Neural Network Predictive Controller for Improved Operational Efficiency of Shiroro Hydropower Plant

Ojo O. Adedayo, Gbadamosi S.L and Ale D.T

Abstract— The development of efficient models and controllers is central to better understanding and analysis of operational efficiency of modern hydropower plants. In this work, an intelligent Levenberg-Marquardt based Neural Network Predictive Controller (NNPC) was developed for Shiroro hydroelectric power station using actual data obtained from the plant operation. Results obtained after training and simulation of the system show that neural network technique serves as an efficient approach of designing hydroelectric power station models and controllers.

Index Terms— Neural Network Predictive Controller, hydropower plant, operational efficiency, Artificial Neural Network.

1 INTRODUCTION

Since the advent of electricity, the world has relied on it for its ever-growing energy needs for domestic and industrial applications. This increasing demand for electric energy has necessitated the design and implementation of robust and complex networks of power stations and distribution outlets. However, this trend comes with its inherent challenges; security, efficiency and stability. These challenges are even more evident in the aspect of rotor oscillations and difficulty in re-attaining stability after the introduction of disturbances.

To address these problems, Power System Stabilizers (PSS) have been employed. These PSSs are quite common but have a prominent shortcoming of yielding unsatisfactory performance especially when employed for multi-area applications. This necessitated yet another requirement for improvement. As Proportional Integral Derivative (PID) controllers became popular, they quickly found their use in power systems, this is partly due to their versatility and high reliability. However, the overall performances of power system PID controllers do drop significantly with varying degrees of input range and uncertainties.

The standard form of representing the PID control $u(t)$ is expressed as:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{de(t)}{dt} \quad (1)$$

Where the quantities $y(t)$, $n(t)$, $d(t)$ are the measurable output, the sensor noise, and the system disturbances respectively. While the parameters K_p, K_i and K_d are carefully selected values that models the plant in terms of settling and rise times, steady state error and overshoot in response to alterations in demand signal, and $e(t)$is the error in deviation from the desired reference output [1].

The search for power system stability and efficiency has turned the attention of researchers to various softcomputing and intelligent approaches [2]-[10] in the design of power system controllers. These include Artificial Neural Network (ANN) controllers which attempt to replicate the pattern matching capacity of the human brain [11], the fuzzy Inference System (FIS) which employs a gradually varying logic ranging from complete exclusion to complete inclusion as against crisp binary logic of 1s and 0s, and a host of other hybrid systems like Adaptive Neurofuzzy Inference System (ANFIS).

The unique advantage offered by the introduction of ANN (and other forms of softcomputing) controllers is the ability to eliminate the computational intricacies and explicit mathematical relations within its hidden layers and yet produce accurate results at its output. This property is one of the reasons that informed the idea of this work.

In this work, an intelligent Levenberg-Marquardt based Neural Network Predictive Controller (NNPC) was developed for Shiroro hydroelectric power station. This power station is situated in the Shiroro Gorge on the Kaduna River, approximately 60 km from Minna, capital of Niger State, in close proximity to Abuja, Nigeria's federal capital territory. Commissioned in 1990, the station has an installed capacity of 600MW [12].

- *Ojo O. Adedayo is a lecturer in the Department of Electronic Electrical & Computer Engineering, College of Engineering, Afe Babalola University, Km 5454, Afe Babalola way, Ado-Ekiti, Nigeria. Email:ojoao@abuad.edu.ng*
- *Gbadamosi S.L. is also a lecturer in the Department of Electronic Electrical & Computer Engineering, Afe Babalola University, Km 5454, Afe Babalola way, Ado-Ekiti, Nigeria. Email:gbadamosisl@abuad.edu.ng*
- *Ale D.T. is a lecturer in the Department of Electronic Electrical & Computer Engineering, College of Engineering, Afe Babalola University, Km 5454, Afe Babalola way, Ado-Ekiti, Nigeria. Email:aledt@abuad.edu.ng*

1.1 HYDROPOWER PLANT MODEL REPRESENTATION

Broadly, hydropower plant model representations are of two forms: Non-elastic (linear) models and elastic (nonlinear) models. The corresponding representative equations in both cases highly depend on the system complexity and can be classified using same criterion [13]. A fairly less complex linearized representation of the incremental torque with respect to the gate deviation for a hydropower station can be written as:

$$\frac{\Delta P_m(s)}{\Delta Z(s)} = \frac{(1 - sT_w)}{(1 + s0.5T_w)} \tag{2}$$

This equation is valid under several assumptions on hydraulic resistance, volume flow, water flow compressibility and gate position.

