

Implementation of an Intelligent Algorithm in a PLC: A Practical Approach

Arturo Vázquez Gálvez^a, Oscar Mario Rodríguez Elias^a, Guillermo Valencia Palomo^a, Germán Alonso Ruiz Domínguez^a

^aDivisión de Estudios de Posgrado e Investigación, Instituto Tecnológico de Hermosillo
Avenida Tecnológico, El Sahuaro, Hermosillo, Sonora, México, 83170.
arturov47@yahoo.com.mx, {[omrodriguez](mailto:omrodriguez@ith.mx), [gvalencia](mailto:gvalencia@ith.mx), [gruiz](mailto:gruiz@ith.mx)}@ith.mx

2013 Published by *DI^FU₁₀₀ci* @ <http://www2.uaz.edu.mx/web/www/publicaciones>
Selection and peer-review under responsibility of the Organizing Committee of the CCOMP-2013, www.cicomp.org

Abstract

This paper focuses on finding a way to implement an intelligent algorithm in an Allen Bradley PLC. It demonstrates such implementation with the purpose of improving a process, particularly water level control. The processing module followed during this research was the artificial neural networks (ANN). To achieve the programming, MATLAB was selected as the main support tool due to the powerful modules that it offers. In the paper, the steps that were necessary in order to achieve the proposed objective are described.

Keywords: Intelligent Algorithms, Neural Networks, Programmable Logic Controller, Intelligent Control

1. Introduction

Through history mankind has shared an interest on creating a machine that could imitate human behavior. It's not surprising that initial projects where conducted by the military during World War II [1]. Advances on automation technologies for industry solutions have been evolving at a great pace since the industrial revolution. Without a doubt one of the main "devices" that have had major impact on the industry is the Programmable Logic Controller (PLC). Since its beginnings on the 70's it has offered great advantages in process optimization that could not be done before. One of its main advantages is its adaptability to different

process at low cost. Even when the PLC performs an extraordinary work in diminishing the workload, industrial processes have become more complex, require to be performed at greater speeds and have better accuracy. Additionally, processes are submitted to revisions more frequently, which cause a shift in the variables involved [2]. To tackle these issues, new techniques have been implemented in order to reach better solutions; intelligent algorithms are among these new techniques.

Intelligent systems offer a great diversity of solutions; one of the most popular techniques for developing intelligent systems in industry during the last years is the artificial neural networks (ANN). ANN have been applied to several fields, like: medicine, biology, economy,

software development, etc. [3]. The conjunction of an intelligent algorithm in a PLC has found a solution to problems found every time more frequently as the processes become more complex. These problems relate to the fact that commonly PLC processor run the program based on variable values without any adaptability, even though the variables of the process have changed. Therefore, by implementing an intelligent algorithm the solution could adapt itself to changes occurring in the process.

In this paper we look for a solution that aids us to implement an intelligent algorithm, based in ANN, to a PLC that does not have a specialized module for it. The PLC model used in this research was the Compact Logix Allen Bradley #1768L43.

Following, in this paper, there is a brief explanation and definitions on different subjects that will be referred during the rest of the paper. Then the tools used on the experiment are listed and explained, to later present the steps required from the experiment that achieves the objective of the research. Finally the conclusions of the research are depicted.

2. Theoretical Framework

For choosing the intelligent algorithm to be used in this work, and to define the method for implementing it into a PLC, a literature survey was conducted following two main objectives:

1. To identify the type of intelligent algorithms being used to aid in industrial processes.
2. To identify the best way for programming the intelligent algorithm into the PLC.

Next we describe the main results of this literature survey. Particularly, we found the ANN as the best way to show the manner in which an intelligent algorithm can be programmed into a PLC for supporting complex process control. As well, we identified that ANN are not commonly programmed into PLCs, but are programmed in regular computers or in special hardware. So we required investigating and defining a method for implementing an ANN into the chosen PLC, which is described later in this paper.

2.1. Intelligent systems in industrial PLCs

The subjects that were analyzed belonged to article publications, books and academics projects that implement an intelligent algorithm in to a PLC in order to find solutions to real industry problems. The criterion that was followed in order to filter the information was search

terms, title content, abstract, etc. The results concluded that there are several ways to implement an intelligent algorithm into a PLC, which can be achieved according to the program languages that the PLC accepts.

