Estimation of Contact Tip to Work Distance (CTWD) using Artificial Neural Network (ANN) in GMAW

Fuad Mahfudianto[1,*], *Eakkachai* Warinsiriruk[1], and *Sutep* Joy-A-Ka[2]

[1]Department of Industrial engineering, Faculty of Engineering, Mahidol Engineering, Thailand
[2]Material Properties and Failure Analysis Laboratory, Material Properties Analysis and Development Centre, Thailand Institute of Scientific and Technological Research, Thailand

Abstract. A method for optimizing monitoring by using *Artificial Neural Network* (ANN) technique was proposed based on instability of arc voltage signal and welding current signal of *solid wire electrode* (GMAW). This technique is not only for effective process modeling, but also to illustrate the correlation between the input and output parameters responses. The algorithms of monitoring were developed in *time domain* by carrying out the *Moving Average* (M.A) and *Root Mean Square* (RMS) based on the welding experiment parameters such as *travel speed, thickness of specimen, feeding speed, and wire electrode diameter* to detect and estimate with a satisfactory sample size. Experiment data was divided into three subsets: *train (70%)*, *validation (15%)*, and *test (15%)*. *Error back-propagation of Levenberg-Marquardt algorithm* was used to train for this algorithm. The proposed algorithms on this paper were used to estimate the variety the *Contact Tip to Work Distance* (CTWD) through *Mean Square Error* (MSE). Based on the results, the algorithms have shown that be able to detect changes in CTWD automatically and real time with takes 0.147 seconds (*MSE* 0.0087).

1 Introduction

Solid wire electrode (GMAW) is known as the vital need of industry with proven nearly 70% of the world installations are concentrated in the five leading industrial nations *Japan, US, Germany, Italy, and France* [3].
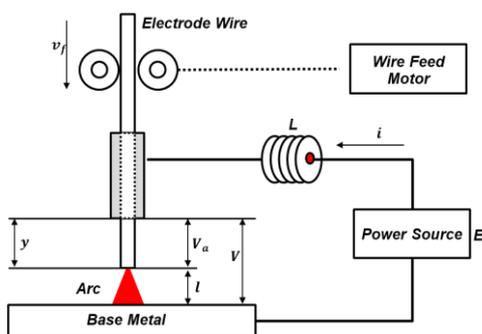

Fig. 1. Welding circuit.

Based on the power sources characteristics, *solid wire* is known as constant voltage power sources which can be defined as power sources has the characterized by a static voltage (*due to fluctuations in arc length*) causes significant change in welding current (*Figure 1*) [4]. By this phenomenon, it is important to control arc length to obtain the high quality welding [1]. Since there are some disturbance, such as *unstable arc*, during the welding, the measurement and *estimation of a contact tip to work distance of the electrode wire and base metal* with

proportion *arc voltage and welding current* may be difficult. To tackle of these problems, the realizing of estimating weld system in real-time by providing *low cost and fast response* becomes an important requirement a nowadays. Much effort has been contributed for estimation weld system with different types of strategy by providing several intelligent evaluating methods, such as *fuzzy logic* [5] and *genetic algorithm* [6], and *artificial neural network*, [1] are utilized to resolve the intelligence of arc welding in terms of estimation to obtain a good quality of the weld. Aside from the intelligent methods, a variety of control commercial product has been developed in this respect, even though a good physical product still has expensive categories in some areas (*Figure 2*).
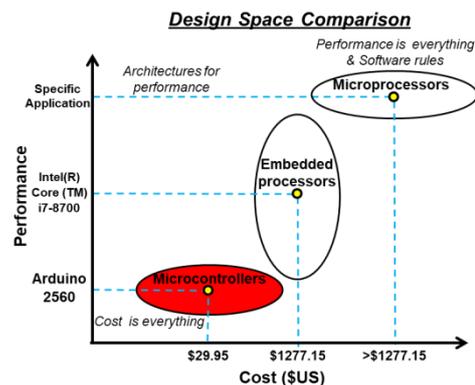

Fig. 2. Correlation between performance and cost in different design space comparison.

---

*
Corresponding author: fuadengine@gmail.com

YAMANE, et al. [1] have been developed an estimation of welding voltage using a neural network in GMA welding. In their study investigation have propose to confirm the validity of system to obtain a good welding quality in spite of an arc fluctuation by estimating CTWD with arc voltage signal.