For the nonlinear model (with elastic water column) however, the input-output relationship with respect to incremental torque and guide vane positional changes is expressed as:

$$\frac{\Delta P_m(s)}{\Delta Z(s)} = \frac{a_{23} + (a_{11}a_{23} - a_{21}a_{13})\frac{T_w}{T_e} tanh(sT_e + F)}{1 + a_{11}\frac{T_w}{T_e} tanh(sT_e + F)} \tag{3}$$

Where F is the loss due to the friction within the hydraulic structure, a_{nm} are the partial derivatives of flow and torque with respect to head, speed and guide vane position respectively [13], and T_e is the travel time.

2. LEVENBERG-MARQUARDT (LM) ALGORITHM

Levenberg-Marquardt (LM) Algorithm second order algorithm that trains an artificial neural network by repeated update of network weights and biases by an optimization technique. The algorithm (which is essentially a trust region type of the Gauss-Newton method) is fast, efficient and often the most recommended choice in supervised training [14]. However, LM algorithm consumes relatively more memory and processing resources than GDA and RP algorithms.

The aim of the Levenberg Marquardt layer-by-layer backpropagation algorithm is to modify the weight and bias values of the different layers of the network in order to repeatedly minimize an error function *E(w)* until a specified minimum value or a stop criterion is reached [15]-[16].

$$E(\boldsymbol{w}) = \frac{1}{2}\sum_{p=1}^{P}\sum_{m=1}^{M} e_{pm}^2 \tag{4}$$

where P is the number of input-target training patterns, and M is the number of outputs.

For the ANN model framework for this work, there are [8×3] elements each in the input weight matrix and output weight matrix, [8×1] elements each in the input layer bias and output layer bias. These correspond to 64 elements. The network error e_{pm} is expressed as

$$e_{pm} = t_{pm} - o_{pm} \tag{5}$$

Where t_{pm} is the desired target, o_{pm} is the actual network output, m is the output node and p is the training pattern.
And,

$$\boldsymbol{o}^k = \boldsymbol{w}^k\boldsymbol{a}^{k-1} + \boldsymbol{b}^k \tag{6}$$

$$\boldsymbol{a}^k = f^k(\boldsymbol{o}^k) \tag{7}$$

where $\boldsymbol{a}^k$ and f^k are the outputs and activation function in the k^{th} layer. The adaptation in the weight is determined by:

$$\boldsymbol{\Delta w} = (\boldsymbol{J}^T\boldsymbol{J} + \mu\boldsymbol{I})^{-1}\boldsymbol{J}^T\boldsymbol{e} \tag{8}$$

where $\boldsymbol{J}$ is the Jacobian matrix as expressed by Equation 9, and $\boldsymbol{I}$ is an identity matrix.

$$\boldsymbol{J}(\boldsymbol{w}) = \begin{bmatrix} \frac{\partial e_{11}}{\partial w_1} & \frac{\partial e_{11}}{\partial w_2} & \cdots & \frac{\partial e_{11}}{\partial w_2} \\ \frac{\partial e_{12}}{\partial w_1} & \frac{\partial e_{12}}{\partial w_2} & \cdots & \frac{\partial e_{12}}{\partial w_N} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial e_{1M}}{\partial w_1} & \frac{\partial e_{1M}}{\partial w_2} & \cdots & \frac{\partial e_{1M}}{\partial w_N} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial e_{P1}}{\partial w_1} & \frac{\partial e_{P1}}{\partial w_2} & \cdots & \frac{\partial e_{P1}}{\partial w_N} \\ \frac{\partial e_{P2}}{\partial w_1} & \frac{\partial e_{P2}}{\partial w_2} & \cdots & \frac{\partial e_{P2}}{\partial w_N} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial e_{P2}}{\partial w_1} & \frac{\partial e_{PM}}{\partial w_2} & \cdots & \frac{\partial e_{PM}}{\partial w_N} \end{bmatrix}$$

$$e = \begin{bmatrix} e_{11} \\ e_{12} \\ \cdots \\ e_{1M} \\ \cdots \\ e_{P1} \\ e_{P2} \\ \cdots \\ e_{PM} \end{bmatrix} \tag{9}$$

The Levenberg Marquardt algorithm for the feedforward ANN proceeds thus:

Step 1: Supply the network with all the inputs and evaluate the corresponding outputs using Equations 6 and equation 7.

