International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 5 Number 13 (2010) pp. 2355–2366

https://dx.doi.org/10.37622/IJAER/5.13.2010.2355-2366

Neural Networks Design for Control in CO_2 Welding using Backpropagation and Levenberg-Marquardt Algorithm

K. Lalitnarayan[1], M.M.M. Sarcar[2] and K. Mallikarjuna Rao[3]

[1]*Assoc. Prof., Dept. of Mech. Engg., Sir C.R. Reddy College of Engg., Eluru, India*
[2]*Prof., Dept. of Mech. Engg., A.U. College of Engg., AU, Visakhapatnam, India*
[3]*Prof., Dept. of Mech. Engg., J.N.T.U. College of Engg., Kakinada, India*

Abstract

CO_2 welding is a joining process which is used to produce high quality joints and has a capability to be utilized in automation systems to enhance productivity. Despite its wide spread use in the various manufacturing industries, the full automation of the CO_2 welding has not yet been achieved partly because mathematical models for the process parameters for a given welding task are not fully understood and quantified. In this paper a neural network model is developed to predict the weld bead width as a function of key process parameters in CO_2 welding. The neural network model is developed using two different training algorithms, namely, the error back-propagation algorithm and the Levenberg-Marquardt approximation algorithm. The accuracy of the neural network models developed in this study has to be tested by comparing the simulated data obtained from the neural network model with that obtained from the actual CO_2 welding experiments. The result will show that the Levenberg-Marquardt approximation algorithm is the preferred method, as this algorithm reduces the root of the mean sum of square (RMS) error to a significantly small value.

Keywords: Error back-propagation algorithm, Levenberg-Marquardt approximation algorithm, RMS error, Factorial technique.

Introduction

In CO_2 welding, the weld quality is greatly effected by the welding parameters. Especially, the welding process parameters are closely related to the geometry of the back bead. The optimal welding conditions are determined by a combination of factors such as the type of base metal, the welding process and the geometry of the welded parts. Several mathematical models to control welding quality, productivity, microstructure and weld properties in arc welding processes have been studied [1]. However, it is not an easy task to apply them to the various practical situations because the relationships between the process parameters and the bead geometry are non-linear and are usually dependent on the specific experimental results. It is difficult to establish a mathematical model that can predict the result of the actual welding process and determine the optimum welding condition under typical process constraints. In recent years, artificial neural networks (ANNs) have become very useful tools to develop models which express the interrelationship between the input and the output of complicated systems [2]. The key benefit of ANNs in the domain of engineering design and group technology is in their ability to store a large set of parameter patterns as memories for the system which can be later recalled. When recalled, these memories will be excited with a key pattern containing a part of information on a particular member of a stored pattern set. This particular set of patterns can be evoked through the association of the key pattern and the stored information.

In addition, the neural network has learning and generalization capabilities so that the prediction of the correlation between the input examples and the expected output and the generalization of the relationship between two can be established systematically. Because of this after a certain amount of training, the neural network can generate appropriate outputs in response to new inputs [3]. This capability makes the neural network a useful tool in many applications in manufacturing industry. A very large amount of data is needed in order to determine the optimal welding conditions. It is impossible in continuous butt welding to check whether each back-bead of the welded part has been formed in the desired bead size. Also working conditions do not allow for the installation of the costly vision sensors in each area. Therefore, geometry prediction system of back-bead [4] is needed in order to predict the size of the back bead without the use of separate measuring systems. Numerous attempts have been reported to develop mathematical models relating process variables and bead geometry for the selection and control of the procedural variables [5-7]. Chandel [8] first applied this technique to the GMA welding process and investigated relationships between process variables and bead geometry of bead on plate welds deposited by the GMA welding process. Technique of neural network offers potential as an alternative to standard computer techniques in control technology, and has attracted a widening interest in their development and application. Development of the intelligent system for prediction of process parameters for robotic arc welding has been described in the literature [9]. Cook has preliminary worked at the development of intelligent control systems incorporating ANN [10]. Also, srikanthan and chandel [11] proposed the steps adapted to construct the neural network model for GMA welding and evaluated the proposed neural network model.