Yoshiro, et al. [2] has been developed a design of neural network based FRIT PID controller and applications. In their research, the neural network and PID control have been applied in their strategy. However, because of both using the parameter voltage signal only, the high accuracy information become consideration. Added, many PID (*proportional-integral-derivative*) control algorithm calculate their PID parameters based on the estimates by LSM (*least squares method*), it is difficult to employ for the system, especially in adapting for nonlinearity, strong coupling, and uncertainty of complex multivariable nonlinear systems [7], which normally is used for the linear mathematical model [2]. This because the PID controller is very difficult to generate the reference table. On the other words, the generation of this table is required *cat-an-try* for each process [2].

In this respect, the demonstration more advantages when it has the simple device in terms of acquiring real-time signal (*i.e with a satisfactory sample size*) which is not easy to be affected by complex welding environment, it also has more stability, and robustness. Based on the variety of microcontroller existing, the Arduino microcontroller has become popular because of its advantage; such as *save time and equipment cost*. Besides of that, the Arduino is a microcontroller board founded on easy to use *open hardware and software*, which nowadays is used in many areas; such as education activities, electronic lab, and even welding lab [8]-[12]. Therefore, in this research has proposed to estimate of contact tip to work distance (CTWD) using artificial neural network (ANN) in GMA welding on a very inexpensive and low end microcontroller.

2 Experimental system

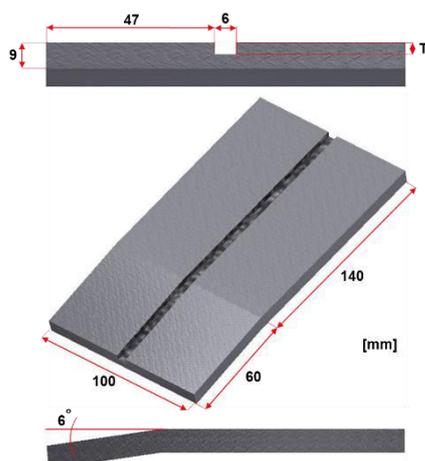Carbon steel JIS 3101 SS400 with dimension 100 mm x

Fig. 3. Base metal.

200 mm and a thickness of 9 mm were selected as the base metal (T=1, 2, 4, 6 mm). A 1.2-mm-diameter-YM-28-carbon-steel was used as the welding wire electrode. The base metal and chemical composition of the base metal can be seen in figure 3 and table 1, respectively.

Table 1. Carbon steel JIS G3101 SS400 (wt)%)

C	Mn	Si	S	P	Wire Electrode
0.14	0.80	0.15	0.035	0.035	
Max	**Max**	**Max**	-	-	**YM-28**
0.20	1.00	0.30	-	-	**(Ø1.2)**

The experiment of this study which consists of four parts: *real-time monitoring, real-time signal processing, welding system (DCEP), and weld process control (actuator)*. The half-fractional central composite design was performed which requires 32 experimental runs. MINITAB was used to setup the design matrix.

During welding, the mock-up specimen (*carbon steel*) is moved by speed control motor, whilst the torch of the welding machine is assembled with actuator of the system. To gain the signal for CTWD estimation, two kinds of the welding signals have been collected, such as *arc voltage and welding current,* respectively. Arc voltage and welding current are obtained from the positive and negative of the machine welding and *Hall Current sensor.* However, because of the machine welding voltage too large which can be destroying the electronic component, the *shunt resistor* has applied to tackle of these problems. After gaining of both signals, the signals are divided into two directions. The first signals go to the *Data Logger* and CPU which are used to confirm the signal processing by *offline path*. Second, the signals go to the *microcontroller* for CTWD estimation by *real time* signal processing. The output signals will change when the CTWD changes by mock-up specimen (*slope up or slope down*).

