Networked control system for electrohydraulic flow control positioner using Neural Controller and Collaborative Network

Audu Eliazar Elisha, Dr. Lalit Garg

Abstract—Electrohydraulic flow control valve is an essential element of an automated process industry where fluid control is applicable. The use of conventional controllers overan IP-communication network for controlling electrohydraulic flow control positioner to regulate mainline pressure and flow rate in pipeline transportation of petroleum products between two stations where downstream pressure of the pumping station fluctuates significantlyposes a problem of instability on the flowrate and the mainline pressure of the pipeline. Additionally, the effect of network induced, time-varying delay between the controller and the electrohydraulic flow control valves induces a problem of poor quality of control and inefficient system performance of the control loop. In this paper, we presented an application of neural network in processflow control using an electrohydraulic valve positionerand proposed a concept of collaborative network for networked control systems over IP-based networks.

Index Terms—Artificial Neural Network (ANN), Collaborative network,Linear time invariant (LTI),Network Control System, PID-controller, Smith Compensator, Time-varying delay.

1 Introduction

THEinnovative transformation from classical control systems to networked-based control systems (NCSs) has presented new possibilities in terms of systems integration and unlimited opportunities in its applications. NCS are widely applied in wireless sensor networks, transportation networks, automobile as Controller Area Network (CAN), aircraft control systems, and electrical power networks due to their reduced cost, ease of maintenance and installations, flexibility and scalability [1], [2]. In process plant and manufacturing industries, NCS provides the mechanisms forintegrating various processes into a single view and allow sub-systems that are geographically distributed within the plant to communicate over a shared data media. The collective performance of the NCS elements (or subsystems) helps to keep the process variables within tolerable limits. This implies that NCSloop must be stable and quality of control must be guaranteed for effective operations. However, the application of NCS to process control has brought new sets of challenges due to the effect of network and shared communication channels, which cause non-linear time varying delay, packet dropout, and data congestions in the control loop [1], [2], [3], [4], [5]. These network effects affect controller performances andintroduce instability in maintaining process variables due to loss of quality of control by the NCS [5], [6], [7]. In real-time control system, data transmission from controller to actuating mechanism, and from sensors to the controller is time bound. Failure to meet these deadlines has significant consequences onthe performance and quality of service of the NCS. To deal with the problem of time-varying delay, packet dropout during routing and traffic congestion, different methodologies have been developed to ensure optimal stability of the NCSs. Analysis and synthesis on NCSs stability were conducted by researchers over the years to provide framework for optimizing network performance and improving quality of control. In [1], a new approached was presented based on discrete-time representation of the NCS using Lyapunov-based stability criterion expressed in terms of linear matrix inequalities (LMI). Perturbed frozen time (PFT) and point-wise stability approached have, also,been used to investigate the problem of stability in NCS by assuming linear time varying system [3]. Before the emergence of modern mathematical treatment of networked control systems with non-linear time varying delay, the solution to the problem of stochastic delay (or dead time) was first offered by Smith in 1957 by eliminating the time-delay variable from the control loop characteristic equation [8]. The Smith predictor method of compensation and robust control stability theory was used to improve the quality of performance (QoP) of NCS on shared communication networks [9]. Smith predictor is largely applied to either structurally or parametrically optimize the performance of a classical control system over networks with unpredictable time delay [8], [9], [10]. An improved version of Smith compensator based on Internal control module and dynamic matrix controller was used to deal with the problem of stochastic network delay in real-time and internet based NCS [10], [11]. One of the benefits of this method is that it requires minimal knowledge of plant under control [12].

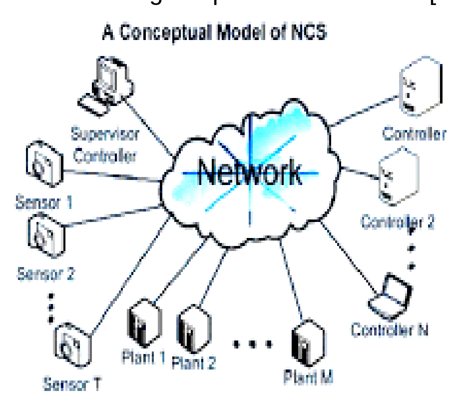


Fig.1. Networked-based Control System [13].

The development of Smith Predictor model opened a Pandora box in developing a technique that can minimized nonlinear time-delay in control networks. However, model mismatch as-

ISSN 2229-5518

sociated with the predictor can result to closed loop instability and degrade the quality of control of a networked control system [8].

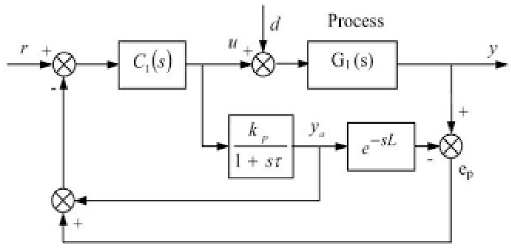


Fig. 2. Block diagram of a Smith Predictor [14].

Fuzzy logic has been successfully applied in NCS to provide tuning parameters and non-linear mapping for Proportional-Integral-Derivative (PID)-based controllers. Fuzzy logic was developed to copy some of the attributes of human experts by encoding specialized knowledge of process control using linquistic rules [13].The traditional PID controllers tuning technique such as Ziegler-Nichol method provides unsatisfactory performance of NCS where the time delay exceeds the critical value [14]. This situation can result to loss of control efficient and induced instability in the process[15]. The PID-based fuzzy controller allows both the input variables and the control action to be defined in terms of linguistic rules and inference engine [16]. As an improvement, the fuzzy set weighted controller was also proposed to deal with the problem of stochastic time-delay in a control network. The weighted approach is based on parameter setting of the proportional action of the controller to a constant value, typically, of less than unity. Using the inference engine of the fuzzy approach, one part of the controller controls the attenuation of load variation and the other part is devoted to set point [17].

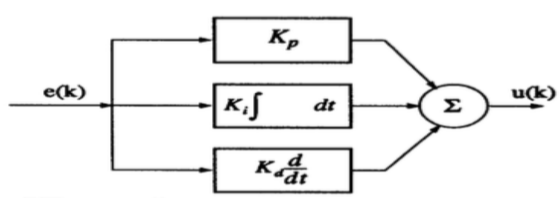


Fig.3. General Structure of PID Controller [15].

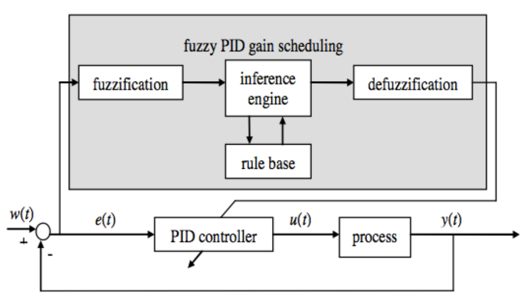


Fig.4. Fuzzy-PID Controller [16],[17].

The application of fuzzy logic to PID controllers in network control offers a promising path to the minimization of time varying delay for NCS. However, the tuning process can be time consuming and three parameters are required to be precisely tuned to achieve the desire performance.

2 NCS MODELLING AND TIME-VARYING DELAY

In this section, the NCS problem is shaped into control and optimization framework for analysis, and the time-varying delay of the system is presented in the model as a variable.

2.1 NCS Modelling

The NCS plant model is represented as a continuous-time, linear time invariant (LTI) system. This assumption simplifies system analysis. In figure below, time taken for data to travel from controller to actuator, and data from sensor to controller are the network time delays. These two time delays are non-linear time-varyingwith significant impact on control performance.

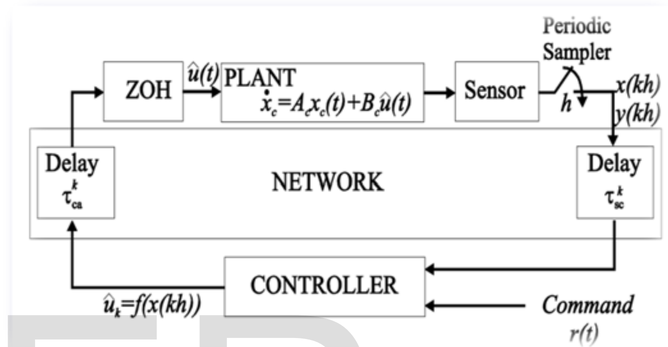


Fig.4. Model of a Continuous-time Plant, Discrete Time Controller with network delay along the communication paths [17].

$$\tau = \tau_{ca} + \tau_{sc} + \tau_p \dots\dots\dots\dots\dots\dots\dots i$$

Where: $\tau_{ca}$is the controller to actuator time delay, τ_{sc} represents sensor to controller delay and $\tau_p$is the total processing time delay. $\tau_p$is not represented in the diagram and is assumed to be negligible since the processing delay has a relatively small impact on the control loop especially with fast processors and improved system architectures. Therefore, we considered, in this networked control system modelling, the sensor to controller, and controller to actuator end to end stochastic time delays as described in [19], [20]. Hence,

$$\tau = \tau_{ca} + \tau_{sc} \dots\dots\dots\dots ii$$

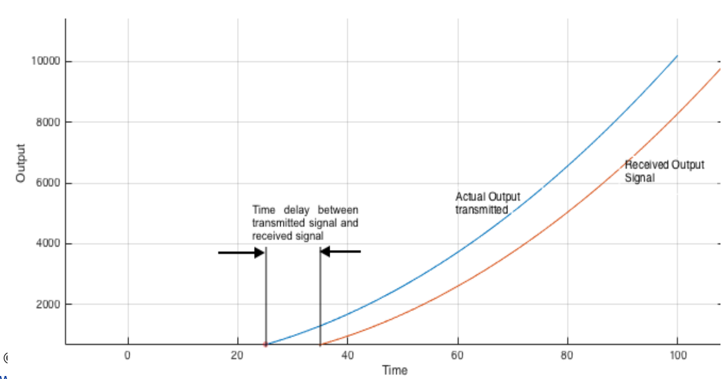


Fig.5. Time delay between a controller and an actuator in NCS.