The type of intelligent algorithms that are being implemented varies depending on the situation that is being managed, for example, genetic algorithms were employed to manage comfort variables (temperature, illumination, air quality, etc.) in the control of an intelligent building [4].

ANN were implemented to manage the control of a character recognition system [5]. Another project that implemented ANN used them for failure prevention in a conveyor belt, which only had a PLC to control the process. The processing capacity of the PLC was improved since an ANN can run its process in parallel, which caused a significant reduction in the processing time of the variables [6].

There are other types of algorithms that follow different techniques such as fuzzy logic, predictive control, etc. And the solutions are not restricted to implement only one type of algorithms; integrations of two or more intelligent algorithms can also be achieved. The benefits obtained vary depending on the areas that are being worked on, among the more relevant we can find are energy saving, time improvements, battery consumption, among others.

Since it is a relatively new method of improvement, documentation to achieve the implementation of an ANN into a PLC it is currently scarce. With the information and tools that were available at the moment in this project to achieve the objective, the ANN was selected as the intelligent algorithm that was implemented in a PLC to manage the level control in a cylindrical tank.

2.2. PLC Programming

Through its history the basic programming form of a PLC has been ladder programming; this language provides a graphic representation of the selected elements, making it easy to understand [7]. The graphic representation of the ladder programming is similar to the one that can be found in a relay circuit, this easy to understand language was made by design so its acceptance was wider. This wide acceptance helped to break the paradigm that existed at the time in process control [8]. The IEC 1131-3 norm was created by the International Electronic Commission to standardize the distinct languages that were being applied to program a PLC by several brands and have a broader compatibility between devices notwithstanding the manufacturer [9].

Among the basic types of programming one can find

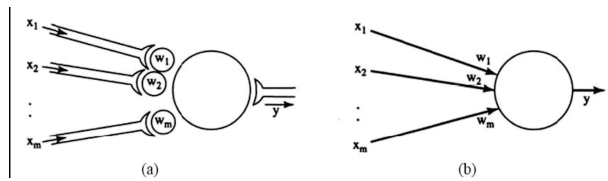


Figure 1. Neural model. a) Natural form representation. b) Abstract representation [13].

the following:

- Block Diagram, mostly used in control systems in sequences.
- Structured Text, used when the operations described in its programming are repeated constantly.
- Ladder Diagram, recommended for systems based on events.

The PLC used in this project manages RSLogix 5000 as its programming software; taking advantage of the Ethernet communication port of the PLC, and as it will be shown on chapter 4.2, the created ANN was imported as an “add-on” block to RSLogix 5000. To establish the communication between the PC and PLC, Matrikon OPC and OPC module from MATLAB were used so all the variables could be transferred. Ladder diagram language was used as the programming language to manage the variables in the control system as well as the ones inside the created ANN.

2.3. Artificial Neural Networks

The “traditional” concept of a neural network is utilized to describe the functions of the way human brain processes information that it receives through the human senses, so it can take a pertinent decision [10]. These decisions will improve according to the experience that a person gathers every time she faces the same situation. An ANN can also “learn” to take pertinent decisions by adjusting declared variables [11]. An ANN is composed of artificial neurons that can be related as nodes in an electrical circuit. These neurons are the main component of an ANN. Figure 1 presents a graphical interpretation of a single neuron model, where it can be noticed the distinct input variables (X_1, X_2, \dots, X_m) that stimulate the neurons depending on the declared weights (W_i). Inside a neuron there is a “black box” where the neuron generates a function with an output variable (Y) [12].

The following are several characteristics that ANN can offer when compared to other tools [13]:

- Applying an ANN is rather simple as its interpretation.

- Unlike statistical methods, ANN does not need to know every single data that is going to be involved in the system.
- An ANN offers the opportunity to divide its network into distinct parallel circuits, which allows having faster responses.
- An ANN can adjust the results when the data being received is updated.

