Journal of Scientific & Industrial Research
Vol. 69, May 2010, pp. 350-355

Neuro hybrid model to predict weld bead width in submerged arc welding process

J Edwin Raja Dhas[1]* and Somasundaram Kumanan[2]

[1]Department of Automobile Engineering, Noorul Islam University, Nagercoil 629 180, India
[2]Department of Production Engineering, National Institute of Technology, Tiruchirappalli 620 015, India

Received 09 November 2009; revised 09 February 2010; accepted 10 February 2010

This paper presents development of neuro hybrid model (NHM) to predict weld bead width in submerged arc welding. Experiments were designed using Taguchi's principles and results were used to develop a multiple regression model**.** Data set generated from Multiple Regression Analysis (MRA) was utilized in ANN model, which was trained with backpropagation algorithm in MATLAB platform and used to develop NHM to predict quality of weld. NHM is flexible and accurate than existing models for a better online monitoring system.

Keywords: Bead width, Hybrid neuro model, Multiple regression model, Submerged arc welding

Introduction

Submerged arc welding (SAW) process[1-7] is widely used in heavy fabrication industries due to its high quality, reliability, attaining deeper penetration and a smooth bead. Mathematical models were used to model welding process[8-10]. Fractional factorial technique[11], statistical control technique[12], curvilinear regression models[13] and linear regression models[14] have been used to model welding process. Response surface methodology (RSM) was used to predict weld bead quality in SAW of pipes[15]. Multiple regression analysis (MRA) technique[16] has been applied to predict process parameters for gas metal arc welding. Taguchi method and regression analysis were used to determine weld process parameters[17,18]. Artificial neural network (ANN), fuzzy logic and expert system have emerged as competent tools for real time monitoring of weld process. ANNs with back- propagation learning algorithm[19-21] has been used to model different arc welding processes but with no guarantee of an optimal solution. Genetic algorithm (GA), which gives near optimal solutions for any kind of problems[22,23], guides weight updating process in ANN technique to build fast, accurate predictive networks. Hence, back propagation weight changing mechanism of ANN is replaced with GA operators in neuro hybrid system. A neurofuzzy (ANFIS) system has been used to predict weld bead width (WBW) in SAW process[24]. This study presents development of a neuro hybrid model (NHM) to predict WBW in SAW process for a given set of welding parameters.

Proposed Methodology [Neuro Hybrid Model (NHM)]

Data Acquisition

Experiment was carried out at M/s Ind Auto Products, Tiruchirappalli, India in semiautomatic SAW machine (SURARC of type XRCP 1200) with Taguchi's design of experiments using L_8 orthogonal array (factors, 4; levels, 2). Two mild steel plates (500 mm x 50 mm x 6 mm) of IS 2062 grade (Fe, 98.8%; C, 0.25%; Si, 0.20%; Mn, 0.75%) were welded with a square butt joint at a single pass in a SAW machine keeping electrode (diam, 3.15 mm) positive and perpendicular to plate. Samples (width, 10 mm) were cut from test piece. Then specimens were cleaned, polished and etched. Profile projector measured WBW (Table 1).

Multiple Regression Analysis (MRA)

MRA was performed using a statistical package. Relationship between WBW and its dependent variables (Fig. 1) is given as: WBW (mm) = -34.833 + (6.667×10^{-2} × welding current, amperes) + ($0.750\times$arc voltage, volt) + (1.25×10^{-2} × welding speed, mm/min) - ($4.17\times$