Step 2: Find the Jacobian matrix $\boldsymbol{J}(\boldsymbol{w})$ from Equation 9;

Step 3: Compute the increment vector $\boldsymbol{\Delta w}$ from Equation 8;

Step 4: Determine the new performance index $E(\boldsymbol{w} + \boldsymbol{\Delta w})$;

Step 5: If performance index in step 4 is smaller than the previous one (step 1), then multiply μ_n by $\frac{1}{\beta_n}$ and obtain the

ISSN 2229-5518

new weights $\boldsymbol{w} = \boldsymbol{w} + \boldsymbol{\Delta w}$ and proceed to step 6, where μ_n is the coefficient of the identity Matrix in Equation 8 and β_n is a real number ranging between 0 and 1. Else, multiply μ_n by β_n and proceed to step 3;

Step 6: Decrease n by 1 and repeat steps 1 to 5
Step 7: Repeat the entire steps until the stop criteria is met.

3. NEURAL NETWORK SYSTEM IDENTIFICATION

In order to model the NNPC in this work, the following steps were taken: firstly, a neural network architecture was designed to closely and satisfactorily model the forward dynamics of the generating power plant, the network parameters were optimized for optimum values of layers and connection weights, and the network was simulated and the performance was evaluated. To achieve these, a Single-Input Single-Output (SISO) neural network was designed for the Shiroro hydropower plant. The multilayered network consists of 60 neurons in the hidden layer and 1 neuron in the output layer and appropriate input and layer biases were applied. The tansig transfer function was applied to the first layer while the purelin transfer function was applied at the output layer. A total of 778 data samples were used in training the feedforward neural network which was divided into training, validation and testing data in the ordered ratio of 70:15:15 respectively.

In other to attain rapid convergence, elimination of local minimal convergence as well as obtaining satisfactory overall network accuracy, the Levenberg Marquardt (LM) algorithm was employed in training the NNPC. More details on the model are listed in table X. The deviation in value between the plant output and the neural network actual output was applied as a training signal for the neural network. This process is highlighted by the NNPC algorithm represented in Figure 1.

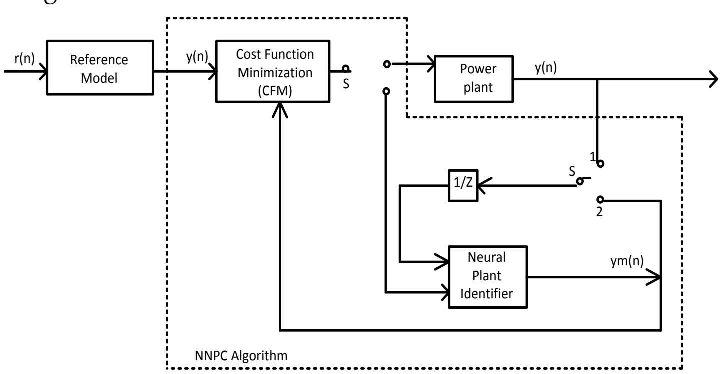

Figure 1: The predictive controller logic flow

For the control input for the plant, the algorithm involved (Figure 2) first generates a reference signal which is supplied to the plant performance prediction submodel. Then new control input is generated. This step is aimed at generating a set of control input that effectively minimizes the cost function. If having evaluated the cost function, the desired performance is still not attained in terms of cost function minimization, the algorithm returns to the plant performance prediction submodel and the steps are repeated until a satisfactory performance is deemed attained. Once this stage is reached, the plant is supplied with the plant control input.