It was necessary to analyze the characteristics of the particular PLC that is going to be used. In this research was the PLC 1768L43 compact Logix from Allen Bradley. According to its manual it is compatible with the IEC 1131-3 norm.

3. Methods and Materials

By following the IEC 1121-3 norm, the PLC used on this research allows its processor to be programmed by the distinct languages listed on Table 1, where some of their characteristics are mentioned.

Among the characteristics of the PLC that helped to reach the objective on this research, the following can be noticed: the PLC version and brand are compatible with MATLAB and even has a specialized option to select it in the OPC module. Additionally, Matrikon OPC also has a specialized module to configure PLC of the used brand. By using RSLogix 5000 to program the PLC, it was possible to import the ANN in to the ladder diagram.

The Compact Logix PLC allows importing add-ons blocks with additional instructions inside its programming. These modules allow for a more advanced programming like Visual Basic or C++. These tools allow for a more complex syntax that can support the program of an ANN.

In order to manage information between the PLC, with RSLogix 5000, and the PC, with MATLAB, a communication between these devices has to be established. This communication was handled by using the software Matrikon OPC, the user friendly interface of this software makes the communication easy to configure.

Other tools that were used in this project are modules contained in MATLAB. The Neural Network Toolbox was used to create the implemented ANN. SIMULINK managed the export of the ANN in to code that is compatible with RSLogix 5000.

The ANN was implemented in the PLC to control a cylindrical tank level process. The brand of these cylindrical tanks is Quanser, which contains a pump for water extraction, an amplifier and a data acquisition board. The

Table 1. Languages Allowed by the Compact Logix PLC Allen Bradley #1768L43

Language	Characteristics
Ladder Diagram	<ul style="list-style-type: none"> Graphical representation of the elements. Evaluates the networks based on the interconnected elements. Allowed elements: contacts, functions, functional blocks, among others.
Block Diagram Function	<ul style="list-style-type: none"> Graphic interface that allows the user to program the elements in a form that is similar to an electric circuit. Based on representation of interconnected functions. Similar interpretation, compared to traditional electric circuits.
Sequential Function Graphic	<ul style="list-style-type: none"> Mainly used to describe functions of sequential control. Based on GRAFCET norm (IEC848). Action properties that allows managing actions, pulses, memories, among others.
Structured Text	<ul style="list-style-type: none"> PASCAL like syntax. Variable designation, function control is managed with text succession.



Figure 2. Quanser data acquisition board [14].

data acquisition card send and receives the inputs and outputs of the ANN, this data controls the voltage that powers the pump for level control. This data acquisition card is shown in Figure 2.

4. Implementation of the ANN into the PLC

The ANN was implemented in the PLC to control a cylindrical tank level process. This was done with the objective of gather real data of the implemented ANN working on a real system. The PLC that was used in this project doesn't have a data acquisition card, therefore a constant communication with a PC was established, and so the sensor data that was sent by the Quanser tank could be read in real time. This communication was handled with MATLAB, which managed communications

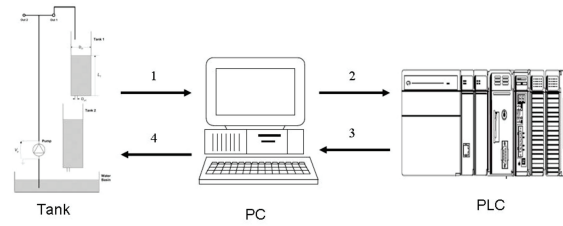


Figure 3. Data flow on the implemented system.

between the Quanser tank and the PLC.

The Quanser system is compatible with MATLAB, which makes possible to establish and analyze different configurations. However it does conflict with any external elements added on the working window. This occurs when the data acquisition card tries to convert the program from the working window into "Quanser code" for its data acquisition card. To achieve the implementation of the ANN, several adaptations where necessary in the MATLAB project so the Quanser system could communicate effectively with the rest of the devices like the Allen Bradley PLC.

Another problem we faced was that the PLC we used does not have an input/output card for managing analog data. So it was required a configuration based on the use OPC communication, as illustrated in Figure 3. First, the sensors on the tank send the data to a PC, which trough MATLAB sends the data to the PLC that contains the created ANN. Once the ANN finishes its process, the data is sent back to the PC, which sends the data back to the Quanser tank trough MATLAB.