3 Artificial neural network model

3.1 Sampling data and hardware Implementing

Implementing *artificial neural network* on a 10-bit microcontroller with limited computing power and data
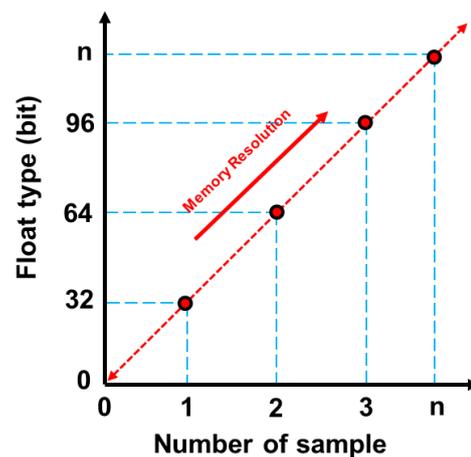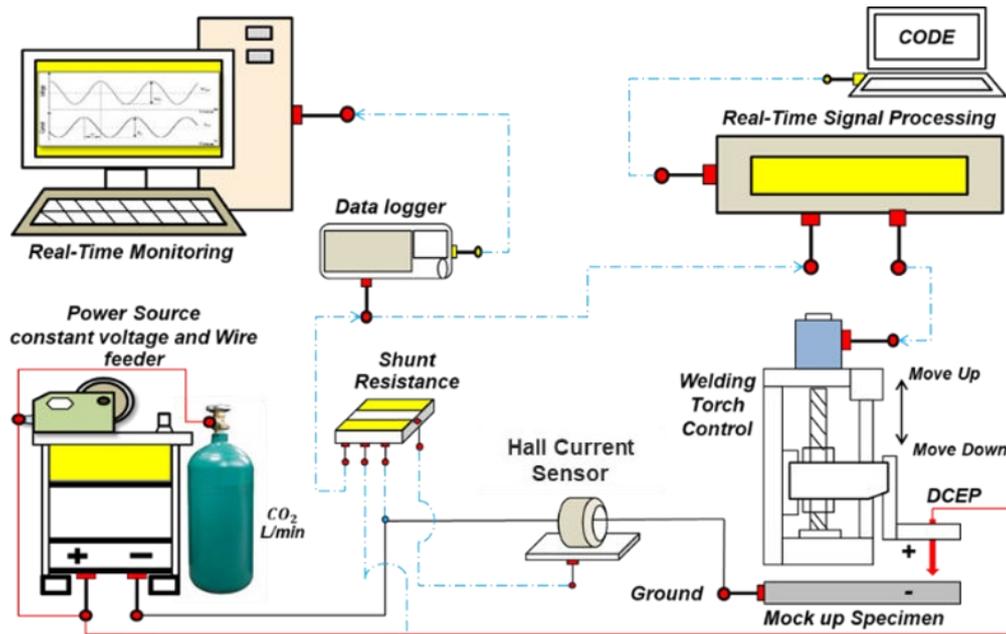

Fig. 4. Correlation between resolution and number of sample.

Fig. 5. Experiment setting.

present become several programming challenges.Then, the need of quickly execution as possible, reliable training, and more efficient were selected to develop the software (*embedded system*). By that reason, the hardware portability was also a motivating factor and more generic C implementation was also developed. *This because it is important to manually manage a very limited amount of data memory.* On the flip side, because of data collection using array path, the memory static of SRAM is used to store the sample data. In this context, aside has the goal to obtain a more velocity in learning process, the accuracy data (*provides decimal number*) by selecting data type also considered. Float data type was selected to collect every each data by giving 32 bits and more clearly can be seen in figure 4. An *Integrated Development Environment* (IDE) was developed exclusively for neural network calculations.

In practically, the clock frequency of Arduino is 16 MHz, a low frequency, but enough is used to collect data for estimating system by using *digital communication* (ADC and DAC). In case of sample collection in ADC, by increasing the sampling rate will increase the number of resolutions, it make the low memory and performance eventually the negative impact occurs such as in sampling time (*delay detection*). Aside from the sampling rate, sampling time mainly depend on cycle time of the program running in Arduino. Furthermore, the data sample through to DAC and the resolution of n-bit *Digital Analog Converter* (DAC) can be calculated as follows:

$$R_1 = 2^n \tag{1}$$

Table 2. Resolution in arduino microcontroller

No	ARDUINO Type	Size	Resolution
1	ARDUINO AT-MEGA 2560	10 Bits	1024

$$R_2 = \frac{V_0 FS}{2^n - 1} \tag{2}$$

$$A = \frac{V_0 FS}{(2^n - 1)2} \tag{3}$$

Table 3. Resolution and accuracy in arduino

No	ARDUINO Type	Full Scale Output Voltage	Resolution	Accuracy
1	ARDUINO AT-MEGA 2560	5 Volt	0.0048 /LSB	0.0024 V

Where, R_1 and R_2 are resolution, A is accuracy, $V_0 FS$ is full scale output voltage. Addition, the A/D and D/A converters in Arduino using *Serial Peripheral Interface* (SPI). On the other words, the sampling time communication depends on the number chips used.