Description of CO_2 Welding for Mild Steel Plates

The CO_2 welding process, sometimes called metal active gas (MAG) welding is a welding process that yields coalescence of metals by heating with a welding arc between the continuous filler metal (consumable) electrode and the work piece. The continuous wire electrode is drawn from a reel by an automatic wire feeder and then fed through the contact tip inside the welding torch. In operation, the filler wire of CO_2 welding is melted mainly by the heat generated from the welding arc. Internal resistive power plays only a minor role. The heat is concentrated by the welding arc from the end of the melting electrode to the molten weld pool and by the molten metal that is being transferred to the weld pool. The molten weld pool and the electrode wire are protected from contaminants in the atmosphere by the externally supplied shielding gas such as CO_2 or mixtures Ar with O2 in various combinations.

In this paper, the measurement of the bead width is considered as an output index to evaluate the accuracy of the model and algorithm. A statistically designed experiment based on a factorial technique was chosen because it is much more efficient than the other common approaches. The process parameters considered in this study were the Number of passes—-three levels; 2, 3 and 4 (seeFig. 1), the welding current (170, 220 and 270 A), the arc voltage (23, 26 and 28 V) and the welding speed (12-50 cm/min). All other parameters were fixed. The base material used for this study was the BV-AH32 steel 12mm in thickness for multi pass CO_2 welding process. Chemical compositions and mechanical properties of BV-AH 32 steel are shown in Tables 1 and *2.* The plate was cut into 300mm, 200mm, 2 pieces, and the joining surfaces were blasted to remove dirt and oxides.

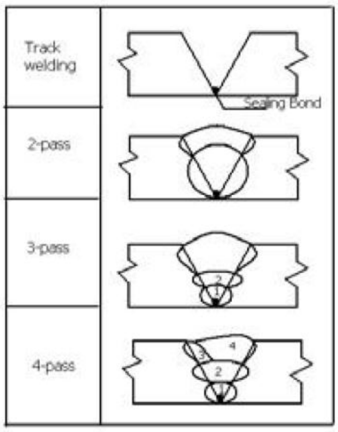

Figure 1: Welding specimen of pass number.

Table 1: Chemical Compositions of BV – AH 32 Steel (wt %).

C	Si	Mn	P	S	Cr	Ni	cU	Nb	V	Mo
0.16	0.42	1.5	0.018	0.005	0.03	0.03	0.02	0.0036	0.005	0.03

Table 2: Mechanical properties of BV – AH 32 steel.

Yield Strength (kgf / mm2)	Tensile strength (kgf / mm2)	Elongation (%)	Young's Modulus (kgf / mm2)
41.02	57.35	20	21,740

The CO_2 welding system and an automatic traveling unit were combined to make an automatic process system. The shielding gas composition was Ar 80% + CO_2 20%. Experimental test plates were located in the fixture around the welding robot and the required weld conditions were led to the particular weld steps in the programmed path of the robot. After welding was completed, the transverse sections of each weld were cut using a power hacksaw at the middle position of welds. Specimen end faces were machined polished and then etched using a 2.5% natal solution to display bead width.

The schematic diagrams of the bead width on top of bead surface were analyzed using a metallurgical microscope interfaced with an image processing system. The images are represented by a 256 level gray scale. An algorithm Lining Auto matrix j [12] was developed to identify and measure the bead width. The fractional factorial matrix was constructed to link the mean values of the measured results with changes in the four process parameters to predict the optimal bead width. Fig.2 shows the major input and output parameters associated with the quality characteristics in robotic CO_2 welding process.

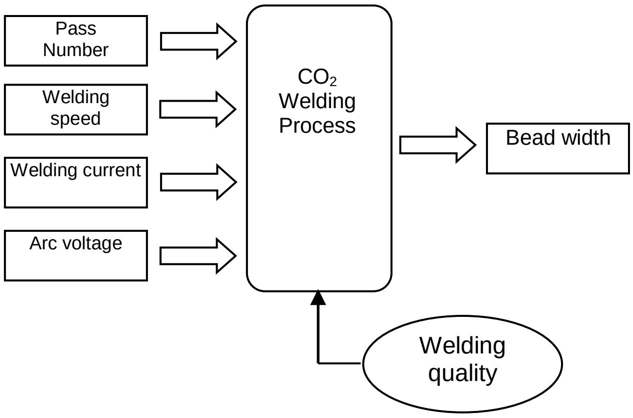

Figure 2: Input and output parameters of the CO_2 welding process.