4.1. Controlled System

The system to be controlled is a set of two coupled tanks. The cylindrical tanks are configured as shown in Figure 4. One tank is above the other so the output of the first one becomes the input of the second one. At the output of both cylinders there is a recipient that holds the liquid. The ANN created in this research controls the level of the top tank by managing the necessary voltage that is directed to the pump.

4.2. Design of the ANN

In order to create an ANN that controls the variables involved in the tanks process, first it is required to identify the variables as the desired outputs. Additionally, the appropriate ANN type to manage this process needs to be identified.

When implementing the ANN it has to be considered that this algorithm can handle a wide variety of ways to manage the variables. Once the inputs values are



Figure 4. Tank configuration.

defined, the value of the outputs has to be declared and by doing so, the activation level of the pump can be controlled.

As it was previously explained, the creation and manage of an ANN is easier to handle with the different tools that the modules of MATLAB offer. The ANN was created with the graphical interface, from where it is exported to the command window; then it is exported one more time to SIMULINK where it can be coded to RSLogix. The ANN designed is a multilayer perceptron network containing 4 neurons as input and 4 neurons as output which represents a number in binary code. The values were represented in binary code so the variable management in the PLC would be easier to handle. Additionally a hidden layer with 4 neurons is declared. Since the PLC was not made with the idea of handling ANN, it is possible to find errors when importing a complex ANN into RSLogix. In order to simplify the networks and achieve the control for distinct levels, different neural networks were designed (one for each level). As it is shown in Figure 5, when selecting a level on the system, a variable will select the proper ANN from the ones implemented on the PLC.

It is important to have a reliable database of possible values that can be obtained in the system. This data has to represent the behavior of the system so it can be properly simulated when creating the ANN. In this system, the water level can fluctuate between 0 to 30 centimeters for the level which is measured by a sensor with outputs values that vary from 0 to 5 V DC. The levels that are going to be managed in this project vary from 0 to 4 V DC. Regarding the output values of the ANN, they will be managed as voltages with a value of 9 V; the pump will raise the water level at an acceptable speed without overheating the pump motor; a value of 4 V will be declared for the minimum value, by doing so, the water level will drop at a controlled pace and the signal adjustment will be minimum.

In Table 2, the input and output values are listed for

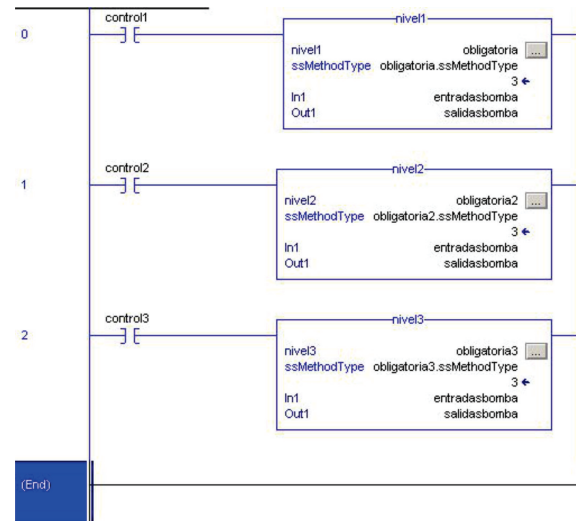


Figure 5. RSLogix code that shows three ANN imported as individual add-ons.

Table 2. Input/Output values, inferior level.

Level	Table of voltage values	
	Input	Output
Inferior Level	0	9
	1	9
	2	5
	3	4
	4	4

the inferior level so it shows the logic that is being followed in the other levels.

It is common to wonder if by adding more neurons in the hidden layer it will optimize the results but this is not always the case; when adding more neurons, it is possible that the system will require more learning cycles. In a similar way, if the learning curve is diminishing in an unnecessary manner, it can reflect in unwanted results [15].

The training parameters will remain the ones that MATLAB has as default values for this type of ANN. The weights for the layers are also managed by the neural network toolbox of MATLAB.