Controlled Process equation

$$x'_c = Ax_c(t) + B_c u_c(t) \ldots\ldots\ldots\ldots\ iii$$
$$y(t) = Cx_c(t) \ldots\ldots\ldots iv$$

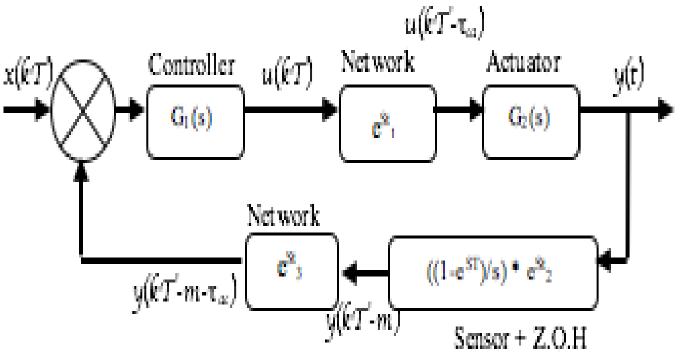


Fig.7. Simplified model of continuous time plant of a NCS.

In figure7 above, the controller function can be determined using the following mathematic relationship:

$$G(s) = \frac{Y(s)}{X(s)} \ldots\ldots\ldots\ldots\ldots\ldots iv$$

Where $X(s)$ is the Laplace transformation of the input function and $Y(s)$ denotes the Laplace transformation of the output function. To simplify control analysis, we assumed that the NCS in figure 7 is a linear time-variant (LTI) system. Therefore, the overall transfer function can be expressed as:

$$G(s) = \frac{G_1(s) * G_2(s) * e^{s\tau_1}}{1 + \left(\frac{(1 - e^{sT})}{s} * e^{s\tau_1} e^{s\tau_3}\right) * \left(G_1(s) * G_2(s) * e^{s\tau_1}\right)} \ldots\ldots v$$

2.2 IP-BASED NCS TIME-VARYING DELAY AND CHANNEL OPTIMIZATION

In IP-based networked control systems, control information passes through series of OSI layers and network infrastructures from one node to another. Routing of information through routers and switches contribute to the overall processing delay of the NCSs. Furthermore, the selection of transport layer protocol is crucial in NCS for guaranteed stability and reduced network delay.

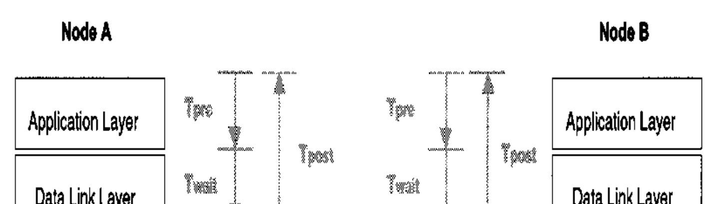


Fig. 8. Diagram showing delay along data flow path at devices OSI layer level [20].

The use of transmission control protocol (TCP) offers better congestion and flow control mechanisms for reliable delivery of data than user datagram protocol (UDP) but increases the transmission latency when used for networked control system application [21]. The emphasis of UDP is packet-based, connectionless, best effort services that deliver continuous stream of data over an IP-network, which makes UDP an ideal protocol for real-time networked control systems communication [21]. However, networked control application using UDP protocol must be implemented with flow control and error correction mechanisms since they are not featured in the UDP software routines.

From optimization perspectives, network can be represented as a directed graph, and access to network can be viewed as a problem of distributed resources sharing [22], [23]. Let V be a set of nodes and E be a collection of links connecting the nodes on a communication network. Therefore, the network can be represented as G= (V, E). In networked control system, controllers and sensors are regarded as traffic sources, and from information theoretic perspectives, for a reliable and successful data delivery in NCS, transmission from source to sink should not exceed the link capacity [24].

let n_{ij} be a link between controller v_i and actuator v_j where $n_{ij} \in E$, $v_i, v_j \in$ V, and i, j $\in \{1,2, \ldots\ldots, k\} \in \mathbb{Z}$. For each link, n_{ij}, let u_{ij} be the link non-negative, finite transmission capacity. Therefore, controller v_i transmitting data over a link n_{ij} to an actuator v_j at *m*-data rate should have a channel utility ($m \leq u_{ij}\ \forall u_{ij} > 0$). Assuming that the channel utilization is a continuous function, then the controller computes the following optimization problem [23].

$$\max_{u_v} \sum_{v \in V} y(u_v) \ldots\ldots\ldots vi$$

where $v \in$ V

If M is the overall network capacity, then the optimization constraint is given by:

$$Total\ links\ transmission\ capacity, \sum_{i,j=1}^{M-1} u_{ij} \leq M \ldots vii$$

The constraint is required for reliable data transmission over NCS communication channel. Beside the problem of networked induced delay and the need for robust dynamic characteristic in the controller design, the problem of bandwidth optimization must be dealt with by the controller algorithm. In addition to the above mentioned problems in network control, NCS operating over IP-networks must take into consideration the probabilistic nature of the network. With UDP, the tendency of network congestion and inefficient utilization of bandwidth arises. This also raise the probability of packet dropout, which can lead to control system instability and poor quality of system performance. In this paper, we explore the use of artificial neural network and collaborative network to propose a methodology for dealing with time-varying delay in NCS in relation to electrohydraulic flow control valve for a stable flow rate in pipeline transportation of petroleum products from a pump station to a distant storage and loading facility.We introduced the use of collaborative network to enhance intelligent communication between local nodes (or neighbours) and provides a medium for scheduling and prioritization over IP-based networks.

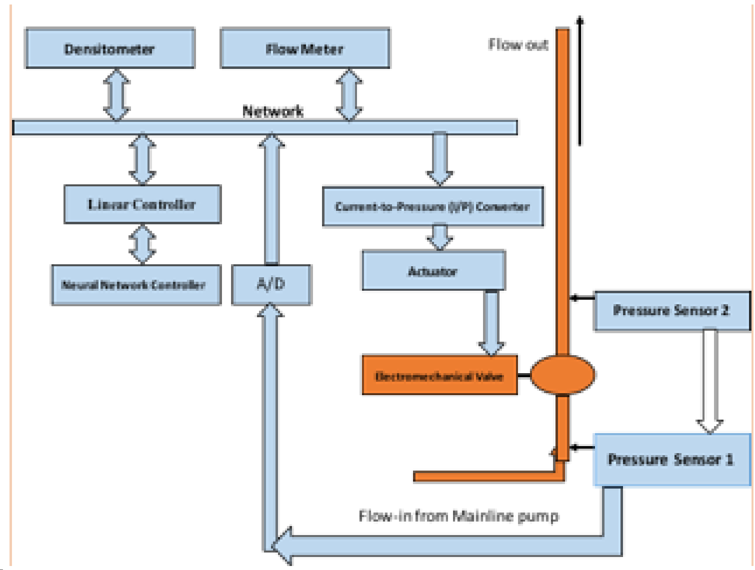


Fig. 9. Networked Control System for Electrohydraulic flow control positioner using Neural Network Controller.

3 PROBLEM FORMULATION AND CONTROLLER DESIGN

In the distribution of petroleum products between two stations through multi-product pipeline, maintaining a correct positioning of electromechanical flow control valve at the pumping station is challenging especially where the upstream stream pressure of a flow control valves fluctuates significantly due to process activities at the receiving station such as adjustment of valves to meet product reception requirements. These flow control activities at the receiving station affects the main line pressure and the pumping rate between the two stations.Pressure variation across electrohydraulic flow control at the discharge of a mainline pump affects the fluid volumetric flow rate of the mainline. The need to maintain correct (or almost constant) flow rate in the transportation of petroleum products through pipeline has both economic and technical significance. Furthermore, the pressure differential of the electrohydraulic flow control valve must be maintained in relation to the mainline pump motor amperage (or power), actual flow rate, and valve downstream pressure. The flow rate through electrohydraulic flow control positioner can be determined by the following equation.

$$Cv = \frac{Q}{N_1 F_P \sqrt{\frac{(P_1 - P_2)}{G_f}}} \dots\dots\dots\dots\dots viii$$

The equation above established the relationships between flow rate (Q), specific gravity (G_f), downsteram pressure (P_1), upstream pressure (P_2), numeric constant (N_1), and piping geometric factor (F_P) . The use of PID controller, in this case, can be challenging due to the difficulty associated with finding appropriate tuning parameters to provide optimal process regulation under uncertain downstream or upstream pressure variation [25].

