© 2018 IJSRSET | Volume 4 | Issue 1 | Print ISSN: 2395-1990 | Online ISSN : 2394-4099
Themed Section : Engineering and Technology

Revamping of PID Controller via Artificial Intelligent Technique

Uchegbu Chinenye E*[1], Inyama Kelechi,[2] Ekwuribe Michael J[2]
*[1]Department of Electrical and Electronics Engineering, Abia State University, Uturu, Nigeria
[2]Department of Electrical and Electronics Engineering, Abia State Polytechnic, Aba, Nigeria

ABSTRACT

There are comparative advantages derived from the revamped proportional integral derivative PID Controller using Neural Network over the traditional controller. A few of them include increase in precision, cutback in time and uncomplicated hardware implementation. Ultimately, these advantages improve the control system in our industries. This paper recommends a non-linear control of stochastic differential equation to Neural Network matching. There was a validation, evaluation and comparison with other existing controllers. The essence is to get control systems suitable enough to achieve efficiency and improve on the performance of the traditional control systems. It is also to have control systems that reduce wastage and be more elastic in the level of conversion. More so, the initiative is to produce control systems that are competent in tracking set point change and discard load disturbance in our production industries. This paper is groundwork to devise a basic neural network and proportional integral derivative PID control system to model its operational distinctiveness for a class of straightforward process. Eventually, we recorded a laudable outcome by revamping the proportional integral derivative PID controller with Neural Network technique. The plant process control was also connected and the unique characteristics of the traditional proportional integral derivative PID were maintained. There was also an enhancement of PID controller. Finally, a rewarding result was recorded as the loss due to wastage encored by the process industries condensed significantly
Keywords: Neural network, PID controller, Mat-lab, ANN,BNN.

I. INTRODUCTION

The PID controller can be used to regulate events in a process such as weight, speed level, flow, temperature, pressure, et cetera. PID is extensively used in industrial control system as a three terms control loop controller. Similarly, it computes an error value as the margin between a measured process variable and a desired set point. There are varying ways to apply PID controllers which can be either as a stand-alone regulator or as a dispersed component of a control system. The PID controller offers varied alternatives over the dynamics of the system, which includes the pace at which it responds to input pointer. PID controller has a disadvantage which is performing poorly real-time in industrial workstation. A way to combat this weakness is to design a corresponding neural network model. This model is trained to loom a PID controller through many of input and output data pairs form PID controller. The outcome is an enhanced applications PID with hardware execution such as found in the Digital Signal Processing (DSP).

The objective of Neural Network (NN) supported PID is to reduce the difficulty in calculation and boost the poor real-time performance in traditional PID control algorithm. A corresponding neural network model is adopted in revamping a known PID controller. The PID controller and the corresponding neural network model which control the plant model are simulated with varied reference inputs respectively.

The industrial application controls are achieved using neural networks. The aim for applying them in such sense is that they can act as associative memories. In other words, they can learn from historical patterns. In addition, appropriate outputs can be realized from patterns dissimilar to the one used during the training stage. In this sense, they offer good quality outcomes. Neural networks can create input-output maps from records devoid of identified relation as well as adapted to partial or perturbed data. Generally, there are various algorithms which can be found online or off-line and can be tailored to a specific complexity

1.2 NEW EFFICIENT MODEL-BASED PID DESIGN METHOD

A new simple and efficient model-based PID design method for achieving an important design is presented compromise; acceptable stability, and medium fastness of response.

The proposed method was test for different systems including first, second, third and fourth order systems and first-order process with dead-time, the numerical results and response curves are plotted and some are compared with other PID design methods.

Analysis of testing and simulation results show that an important design compromise in the form of acceptable stability and medium fastness smooth and without overshoot response, is achieved, to speed up the response and reduce (remove) the overshoot, a gains tuning factor is introduced.

II. ARTIFICIAL NEURAL NETWORK

Artificial neural network is a network that is biologically inspired and has the ability to learn from empirical data or information. They have the ability to learn and recall the main functions of the brain in order to make decisions and draw conclusions when presented with complex and noisy information.