Once the ANN is completed, a diagram that shows a preview of it is displayed, as shown in Figure 6, where the layers are displayed along with their respective neurons.

4.3. Coding the ANN in RSLogix

Once the ANN was created, the next step was to translate it into a code recognizable by RSLogix 5000. To achieve this, the ANN needs to be exported to the MATLAB command window where it can be exported to SIMULINK module with the following command:

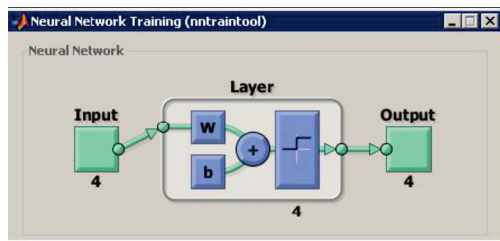


Figure 6. ANN preview, displayed by Neural Network Toolbox in MATLAB.

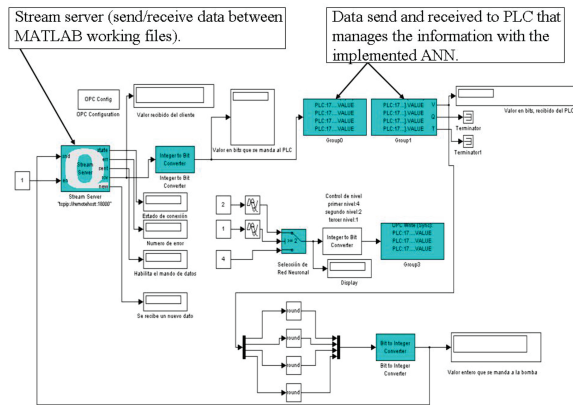


Figure 7. MATLAB working file that handles data between the PC and the PLC.

“gensim(name of the network)”. In SIMULINK, the imported ANN needs to be compress into a subsystem and then, by configuring the available SIMULINK, to export the subsystem into a code compatible with RSLogix 5000.

The created ANN can be imported to RSLogix 5000 as add-on and can be added to a ladder logic program. Once the ANN has been imported, all the different variables that are involved in the process need to be configured in RSLogix 5000. It is worth noticing that the variables that represent the inputs and outputs of the ANN need to be declared as REAL with a $1 \times n$ dimension so they work as a matrix, where n is the number of input or output neurons.

Next, the working file which handles the project that controls the tank level in MATLAB needs to be configured, Figure 7 shows this file project in MATLAB. Since the ANN was trained using binary code representing the real values, it is necessary to add a block that converts the decimal values into binary code and vice versa. These values represent the inputs and outputs of the ANN, so they will need to be managed as a 1×4 matrix, so it continues with the logic that was used when the ANN was created.

Additionally, it is necessary to configure the OPC communication blocks that connect the working files, the PLC, and the tanks process system.

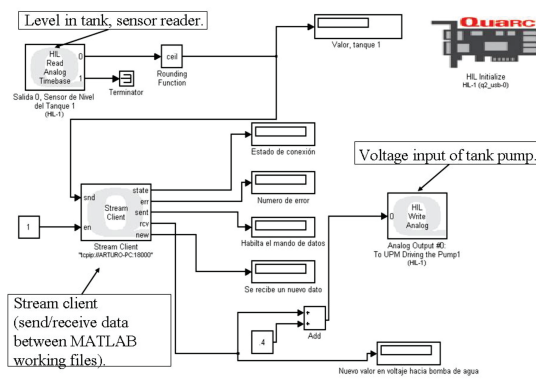


Figure 8. MATLAB working file that handles data between PC and Quanser tank.

All the blocks that are involved on the working window with the system need to be adjusted at the same working simulation time, in our case it was 0.05, so they synchronize. If they are not synchronized, the system returns unwanted results.

Once all the programming has been completed, the MATLAB working file that manages data between PC and the Quanser tank has to be downloaded into the Quanser data acquisition board, Figure 8 shows the MATLAB working file that handles data between PC and the Quanser tank as displayed in the PC window.