3.2 Input and output artificial neural network

The three layer back propagation algorithm is set up to estimate the CTWD in this study. However, because of this method may cause "*over fitting*" problem which is characterized by slowing convergence, the *Levenberg–Marquardt* (LM) is used to overcome of this lack by providing the *lower mean square error* (MSE) than any other algorithm in case of function approximate problems [9]. Convergence is the eventual minimization of error between the desired (*target*) and computed output.

A 32 run experiments were available by d*ividing data into 3 subsets-train (70%)-validation (15%)-test (15%)*. The eight parameters are set up become input with one output parameter, more clearly can be displayed in figure 6. By seeing the figure 6, it can be understood that the neural network has been divided into 3 layers with technically terms are called *input layer, hidden layer, and output layer* [10].
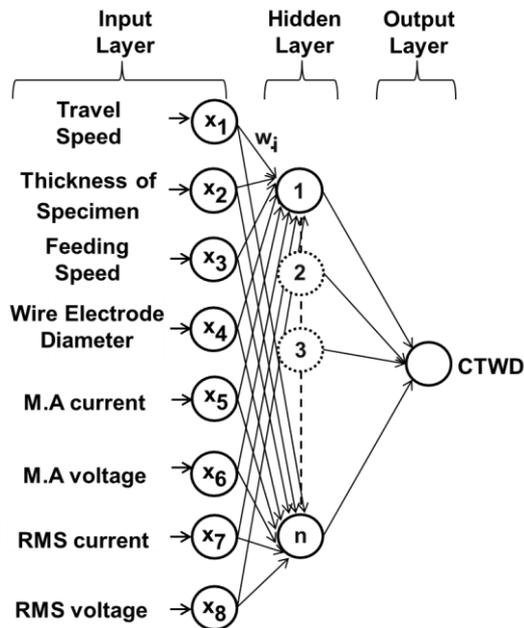
Fig. 6. Topology of ANN.

To estimate a CTWD at present time, this research figure out the total net input to each hidden layer neuron, then squash the total net input using an activation faction (*sigmoid function*) and repeat the process with output layer neurons. The characteristic of sigmoid function is produces outputs in the range 0 to 1 [11]. However, because of sigmoid function actually never reaches 0 or 1, the neural network input parameters need to be normalized before calculation. The weight sum I_i of all signal which are active at the input connection is employed as the input function.

$$I_i = \sum_j w_{ij} x_j \tag{4}$$

Where w_{ij} component represent the connecting weights between neuron j and neuron i, and the x_j component represent the signals at the connection concerned. The sigmoid function can be calculated as follows.

$$f(z) = \frac{1}{1+e^{-z}} \tag{5}$$

$$o_i = a_i = f(\sum_j w_{ij} x_j) \tag{6}$$

Where o_i the output of the neuron is, a_i, is its activation, x_j is identical to the output of preceding neuron with index j of the observed element.

After getting network output by propagating through the network in forward direction, the comparison of the network output with a target output value generating an error for each output neuron.

$$E = \frac{1}{2}\sum_i (z_i - o_i)^2 \tag{7}$$

Where z_i are the target output. The effect of the adaptation of the weights can be seen in the algorithm as follows:

$$\Delta w_{ij} = w_{ij}(k+1) - w_{ij}(k) = -a\frac{\partial E}{\partial w_{ij}} \tag{8}$$

Where a is defined as the learning rate, w_{ij}, is the connection weights.