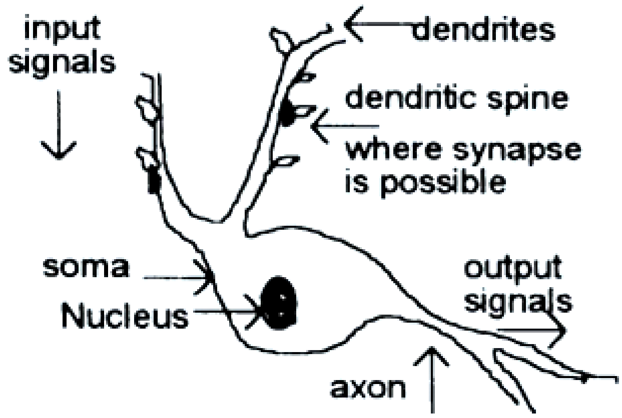

Figure 1. Biological Neural Network (BNN). (source: http://www.ias.ac.in/resonance/Volumes/01/02/0047-0054.pdf)

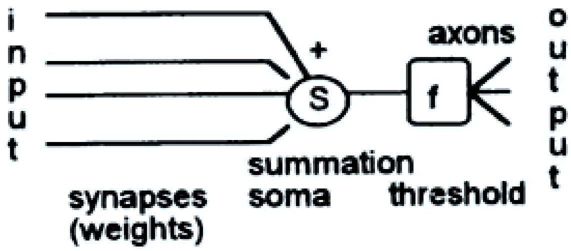

Figure 2. Artificial Neural Network (ANN). (source: http://www.ias.ac.in/resonance/Volumes/01/02/0047-0054.pdf)

BNN and ANN models have certain features in common. Weights in ANN model represent synapses of BNN.

For Biological Neuron System, Dendrites input branching tree of fibers - connect to a set of other neurons-receptive surfaces for input signals. Soma cell body is where all the logical functions of the neurons are realized. Synapse specialized contacts on a neuron - interfaces some axons to the spines of the input dendrites - can increase/dampen the neuron excitation Axon nerve fiber is the final output channel - signals converted into nerve pulses (spikes) to target cells.

In Artificial Neuron System, the input layer the layer of nodes is for data entering on ANN. Hidden layer is the layer between input and output layers and output layer is the layer of nodes that produce the network's output responses. Between the nodes are weights strength or the (gain) value of the connection.

III. DESIGN METHODOLOGY

DESIGN OF PID CONTROLLER VIA NEURAL NETWORK

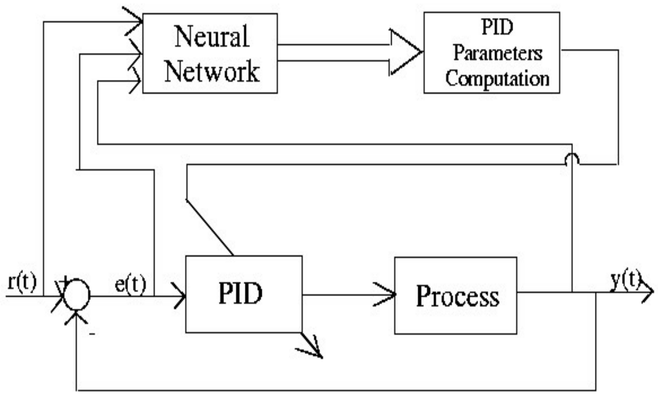

Figure 3. The Two Controller Structures OF PID AND NEURAL NETWORK

A neural network is a powerful data modeling tool that is able to capture and represent complex input/output relationships. The motivation for the development of neural network technology stemmed from the desire to develop an artificial system that could perform "intelligent" tasks similar to those performed by the human brain. Neural networks resemble the human brain in the following two ways:

A neural network acquires knowledge through learning.

