

Implementation of a New Neural Network Function Block to Programmable Logic Controllers Library Function

Hamid Abdi, Abolfazl Salami, and Abolfazl Ahmadi

Abstract—Programmable logic controllers are the main controllers in the today's industries; they are used for several applications in industrial control systems and there are lots of examples exist from the PLC applications in industries especially in big companies and plants such as refineries, power plants, petrochemical companies, steel companies, and food and production companies. In the PLCs there are some functions in the function library in software that can be used in PLC programs as basic program elements. The aim of this project are introducing and implementing a new function block of a neural network to the function library of PLC. This block can be applied for some control applications or nonlinear functions calculations after it has been trained for these applications.

The implemented neural network is a Perceptron neural network with three layers, three input nodes and one output node. The block can be used in manual or automatic mode.

In this paper the structure of the implemented function block, the parameters and the training method of the network are presented by considering the especial method of PLC programming and its complexities. Finally the application of the new block is compared with a classic simulated block and the results are presented.

Keywords—Programmable Logic Controller, PLC Programming, Neural Networks, Perception Network, Intelligent Control.

I. INTRODUCTION

DEVELOPING the new control strategies and implementation and extending the new calculation functions in PLC systems are the interesting subjects for PLC manufacturers. Manufacturers presented new functions by introducing their new PLCs.

They do try to improve their last functions, simple to understand for applying for PLC users [6],[7]. So declaration of the new function blocks is an applicable research field in industrial control, however industrial applications of the neural network are limited by needs of the high level engineering knowledge in control, intelligence and level of automation. Moreover it needs to increase the reliability of neural network function block especially in industrial applications.

In PLCs, there are several mathematical and operational function blocks which can be used easily but the neural network function block has not be used in PLC systems [1],[6],[7]. In this paper in the first section the importance of the subject and prior art are discussed.

Then in part II and III neural networks are reviewed and next in Part IV PLCs and PLC programs are studied and in part VI and VII structure of the implemented neural network for PLC application and its parameters are introduced then in part V the function block operation is analyzed in comparison with a simulated system and finally some conclusion have been made and the results are presented.

II. NEURAL NETWORKS

In an artificial neuron, input information and parameters excite the input channel of a mathematical model of the real neuron as an electric signal. Then these input signals are multiplied by the synapse weights (Weight) and the effect of them with impact of the other parameter (Bias) is added and excites the output of the neuron. If the summation is enough for neuron due to the decision function of the neuron cell. So it produces a constant current in the output paths of the neuron [3], [6]. A neural network is a network of neurons which including some mathematical and graphical models. These models are used to show the structure of artificial neural network.

There are two main topologies in neural network. The first one is feed forward and the second one is feedback topology. Most of the recent applications of neural network use the feed forward topology. The most conventional method for their training is the back error propagation method [4],[5]. Perceptron is one of the feed forward neural networks. Fig. 1 shows a feed forward neural network.

Multilayer Perceptron networks are another example of feed forward neural network which has several applications in complex mathematical problem solving and nonlinear equation solving. Multi layer Perceptron neural network can be used by proper selection of the network parameters and decision functions [3],[2] for a lot of problems .

In these networks there are an input layer as an input part of the network, an output layer as the output part of the network and some hidden layers which are used between input and output layers. Fig. 1 shows this network structure. In these networks the information passes straightforward the network to produce the output. There are considerable contributions on the use of Perceptron neural networks for industrial control or calculation problems. [3], [4]