3.1 Artificial Neural Network Controller

Neural controller is emerged from the intense study and research of how simple and highly interconnected neurons of human brain collectively solve complex problems. Neural network is very robust, adaptive, flexible, and fault tolerant with self-learning ability [26]. These unique abilities give the neural controllers the capability to provide effective control performance without having a prior knowledge of the plant mathematical model. The highly structural property and massively parallel distributed capability of artificial neural network makes it suitable for practical implementation of parallel processing systems [27], [28]. The behaviour of neural networks can be altered by changing the connection weights of individual neurons that made-up the network and layer activation function. These behavioural features of an ANN is very essential in optimization and solving non-linear problems in control systems. The most commonly implemented artificial neural network is feed-forward with Backpropagation learning algorithm [26], [29]. Backpropagation method is widely used in control systems to solve nonlinear control problems where input parameters are unpredictably unstable. They are also used where conventional computational models such as fuzzy logic and PID proved to be inadequate. Artificial neural network has been successfully applied in real-time dynamical adaptive systems to minimize cost functions (error) to the desire accuracy [28], [30]. Problems involving finding unknown functions using ANN was demonstrated in [31], [32].

For neural network to function, neurons must be trained on the dynamics of the system or process under control. Psaltis et al. proposed two learning schemes for neural network, the general and specialized learning [26], [30]. In generalized learning method, neural network is trained off-line with the controlled process dynamics before deployment. Once trained, the network can perform control functions based on the preprogrammed training dynamics. Specialized learning algorithm provides conditions for training a neural network online. It constantly monitors the difference between the actual output of the neural network and the expected output such as in the case of gradient descent. The output difference is used to adjust the synaptic weight of the network to produce the desire output. This process of self-correction by minimization of error is an adaptation process. Figure 10 below shows a conceptual model of a three-layer neural network.

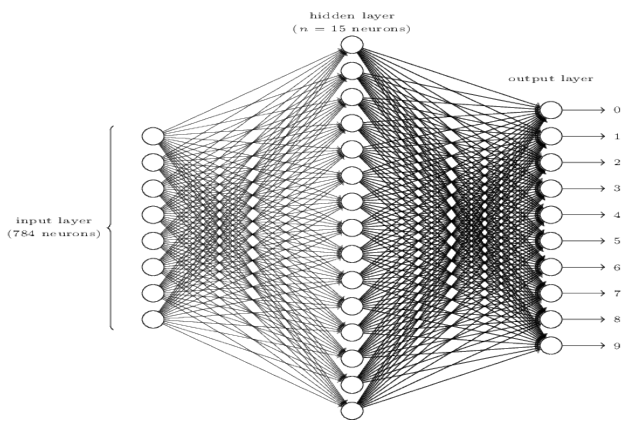


Fig.10. Conceptual model of a Artificial Neural Network with Input, hidden and output layers [31].

$$\text{Neuron output } = \begin{cases} 1 \quad if \sum_{i=0}^{N} w_{ji} x_i > Threshold \\ 0 \quad if \sum_{i=0}^{N} w_{ji} x_i \leq Theshold \end{cases}$$

Where:
w_{ji} and x_i are the network weights and the input respectibely

The network weight can be iteratively updated during training to provide the optimal weights required to solve the optimization *problem.*

Let $y_i = \sum_{i=1}^{N} w_{ji} x_i \ldots \ldots \ldots \ldots . ix$

Let $f\,(w, x)$ be any arbitrary differentiable activation function and let $Error, E, = \frac{1}{2}\sum_{i=1}^{N}(d_i - y_i)^2 \ldots \ldots \ldots x$

$$i = \{1,2,3, \ldots .\} \in \mathbb{Z}$$

Therefore, using gradient descent method, the weight can be updated using:

$$\widehat{w_{ji}} = w_{ji} - \eta \frac{\partial E}{\partial w_{ji}} \ldots \ldots . \ldots xi$$

Where $\widehat{w_{ji}}$ *and* w_{ji} *are the new and old weights respectively*

Feed-forward multilayer perceptron using back-propagation learning algorithm is the most commonly applied form of artificial neural network. The algorithm employs gradient descent minimization to decrease the error cost function that exist between the training set and the actual trajectory of function. The error cost function is computed using the mean square error (MSE) method and propagated backward to the network in an iterative fashion until an optimal result is reached. The optimal weights that minimized the error function is considered to be a solution to the learning problem and the algorithm stop searching or terminated [31]. On each iteration, the mean squared error is evaluated and compare with the performance goal. If performance goal is not meet, then the output error is propagated backward towards the input by partially differentiating the error with respect to the weight of a node to find a new synaptic weight for optimal result [29], [33].

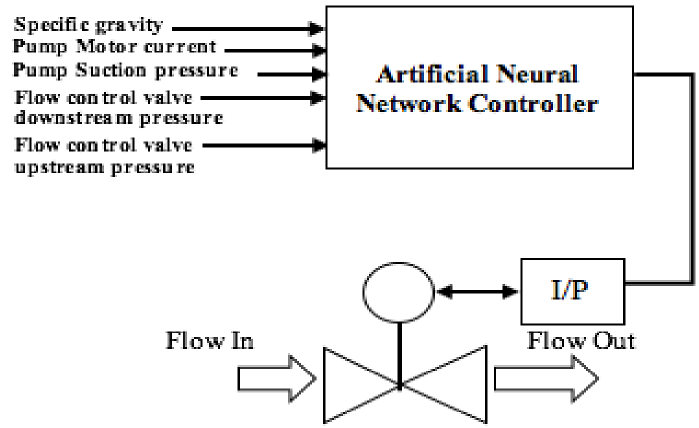


Fig.11. Simplified model of the proposed Electrohydraulic flow control using Neural Network controller in Networked Control System.

ANN is highly suitable for solving non-linear control problems, and it is very robust, self-learning, adaptive, flexible, and fault tolerant in its application [34]. The self-learning ability gives neural controllers the capability to provide effective control performance without having the prior knowledge of the control dynamics [34], [35]. The structural property of ANN makes it practically feasible for implementation of parallel processing systems, and the behavior of the system can be altered by adjusting the network weights [35], [36].

3.2 Collaborative Network

The central idea behind the use of collaborative network is to foster intelligent communication among local nodes prior to data transmission and allow deterministic access to communication medium by NCS over probabilistic (IP) network for real time control. With collaboration, nodes can monitor the data rate of their neighbourhood and priority level. The inclusion of this network allows for the implementation of scheduling and prioritization scheme without using the main IP-communication network. This intelligent exchange of information provides the neural controller nodes with input information such as data rate and scheduling details of neighboring nodes to regulate data rate over the communication network and minimise packet dropout. As a demonstration,EIA-485 (RS-485) was used as a collaborative network protocol to interlinked various nodes, called clusters, connected to the network. EIA-485 is a balanced line, differential voltage, digital transmission system designed to operate directly over a physical layer. It is capable of supporting up to 32 devices in a bus or ring topology in asynchronous communication and can operates at a data speed of up to 10Mbps [37], [38]. It is widely used in the control communications and DeviceNet data networks for their ability to reject common mode noise (noise immunity). In this implementation, EIA-485 is used to provide communication protocol

ISSN 2229-5518

for self-awareness between various nodes on an IP-network and no master controller is required to control access. The nodes are self-driven by scheduling algorithm that keeps track of other nodes communicating over a physical layer by means of collaboration. Communication is central to effective scheduling and prioritization of access to a network resources or media in IP-based networked control systems.

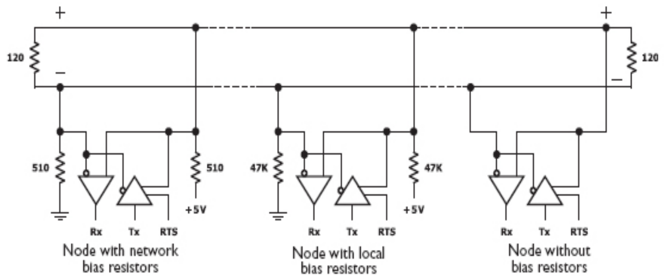


Fig.12. EIA-485 network showing nodes in Bus-mode [37].

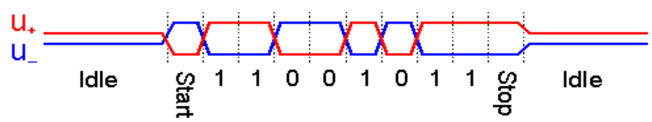


Fig.13. Timing diagram of the EIA-485 over communication network [40], [41].

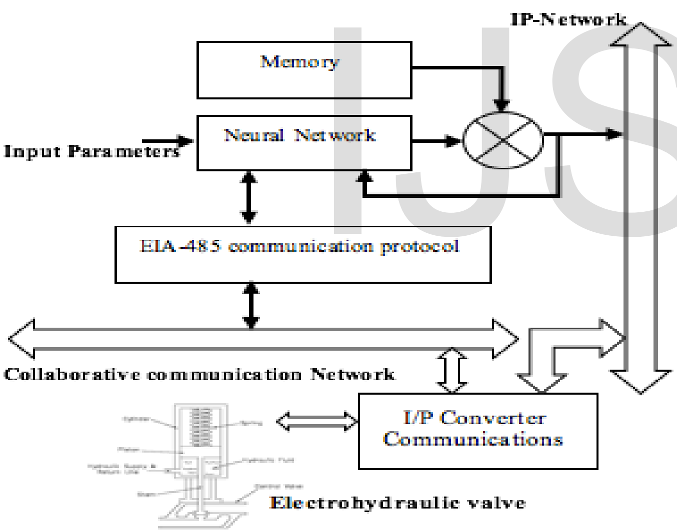


Fig. 14. A Simplified diagram of NCS for electrohydraulic flow control valve using Neural Network controller and collaborative network.