*Author for correspondence
E-mail: edwinrajadhas@rediffmail.com

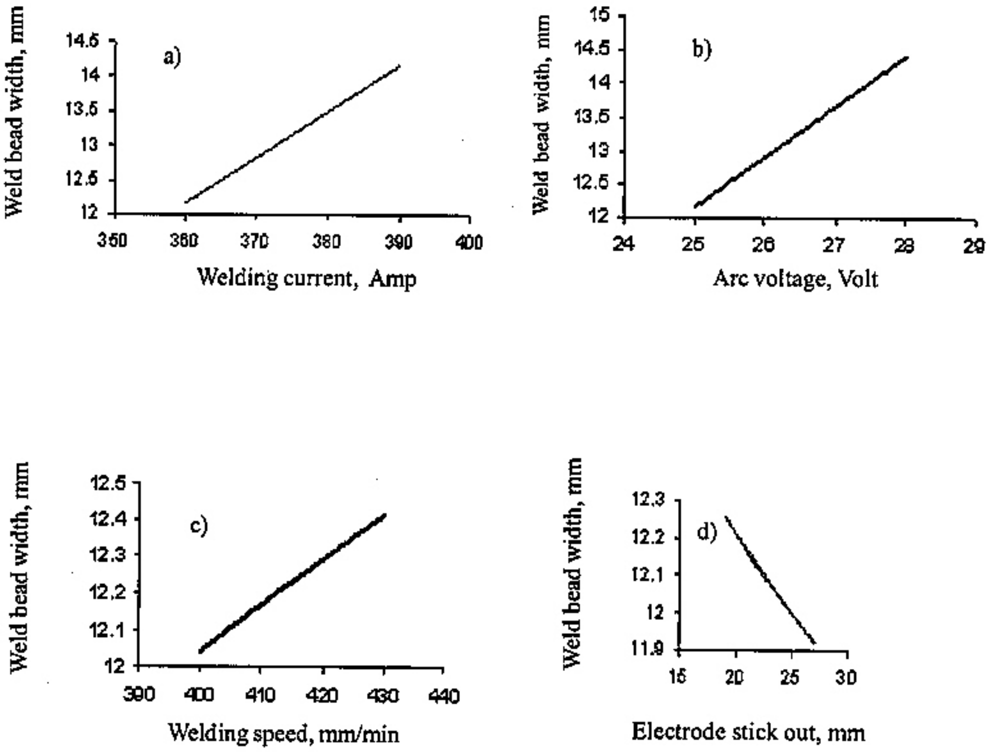

Fig. 1— Relationships between weld bead width from MRA versus : a); welding current; b) arc voltage; c) welding speed and d) electrode stick

Table 1—Weld bead width observed on different trials of experiment

Welding current A	Arc voltage V	Welding speed mm/min	Electrode stickout mm	Weld bead width mm
360	25	400	19	13.0
360	25	400	25	11.0
360	26	420	19	12.5
360	26	420	25	13.5
390	25	420	19	14.5
390	25	420	25	14.0
390	26	400	19	14.5
390	26	400	25	15.0

10^{-2} × electrode stick-out, mm). Validity of this equation is confirmed by calculating R^2-value (0.814), which has high coefficient of correlation.

Development of ANN for Weld Quality Prediction

Back propagation neural network (BPNN) is to map a set of input patterns to a corresponding set of output patterns by learning from a series of input – output data set. Then network performance is tested for patterns to predict appropriate outputs. In SAW, WBW is changed due to complicated welding conditions, and accurate mapping is needed to produce desired WBW according to welding parameters. Architecture of developed ANN (Fig. 2) is feed forward BPNN trained with Levenberg-Marquardt back propagation algorithm in MATLAB platform (Appendix 1). Number of samples for training and testing were 51 and 5 respectively. Learning function was gradient descent algorithm with momentum weight and bias learning function. Number of hidden layers and neurons were determined through a trial and error method, in order to accommodate converged error. Structure of proposed ANN is 4-12-9-1 (4 neurons in input layer, 12 neurons in 1st hidden layer, 9 neurons in 2nd hidden layer and 1 neuron in output layer) . With a learning rate of 0.55 and a momentum term of 0.9, ANN was trained for 10000 iterations. Error between desired and actual outputs was < 0.001 after training ANN.

Development of NHM for Weld Quality Prediction

Flowchart (Fig. 3) depicts stages of NHM for weld quality prediction. Topology of ANN model developed