ANNS For The Robotic Welding

ANNs are widely accepted in the artificial intelligence (A1) research where a non-linear mapping between input and output parameters is required for a function approximation [3]. In this research, a multi-layer back propagation network was employed as a tool for mapping the complex and highly interactive process parameters such as pass number, welding speed, welding current and arc voltage into the bead width to predict the optimal bead width in robotic CO_2 welding process. Fig. 3 shows a schematic representation of a multi layer neural network architecture employed in this research. Two specific training algorithms the error back propagation algorithm and the Levenberg-Marquardt approximation algorithm – are employed to establish the network.

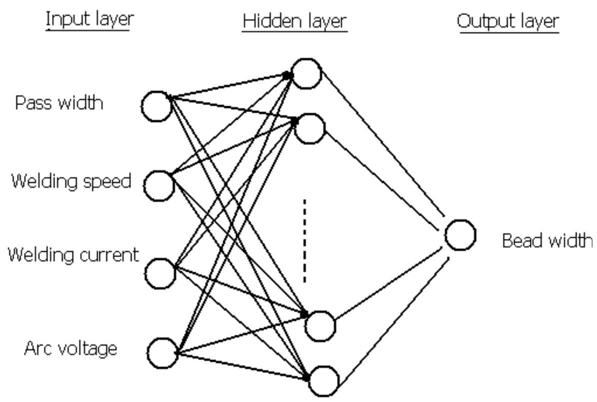

Figure 3: Optimal Neural network architecture for predicting bead width.

The process parameters of the roboticCO$_2$ welding process are inter-dependent and constantly changing in a complex way. Therefore a feed-forward structure for the neural network was adopted in this work. There are several factors that influence the bead width in the robotic CO_2 welding processes. However, this work considered only four factors; the number of Passes, welding speed, welding current and arc voltage. A four layer feed-forward network is constructed with four input neurons in the input layer and one neuron in the output layer to map the output bead width to four input variables. In the developing stages of the neural network, initialization of weights, selection of the activation function and selection of the number of neurons to be used in the hidden layer are considered as the learning factors. The understanding of these factors is important because they affect not only the network's convergence, but also the accuracy of the prediction. In order for a neural network to be trained properly, the weights of the neural network have to be initialized at a set of small random values. Unless the network is distributed by random factors during the training, the internal representation will continuously result in symmetric weights. If this happens, the net work may fail to learn the training examples with the error stabilizing or even increasing as the learning continues.

Network Trained with the Error Back Propagation Algorithm

The training of the neural network using an errorback propagation algorithm was carried out according to the procedure presented in Section 3. The network configuration setup for the training is shown in Table 3. Many attempts are made to find a result showing the minimum error or the maximum prediction accuracy.

Table 3: Neural network configuration for the training with the error back-propagation algorithm.

Four layer network	**Four neurons in input layer, one hidden layer, one neuron in output layer**
Max epoch	500
Error goal	1.0 x 10-6
Learning rate	0.05
No. of neurons in hidden layer	2 – 16 (in steps of 2)
Number of training sets used	One

The discussion presented here is based only on the case of the optimal results obtained with different configurations. In order to understand the effect of the network parameters on the bead width, the number of neurons is intentionally chosen from 2 to 16 neurons in a hidden layer. The accuracy of the network was evaluated by the root of the mean sum of squared (RMS) error between the measured and the estimated values for the training and the testing. Table 4 presents various network configurations using the error back propagation algorithm.

Table 4: The result of various network configurations with the error back-propagation Algorithm.