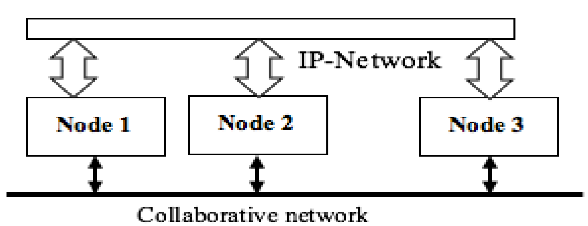


Fig.15. A simplified NCS for electrohydraulic valve showing IP-network and Collaborative network.

4 RESULT

The Neural network controller was trained, tested and simulated using a set of data recorded during pumping operations between PPMC Area Office pump station PortHarcourt Nigeria and Aba depot receiving and storage facility. The data was taken from 2001, 2002, and 2014 pumping charts for Premium Motor Spirit (PMS), Dual Purpose Kerosene (DPK), Automotive Gas Oil (AGO) and Water. Discharge pressure, pipeline pressure (between Area Office and Aba), pump electric motor amperage, and specific gravity of the product are the input parameters to the Neural network while the flow control valve sizing coefficient expressed in terms of flow rate is the corresponding output of the controller. The controller is trained to adapt to significant differential pressure changes between the downstream and upstream pressure, and generate a proportionate control signal for the electrohydraulic flow control valve. Three different learning methods was used during training to compare their performances and determine the appropriate training algorithm that provides best case scenario (approximations). The algorithms are Levenberg-Marquardt (trainlm), Resilient Backpropagation (trainrp) and the variable learning rate backprogation (traingdx), and inbuilt functions of MATLAB application. The collaborative network communication using EIA-485 between two nodes is simulated using PIC microcontroller on Proteus schematic software environment. We built ANN with four input, ten hidden layer neurons and one output as shown below.

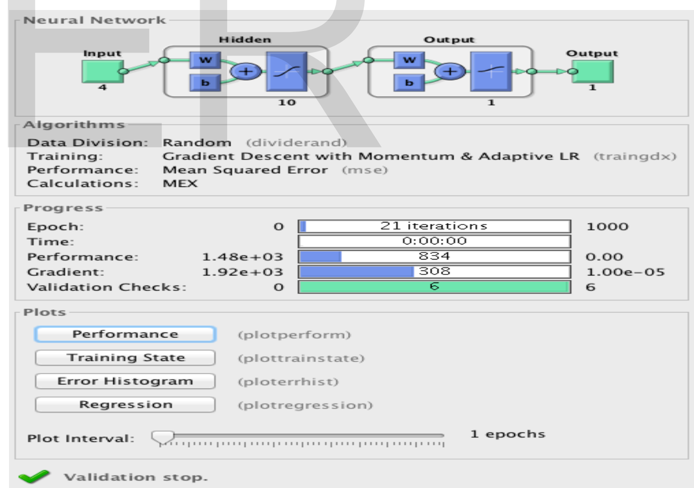
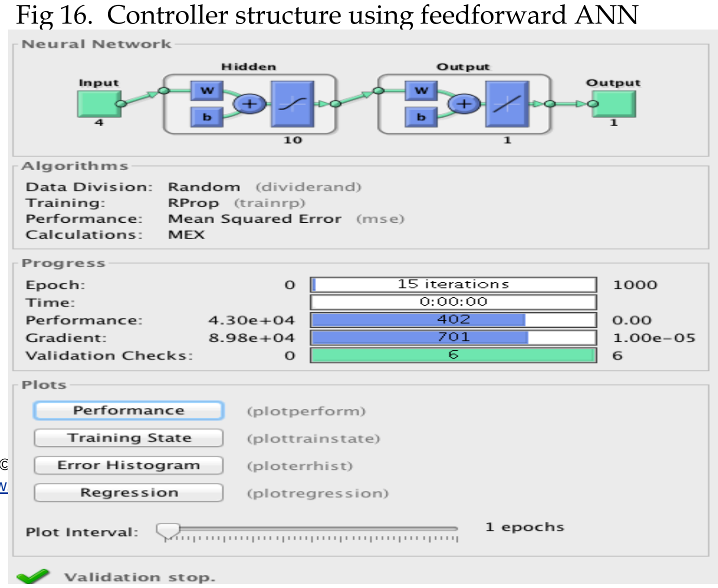


Fig 16. Controller structure using feedforward ANN



ISSN 2229-5518

Fig 16. Controller structure using feedforward ANN trained with Resilient Backpropagation (trainrp).

The data above was randomly extracted from same set of data use for training and validating the neural network. However, this data was not part of the neural network training of the controller. P1 is the electrohydraulic valve down stream pressure, P2 is the upstream pressure, and A is the pump motor amperage.

TABLE 1
ACTUAL PUMPING DATA FOR SIMULATING THE ANN

TEST DATA					
Product	**P1**	**P2**	**A**	**Density**	**Flow Rate**
AGO	65	24.5	115	0.83	183
AGO	65	24.5	115	0.83	183
AGO	65	24.5	115	0.83	183
AGO	65	24.5	115	0.83	183
AGO	65	25	115	0.83	183
AGO	65	25	115	0.83	183
HHK	64	18.6	115	0.805	186
HHK	64	24.7	112	0.805	193
HHK	64	28.8	112	0.805	201
HHK	64	32.6	112	0.805	182
Water	85	11.1	134	1	193
Water	85	17.7	133	1	192
Water	85	17.9	133	1	193
Water	85	17.7	134	1	193
PMS	61	23.2	110	0.73	230
PMS	61	19.5	116	0.73	229
PMS	61	11.1	108	0.73	210
PMS	62	27.2	104	0.73	209

TABLE 2
ANN RESULT USING TRAINLM ALGORITHM

ANN TESTING RESULT			
Actual Flow Rate	**ANN with Default MATLAB activation function and trainlm training method**	**ANN with trainlm training method and tansig-logsig activation function**	**ANN with trainlm training method and logsig-logsig activation function**
183	179.4571	177.4211	179.8968
183	179.4571	177.4211	179.8968
183	179.4571	177.4211	179.8968
183	179.4571	177.4211	179.8968
183	181.0215	176.8801	179.8885
183	181.0215	176.8801	179.8885
186	178.9075	197.2158	182.9737
193	190.0291	189.2153	204.3044
201	193.0076	185.4825	191.4642
182	189.4029	180.4791	185.295
193	180.1546	168.399	194.6979
192	185.4379	181.974	194.6979
193	185.7385	182.4782	194.6979
193	191.0662	188.2279	194.6979
230	230.8599	224.981	212.8489
229	228.3767	230.8168	241.9356
210	206.3306	211.6307	151.273
209	203.9613	205.7758	209.1083

TABLE 3
ANN RESULT USING TRAINRP ALGORITHM

Actual Flow rate	ANN with Default MATLAB activation function and trainrp training method	ANN with tansig-logsig activation function and trainrp training method	ANN with trainrp training method and logsig-logsig activation function
183	171.3896	178.8079	186.8257
183	171.3896	178.8079	186.8257
183	171.3896	178.8079	186.8257
183	171.3896	178.8079	186.8257
183	173.8061	179.0287	186.7563
183	173.8061	179.0287	186.7563
186	165.5208	184.9917	193.2041
193	190.661	187.9602	193.9034
201	203.7871	186.3735	194.8236
182	202.5861	184.9721	192.3959
193	177.4965	184.12	167.7326
192	188.1598	183.8858	178.829
193	188.2027	183.9454	179.2963
193	189.4719	187.2432	183.4058
230	210.6172	220.3367	220.0148
229	210.3427	221.6763	222.7025
210	204.7382	200.8202	184.1487
209	225.8038	214.9605	211.2334

TABLE 4
ANN RESULT USING TRAINGDX ALGORITHM.

Actual Flow rate	ANN with Default MATLAB activation function and traingdx training method	ANN with tansig-logsig activation function and traingdx training method	ANN with traingdx training method and logsig-logsig activation function
183	189.0306	174.4177	203.8547
183	189.0306	174.4177	203.8547
183	189.0306	174.4177	203.8547
183	189.0306	174.4177	203.8547
183	190.9316	174.5444	203.0464
183	190.9316	174.5444	203.0464
186	175.1273	196.6333	207.0924
193	193.1001	200.7008	193.621
201	205.0969	195.9114	189.4919
182	215.5637	194.3293	187.1235
193	175.7653	151.0006	159.1366
192	187.1846	151.0021	164.4285
193	187.2125	151.0021	164.5291
193	188.2482	151.0018	163.5584
230	207.2949	223.707	187.2188
229	193.1041	214.659	190.4119
210	163.836	227.2707	193.9151
209	196.0743	227.7397	185.1539