$$\Delta w_{ij} = a\delta_i x_i \tag{9}$$

Where the local error of hidden element is calculated by:

$$\delta_i = f'(I_i) \sum_j \delta_i w_{ij} \tag{10}$$

δ_i represent the errors of the elements in the following layer, whilst w_{ij} represents the connection weights for these elements. Then, the error of the output layer is obtained by given:

$$\delta_i = f'(I_i)(z_i - o_i) \tag{11}$$

This error is first of all calculated and the back propagated into the hidden layer until the input layer of each element is reached which located before the output layer. To adjust the speed of convergence and extinguish the danger "*getting stuck*", a momentum and learning rate are introduced in this algorithm by:

$$\Delta w_{ij}(k) = a(k)\delta_i x_j + \mu \Delta w_{ij}(k-1) \tag{12}$$

Where Δw_{ij} (k) is the kth connection weight variation in the algorithm, $\Delta w_{ij}(k-1)$ is $(k-1)$th connection weight variation, μ is momentum [10].

4. Result and discussion

4.1 Optimization number neuron of hidden layer

In real-time signal processing, *the computation time, memory storage, and mean square error* (MSE) by consideration the number neuron in hidden layer become vital point to obtain the best performance of a neural network. *This because if the number neurons is too small, the network cannot learn the entire information well and will result in a big error* [10] *and if the number of the neurons is too big it will dramatically increase the computational time and may result in over-train* [10]-[13].
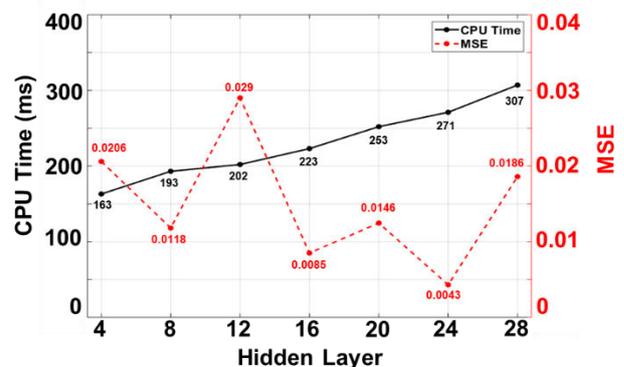
Fig. 7. Effect number neuron in hidden layer to the total time and MSE.
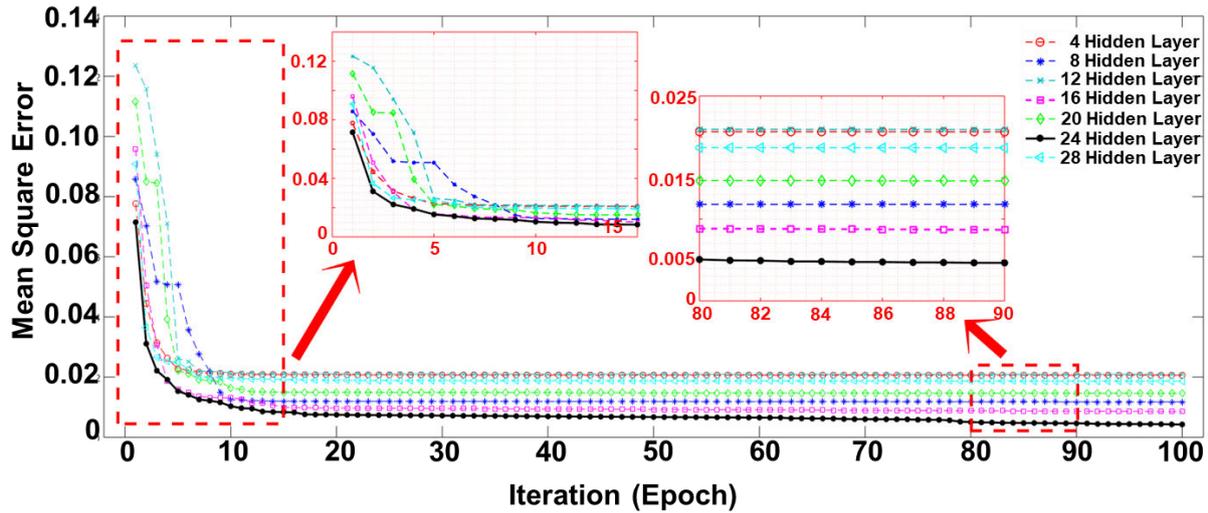
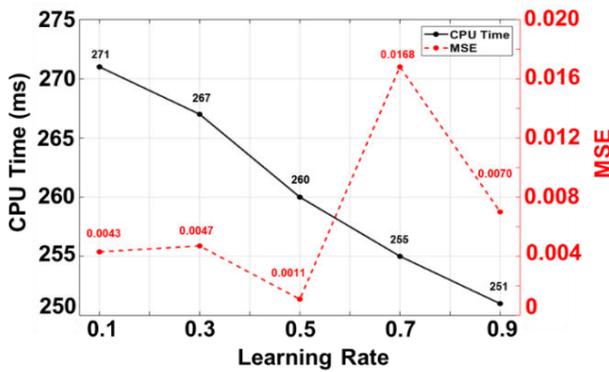**Fig. 8.** Convergence process in difference number neuron in hidden.