A neural network's knowledge is stored within inter-neuron connection strengths known as synaptic weights

Types of ANNs

The two main types of ANNs are;

(i) Feed Forward Neural Network (FFNN).

(ii) Recurrent Neural Network (RNN).

In FFNN there are no feedback loops from the output back to the input. The signal and information flow is in the forward direction only and the behaviour of FFNN does not depend on past input. This means that the network responds only to its present input.

On the other hand, RNN has feedback loops from its output to its input.

3.1 TYPES OF NEURAL NETWORK ARCHITECTURES.

(i) Single-layer feed forward network:

This NN architecture has only one layer of computational node which does not have any feedback loop.

(ii) Multi-layer feed forward network:

This ANN is feed forward with one or more hidden layers. The input layer which is the source node supplies inputs to the neurons of the first hidden layer, then the output of the first hidden layer neurons supplies inputs to the neurons of the second hidden layer and so on. The network is said to be fully when all the nodes in each layer of the network is connect to every other node in the adjacent forward layer.

(iii) Recurrent neural networks:

This type has at least one feedback loop. The different type of this network depends on the way in which the feedback is loop back to the input. Example is the typical case where a single layer of neurons with each neurons feeding its output back to the inputs of all other neurons.

(iv) Lattice Networks;

A lattice network is a feed forward network with the output neurons arranged in rows and columns. It can have one-dimensional, two-dimensional or higher dimensional arrays of neurons with a corresponding set of source nodes that supply the input signals to the array.

3.2 MODIFICATION OF THE PID ALGORITHM

The basic PID algorithm presents some challenges in control applications that have been addressed by modifications to the PID form.

Integral wind: One common problem resulting from the ideal PID implementations is integral windup. Following a large change in set point the integral term can accumulate an error larger than the maximal value for the regulation variable (windup), thus the system overshoots and continues to increase until this accumulated error is unwound.

This problem can be addressed by:

a. Increasing the set point in a suitable ramp
b. Disabling the integration until the PV has entered the controllable region
c. Preventing the integral term from accumulating above or below pre-determined bounds
d. Back-calculating the integral term to constrain the regulator output within feasible bounds.
e. Overshooting from known disturbances For example, a PID loop is used to control the temperature of an electric resistance furnace where the system has stabilized. Now when the door is opened and something cold is put into the furnace the temperature drops below the set point. The integral function of the controller tends to compensate this error by introducing another error in the positive direction. This overshoot can be avoided by freezing of the integral function after the opening of the door for the time the control loop typically needs to reheat the furnace.

Basic block of a PI controller A PI Controller (proportional-integral controller) is a special case of the PID controller in which the derivative (D) of the error is not used.
The controller output is given by

$$K_P\Delta + K_I \int \Delta \, dt \text{ ---------------------3.1}$$

where is the error or deviation of actual measured value (PV) from the set point (SP).

$$\Delta = SP - PV \text{ ---------------------------3.2}$$

A PI controller can be modeled easily in software such as Simulink or using a "flow chart" box involving Laplace operators:

$$C = \frac{G(1+\tau s)}{\tau s} \text{ ---------------------------3.3}$$

Where
$G = K_P$ = proportional gain
$G/\tau = K_I$ = integral gain

Setting a value for is often a trade off between decreasing overshoot and increasing settling time.
The lack of derivative action may make the system steadier in the steady state in the case of noisy data. This is because derivative action is more sensitive to higher-frequency terms in the inputs.
Without derivative action, a PI-controlled system is less responsive to real (non-noise) and relatively fast alterations in state and so the system will be slower to reach set point and slower to respond to perturbations than a well-tuned PID system may be.
Dead band many PID loops control a mechanical device (for example, a valve). Mechanical maintenance can be a major cost and wear leads to control degradation in the form of either striation or a dead band in the mechanical response to an input signal. The rate of mechanical wear is mainly a function of how often a device is activated to make a change. Where wear is a significant concern, the PID loop may have an output dead band to reduce the frequency of activation of the output (valve). This is accomplished by modifying the controller to hold its output steady if the change would be small (within the defined dead band range). The calculated output must leave the dead band before the actual output will change.
Set Point change the proportional and derivative terms can produce excessive movement in the output when a system is subjected to an instantaneous step

increase in the error, such as a large set point change. In the case of the derivative term, this is due to taking the derivative of the error, which is very large in the case of an instantaneous step change. As a result, some PID algorithms incorporate some of the following modifications:

Set point ramping

In this modification, the set point is gradually moved from its old value to a newly specified value using a linear or first order differential ramp function. This avoids the discontinuity present in a simple step change.