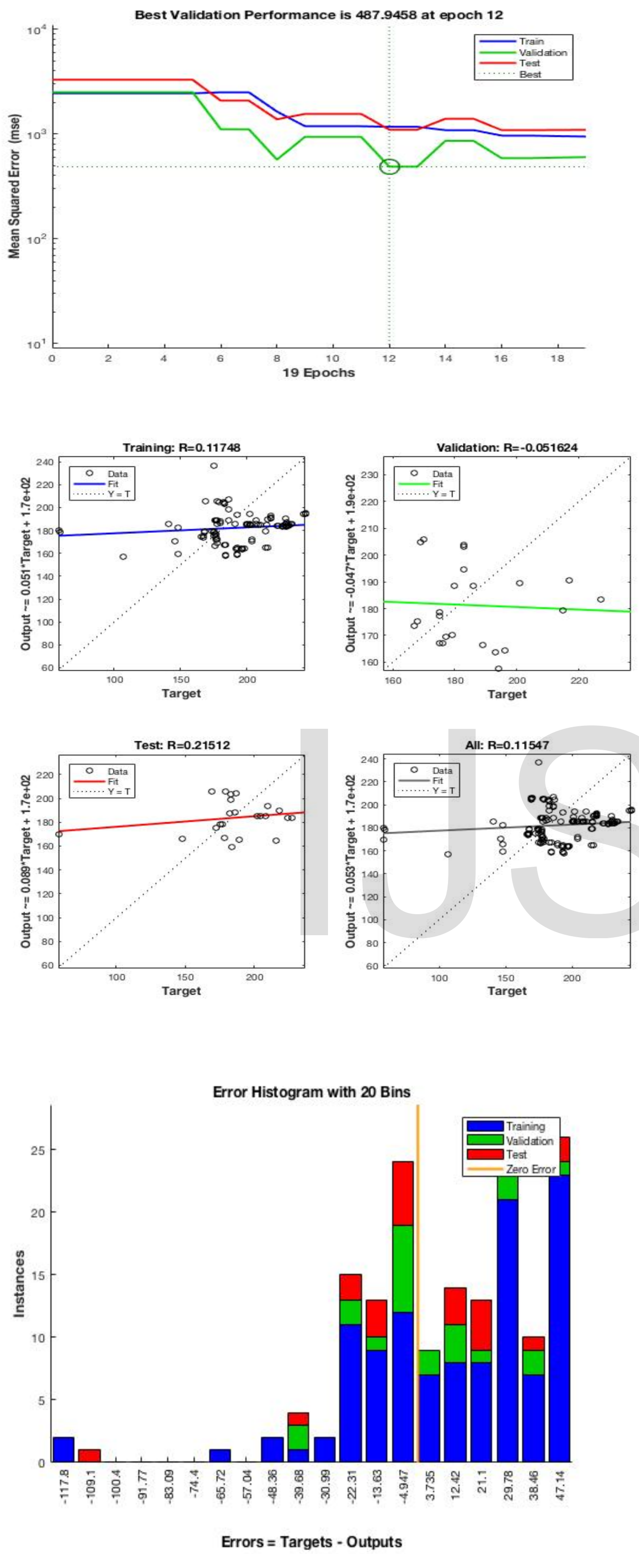


Fig.17. ANN Controller training, validation and testing using Traingdx Algorithm and logsig-logsig activation function.

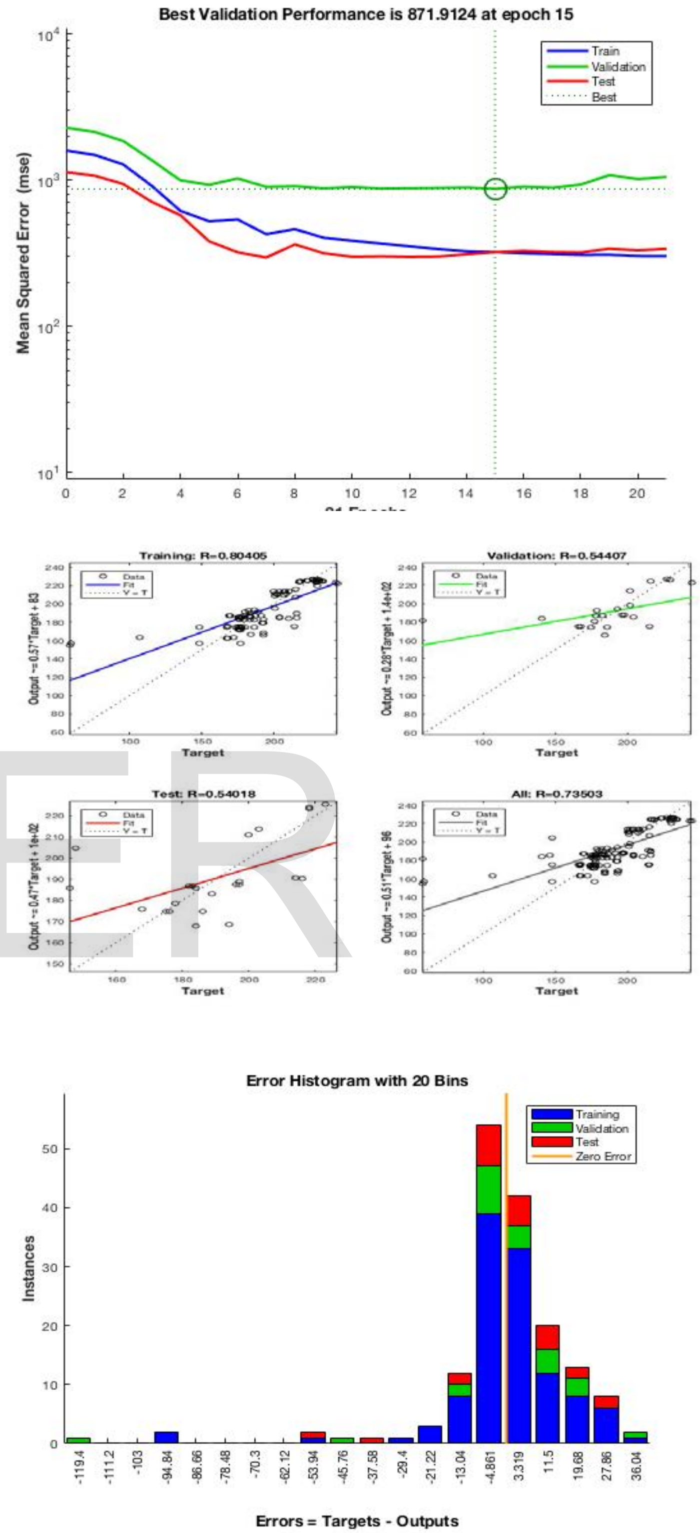


Fig. 18. ANN Controller training, validation and testing using Trainrp algorithm logsig-logsig activation function.

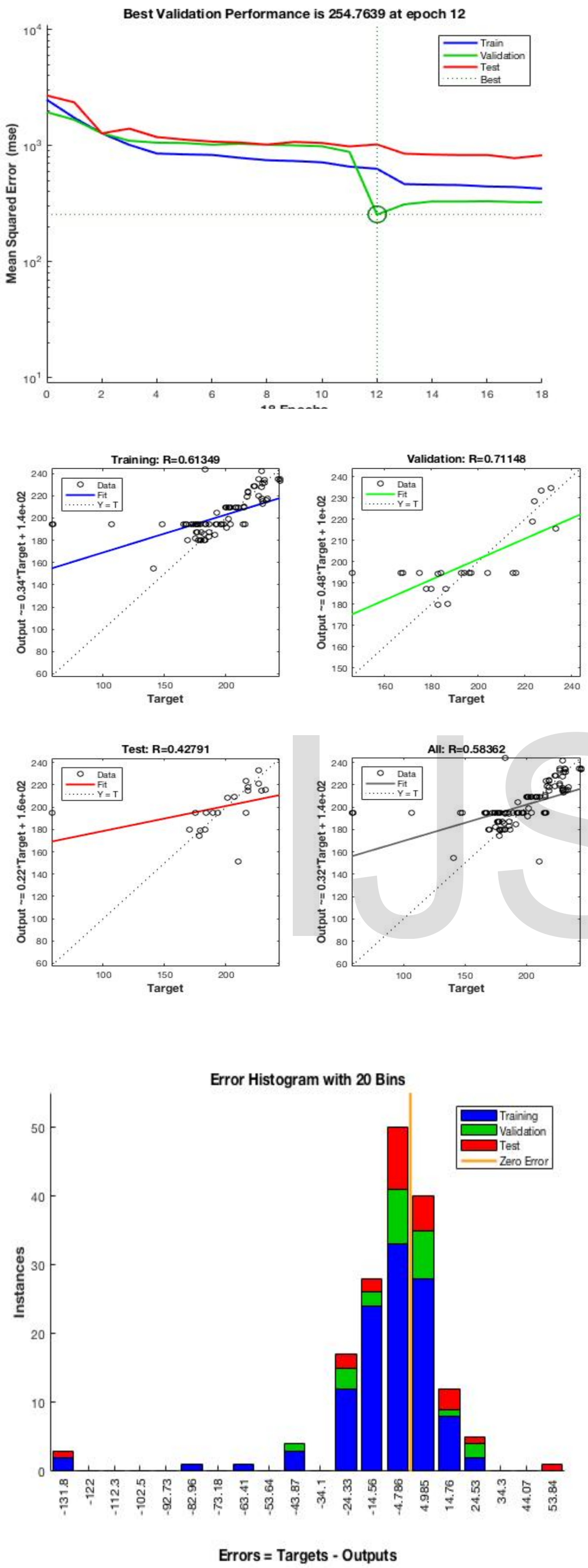


Fig. 19. ANN Controller Training, validation and testing using Trainlm algorithm logsig-logsig activation function.