Fig. 9. Effect learning rate to the total time and MSE.

Therefore, several combinations was tried out to gain an optimal number of hidden layer in neural network. By consideration the three of them, the best number neuron of ANN finalized was 8-24-1 (*8 neuron in the input layer, 24 neuron in hidden layer, and 1 neuron in output layer*) which is decided with obtain mean square error 0.0043 and takes time 271 milliseconds in 100 iterations for selecting in CPU the result more clearly can be seen in figure 7. Aside from the CPU time and mean square error, the figure 8 has shown the smooth convergence by increasing and decreasing weight to minimize of error between target value (*CTWD Measurement*) and output value (*CTWD Estimation*) of ANN in every iteration.

A key point to keep in mind that, the computation time of the algorithm can be reduced by adjusting the varying learning rate with Eq. 8-12. However, to avoid *over learning* and *overshooting*, the learning rate has been selected with 0.5 and the computation time decrease from 271 to 260 milliseconds while the mean square error has shown 0.0011 in 100 iterations, more clearly can be seen in the figure 9.

The commercial software MATLAB is used to develop a neural network in offline path. Furthermore, before a neural network is used to estimate, the training,

validation, and test set are used to ensure over fitting and under fitting were avoided.

4.2 CTWD estimation result

To estimate the CTWD, the training data set is used to teach the classifier to determine the target value. Then, the un-training (*during training*) results were compared with after training results to check the estimation performance.
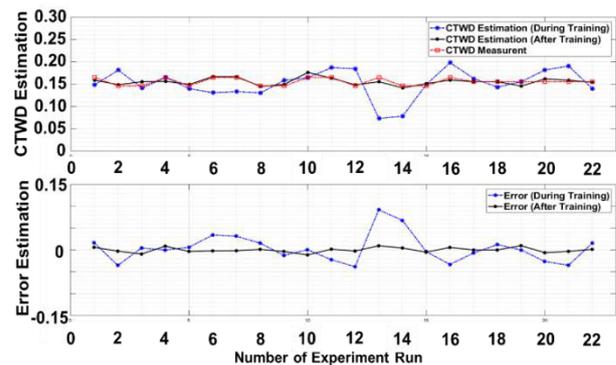

Fig. 10. CTWD and error estimation.

Table 4. CTWD measurement and CTWD estimation.

No	CTWD Measurement (Real [mm])	CTWD Estimation (Normalization)
1	10	0.134
2	12.5	0.143
3	15	0.153
4	17.5	0.162
5	20	0.172

Based on the results, *after training results* has shown the mean square error 0.0013 which takes time 261 millisecond in 100 iterations, whilst the un-training has shown the mean square error 0.028 which takes time 265 millisecond and more clearly can be seen in the figure 10. Furthermore, the validation and test data set are used to evaluate the prediction on unseen data to confirm the

classification capability of the developed model, respectively. Based on the result the validation value and test has shown mean square error 0.0023 and 0.0031 with take time 269 and 262 milliseconds in 100 iterations, respectively.

To confirm that parameters of a neural network in CPU computation, the Arduino microcontroller is implemented. Based on the training computation time results, the Arduino microcontroller has shown 35700 millisecond in 100 iterations, while the mean square error 0.0099

Table 5. CPU and microcontroller comparison in training.