Configu-ration	Training RMS error	Testing RMS error	CPU time (sec)
4-2-1	2.3272 e -2	0.6404	6.98
4-4-1	2.1782 e -2	0.4745	7.00
4-6-1	2.6664 e -2	0.5237	7.11
4-8-1	2.7203 e -2	0.7768	7.20
4-10-1	2.8318 e -2	0.7638	7.24
4-12-1	2.8541 e -2	0.4172	7.25
4-14-1	2.8673 e -2	0.8236	7.27
4-16-1	3.1151 e -2	0.9488	7.40

According to Table 4, the lowest RMS error is achieved in the 4-4-1 case, while the highest RMS error is obtained from the 4-16-1 case. During the training process, it was found that an increase of the number of neurons in the hidden layer is not directly related to the decrease of the RMS error. The result shows that the average

RMS error at the training phase is reduced to about 0.0695. No further reduction of the RMS error is achieved even if the number of iteration was increased. The RMS error at the testing phase is higher than that of the training phase, meaning that the network has good generalization ability. From the results obtained, it can be observed that the RMS error from the different network configurations is able to converge to the final error goal value of 0.01 within the pre-set maximum training cycle of 500 epochs. The lowest RMS error obtained using this algorithm is 0.0218 with 4 neurons in the hidden layer. It can also be noted that the accuracy of prediction is decreased with the increase of the number of neurons in the hidden layer with the exception of 2 neurons. This could indicate that the increase of the number of neurons would not directly improve the capability of the function approximation of the network. Based on the RMS error of the training examples and the testing examples, it is clear that the 4-4-1 configurations provided the lowest RMS error among all the structures with one hidden layer.

Network trained with the levenberg – Marquardt algorithm

The Levenberg-Marquardt approximation algorithm is employed to further enhance the overall accuracy of the network. The adjustment of weights and biases are done according to transfer function:

$$DW = (J^TJ+mI)-1J^Te;$$

Where 'J' is Jacobian matrix of derivation of each error, 'm' is a scalar and 'e' is error function. The Levenberg- Marquardt approximation algorithm is employed in this paper to further improve the overall accuracy of the neural network because generally this algorithm could provide a faster convergence than the gradient descent algorithm used in the error back propagation algorithm. The training process continues until either the error goal is met, the maximum number of epochs is completed, or 'm' reaches a maximum value. The variable m determines whether learning processes is according to Newton's method or by gradient descent. The network configuration setup for the training is shown in Table 5, the neurons of the input and the interpolation layers are employed to accept the process parameters and then generate estimated bead width. Table 6 shows the performance of various, configurations of the neural network using the Levenberg Marquardt approximation algorithm. As can be seen in the Table 6, the RMS error in the training process is not directly related to the increase of the number of neurons in the competition layer. The learning process is terminated as a scalar reaches its maximum value, i.e., when the network configuration is adjusted to the parameters shown in Table 5. However, the average RMS error of the testing set is larger than that of the training set.

Table 5: Neural network configurations for the training with the Levenberg – Marquardt approximation algorithm.

Four layer network	Four neurons in input layer, one hidden layer, one neuron in output layer
Max epoch	500
Error goal	1.0 x 10-6
Learning rate	0.05
Multiplier for increasing MU	10
Multiplier for decreasing MU	0.1
Maximum value for MU	1.0 x 1010
No. of neurons in hidden layer	2 – 16 (in steps of 2)
Number of training sets used	One

Table 6: The result of various network configurations with the Levenberg – Marquardt approximation algorithm.

Configu-ration	Training RMS error	Testing RMS error	CPU time (sec)
4-2-1	3.7405 e – 3	0.7570	10.70
4-4-1	8.4505 e – 5	0.0357	11.20
4-6-1	5.6373 e – 4	0.0401	12.76
4-8-1	1.6977 e – 4	0.9835	14.84
4-10-1	2.8743 e – 4	1.7903	17.43
4-12-1	5.6296 e – 4	1.7822	20.60
4-14-1	6.8854 e – 4	1.6404	24.03
4-16-1	1.7223 e – 4	1.7546	30.52

The lowest RMS error obtained by using this algorithm is 0.0000845 with 4 neurons in hidden layers in 500 training cycles. Since all of the predicted data are well within the acceptable limit, the network with this configuration is used to predict the process parameters for the robotic CO_2 welding. With the use of the Levenberg—Marquardt approximation algorithm, the network convergence is much faster and the error goal value is able to set to a much lower value compared to the error back propagation training algorithms. It is also noted that unlike the previous technique, the network's mapping ability is not necessarily improved with the increase of the number of neurons in the hidden layer. In conclusion, the 4-4-1 network configuration seems to be best suited for the prediction of the bead width with the lowest RMS error. Also, the Levenberg Marquardt approximation algorithm provides a better learning ability for the robotic CO_2 welding process than the error back propagation algorithm.