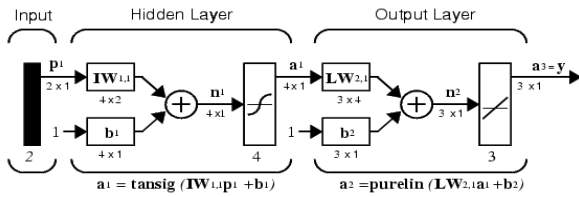


Fig. 1 Multi layer Perceptron neural network

TABLE I
TRAINING ROUTINE OF THE PERCEPTRON NEURAL NETWORKS

1	$E(w) = \frac{1}{2} \sum_{j=1}^p (y_j - d_j)^2$	y_j : network j -th output value, d_j : target value of j -th output, p : number of neurons in the output layer
2	$\Delta w_{ij} = -\eta \cdot \frac{\partial E}{\partial w_{ij}} \quad \Delta w_{ij} = -\eta \cdot \frac{\partial E}{\partial w_{ij}}$ $\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial y_j} \cdot \frac{\partial y_j}{\partial s_j} \cdot \frac{\partial s_j}{\partial w_{ij}}$	y_j - j -th neuron output value, s_j - weighted total of output signals determined by the formula (1).
3	$\frac{\partial s_i}{\partial w_{ij}} = x_i$	x_i is the neuron i -th input value
4	$\frac{\partial E}{\partial y_j} = \sum_i \frac{\partial E}{\partial y_i} \cdot \frac{\partial y_i}{\partial s_j} \cdot \frac{\partial s_j}{\partial y_j} = \sum_i \frac{\partial E}{\partial y_i} \cdot \frac{\partial y_i}{\partial s_j} \cdot w_{ij}^{(n)}$	k is the number of neurons in layer $n+1$
5	$\delta_j^{(n)} = \frac{\partial E}{\partial y_j} \cdot \frac{\partial y_j}{\partial s_j}$	$\delta_j^{(n)}$ n -th layer if we know $\delta_k^{(n+1)}$ of the next $(n+1)$ layer.
6	$\delta_j^{(n)} = \left[\sum_k \delta_k^{(n+1)} \cdot w_{kj}^{(n+1)} \right] \cdot \frac{\partial y_j}{\partial s_j}$	$\delta_j^{(n)}$ for the last layer of the neural
7	$\delta_j^{(0)} = (y_j^{(0)} - d_j) \cdot \frac{\partial y_j}{\partial s_j}$	Network the vector of those values that the neural network must generate with the given set of input values.
8	$\Delta w_{ij}^{(n)} = -\eta \cdot \delta_j^{(n)} \cdot x_i^n$	And finally let's write down the formula (6) in expand form

III. PROGRAMMABLE LOGIC CONTROLLERS

PLCs are the most applied controllers in industrial application especially for interlocking and supervising control to make sequences of operation or safety logics.

However the today's PLCs have control abilities for applying in continuous systems (like PID facility).

It should be noticed that there are more than 100 manufacturers in the PLC forum. They produce various types of PLCs in different sizes. PLCs are categorized in three main groups including small PLCs, medium PLCs and large PLCs. The small sizes have almost less than 40 inputs/outputs, medium sizes have almost less than 100 I/Os and the large sizes have more than 500 I/Os. Figs. 2 and 3 show the application strategy of a large PLC for automation and monitoring purposes.



Fig. 2 A large PLC

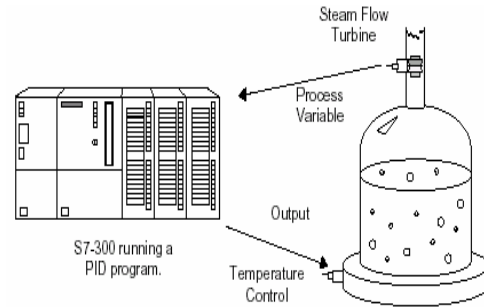


Fig. 3 PLC application terminology

It should be noted that for comparing PLCs there are several other items. Some characteristics such as CPU, Cycle time, simplicity, reliability, expansion capability, peripherals, networking features are some considerable items.

The PLC systems can be programmed with three languages. These are Ladder programming (LAD), function block diagram programming (FBD) and statement list programming (STL). These methods of programming vary in different PLCs but they have a little difference between most of the PLCs. To write a structural PLC program, we have to use some basic instructions or some programmed blocks. It also needs to write some new blocks which should be programmed for the application.

The organization blocks (OB), program blocks (PB) and some function blocks (FB, FC) are some of these blocks in Siemens PLCs. For example OB blocks are function blocks manage the PLCs and they act like an operating system of a computer. They control PLC operation sequences and hardware parts of the PLC.