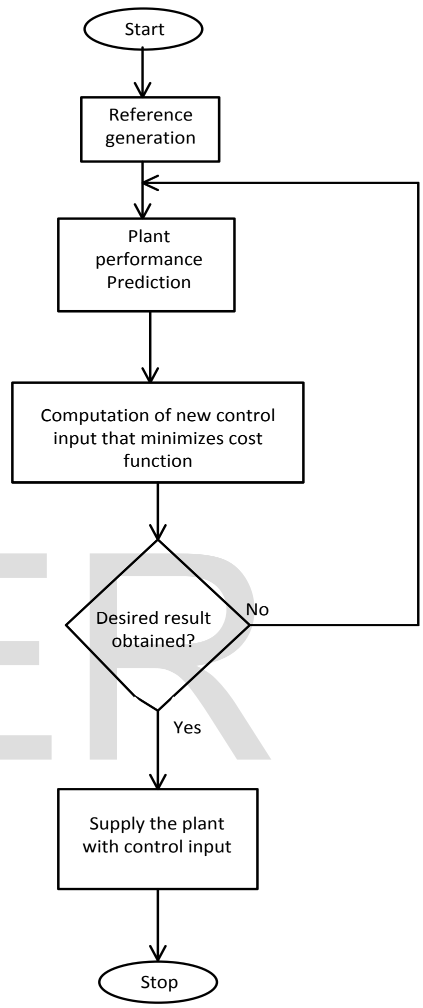

Figure 2: The flow chart for minimization of cost function.

4. RESULTS AND DISCUSSION

The essence of validation in neural network data training is to ensure a separate set of data is used to evaluate the performance of the network outside the training data set. This is operation reduces the chances of the network being overtrained by stopping the training when the minimum validation error set is achieved. If overtrained, the network could return a minimal error for the training data set but an unsatisfactory error margin for other data leading to poor generalization.

The essence of performance evaluation of a softcomputing network is to measure the degree of closeness of the network's output to the actual values as obtained from the physical phenomenon. In this work, the performance index employed is the Root Mean Square Error (RMSE) and the regression coefficient R. After the training phase, the neural network attained an R-value of 0.99374; this value indicates a very accurate neural network output in response to the training data. For the validation data set, a R-value of 0.99152 was observed as shown in Figure 4. This also represents a very high accurate network output in response to the validation data set impressed at the network input. In the same vein, the network yielded a R-value of 0.97122 when fed with the test data set which is a total of 116 plant sample data. Again, this shows that the network performed appreciably well for the

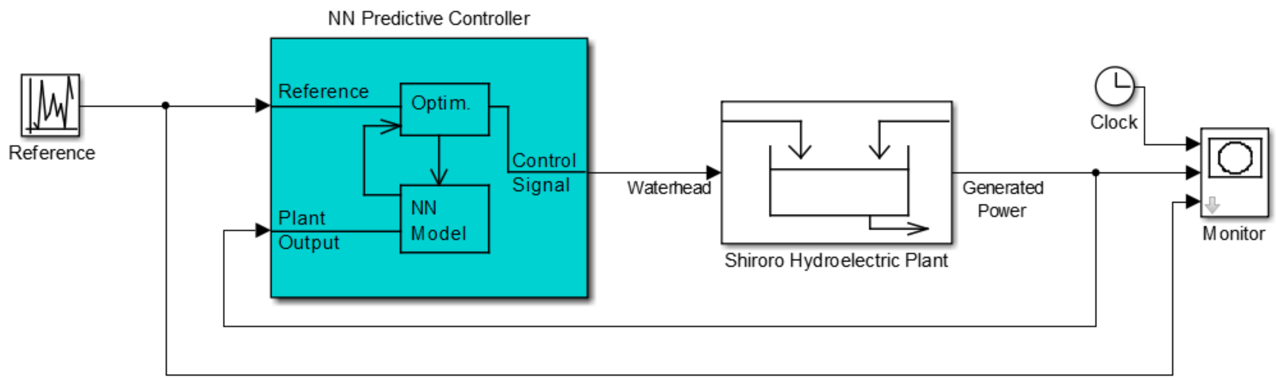

Figure 3: Overall model for the hydropower plant showing the NNPC

test data set. Examining the overall performance of the NNPC over the entire data range, it was observed that the network yielded a R-value of 0.99353. This value is quite close to unity and it indicates a very high overall network accuracy in modeling the Shiroro hydropower plant in terms of the water head as the input and the generated power as the output.