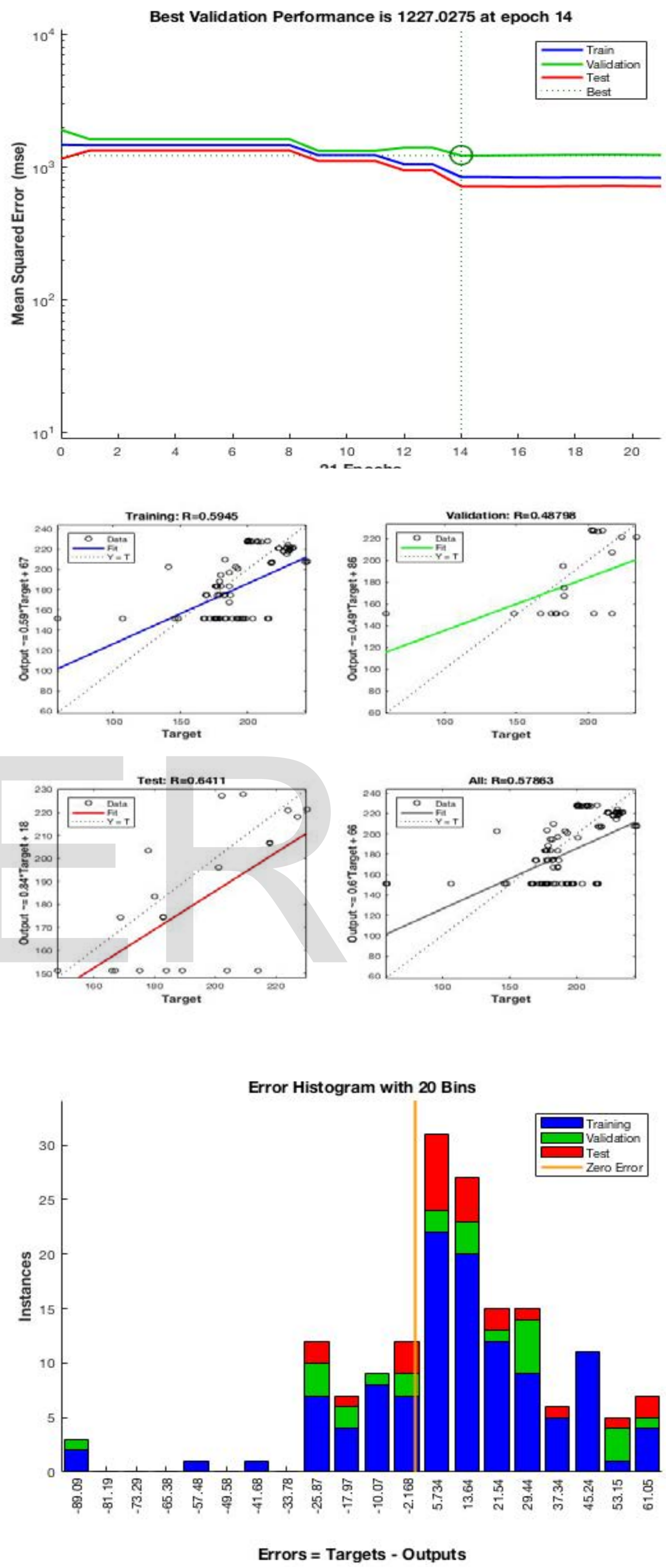


Fig. 20. ANN Controller Training, validation and testing using Traingdx algorithm tansig-logsig activation function.

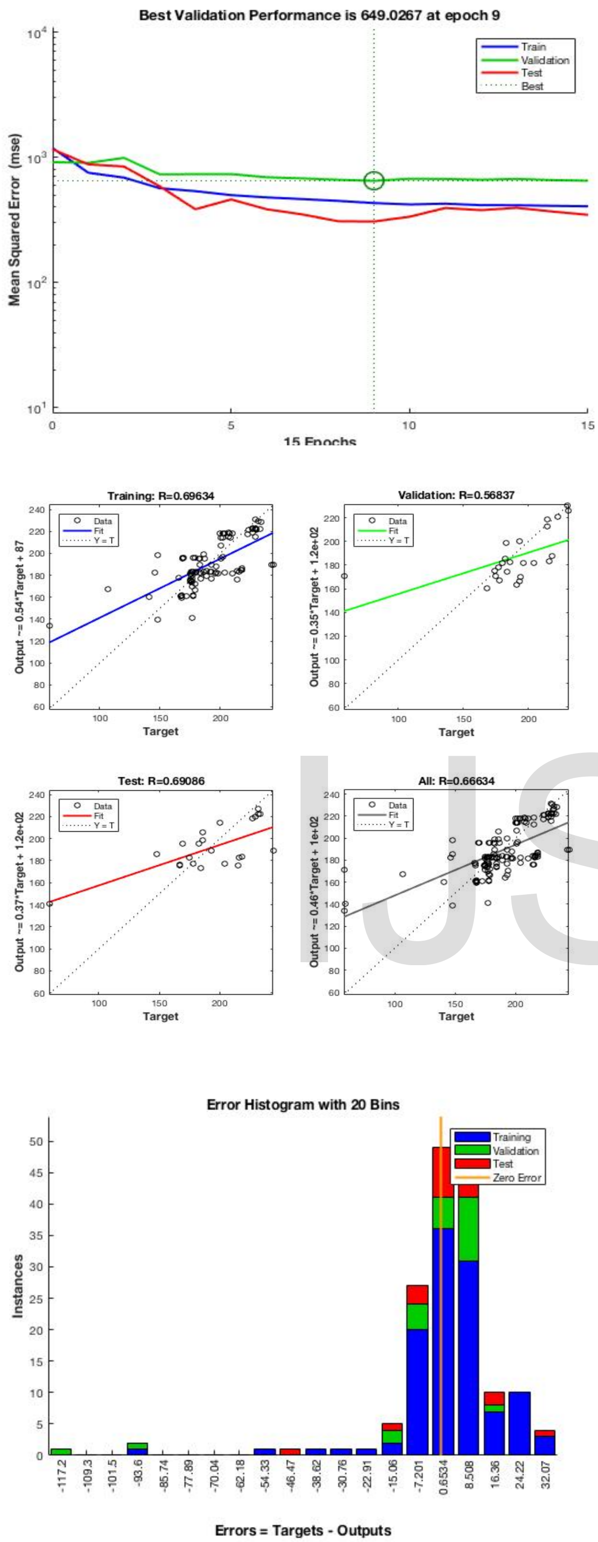


Fig.21. ANN Controller Training, validation and testing using Trainrp algorithm tansig-logsig activation function.

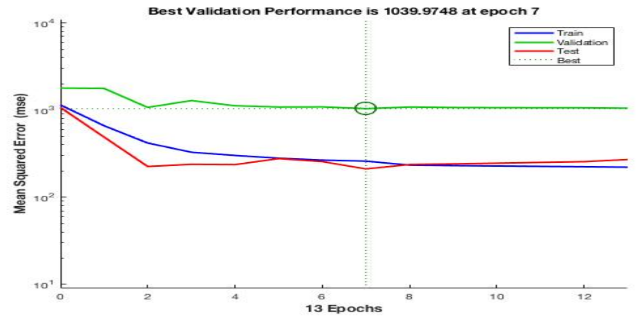


Fig.22. ANN Controller Training, validation and testing using Trainlm algorithm tansig-logsig activation function.

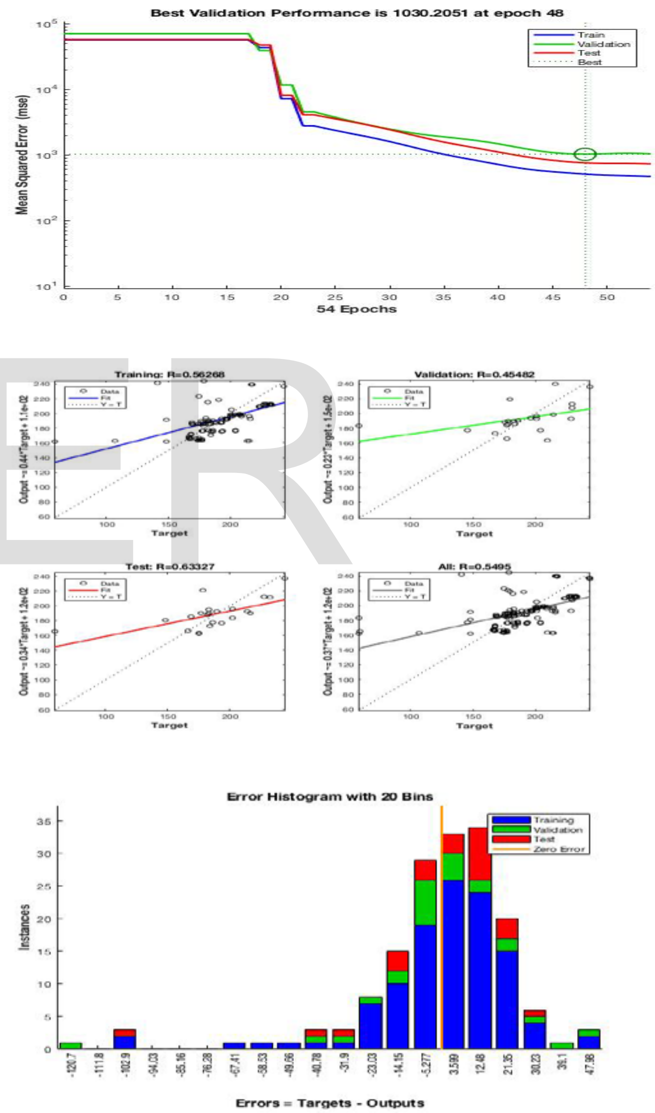


FIG.23. ANN CONTROLLER TRAINING, VALIDATION AND TESTING USING TRAINGDX ALGORITHM AND MATLAB DEFAULT ACTIVATIO FUNCTION.