Selecting the Most Accurate Network

The result of the comparison study in the performance between the two methods with the 4-4-1-network configuration is plotted in Fig, 4. In the figure, the dotted line represents the estimated bead width using Levenberg-Marquardt approximation algorithm, the double dotted line is that obtained from the error back propagation algorithm and the solid line represents the actual bead width obtained from robotic welding operation.

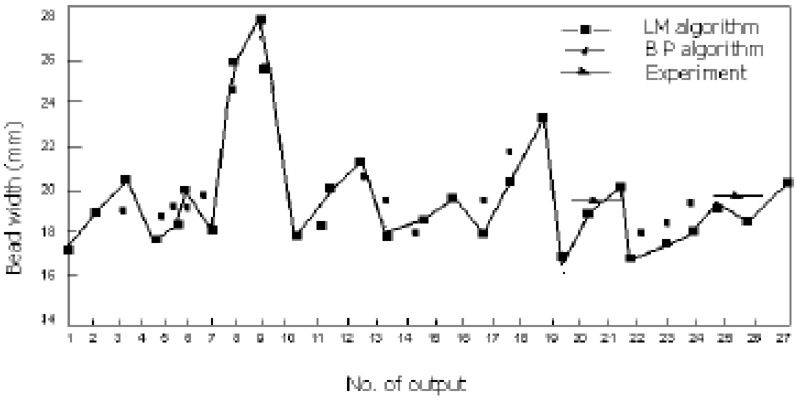

Figure 4: Performance of the error back propagation algorithm and the Levenberg-Marquardt approximation algorithm for bead width prediction.

It was observed that the calculated values obtained by using Levenberg Marquardt approximation algorithm shows better accuracy than those of the developed models using an error back propagation algorithm. It was also shown that the numbers of the errors generated from the error back propagation algorithm are still reasonably small to be accepted in most cases of practical applications. In order to select the most accurate neural network, additional experiments were carried out Table 7 shows the process parameters and the measured results from this additional experiment. The developed neural network model with an error back propagation algorithm and the Levenberg-Marquardt approximation algorithm are compared with their corresponding experimental results.

The welding conditions like the number of passes, welding speed, welding current and arc voltage are employed as the input parameters. The output parameter is the bead width calculated by each model and the corresponding errors of the prediction.

The predicted results from the established models are plotted in Fig. 5 together with the experimental results being listed in Table 7. According to Table 7 and Fig. 5, the neural network model with the Levenberg- Marquardt approximation algorithm gives the best fit to the experimental results and produced better prediction of the bead width than the error back propagation algorithm.

Table 7: Process parameters and results for the additional experiment.

Trial No	Pass No	Arc current (A)	Welding speed 1 (Cm / Min)	Welding speed 2 (Cm / Min)	Welding speed 3 (Cm / Min)	Welding speed 4 (Cm / Min)	Welding voltage (V)
1	2	250	26	26	---	---	27
2	2	200	22	18	---	---	25
3	3	250	34	34	34	---	27
4	3	200	27	27	22	---	25
5	4	250	38	38	41	41	27
6	4	200	28	30	32	32	25