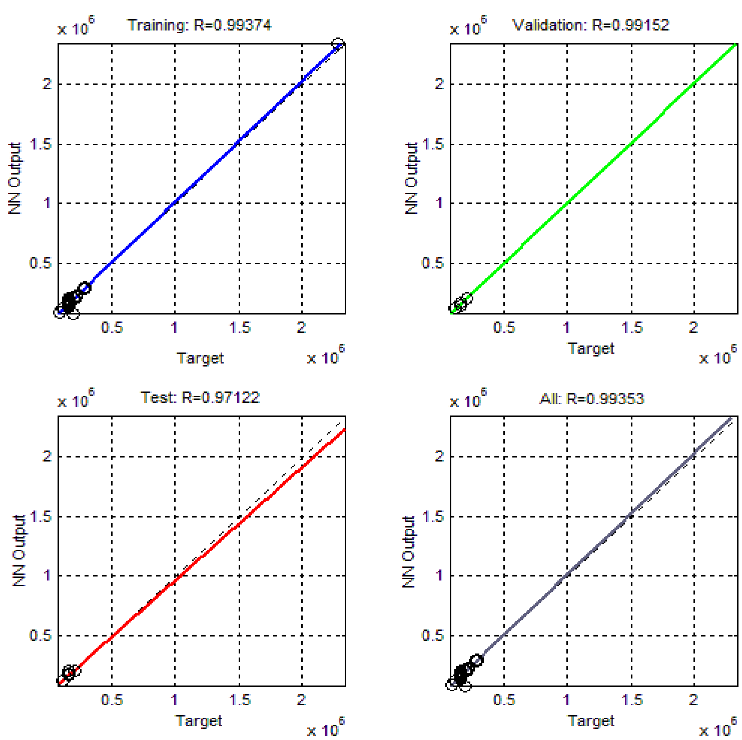

Figure 4: R-values for the neural network for different datasets

The neural network predictive controller that was implemented in this work employs a neural network model of Shiroro hydropower plant with the sole aim of forecasting, with an acceptable accuracy level, the future performance of the hydropower plant. The NNPC thereafter computes the control input values that effectively optimize the plant performance.

The first step taken in this model predictive control process was to determine the neural network plant model, which is a form of softcomputing approach to system identification. Next, the plant model was used by the controller to predict future performance of the plant. The results presented in this work were obtained on the platform of MATLAB's Neural Network Predictive Controller toolbox. The cost horizon (N2) was set at 7. This value controls the number of times over which the prediction errors are minimized. The control weighting factor (ρ) was set at 0.05, this is a function which multiplies the sum of squared control increments in the performance function. The search parameter (α) was set at 0.001and the control horizon (Nu), a factor which is a function of the number of times steps over which the control increments are minimized, was set at 2.

When the simulation was completed, the input, the plant output, the neural network output and the error are shown in figure 5. It was observed that the initial errors were high but the errors were reduced with time which highlights robustness and overall stability and accuracy of the system.

ISSN 2229-5518

Having been widely reported as successful alternative approaches to traditional control systems in several applications, the overall system performance in the current application underlines these findings.