ISSN 2229-5518

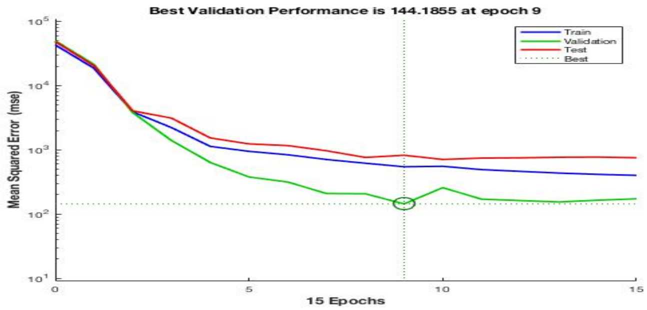


Fig. 24. ANN Controller Training, validation and testing using Trainrp algorithm and MATLAB default activation function.

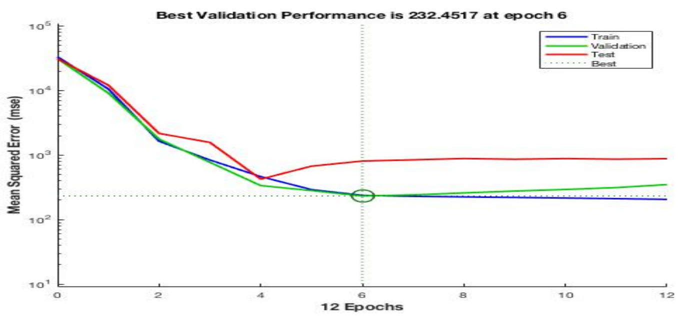
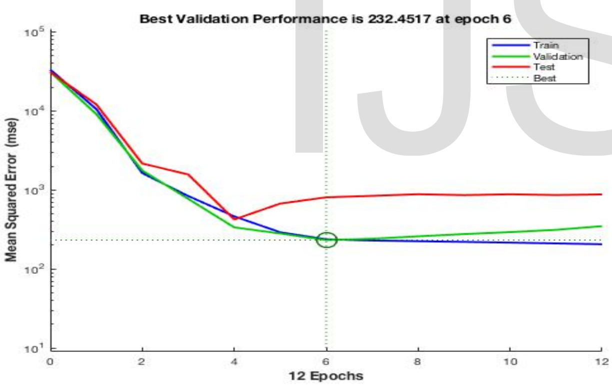
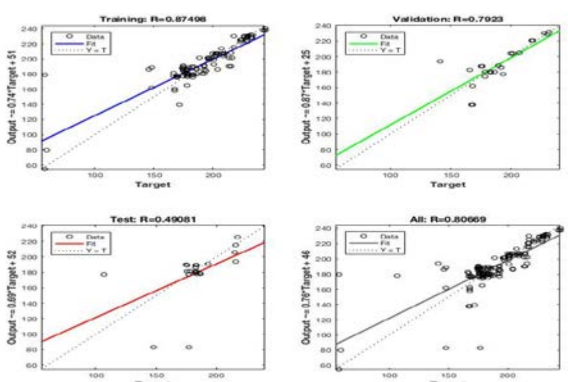


Fig.25. ANN Controller Training, validation and testing using Trainrp algorithm and MATLAB default activation function.





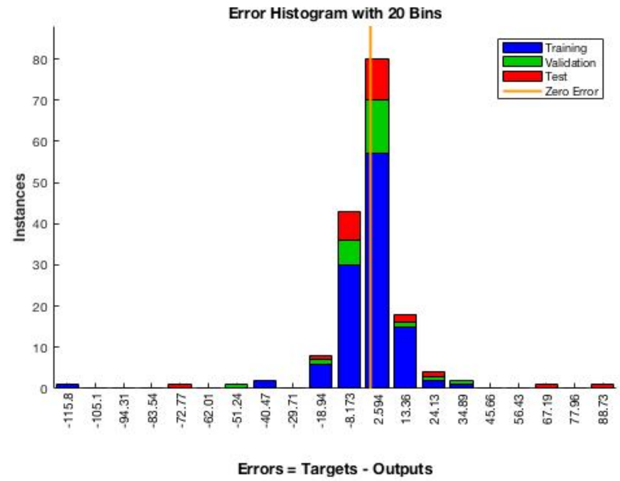


Fig.26 ANN Controller Training, validation and testing using Trainlm algorithm and MATLAB default activation function.

The design, testing and simulation of the EIA-485 node was carried out on Proteus ISIS schematic and simulation application software (Professional edition). In this implementation, two lines are used. One line for transmission and reception while the other line for control purposes. With this arrangement, only one node can transmit over the collaborative by holding the control line "high (or 1)". Thus prevent bus-contention on the network. When control line is set to "1", all other nodes "listen" to the transmit and receive line for scheduling and control programs. Figures 27-29 below show two node communicating over EIA-485 collaborative network. Other communication protocols can be implemented to established intelligent communication between neighboring nodes to regulate data rate of NCS using UDP over IP-network.

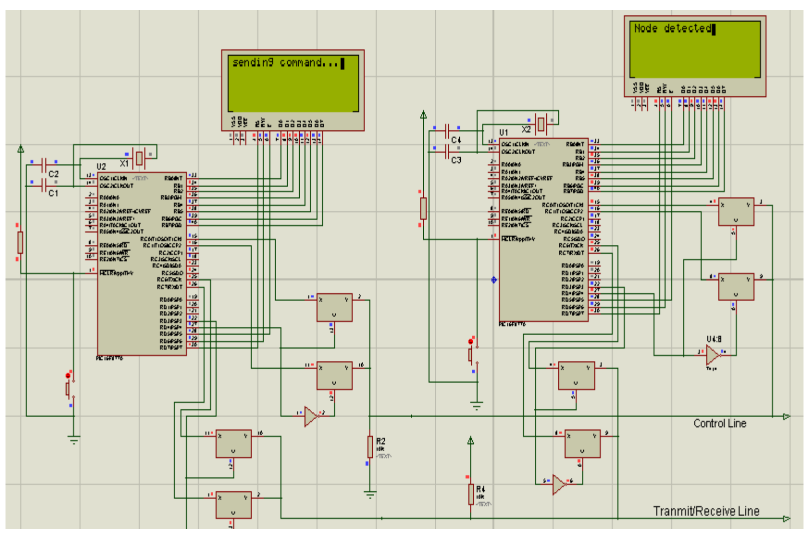


Fig.27. Two nodes communicating over collaborative network EIA-485.

ISSN 2229-5518

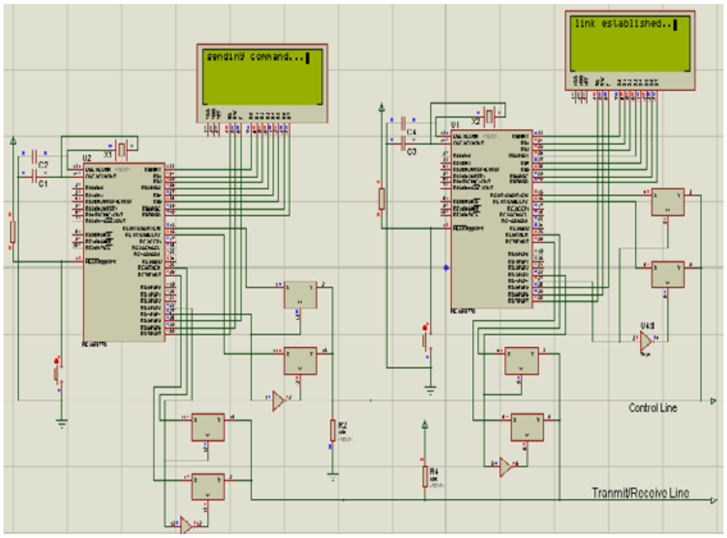


Fig.28. Two nodes establishing link over EIA-485 collaborative network.

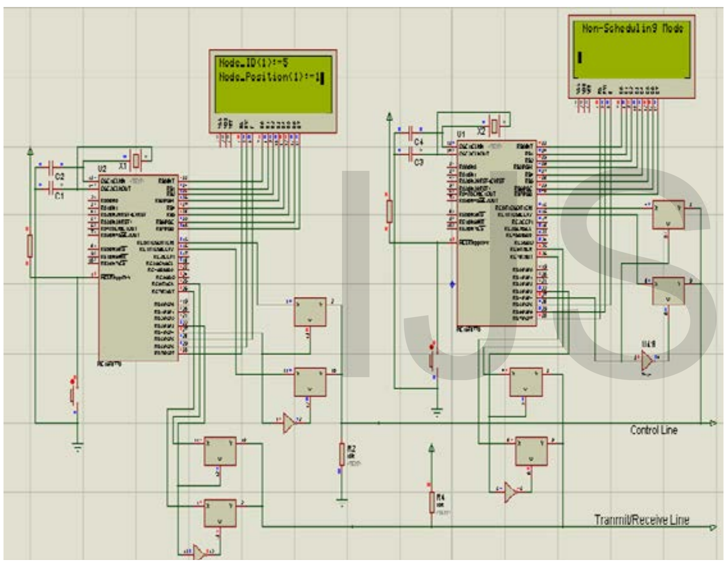


Fig. 29. Two nodes exchanging data over EIA-485 collaborative network.