No	Parameter	CPU	Microcontroller
1	Time (ms)	0.123	17
2	Mean Square Error	0.0011	0.0099

Because the input of a neural network is collected by using real-time signal processing, the increasing computational cost of the time will be occurs during estimating. This because the algorithm will through the *low pass filter, moving average and root mean square of both signal, arc voltage and welding current, respectively*. Based on the estimating time computation results takes 147 milliseconds in one detection (*MSE*: 0.0087). Since the estimation result was good agreement with the experimental results by consideration limited computing power and data present in microcontroller, a neural network was useful as the model which grasps the action of the welding.

5 Conclusion

The CTWD has been estimated by using a neural network. The results of the research can be concluded as follows:
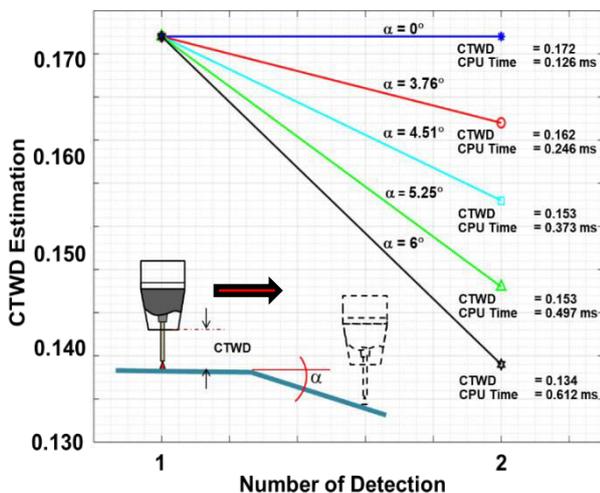
Fig. 11. CTWD estimation in different alpha.

1. The optimization number neuron of hidden layer has been developed with the best number neuron of ANN finalized was 8-24-1 by considering the *computation time*, *memory storage, and mean square* (*MSE*).

2. CTWD has been estimated by using the neural network in which the CTWD estimation (0.172; 0.162; 0.153; 0.143; 134) is kept constant according parameter input setting between the base metal and the tip (*Figure 11*). The minimum time response for CTWD changing (*sensitivity*) has shown in alpha (α = 3.76°, 0.246 millisecond), whilst the maximum time response in alpha (α = 6°, 0.612 millisecond).

References

1. YAMANE, et. al., "Estimation of welding voltage using neural network in GMA Welding". P. 27s-31s. (2009)
2. Yoshiro, et. al. "Design of neural network based FRIT PID controller and its application". 11th IFAC international workshop on adaptation and learning in control and signal processing. (2013)
3. Weman, K,. "MIG welding guide". Boca Raton. CRC Press.
4. Scotti, et. al., "A scientific application oriented classification for metal transfer modes in GMA Welding". Journal of materials processing technology. 6, 1406-1413. (2012)
5. Naso D & Turchiano B., "A fuzzy logic based optical sensor for online weld defect detection". IEEE transaction on industrial informatics, 1. 259-273. (2005)
6. Wand, J.F., "Feature extraction in welding penetration monitoring with arc sound signals". Journal of engineering manufacture, 225. 1683-1691. (2011)
7. G.-Q. Zeng et al,. "Adaptive population extremal optimization-based PID neural network for multivariable nonlinear control system". Swarm and evolutionary computation. (2018)
8. Siriparapu et. al,. Development of low-cost plotter for educational; purposes using Arduino". IOP Conf. Series. Material science and engineering. (2017)
9. M. ahmadzadeh, et. al., "Prediction of residual stresses in gas arc welding by back-propagation neural network". NDT&E International 52, 136-143. (2012)
10. C.S WI et. al,. "A neural network for weld penetration control in gas tungsten arc welding". ACTA Metal. Sin. (Engl. Lett). Vol. 19. No. 1 pp27-33. (2006)
11. H. Taghavifar & A. Mardani,. "Application of artificial neural network for prediction of traction performance parameters". Journal of the Saudi society of agriculture sciences, 13, 35-43. (2014)
12. Luiz, I et. al,. "Low cost surface Electromyography signal amplifier based on Arduino microcontroller". International journal of electrical. Robotic, electronic, and communication, 8. 310-314. (2014)
13. D.A Patel & K.N. Jha,. "Neural network model for the prediction of safe work behavior in construction projects". J. Constr. Eng. Manage. 141.04014966. (2015)