gayatri Vidya Parishad College of Engineering, Visakhapatnam, Andhra Pradesh, India

ABSTRACT

This paper presents the development of a back propagation neural network model for the prediction of weld bead geometry in pulsed gas metal arc welding process. The model is based on experimental data. The thickness of the plate, pulse frequency, wire feed rate, wire feed rate/travel speed ratio, and peak current have been considered as the input parameters and the bead penetration depth and the convexity index of the bead as output parameters to develop the model. The developed model is then compared with experimental results and it is found that the results obtained from neural network model are accurate in predicting the weld bead geometry.

1. Introduction

The GMAW-P process has recently gained wide attention in the welding industry, owing to its comparatively low heat input and precise control over the thermal cycle. This is because, with the pulsed current gas metal arc welding process, spray transfer or more precisely controlled droplet transfer is obtained at a low average current. This provides a smaller and well-controlled weld pool, which allows cladding of thin materials in all positions.

The superiority of the GMAW-P process is mainly related to the nature of the metal deposition, which is governed by the pulse parameters, namely peak current, pulse frequency, wire feed rate, wire feed rate/travel speed ratio, etc. Bead geometry in the arc welding process is an important factor in determining the mechanical characteristics of the weld (Srinivasa Rao, 2004; Srinivasa Rao et al., 2005; Rajasekaran, 1999; Senthil Kumar and Parmer, 1986). Bead geometry variables, such as bead width, bead height, and penetration depth, are greatly influenced by welding process parameters such as wire feed rate, welding current, welding speed, plate thickness etc. Weld shape and size are represented by bead width ($W$), bead height ($H$) and depth of penetration ($P$) as shown in Fig. 1. Convexity index (CI) is defined as the ratio of bead reinforcement height to the bead width. Selection of the most suitable combination of pulse parameters is very difficult owing to the complex interdependence of the above parameters on the pulsed current (Srinivasa Rao, 2004; Srinivasa Rao et al., 2005).

However, costly and time-consuming experiments are required in order to determine the optimum welding process parameters due to the complex and nonlinear nature of the welding process. Therefore, a more efficient method is needed to determine the optimum welding process parameters. Technique of neural networks offers potential as...
an alternative to standard computer techniques in control technology and has attracted a widening interest in their development and application (Rajasekaran, 1999). The advantages of the neural networks is that the network can be updated continuously with new data to optimize its performance at any instance, the networks ability to handle a large number of input variables rapidly, and the networks ability to filter noisy data and interpolate incomplete data.

The back propagation network (BPN) system is one of the family of artificial neural network techniques used to determine welding parameters for various arc welding processes. The network is a multi layer network that contains at least one hidden layer in addition to input and output layers. Number of input layers and number of neurons in each hidden layer is to be fixed, based on the application, the complexity of the problem, and the number of inputs and outputs. Use of non-linear log-sigmoid or tan-sigmoid transfer function enables the network to simulate non-linearity in practical systems. Due to its numerous advantages, back propagation network is chosen for present work (Kim et al., 2001, 2002, 2005; Chan et al., 1999; Lee and Um, 2000; Krose and Van der Smagt, 1996).

2. Feed forward back propagation network

Back propagation learning is a supervised learning where it needs to know the inputs and the desired outputs in advance. This network is well established as a method for data mapping. In this work, the welding parameters are mapped to the weld dimensions through the internal representation of BPNs. Data obtained from the experiments and regression models are provided to a network at the learning stage, e.g., welding parameters and weld bead geometry dimensions. During network learning, the network output is compared with the desired output and the connector weights inside the network adjust to minimize the difference. The error is then propagated backwards through the network and weights are changes based on the back propagation-learning algorithm. This learning process is an iterative process and learning will stop once an acceptable error is achieved. When the trained network is presented with new input (beyond training), the network responds according to the knowledge it has acquired (Hagan et al., 1996; Yagnanarayana, 1999; Hammerstrom, 1993; Rukmini Srikant, 2003). Fig. 2 shows the architecture of the proposed system.

The process parameters of the pulsed GMA welding process are fed as input to the input layer of the network. Each of the inputs, \( I_i \) is multiplied by weights on inter-layer connections of a hidden layer neuron, \( W_{ij} \) and added to bias, \( \phi \) to produce activation, \( a \)

\[
a_1 = W_{ij}I_i + \phi \tag{1}
\]

where \( I_i \) is the output of the input layer, \( W_{ij} \) is the weight structure between input and hidden layers.