5 CONCLUSIONS

The application of artificial neural network and collaborative network to the control of electrohydraulic flow control valve in a process industry provides a powerful method in flow regulation where several observable variables cannot be mapped mathematically to a single output. We have shown that training a neural network controller with a noisy (parameters at start-ups and shutdowns are not remove from the training data set) can produce a remarkable output to regulate the behaviour of a process, in this case, electrohydraulic valve. The results have shown that the choice of training algorithm and layer activation function affects the accuracy of Neural network controller.

The use of collaborative network has the potential to implement highly intelligent scheduling and prioritization algorithm over IP-based networks for networked control systems using UDP communication protocol. This provides an alternative way of achieving flow control mechanism of UDP packets For NCS.

References

[1] B.G.C. Marieke, V.W Nathan, W.P.M Heemels and H. Nijmeijer, “Stability of Networked Control Systems with uncertain Time-varying Delays”, IEEE Transactions On Automatic Control, Vol. 54, No.7, pp.1575-1580, 2009

[2] W.P. Maurice, H. Heemels, A.R. Teel, N.V.D Wouw and D. Nesic, “Networked Controls Systems with Communications: Trade-offs between transmission intervals, Delays and Performance”, IEEE transactions on Automatic Control, Vol.55, No.8, pp.1781-1796, 2010.

[3] L. Jetto and V. Orsini, “Relaxed Conditions for the Exponential Stability of a Class of Linear Time-Varying System”, IEEE Transactions on Automatic Control, Vol.54, No.7, pp.1580, 2009.

[4] X. Wang and M.D Lemmon, “Event Triggering in Distributed Networked Control System”, IEEE Transactions on Automatic Control, Vol.56, No.3, pp.586-601, 2011.

[5] I. Pan, S. Das and A. Gupta, “Tuning of an Optimal Fuzzy PID Controllers with stochastic algorithm for Networked Control System with random time-delay”, ISA Transactions, Vol.50, No.1, pp.28-36, 2011.

[6] Y. Tipsuwan and M.-Y. Chow, “Control methodologies in networked control systems”, Control engineering practice, Vol.11, pp. 1099-1111, 2003.

[7] M.-Y. Chow and Y. Tipsuwan, “Networked-based control systems: A tutorial”, IECON’01: The 27th Annual conference of the IEEE industrial electronics society, pp.1593 – 1602.

[8] S. Shokri, M. Shirvani, A.R. Salmani and M. Younesi, “Improved PI-Controllers Tuning in Time-delay Smith Predictor with Model Mismatch”, International Journal of Chemical Engineering and Applications, Vol.1, No.4, pp.290-293, 2010.

[9] N. Vatanski, J.P. Georges, C. Aubrun, E. Rondeau, and Jamsa-Jounela (2001). Control Compensation Based on Upper-bound delay in NCS. Retrieved from http://arxiv.org/ftp/cs/papers/0609/0609151.pdf (Accessed on 13th April, 2014).

[10] A. O’Dwyer, “A reference guide to Smith Predictor based methods for the compensation of dead-time processes”, ISSC, Dublin, September 1-2, 2005.

[11] S.H. Yang, X. Chen, L.S. Tan and L. Yang, “Time Delay and Data Loss Compensation for Internet-Based Process Control”, Transactions of the Institute of Measurement and Control, Vol.27, No.2, pp.103-118, 2005.

[12] M. Veronesi, “Performance Improvement of Smith Predictor through Automatic Computation of Dead-Time”, Yokogawa technical Report, English Edition, No.5, pp.25-30, 2003.

[13] V. Rajinikanth and K. Latha,”Tuining and Retuning of PID controllers for stable using Evolutionary Algorithm”, International Scholarly Research Network, ISRN, Chemical Engineering, Vol.2012, pp.1-10, 2012.

[14] S. Vardhan and R. Kumar, “Simulation for Time-delay Compensation in Network Control System”, Cyber Journals: Multidisciplinary

[15] Journals in Science and Technology, Journal of Selected Areas in Telecommunication (JSAT), pp.38-43, 2011.

[16] C.C. Hang, “Smith Predictor and its Modification Control System, Robotics, and Automation”, Encyclopaedia of Life Support System-Vol. II.

[17] A.I Al-Odienat and A.A. Al-Lawana, “The advantages of PID fuzzy controllers over the conventional Types”, American Journal of Applied Science, Vol.5, No.6, pp.653-658, 2008.

[18] M. Gaddam and R. Akula, “Automatic Tuning of PID Controller using Fuzzy Logic. 8th International Conference on Development and Application Systems, Suceava, Romania, pp.120-127, May 2006.

[19] V. Tsoulkas, “Networked Control Systems with Delay”, 2nd International conference on Computational Intelligence Communication Systems and Networks, Liverpool, Uk, 29 July, 2010.

[20] H. Gao, T. Chen and J. Lam, "A new delay system approach to network-based control", Automatic, Vol.44(2008), pp.39-52, 2007.

[21] F.-L. Lian, J. Moyne and D. Tilbury," Networked design consideration for distributed control systems', Vol.10, No.2, pp.297-307.

[22] J.F. Kurose and K.W. Ross, Computer networking: A top-Down Approach (6th Ed.). Boston: Pearson Education, pp. 452-453, 2013.

[23] R. Ahlswede, N. Cai, S.-Y.R. Li and R.W. Yeung, "Network infor`mation flow", IEEE transactions on information theory, Vol.46, No.4, pp. 1204-1216, 2000.

[24] S. Shakkottai and R. Srikant, "Network optimization and control", Foundation and trends in networking, Vol.2, No.3, pp.271-379, 2008.

[25] L.S. Coelho, " Tuning of PID Controller for an Automatic regulator voltage using Chaotic Optimization Approach", Chaos, Solitos and Fractals, 39, 1504-1514, 2009.

[26] M. Khalid and S. Omatu, "A Neural Network Controller for a Temperature Control System", *IEEE Control System*, 12(3), pp.58-64, June 1992.

[27] I.J Nagrath and M. Gopal (2009). Control Systems Engineering. 5th Ed. New Delhi: New Age International Publishers.

[28] J. Zhong-Ping and L. Praly,"Design of Robust Adaptive Controllers for Non-linear Systems with dynamic uncertainties", *Automatica*, 34(7), pp.825-840, 1998.

[29] Jain, A.K., Mao, J. and Mahiuddin, K.M. (1996) Artificial Neural Networks: A tutorial [Online]. Available from:http://web.iitd.ac.in/~sumeet/Jain.pdf(Accessed on 13th March, 2013).

[30] D. Psaltis, A. Siderie,and A.A Yamamura, "Multilayered Neural Network Controller", *Presented at IEEE International Conference on Neural Networks (1987)*, *IEEE Control Magazine*, pp.17-21, April 1988.

[31] R. Rojas .Neural Networks; A systematic Introduction. New York: Springer-Verlag, 1996.

[32] S.S. Ge, C.C. Hang and T. Zhang," Adaptive Neural Network Control of non-linear Systems by State and Output feedback", IEEE transactions on System, Man, and Cybernetics-PartB: Cybernetics, 29(6), pp.818-820, December 1999.A.U. Levin and K.S. Narendra, "Control of non-linear dynamical systems using Neural Network", IEEE transactions on Neural Network, 7(1), pp.30-42, 1996.

[33] P. Sehgal, S. Gupta, S., and D. Kumar, "Minimization of Error in Training a neural network using gradient descent method", *International Journal of Technical Research (IJTR)*, 1(1), April, pp.10-12, April 2012.

[34] S. Verdu, "Fifty years of Shannon theory", IEEE transactions on information theory, Vol.44, No.6, pp.2057-2078, 1988.

[35] M. Husken, Y. Jin. and B. Sendhoff, B., "Structure Optimization of Neural Networks for Evolutionary Design Optimization", *Soft Computing*, Vol.9, No.1, pp.21-28, 2005.

[36] H. Hakimpoor, H. Tat, and M. Rahmandoust, "Artificial Neural networks", Applications in Management, *World Applied Sciences Journal*, Vol.14, No.7, pp1008-1019, 2011.

[37] C. Lipe-Heffron (2012) Programming the Brain Neural Network. Retrieved from http://www.braincentersnw.com/concussion-recovery-reprogramming-typical-and-atypical-brains/.

[38] O. Razor (2014). Brain. Retrieved from http://www.abc.net.au/radionational/programs/ockhamsrazor/brain/5362454.

[39] M.A. Nielsen, "Neural Network and Deep learning", Determine Press, 2015.

[40] Anom. (1999) Understanding EIA-485 Networks: A technical Supplement to Control Network [Online]. Retrieved from: http://www.ccontrols.com/pdf/ExtV1N1.pdf (Accessed on 22nd April, 2014).

[41] Chipkin Automation (2007) What is RS-485, EIA-485 [Online]. Available from:http://www.chipkin.com/what-is-rs485-eia-485/(Accessed on 12th April, 2013).

[42] Industrial Ethernet Book Issue 49/40, 2016. Retrieved from http://www.iebmedia.com/?hpid=1&parentid=59&themeid=248.

[43] J.F. Kurose and K.W. Ross, Computer networking: A top-Down Approach (6th Ed.). Boston: Pearson Education, pp. 452-453, 2013.