Derivative of the process variable

In this case the PID controller measures the derivative of the measured process variable (PV), rather than the derivative of the error. This quantity is always continuous (i.e., never has a step change as a result of changed set point). This modification is a simple case of set point weighting.

Set point weighting

Set point weighting uses different multipliers for the set point in the error in the proportional and derivative element of the controller. The error in the integral term must be the true control error to avoid steady-state control errors. These two extra parameters do not affect the response to load disturbances and measurement noise and can be tuned to improve the controller's set point response.

Feed forward. The control system performance can be improved by combining the feedback (or closed-loop) control of a PID controller with feed-forward (or open-loop) control. Knowledge about the system (such as the desired acceleration and inertia) can be fed forward and combined with the PID output to improve the overall system performance. The feed-forward value alone can often provide the major portion of the controller output. The PID controller primarily has to compensate whatever difference or error remains between the set point (SP) and the system response to the open loop control. Since the feed-forward output is not affected by the process feedback, it can never cause the control system to oscillate, thus improving the system response without affecting stability. Feed forward can be based on the set point and on extra measured disturbances.

For example, in most motion control systems, in order to accelerate a mechanical load under control, more force is required from the actuator. If a velocity loop PID controller is being used to control the speed of the load and command the force being applied by the actuator, then it is beneficial to take the desired instantaneous acceleration, scale that value appropriately and add it to the output of the PID velocity loop controller. This means that whenever the load is being accelerated or decelerated, a proportional amount of force is commanded from the actuator regardless of the feedback value. The PID loop in this situation uses the feedback information to change the combined output to reduce the remaining difference between the process set point and the feedback value. Working together, the combined open-loop feed-forward controller and closed-loop PID controller can provide a more responsive control system

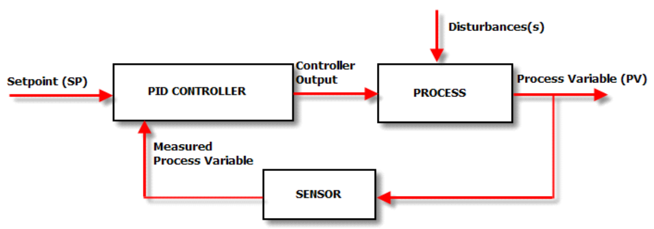

Figure 4. Anatomy of A Feedback Control System

3.3 NEURAL NETWORK.

The word network in the term 'artificial neural network' refers to the inter–connections between the neurons in the different layers of each system. An example system has three layers. The first layer has input neurons which send data via synapses to the second layer of neurons, and then via more synapses to the third layer of output neurons. More complex systems will have more layers of neurons with some having increased layers of input neurons and output neurons. The synapses store parameters called "weights" that manipulate the data in the calculations. An ANN is typically defined by three types of parameters:

1. The interconnection pattern between the different layers of neurons
2. The learning process for updating the weights of the interconnections
3. The activation function that converts a neuron's weighted input to its output activation.

Mathematically, a neuron's network function is defined as a composition of other functions, which can further be defined as a composition of other functions. This can be conveniently represented as a network structure, with arrows depicting the dependencies between variables. A widely used type of composition is the nonlinear weighted sum, where (commonly referred to as the activation function is some predefined function, such as the hyperbolic tangent. It will be convenient for the following to refer to a collection of functions as simply a vector.