A log-sigmoid activation function that transforms activation of hidden layer neurons to a scaled output \( O_1 \) can be written as

\[
O_1 = \frac{1}{1 + e^{-a_1}} + 1 \tag{2}
\]

Outputs from hidden layer neurons are treated as inputs to output layer neurons. Summation of product of all hidden layer outputs and weights between hidden and output layers added to bias constitutes activation \( a_2 \) of output layer neurons. Log-sigmoid transfer function is applied and output \( O_2 \) is computed. Error is estimated as difference between actual and computed outputs. This procedure constitutes forward flow of back propagation phase and error computed is back propagated through same network to update weights. Weights are updated using generalized delta rule as given below:

\[
W_{\text{new}} = W_{\text{old}} - \eta E_T I \tag{3}
\]

where \( W_{\text{new}} \) is weight after modification, \( W_{\text{old}} \) is the weight structure before modification, \( \eta \) is the learning rate, usually taken between 0 and 1, \( E_T \) is the error obtained.

Weight change is calculated for all connections. Errors for all patterns are summed and the algorithm is run till error falls below a specified value. Back propagation algorithms attempts to minimize the error of mathematical system represented by neural network’s weights and thus walk downhill to the optimum values for weights. Unfortunately due to the mathematical complexity of even simplest neural network, there are many minima, some deeper than others. In order to reach the global minimum, the aid of heuristic optimization techniques is generally sought (Hagan et al., 1996; Yagnanarayana, 1999; Hammerstrom, 1993; Rukmini Srikant, 2003).
al., 1996; Yagnanarayana, 1999; Hammerstrom, 1993; Rukmini Srikant, 2003). The present work uses the concept of adaptation learning rate to overcome local minima and speed up the process of attaining stable weight structure, also known as convergence.

### 2.1. Variable learning rate (traingda) algorithm

With standard steepest descent, the learning rate is held constant throughout training. The performance of the algorithm is very sensitive to the proper setting of the learning rate and can be improved if we allow the learning rate to change during the training process. An adaptive learning rate will attempt to keep the learning step size as large as possible while keeping learning stable. First, the initial network output and error are calculated. At each epoch new weights and biases are calculated using the current learning rate. New outputs and errors are then calculated. If the new error exceeds the old error by more than a predefined ratio \( \text{max} \_\text{perf} \_\text{inc} \) (typically 1.04), the new weights and biases are discarded. In addition, the learning rate is decreased (by multiplying by \( \text{lr} \_\text{dec} \)). Otherwise, the new weights, etc., are kept. If the new error is less than the old error, the learning rate is increased (by multiplying by \( \text{lr} \_\text{inc} \)). This procedure increases the learning rate, but only to the extent that the network can learn without large error increases. The gradient is computed by summing the gradients calculated at each training example, and the weights and biases are only updated after all training examples have been presented (MATLAB 6.1, 2001).

### 3. Model development

#### 3.1. Process parameters used in the study

Five input process parameters, plate thickness \( A \), pulse frequency \( B \), wire feed rate \( C \), wire feed rate to travel speed ratio, i.e. \( \text{WFR} / \text{TS} \) ratio \( D \) and peak current \( E \) are used in the present study to predict two output parameters, the depth of penetration \( \text{FD} \) and the convexity index \( CI \). Experiments have been carried out for different combinations of inputs and the bead penetration depth and convexity index have been found (Srinivasa Rao et al., 2005).

The process parameters included in performing the experiment are three levels of plate thickness \( \text{A'} \) \( (6, 8, 10 \text{mm}) \), three levels of frequency \( \text{B'} \) \( (50, 101, 152 \text{Hz}) \), three levels of wire feed rate/travel speed ratio \( \text{D'} \) \( (15, 20, 25) \), three levels of peak current \( \text{E'} \) \( (440, 480, 520) \) and varying wire feed rate \( C \) at \( 3.0, 4.0, 4.5, 5.0, 5.5, 6.0, 6.5, 7.0 \) and \( 8 \text{ m/min} \). All other parameters except those under consideration are fixed.

#### 3.2. Development of neural network model

Various experiments have been carried out for different combinations of inputs and the bead penetration depth and convexity index have been found. Fifty-four samples are taken to develop the neural network model, of which, 48 samples are used for training the neural network model and 6 samples are used for testing.

### 4. Network experimentation

In this study, an attempt is made to construct a single multi output BPN model to predict both weld dimensions with one network, as it is believed that the welding parameters and weld dimensions are interrelated in such a way that the solution should always be considered as a set (Manikya Kanti, 2006). Therefore solving all with one network is a more logical approach in this case. The neural network development and the training are carried out using MATLAB 6.1 application tool.

A schematic representation of the BPN model is shown in Fig. 3. The hidden network structure contains two layers and a bias node. The neural network employed in this case is a feed forward back propagation rule with ‘traingda’ as learning algorithm.

The input layer has five neurons (since five input parameters) and the output layer has two neurons (depth of penetration and convexity index). Various network structures with one and two hidden layers with varying number of neurons in each layer were examined. Out of all the networks tested, a network with two hidden layers with five and four hidden neurons in each layer reached the best performance goal when compared with other networks. This particular network also gave accurate results for test data compared to other networks. Hence the network with two hidden layers with five and four hidden neurons is selected in this work.