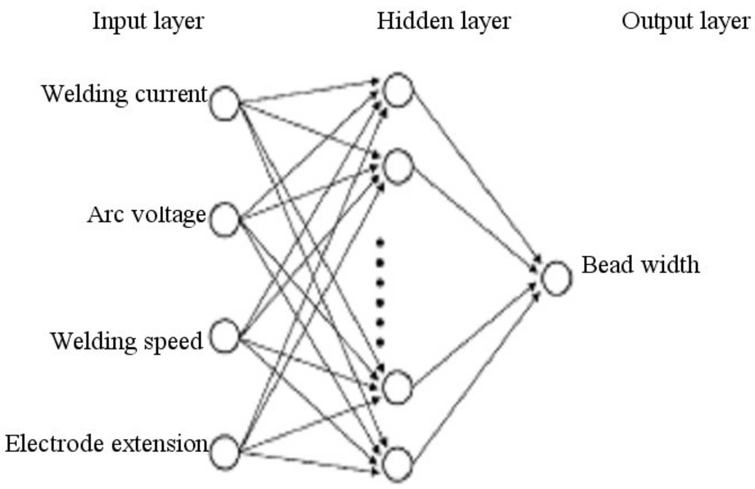

Fig. 2— ANN architecture

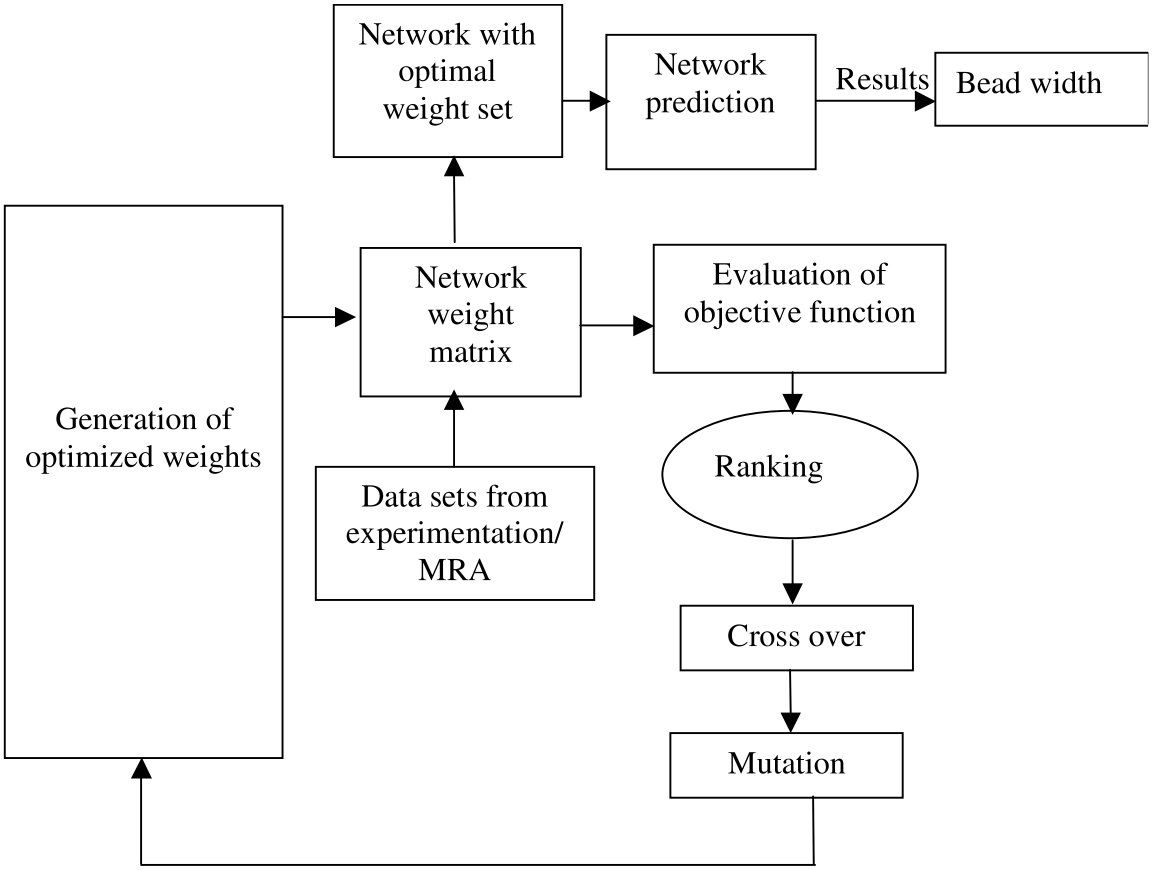
Network with optimal weight set
Network prediction
Results
Bead width
Generation of optimized weights
Network weight matrix
Evaluation of objective function
Ranking
Data sets from experimentation/ MRA
Cross over
Mutation

Fig. 3— Schematic diagram of proposed neuro hybrid model (NHM)

Table 2— Weld bead width predicted by MRA, ANN and neuro hybrid model (NHM)