3.4 A FLOW CHART REPRESENTING THE SIMULATED WORK

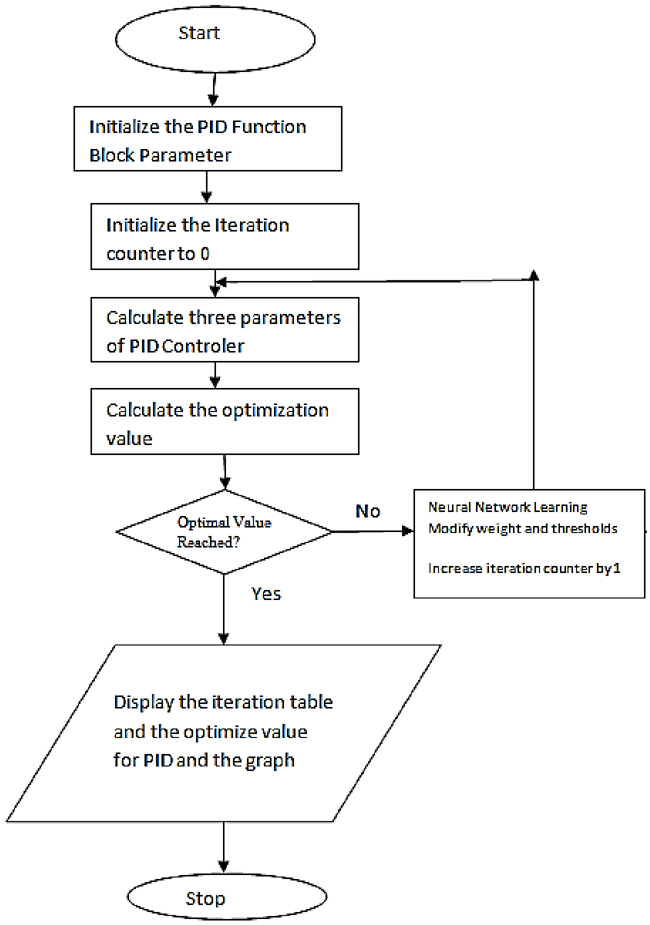

Figure 5. PID Controller based on Neural Network Model Simulation

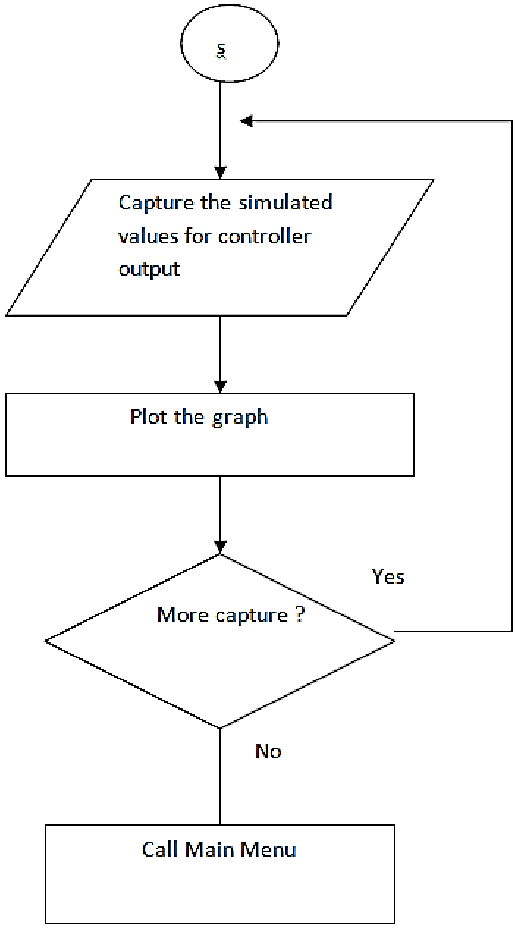

Figure 6. Time offset - Controller Output

IV. DATA PRESENTATION AND ANALYSIS STEPS RESPONSE OF PID CONTROLLED BY NN

This process involve the PID and NN, on this process date is input on gain block and followed by a click on simulink double click on NN block which gives an output result after a process of optimization. See output of explanation on the graph below.