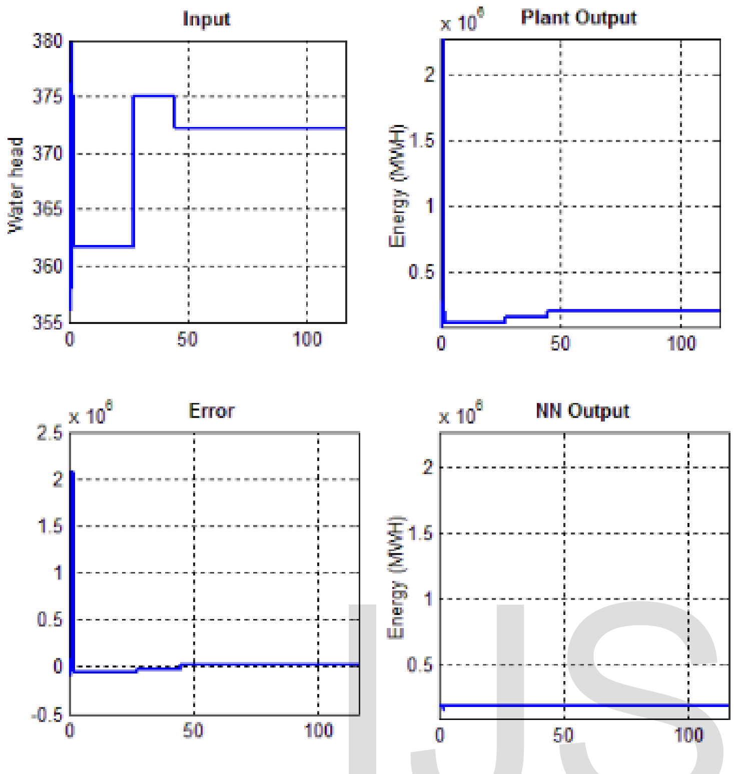

Figure 5: Plant simulation and NN output

5. CONCLUSION

The development of efficient controllers and models are crucial in the quest of better understanding, control and analysis of operational efficiency of modern hydropower plants. In this work, the robust pattern matching and learning ability of neural network based systems were demonstrated through the design, optimization and simulation of an intelligent Levenberg-Marquardt based Neural Network Predictive Controller (NNPC) for Shiroro hydroelectric power station using actual data obtained from the plant operation. Results obtained after training and simulation of the system show that neural network technique serves as an efficient technique of designing hydroelectric power stations.

REFERENCES

[1] Jurado, F., Ortega, M., & Cano, A. (2002). Neuro-fuzzy controller for gas turbine in biomass-based electric power plant. *Electric Power Systems Research*, *60*, 123–135.

[2] Cui, X. I. A. N. Z. H. O. N. G., & Shin, K. G. (1992). Contributed Paper Application of Neural Networks to Temperature Control in Thermal Power Plants, *5*(6), 527–538.

[3] Farahani, M., & Ganjefar, S. (2015). Neurocomputing An online trained fuzzy neural network controller to improve stability of power systems. *Neurocomputing*, *162*, 245–255.

[4] Henriques, J., Gil, P., Cardoso, A., Carvalho, P., & Dourado, A. (2010). Control Engineering Practice Adaptive neural output regulation control of a solar power plant. *Control Engineering Practice*, *18*(10), 1183–1196.

[5] Khajavi, M. N., & Menhaj, M. B. (2002). A neural network controller for load following operation of nuclear reactors, *29*, 751–760.

[6] Ko, H., Lee, K. Y., Kang, M., & Kim, H. (2008). Power quality control of an autonomous wind – diesel power system based on hybrid intelligent controller, *Neural Networks*, *21*(10), 1439–

[7] Sisworahardjo, N. S., El-sharkh, M. Y., & Alam, M. S. (2008). Neural network controller for microturbine power plants, *Electric Power Systems Research*, *78*, 1378–1384.

[8] Tiwari, S., Naresh, R., & Jha, R. (2011). Neural network predictive control of UPFC for improving transient stability performance of power system. *Applied Soft Computing Journal*, *11*(8), 4581–4590.

[9] Tofighi, M., Alizadeh, M., Ganjefar, S., & Alizadeh, M. (2015). Direct adaptive power system stabilizer design using fuzzy wavelet neural network with self-recurrent consequent part. *Applied Soft Computing Journal*, *28*, 514–

[10] Wang, J., Huang, W., Ma, G., & Chen, S. (2015). Electrical Power and Energy Systems An improved partheno genetic algorithm for multi-objective economic dispatch in cascaded hydropower systems. *International Journal of Electrical Power and Energy Systems*, *67*, 591–597.

[11] Adedayo, O. O., Oladipo, F., & Chris, I. (2015). Softcomputing Techniques for Improved Electroencephalogram Signal Analysis. *International Journal of Current Engineering and Research*, *5*(3), 2181–2186.

[12] Gbadamosi S.L., Ojo O. Adedayo, Nnaa L. (2015). Evaluation of operational efficiency of Shiroro hydro-electric plant in Nigeria. *International Journal of Science and Engineering Investigations*.

ISSN 2229-5518
[13] Nand, K., Saini R., & Singh S. (2007). A review on hydropower plant models and control. *Renewable & Sustainable Energy Reviews*. *11*, 776-796.

[14] Mukherjee, I., & Routroy, S. (2012). Comparing the performance of neural networks developed by using Levenberg–Marquardt and Quasi-Newton with the gradient descent algorithm for modelling a multiple response grinding process. *Expert Systems with Applications*, *39*(3), 2397–2407.

[15] He, S., Sepehri, N., & Unbehauen, R. (2001). Modifying Weights Layer-By-Layer with Levenberg- Marquardt Backpropagation Algorithm. *Intelligent Automation & Soft Computing*, *7*(4), 233–247.

[16] Wilamowski, B. M., & Yu, H. (2010). Improved computation for Levenberg-Marquardt training. *IEEE Transactions on Neural Networks*, *21*(6), 930–7.