Welding current A	Arc voltage V	Welding speed mm/min	Electrode extension mm	Weld bead width, mm			Error MRA ANN	Error MRA NHM
				MRA	ANN	NHM		
360	25	410	25	12.0007	12.0036	12.0011	0.0029	0.0004
360	26	420	25	12.6257	12.6244	12.6214	0.0013	0.0043
360	26	420	21	12.7925	12.7926	12.7922	0.0001	0.0003
370	25	410	25	12.6674	12.3703	12.4215	0.2971	0.2459
370	26	410	21	13.5842	13.9179	13.7491	0.3337	0.1649
370	26	420	25	13.2924	13.5806	13.332	0.2882	0.0396
380	25	400	19	13.7093	14.3760	13.962	0.6667	0.2527

Table 3—Results from confirmatory experiments

Welding current A	Arc voltage V	Welding speed mm/min	Electrode extension mm	Bead width, mm Experimentation	Bead width, mm NHM	Error %
390	26	410	19	14.7	14.900	-0.13
370	25	420	21	12.5	12.508	-0.06
380	26	400	25	14.0	14.209	-1.47
360	26	400	19	13.0	12.522	3.817

remained same. Weights of ANN were updated using GA to achieve optimum weight set. Data set required for training and testing NHM was obtained from experimental results and MRA. Values of different network parameters were as follows: number of training data sets, 43; number of testing data, 8; fitness function, mean squared error; population size, 40; maximum number of iterations, 250; selection, best half of the population; crossover probability, 80%; and mutation probability, 0.5%. Weight updation is explained as follows:

Encoding Weights

Weights and bias of network are represented in the form of chromosomes. Each chromosome represents one neural weight bias set. Initial population of chromosomes were randomly generated. Length of chromosome is number of links between neurons. For four input layer neurons, four hidden layer neurons and one output neuron, there were 20 links thus length of chromosome was 20 to correspond 20 weights.

Fitness Function

Fitness function for evaluating chromosome's performance estimates performance of a given ANN. Training set obtained through performing experiments (Table 1) was presented to ANN, and mean squared error was calculated. GA finds a set of weights that minimizes mean squared errors (difference between target and predicted WBW).

Ranking

Individuals (chromosome) in population were evaluated and ranked. Since number of individuals in each population was kept constant, for each new individual an old one having worst fitness value was discarded.

Selection

Selection of individuals for crossover and mutation was biased towards fittest chromosomes. Top half of ranked chromosomes were selected for production of new offspring. All above steps are repeated till stopping criteria are reached. When ANN reaches with a best chromosome, optimal weights are stored. NHM was trained and tested for different conditions. Results obtained from BPNN and NHM are compared (Table 2) with results from MRA.

Validation of NHM for Weld Quality Prediction

Confirmatory experiments are done to validate NHM to predict WBW (Table 3). Error (%) = [(Experimental value – Predicted value) / Predicted value] X 100. Error (%) between results predicted by NHM and experimental values were not found significant.

Results and Discussion

Experiments were conducted in SAW machine using Taguchi's principles. Results from experimentation were used to generate more sets of data from MRA to train proposed ANN. Time elapsed by NHM to predict weld quality was 32% less than that of BPNN. Computing machine used was Intel Pentium IV 1.8 GHz processor, 512 MB RAM and 80 GB Hard Disk Drive.

Conclusions

This paper presented development of welding quality predictor using neuro hybrid model. Taguchi method was successfully employed to model weld quality using L_8 orthogonal array.

Acknowledgement

Authors thank Ministry of Human Resource Development [MHRD], Govt of India for sponsoring experimental work under its R & D program.

Appendix 1— MATLAB code

Load input and output data sets
Set input and output range

Declare

```
IN = iprange, %declaring input range
OUT = oprange, %declaring output range
IP = input, %declaring input matrix
OP = output, %declaring output matrix
S1 = size(IP), %size of the input matrix
p = S1(1), %number of input parameters(ie no. of rows)
x = p
q = S1(2), % number of training sets(no. of columns)
S2 = size(OP), %size of the output matrix
y = S2(1), % number of output parameters
r = 2, % number of layers
h =14, % number of neurons in the hidden layer
G = 5000, % number of generation
P = 40, % number of population
```

Normalization of input data

```
For i=1:p
For j= 1:q
        IP(i,j) = 0.1+0.8*((IP(i,j)-min(IN(i,:)))/(max(IN(i,:))-
min(IN(i,:))))
End
```

