# *Application of an Artificial Neural Network in friction stir welding*


## Abstract:

The artificial neural network (ANN) can be used as a model for friction stir welding (FSW), predicting the correlation between FSW parameters and weld properties. ANN for predicting properties of welded plates by FSW is explained and previous works in this field are reviewed. Furthermore, some examples of applications of ANN in FSW are presented. The development of sound joints between materials is of great importance in FSW. One of the most challenging problems is choosing appropriate welding parameters in order to produce a sound joint. Traditionally, a time-consuming trial and error development was carried out to determine welding parameters. Additionally, an optimized welding parameters combination was not achieved, because welds can be fabricated with very different ideal welding parameters. Different optimization techniques can be employed to determine the optimized output parameters by specifying the relation between the input and output variables. Applications of optimization methods in FSW are explained, and basic principles of these methods, such as Taguchi, genetic optimization, and multi-objective optimization methods are described

## Methodology:

In this section, an ANN model is developed to simulate the correlation between the FSW parameters and mechanical properties of a butt joint, as a function of weld and rotational speeds. The summary of experimental results is presented in table. As shown in the table, the inputs are the rotational speed and the traverse speed, and the output is hardness.

| Traverse Speed | Rotational Speed | Hardness |
|---|---|---|
| 20 | 1400 | 92.3 |
| 20 | 1200 | 94.2 |
| 20 | 900 | 93.7 |
| 20 | 700 | 96.7 |
| 20 | 565 | 97.9 |
| 36 | 1400 | 90.23 |
| 36 | 1200 | 92.1 |
| 36 | 900 | 94.1 |
| 36 | 700 | 95.25 |
| 36 | 565 | 97.3 |
| 63 | 1400 | 89 |
| 63 | 1200 | 90.1 |
| 63 | 900 | 91 |
| 63 | 700 | 90.9 |
| 63 | 565 | 95.2 |
| 96 | 1400 | 87.21 |
| 96 | 1200 | 88.23 |
| 96 | 900 | 90.1 |
| 96 | 700 | 91.67 |
| 96 | 565 | 92.89 |

## Results obtained:

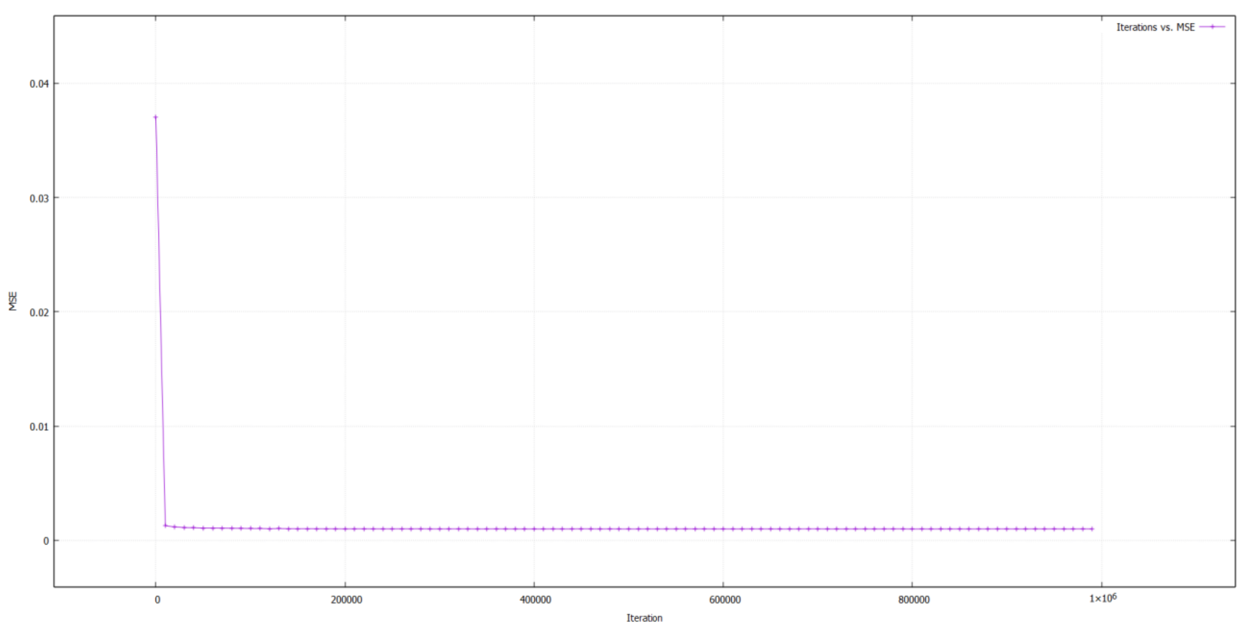
| *Pattern* | *Input-1* | *Input-2* | *Target value* | *Output value* |
|---|---|---|---|---|
| 1 | 20.000000 | 1400.000000 | 92.300000 | 92.660195 |
| 2 | 20.000000 | 1200.000000 | 94.200000 | 93.463885 |
| 3 | 20.000000 | 900.000000 | 93.700000 | 94.553956 |
| 4 | 20.000000 | 700.000000 | 96.700000 | 96.791142 |
| 5 | 20.000000 | 565.000000 | 97.900000 | 97.709624 |
| 6 | 36.000000 | 1400.000000 | 90.230000 | 91.152492 |
| 7 | 36.000000 | 1200.000000 | 92.100000 | 91.961583 |
| 8 | 36.000000 | 900.000000 | 94.100000 | 93.378754 |
| 9 | 36.000000 | 700.000000 | 95.250000 | 95.125998 |
| 10 | 36.000000 | 565.000000 | 97.300000 | 97.535878 |
| 11 | 63.000000 | 1400.000000 | 89.000000 | 88.894383 |
| 12 | 63.000000 | 1200.000000 | 90.100000 | 89.417191 |
| 13 | 63.000000 | 900.000000 | 91.000000 | 90.770963 |
| 14 | 63.000000 | 700.000000 | 90.900000 | 91.886453 |
| 15 | 63.000000 | 565.000000 | 95.200000 | 95.172705 |
| 16 | 96.000000 | 1400.000000 | 87.210000 | 87.872754 |
| 17 | 96.000000 | 1200.000000 | 88.230000 | 88.046611 |

The table represents the training pattern followed to train the ANN.

After training, the ANN is tested on 3 more patterns as followed:

| **Pattern** | **Input-1** | **Input-2** | **Target value** | **Output value** | **Error in prediction** |
|---|---|---|---|---|---|
| 18 | 96.000000 | 900.000000 | 90.100000 | 90.106252 | 0.006252 |
| 19 | 96.000000 | 700.000000 | 91.670000 | 92.404469 | 0.734469 |
| 20 | 96.000000 | 565.000000 | 92.890000 | 92.877978 | 0.012022 |

***The variation of Mean Squared Error (MSE) with the number of iteration is depicted below:***



## Observation & conclusion:

The optimum number of training patterns is calculated using the formula:

$$N_h = \frac{N_s}{\alpha * (N_i + N_o)}$$

Where, $N_h$ is the optimal number of hidden neurons,

$N_i$ is the number of input neurons,

$N_o$ is the number of output patterns,

$\alpha$ is arbitrary scaling factor ranging from 2 to 10

The error in prediction (MSE) turned out to be 0.049494 during the testing phase.

The optimal number of hidden neurons turned out to be 2 and it was implemented in the code of the aforesaid Neural Network's training.