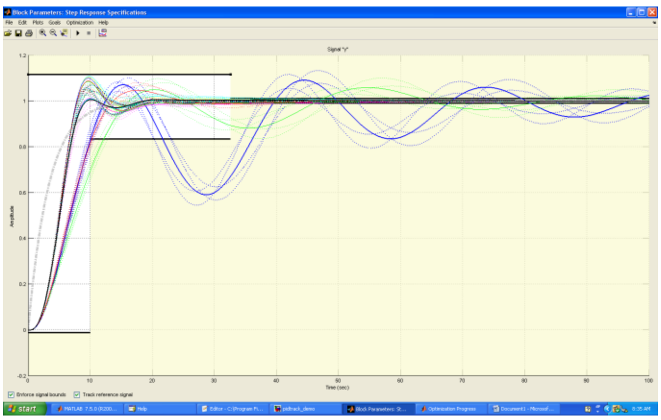

Figure 7. Step Response Specification of PID controlled by NN

and controller response for both ANN and PID controller. For PID controller, the controller setting that gave the best performance was found to be Kp = 0.15, Ki = 0.4 and Kd = 0.8. The response is not without of overshooting, which is very high and small inverse response. For the case of ANN the overshoot is very small but in both cases, they brought the reaction almost into complete conversion.

The next performance test involved a set point tracking problem the set point was allowed to change in random fashion.

The table is a time off set against the control output

TABLE 1: Time offset-Controller Output

S/N	TIME OFFSET (SEC)	CONTROLLER OUTPUT (SEC)
1	100	2
2	200	4
3	300	6
4	400	8
5	500	9
6	600	10
7	700	11
8	800	12
9	900	14

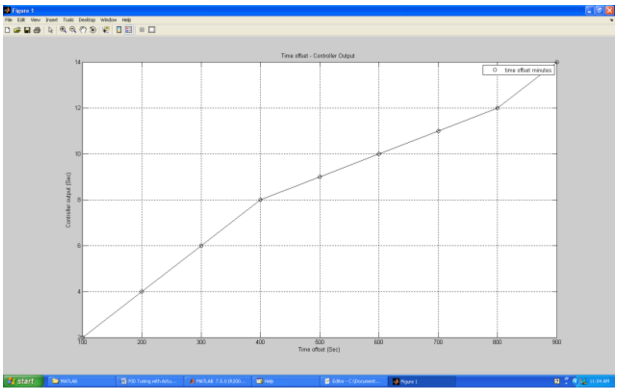

Figure 8 : Time offset - Controller Output

The graph above represents a time offset of the control system. The system proceed in an arithmetical progression which shows that the system was properly controlled by NN programmed

4.1 NEURAL NETWORK MODELING RESULT

The result views of neural network modeling in tune Time Series are grouped into two categories, tables and graphics. The details of them are described here:

4.2 MODEL SUMMARY

This is a précis on revamping of PID controller via artificial intelligent technique which involves the neural network. The analysis of this methodical scheme shows that PID controller and neural network are the principal graphic representations.

V. CONCLUSION

The outcome of this work affirms the effective function of neural networks in the manufacturing and control technology in the present era. An undisputable example is the use of it in the enhancement of the PID controller to work as an intelligent system. However, this work adopted plasticity on the regulation system in the network.

This paper, therefore, showcases the fabrication and refashioning of the proportional integral derivative PID control system using different even parameters. This approach enhanced the precision of the PID controller by the use of an intelligent neural network. One of the advantages is the elasticity gained. The modernized PID offered improved operational qualities by the control of the neural network controller. For instance, the traditional control design (PID) often needs a remodeling using artificial neural network (ANN) to become intelligent and operate efficiently whereas the neural network controller is naturally sensitive to equipment and the system constraints. More so, the quantum of wastage recorded originally was cut down significantly.