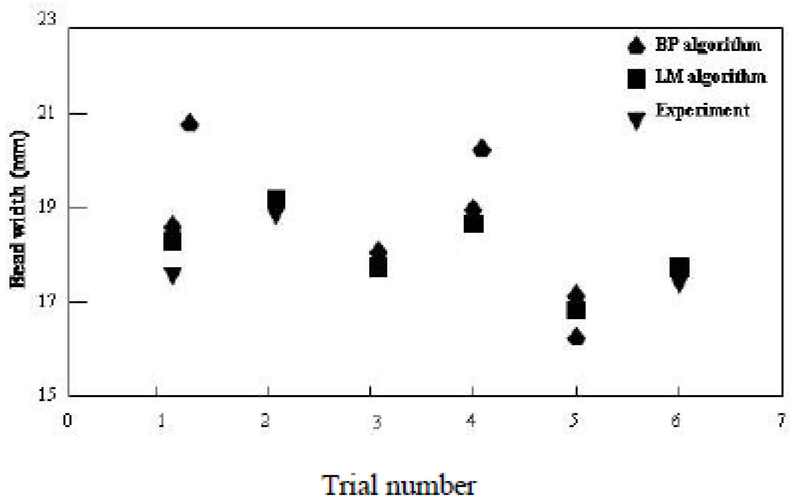

Figure 5: Comparison of Measured and calculated results using two training algorithms.

From the result, it can be concluded that the predictions obtained from the neural network model agree closely with the actual values obtained from the additional experiment, which could prove the feasibility and effectiveness of this research..

Conclusions

A new approach that predicts bead width in the robotic CO_2 welding process using artificial neural networks and factorial design is proposed. The study has employed four process parameters as the input to the network, number of passes, welding speed, welding current and arc voltage.

The neural network model with the error back propagation algorithm and the Levenberg-Marquardt approximation algorithm was trained with data collected from the experiment. The Levenberg-Marquardt approximation algorithm was found the best fit for this application, as it can reduce the RMS error to a significantly small value and thus provide better accuracy of prediction. After the 500 cycles of a training process, the result shows that the use of a one hidden- layer of 4-4-1 configuration model was more effective and accurate than any other structures. The models developed are able to predict the process parameters required to obtain the desired bead width to establish guidelines and criteria for the most effective joint design and further to help the development of automatic control system of arc welding robot. The trained neural network functioned as a mapping mechanism and is capable of predicting the bead width when a new welding condition is given. A rule-based expert system can be incorporated with the developed neural network system to develop an optimized system for the robotic CO2 arc welding process. It has been realized that with the use of the system, the prediction of bead width becomes much simpler to even a novice user who has no prior knowledge of the robotic CO2 welding process and optimization techniques.

References

[1] Feder DK. Computers in welding technology—a look at applications, potentials, welding quality, the role of computers. Vienna, Austria: Pergamon Press, IIW; 198. p. 17-35.

[2] Cook GR, Andersen K. Barrett R.J. feed back and adaptive control in welding. Proceedings of the Second International Conference on Trends in Welding Research. Gallinburg. Tennessee, USA 1989. p. 891 -903.

[3] Freeman JA, Shapura DM. Neural networks—algorithms, applications and programming techniques. New York: Addison-Weslcy; 1991.

[4] I.S kim, W.H. Kwon, C.E. Park, The effects of welding process parameters on weld bead width in GMAW processes, the korea welding society 14(4) (1996) 204-213.

[5] T.Shinoda, J. Dhorety, The relationships between arc welding parameters and weld bead geometry, Welding Institute Report 74/1978/PE,1978.

[6] J.C. McGlone, The submerged arc butt welding of mildsteel. part 1: The influence of procedure parameters on weld bead geometry, Welding Institute Report 79/1978/PE,1978.

[7] J.Doherty, T.Shinoda, J.Weston, The relationships between arc welding parameters and fillet weld geometry for MIG welding with flux cored wires, Welding Institute Report 82/ 1978/PE, 1978.

[8] R.S Chandel, mathematical modelling of gas metal arc weld features, in: Proceedings of the fourth international conference on modeling of casting and welding processes,1988, pp 109-120.

[9] Y.S Tarng, H.L. Tsai, S.S.Yeh ,Modeling, Optimization and classification of weld quality in tungsten inert gas welding, Int. J. Machine tools Manuf. 39 (1999) 1427-1438.
[10] G.E. Cook, Feedback and adaptive control in automated arc welding system, Met.Constr. 13(9) (1990) 551-556.
[11] L.T Srikanthan, R.S Chandel, Neural network based modelling of GMA welding process using small data sets, in: Proceedings of the fifth international conference on control, Automation, Robotics and Vision, Singapore,1988 pp 474-478.
[12] Automatrix Inc, Image Analyst user's manual,Rail, Macrail, AI, U.S.A. 1990.