The PB blocks are program blocks; they can be used as the basic applied control program or the subroutine of the user programs. The FC blocks are basic applied functions in the PLC systems. They perform a simple software or hardware task which generally can be used by the user program. Some of these blocks have been designed or reserved by PLC manufacturers and some others can be programmed by the user of the PLC due to the applications. Usually FC blocks are used in PB Blocks; PB and FC blocks are used in OB blocks. It should be noted that the main program which calls the other PB blocks or FB Blocks or even FC Blocks should be in OB1 block. One of the function blocks which have been used as a classic controller is shown in Fig. 4.

This structure is used in Siemens S7 PLCs generation; in the same manner other PLCs use the like structures for programming. The differences between these PLCs and other manufacturer's products are less than almost 10%. The differences are only in the names of the blocks and some indicative symbols or methods of combination.

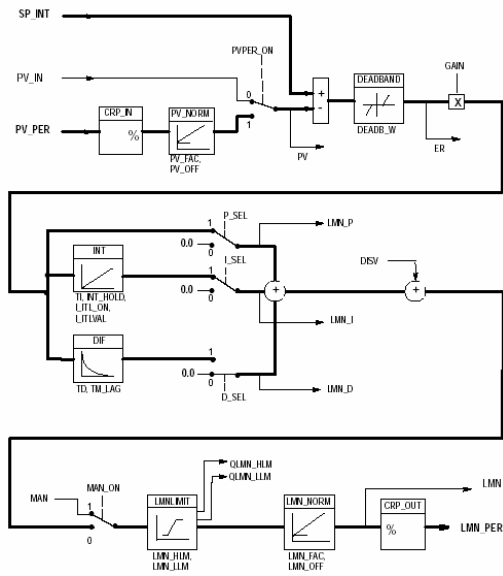


Fig. 4 Classic PID function block in PLC

IV. IMPLEMENTATION OF NEURAL NETWORK FUNCTION BLOCK

The implemented block is a three layers Perceptron neural network. The block includes three input nodes and seven nodes in second layer and one node in output layer. The block in PLC S7300 has two automatic and manual modes. In the manual mode the parameters of the network are declared manually in the program (W01...W21, B01...B21). They are set by the user when the network is used. However if they are not defined some uniform random number are selected. In the automatic mode parameters of the network are calculated automatically by the back error propagation training method.

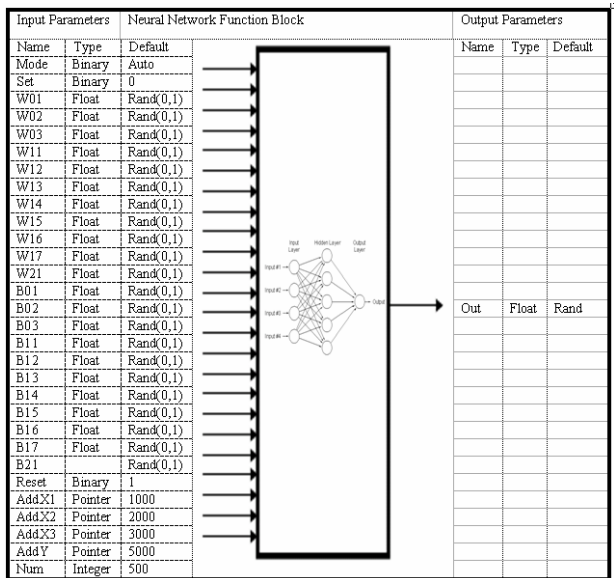


Fig. 5 Implemented function block

It is the most conventional method for training of these networks. The network is completely trained after almost 10 minute of the time which the PLC is placed in RUN mode (for 500 data for training). The training data should be sorted in some parts of the PLC memory and it makes a matrix with four vectors including three column vectors as input data and one other vector as the output date of the network. The inputs and output data are used for training the network parameters. Every training sequence is calculated in one machine cycle of the PLC system. (The machine cycle is almost 1 microsecond for our test program).