VI. REFERENCES

[1]. B. Choi and J.-H. Lee, (2009) "Comparison of generalization ability on solving differential equations using back propagation and reformulated radial basis function networks," Neurocomputing, vol. 73, no. 1-3, pp. 115-118,.

[2]. C. Franke and R. Schaback,(1998) "Solving partial differential equations by collocation using radial basis functions," Applied Mathematics and Computation, vol. 93, no. 1, pp. 73-82,.

[3]. D. R. Parisi, M. C.Mariani, and M. A. Laborde, (2003) "Solving differential equations with unsupervised neural networks," Chemical Engineering and Processing, vol. 42, no. 8-9, pp. 715-721,.

[4]. Ebrahimi M, (2000) Analysis, modeling and simulation of stiffness in machine tool drive [J]. Computers ＆Industrial Engineering,

[5]. G. C. Chen,, Zhang, L., & Hao, N.M. et al. (2003) Application of Neural network PID Controller in Constant Temperature and Constant Liquid-level System [J]. Micro-computer information, , 19(1): 23-24, 42

[6]. I. E. Lagaris, A. C. Likas, and D. G. Papageorgiou, (2000) "Neural network method for boundary value problems with irregular Boundaries," IEEE Transactions on Neural Networks, vol. 11, no. 5, pp. 1041-1049,.,

[7]. K. Narendra, Parthasarathy, K. (1990). "Identification and Control of ynamical Systems using Neural Networks". IEEE Transactions on Neural Nerworks. Vol. 1, No. 1.

[8]. L. Jianyu, L. Siwei, Q. Yingjian, and H. Yaping, (2003) "Numerical solution of elliptic partial differential equation using radial basis function neural networks," Neural Networks, vol. 16, no. 5-6, pp. 729-734,

[9]. M. Suzuki,, Yamamoto, T., & Tsuji, T. (2004). A design of neural-net based PID controller with evolutionary computation. IEICE Trans. Fundamentals. VOL. E87-A, No. 10 October.

[10]. N. Smaoui and S. Al-Enezi, (2004) "Modelling the dynamics of nonlinear partial differential equations using neural networks," Journal of Computational and Applied Mathematics, vol. 170, no. 1, pp. 27-58,

[11]. N. Selvaraju and J. Abdul Samant, (2010). "Solution of matrix Riccati differential equation for nonlinear singular system using neural networks," International Journal of Computer Applications, vol. 29, pp. 48-54,

[12]. O. Dwyer, Aidan (2006). PI and PID controller tuning rules: an overview and personal perspective. Proceedings of the IET Irish Signals and Systems Conference, pp. 161-166, Dublin Institute of Technology.

[13]. S. Mall and S. Chakraverty, (2013). "Regression Based Neural network training for the solution of ordinary differential equations," International Journal of Mathematical Modelling and Numerical Optimisation, vol. 4, pp. 136-149,

[14]. S. A. Hoda and H. A. Nagla, (2011) "Neural network methods for mixed boundary value problems," International Journal of Nonlinear Science, vol. 11, pp. 312-316,.

[15]. S. Haykin, (1999) Neural Networks a Comprehensive Foundation, Prentice Hall, New York, NY,USA,.

[16]. S. He, K. Reif, and R. Unbehauen, (2000) "Multilayer neural networks For solving a class of partial differential equations," Neural Networks, vol. 13, no. 3, pp. 385-396,.

[17]. T. Leephakpreeda, (2002). "Novel determination of differential equation solutions: universal approximation method," Journal of Computational and Applied Mathematics, vol. 146, no. 2, pp. 443-457,

[18]. Uchegbu C. E et al.: (2016) Remoldelling of PID Controller Based on an Artificial Intelligency (Neural Network) American Journal of Science, Engineering and Technology.

[19]. V. P. Singh, S. Chakraverty, R. K. Sharma, and G. K. Sharma, (2009) "Modeling vibration frequencies of annular plates by regression based neural network," Applied Soft Computing Journal, vol. 9, no. 1, pp. 439-447