Normalization of output data

```
For i= 1:y
For j= 1:q
        OP(i,j) = 0.1+0.8*((OP(i,j)-min(OUT(i,:)))/
(max(OUT(i,:))-min(OUT(i,:))))
End

M = (x*h)+(h*y)+h+y, %X = number of elements in a population
For i = 1:M
        R(1,i) = -1, %R = Range of values
        R(2,i) = 1
End

        W = cell(P,2), % weight intialization cell
        HOP = cell(P,2), %hidden layer output intialization
        chrom = initrp(P,R), %chromosome intialization
        chrom1 = chrom, % where chrom1 is intially generated
        chromosomes or wt. set
        n=1, % n = iteration number
```

Determining objective values for all populations

```
For j = 1:P
        W{j,1} = reshape((chrom(j,1:x*h)),h,x), % reshaping
        into first weight matrix
        W{j,2} = reshape((chrom(j,(1+(x*h)):(M-h-y))),y,h), %
        reshaping into second weight matrix
        K{j,1} = W{j,1}*IP
        b{j,1} = reshape((chrom(j,(1+(x*h)+(h*y)):(M-
        y))),h,1), % reshaping into first layer bias        matrix
        b{j,2} = reshape((chrom(j,(1+(M-y)):M)),n,1), %
        reshaping into second layer bias matrix
For r=1:q
        b1 = K{j,1}(:,r)+b{j,1};
        K{j,1}(:,r) = b1;
End

        HOP{j,1} = tansig(K{j,1}), % output data from hidden
        layer
        K{j,2} = W{j,2}*HOP{j,1}
For r=1:q
        bias2 = K{j,2}(:,r)+b{j,2}
        K{j,2}(:,r) = bias2
        HOP{j,2} = purelin(K{j,2}); % output data from
        output layer
```

Calculation of objective function (mean square error)

```
objV(j,1) = (sum(sum((OP-HOP{j,2}).^2))/q;
For i = 1:G
        minerror(i) = min(objV);
        gen = 1;
        while (objV(gen)~=minerror(i))
        gen = gen+1;
% disp('string of leasterror in generation')
        disp(n)
        disp(gen)
        disp(minerror(i))
WLE(i,:) = chrom(gen,:); % where WLE (wt. matrix for least
        error)
```

Condition for stopping the iteration in between generation

```
if minerror(i) < 0.0001
break, end
if n==G
break, end
        n=n+1
        [xx,index] = sortrows(objV)
        newchromes = index(1:P/2,:)
        newchrom1 = chrom (newchromes,:)
        newchrom2 = xovshrs (newchrom1,0.8), % functions
        for crossover
        newchrom = mutate(newchrom2,0.08), %function for
        mutation
        newchrom = mutbga(newchrom2,R,[0.08 1 16])
For j = 1:P/2
```

```
        W{j,1} = reshape((newchrom(j,1:x*h)),h,x); %
        reshaping into first weight matrix
        W{j,2} = reshape((newchrom(j,(1+(x*h)):(M-h-
        y)),y,h);    % reshaping second weight  matrix K{j,1} =
        W{j,1}*IP;
        b{j,1} = reshape((newchrom(j,(1+()+(h*y)):(M-
        y))),h,1);% reshaping first layer matrix
        b{j,2} = reshape((newchrom(j,(1+(M-y)):M)),y,1);  %
        reshaping second layer bias matrix
For r=1:q
        bias1 = K{j,1}(:,r)+b{j,1}
        K{j,1}(:,r) = bias1
        HOP{j,1} = tansig(K{j,1});
        K{j,2} = W{j,2}*HOP{j,1};
For r=1:q
        bias2 = K{j,2}(:,r)+b{j,2};
        K{j,2}(:,r) = bias2;
        HOP{j,2} = purelin(K{j,2});
```

Calculation of mean squared error

```
objv2 (j,1) = (sum(sum((OP-HOP{j,2}).^2)))/q
chrom = [newchrom1;newchrom]
objV = [objv1;objv2];
% display of minimum error
for z = 1:n disp(minerror(z))
```

List minimum error of all generations

```
least = min(minerror(z))
        gen = 1
        while(minerror(gen)~=least)
        gen = gen + 1
End
disp('least error in generation no = ')
        disp(gen)
        disp('weight set  corresponding to least error is')
        wm = WLE(gen,:);% wm is the matrix with least error
```

References

1 Apps R L Gourd L M & Lelson K A, Effect of welding variables upon bead shape and size in submerged-arc welding, *Weld Met Fabr,* **31** (1963) 453-457.