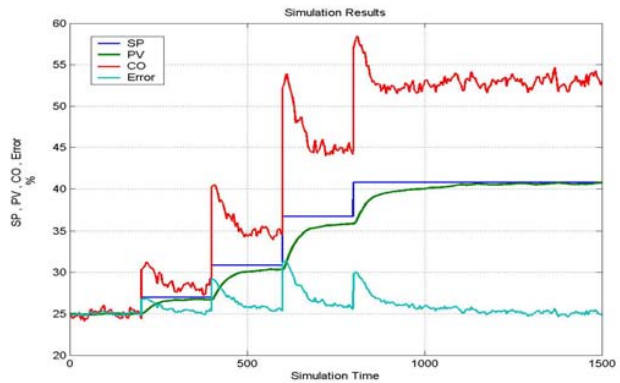


Fig. 6 Neural network block output compare to PID output

V. OPERATION

To test the designed network in a siemens PLC, the behavior of a PID controller for a first order simple model is studied in Matlab software. The model is used to extract data to form four vectors, the samples are gathered from almost 8 minutes analyzing of the PID controller on the model with 1000 sampling points. Table II shows the sampled data variables and related their considerations. The information of the PID controlled systems transferred to the desired places in PLC memory to make the training matrix for the neural network. Then the block is trained by the matrix. After that the behavior of the network is compared with the PID controlled in the model to analyze the operation of the implemented neural network function block. Fig. 6 compares the Function block in PLC with the PID controller due to applying a five level set point both in model and PLC.

TABLE II
TRAINING MATRIX VARIABLES

Type	Name	Address	NN Name
Input	Set Point	MW1000-1999	X0
	Error	MW2000-2999	X1
	Control Output	MW3000-3999	X2
Output	Process Variable	MW5000-5999	Y

VI. CONCLUSION

In this paper the structure of neural networks by aiming of the implementation in the industrial PLCs are studied. The new function block is introduced for the PLCs applications and the operation of the block is studied too. Then the accuracy of the implemented block is investigated regarding to the calculation and programming limits of the PLCs. In addition the back error propagation training method is implemented by PLC software as a complementary PB. Then the implemented function is analyzed by using a computer controlled model. The presented results show that the function performs the desired operation. It should be noted that this application is gathered based on the fully industrial controller

ACKNOWLEDGMENT

The authors would like to appreciate NIOC and Rahavaran's Automation Laboratories and IUST University, Arak Faculty of Engineering.

REFERENCES

- [1] Advances in automatic control, Author VOICU, Publication date: 10-2003.
- [2] Goldman, RM. (1996) *Mathematical Methods for Neural Network Analysis and Design*. MIT Press. ISBN: 0262071746.
- [3] Hinton, G.E., 1992, How neural networks learn from experience, *Sci. American*, 267, 144-151.
- [4] *Neural Network Toolbox User's Guide, "MATLAB User Manual"*, Math Works Inc 1992-2002.
- [5] Beale & Jackson (1990) *Neural Computing: An Introduction*. Inst. Phys. ISBN: 0852742622.
- [6] *Technical Manual References of Siemens PLC- S7300*
- [7] *Technical Manual References of Siemens PLC- S7400*
- [8] *Technical Manual References of Omron PLC- CVM*
- [9] Hamid Abdi, "Control in oil Industries Review and Technology Trend", Sep 2003, Sharif University of Technology, control in oil industry.

Hamid Abdi, Faculty member of Iran University of Science and Technology, Arak Engineering Faculty.
 Member of Iranian rescue robot winner the 2nd rank in Seattle University, July 2001.
 Winner of the golden medal for of national sheikh bahaei festival, Isfahan, 2005.
 UNESCO Certificate for the best national entrepreneur, 2005.
 Best IT Entrepreneur in Markazi Province.
 Inventor of the three national patents and one international Patent, in field of robotic and safety robots "Road Robots".
 Rank 9th in mathematics Olympia for Iranian student, Markazi State, 1991
 Rank 11th in B.Sc. public examination, region 2, 1992.
 Rank 17th in M. Sc. public examination, 1997.
 Member of best group in internship project, Sapco, summer 1998.
IEEE Reviewer at the 2007 IEEE Intelligent Vehicles Symposium, June 13-15, 2007, Istanbul, Turkey.
IEEE Reviewer at the 2007 IEEE Intelligent Transportation conference, ITSC2007, Sep 2007, Seattle, USA.