The inputs and the outputs are normalized in the range \([0,1]\). A log-sigmoid transfer function is used for both the layers as the outputs are ranging between zero and one. The network was created in MATLAB 6.1 using the function net1 = newff([minmax(p),[5,4,2]],’logsig’,’logsig’,’logsig’,’traingda’).

Then it was imported to GUI network/data manager for training and testing the results. The MATLAB command window consisting of sample code and workspace containing the results of the present work is shown in Appendix A. The numbers of the samples for training and testing are 48 and 6, respectively as discussed above. With a learning rate of 0.06, \( \text{lr} \_\text{dec} \) of 0.5, \( \text{lr} \_\text{inc} \) of 1.05 and \( \text{max} \_\text{perf} \_\text{inc} \) of 1.04, and a performance goal set to 0.0002, the network was trained for 150,000 iterations and an error of 0.0006 was reached as connection weights increased and decreased as a neural network settled down to a stable cluster of mutually excitatory nodes as shown in Fig. 4.
5. Results

A back propagation neural network (BPN) model to predict bead geometry is developed in this work. To ensure the accuracy of the BPN model developed to predict bead penetration depth and the convexity index, the experimental results and the predicted results using the developed BPN model are compared. A correlation coefficient is calculated to measure the relationship between experimentally measured output and the output predicted by the BPN model. However, a high correlation coefficient is not necessarily equivalent to accurate predictions since the slope of the measured vs. predicted plot is not reflected by this parameter (Chan et al., 1999). Therefore, percentage difference is also included to measure the spread of prediction error.

The following sections show the results obtained by the BPN model and their comparison with the experimental results.

A comparison plot of the analysis data and verification data of the penetration depth and the convexity index of the bead in the multiple regression analysis is shown in Figs. 5 and 6, respectively. It can be observed from Figs. 5 and 6 that the results obtained by BPN model are in close agreement with the experimental results for training samples.

Fig. 5 – Comparison of predicted and actual results in predicting bead penetration depth while training using BPN model.

Fig. 6 – Comparison of predicted and actual results in predicting convexity index while training using BPN model.

Fig. 7 – Comparison of % error of predicted results using BPN model for depth of penetration.

Fig. 8 – Comparison of % error of predicted results using the BPN model for convexity index.

Fig. 9 – Line of best fit for predicted PD and the actual PD by BPN model for training samples.
The percentage difference to measure the spread of prediction error obtained by the BPN model for all the samples in predicting the depth of penetration and the convexity index of the bead is calculated. It can be observed from Figs. 7 and 8 that the ability of the BPN model to predict bead geometry is well within the allowable error.

The regression analysis is performed to find out the correlation coefficient. The correlation coefficient is used to measure the relationship between the measured and predicted values. And it is observed that a regression coefficient of 0.989 and 0.983 is obtained while predicting the bead penetration depth and the convexity index by BPN model, respectively for training samples proving that accurate bead prediction is possible through the welding input parameters using this model. The line of best fit using the plotted points is calculated using the regression and is shown in Figs. 9 and 10.

To determine the accuracy of the BPN model for predicting the outputs when an unknown input is given a regression analysis is performed on the network and line of best fit is determined (Figs. 11 and 12) for testing samples of NN model. A correlation coefficient of 0.99 and 0.954 are obtained for the testing samples while predicting the depth of penetration and convexity index respectively.

6. Conclusions

The effects of process parameters on bead penetration depth and convexity index in pulsed GMA welding using artificial neural networks has been studied, and the following conclusions have been reached:

- The neural network model developed in this work from the experimental data (for bead penetration depth and convexity index) can be employed to control the process parameters in order to achieve the desired bead penetration depth and convexity index.
- It is observed from the results that a correlation coefficient of 0.99 is obtained between the results obtained by the experimental and the BPN model developed. Also the percentage error obtained for desired bead penetration depth and convexity index is relatively very less. This shows that the developed neural network model is capable of making the prediction of the weld bead dimensions with reasonable accuracy.
Appendix A. MATLAB command window consisting of sample code and workspace containing the results

```
>> klw=thickness
klw=[6.6310];
velref=velocity
velref=[5.010101];
velref2=velocity
velref2=[10 15 20 25];
velref3=velocity
velref3=[440 480 520 440 480 520];

% Neural network setup
net=newff([6 2 1],{'tansig','purelin'},'trainlm',... 'trainbr');
net.inputWeights{1}={w11,w12,w13};
net.layerWeights{1}={w12,w13};
net.bWeights{1}={b1};
net.outputWeights{1}={w21,w22};
net.bWeights{2}={b2};
net=setup(net,48);
net=train(net,netinput,netoutput);
net.out{1}={t1,t2,t3,t4,t5};
net=applynet(net,netinput);
net=applynet(net,netinput);
```

REFERENCES