2 Yang L J Chandel RS & Bibby M J, The effects of process variables on the bead width of submerged arc weld deposits, *J Mater Process Technol,* **29** (1992) 133-144.

3 Yang L J Chandel R S & Bibby M J, The effects of process variables on the bead height of submerged-arc weld deposits, *Canad Metallurg Quart,* **31** (1992) 289-297.

4 Yang L J Chandel R S & Bibby M J, The effects of process variables on the weld deposit area of submerged arc welds, *Welding J*, **72** (1993) 11s-18s.

5 Arminder Pal Singh & Rupinder Singh, Effects of welding variables on mechanical properties of FE410W B steel in SAW, *Cent Manuf Inst,* (2006) 25-28.

6 Kanjilal P Pal T K & Majumdar S K, Combined effect of flux and welding parameters on chemical composition and mechanical properties of submerged arc weld metal, *J Mater Process Technol,* **171** (2006) 223-231.

7 Serdar Karaoglu & Abdullah Secgin, Sensitivity analysis of SAW process parameters, *J Mater Process Technol,* **202** (2008) 500-507.

8 Revaandra J & Parmar R S, Mathematical model to predict weld bead geometry for flux cored welding, *Metal Construct*, **19** (1987) 31R-35R.

9 Chandel R S, Mathematical modeling of melting rates for submerged arc welding, *Weld J*, **66** (1987) 135s-140s.

10 Dilthey U, Reichel T & Heidrich J, Mathematical modelling of welding phenomena 3, *Inst Mater, Lond,* (1997) 390-402.

11 Gupta V K & Parmar R S, Fractional factorial techniques to predict dimensions of the weld bead in automatic submerged arc welding, *J Inst Engr (India)*, **70** (1986) 67-71.

12 Maxwell J E, *Statistical process control for arc welding processes***,** M S Thesis, Vanderbilt University, Nashville, TN, 1990.

13 Yang L J & Chandel R S, An analysis of curvilinear regression equations for modeling the submerged-arc welding process, *J Mater Process Technol,* **37** (1993) 601-611.

14 Yang L J, Bibby M J & Chandel R S, Linear regression equations for modeling the submerged arc welds, *J Mater Process Technol,* **39** (1993) 33-42.

15 Gunaraj V & Murugan N, Application of response surface methodology for predicting weld bead quality in submerged arc welding of pipes *J Mater Process Technol,* **88** (1999) 266-275.

16 Lee J L& Rhee S, Prediction of process parameters for gas metal arc welding by multiple regression analysis, *Proc Inst Mech Engg – B,* **214** (2000) 443-449.

17 Edwin Raja Dhas J & S Kumanan, Determination of SAW process parameters using Taguchi method and regression analysis, *Indian J Eng Mater Sci,* **14** (2007) 103-111,.

18 Hang R K & Kang Y L, Application of Taguchi method to determine hybrid welding conditions of aluminium alloy*, J Sci Ind Res,* 68 (2009) 296-300.

19 Zhang H C & Huang S H, Application of neural networks in manufacturing: a state of the art survey, *Int J Prod Res,* **33** (1995) 705-728.

20 Kim I S , Son J S , Park C E, Lee C W, Yarlagadda & Prasad K D V, A study on prediction of bead height in robotic arc welding using a neural network, *J Mater Process Technol*, **131** (2002) 229-234.

21 Abdulkadir C, Akif K, Ahmet E, Ibrahim M & Guzelbey H, Neural network modeling of arc spot welding, *J Mater Process Technol*, **202** (2008) 137-144.

22 Goonatilake S & Khebbal S, *Intelligent Hybrid System* (John Wiley & Sons, Chichester) 1995, 1-15.

23 Philipp K, *Combining genetic algorithms and neural networks: the encoding problem,* M S Thesis, The University of Tennessee, Knoxville, 1994.

24 Edwin Raja Dhas J & Kumanan S, ANFIS for prediction of weld bead width in SAW Process, *J Sci Ind Res,* **66** (2007) 335-338.