To download the file into the Quanser data acquisition board, the specific board that is being used has to be selected and the simulation parameters need to be configured for the correct system that the PC is running, for example its bits properties (32 or 64 bits). The programmed ANN will be controlling the levels for three different phases, the highest level will be selected (at twenty centimeters) first and remain at it for sixty seconds, then the medium level will be selected (at fifteen centimeters) and also remain at it for sixty seconds, finally the last inferior level is selected (at ten centimeters); to do this, a delay will be added into the MATLAB working file.

4.4. Discussion of results

It is worth noticing that the main objective of this research “implementation of an intelligent algorithm in a Compact Logix PLC #1768L43 by Allen Bradley” was achieved.

By analyzing the PLC user’s manual it was possible to note the advantages and disadvantages that this particular PLC possess, specifically that it follows the IEC 1131-3 norm and communication with other devices can be established via Ethernet.

The Quanser tank system that was controlled had its restrictions regarding the use of external devices but

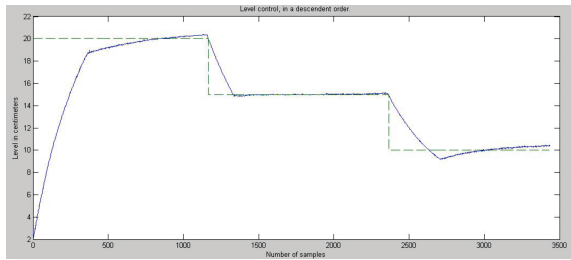


Figure 9. Controlled system behavior in 3 minutes.

a solution was found using support software (Matrikon OPC) and modules that MATLAB includes (OPC toolbox).

Besides noting the benefits that can be obtained by implementing an intelligent algorithm in a PLC, the systematic review of the applications that are being used in the industry helped to confirm that for each type of problem, there is a specific solution; where several factors intervene like the type of algorithm, PLC characteristics, support tools, among others. Therefore, it's not commendable to use a single type of algorithm to optimize every type of problem.

To define the type of intelligent algorithm to be used, different techniques were analyzed, like genetic algorithms, fuzzy logic, among others and due to its characteristics, described on chapter 2.3, the ANN was selected as the one to be implemented. The algorithm was programmed using the neural network toolbox included in MATLAB due to the advantages that its graphical interface offers. Another module included in MATLAB that was important is SIMULINK. With this module, it was possible to export the created ANN into code that is compatible with RSLogix 5000.

Three ANN were implemented into the PLC in order to control three different levels of the coupled tanks system. This solution was employed because, when importing a complex ANN into RSLogix 5000, error messages of invalid expressions in the programming appeared when downloading the project to the PLC, therefore, to avoid this situation, three ANN were managed, one for every single level to be controlled.

The use of all different programs, Quanser, MATLAB, RSLogix, and Matrikon OPC can be difficult for the PC to handle. Unexpected delays occurred when running the process, which caused significant errors. Initially, the PC that was intended to be used on this research had an Intel Dual Core with 1.6 GHz and 1.5 GB of RAM. The PC that was used to run the control process without problems had an Intel Core i5 2.4 GHz processor with 6 GB of RAM.

Figure 9, shows the behavior of the controlled system with a graph that is obtained at real time with MATLAB

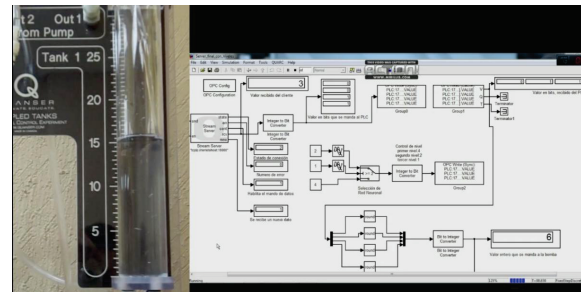


Figure 10. Controlled system at middle level (15 centimeters), showing the cylindrical tank and MATLAB working file.

scopes when the process was running. The Y axis represents the water level in centimeters and X axis represents the number of samples that were taken in a 3 minutes period. As it was previously mentioned in Section 4.2 the voltage level to control the system vary from 9 V as maximum to 4 V as minimum. The transfer from level to level is at the minute mark that passes in the total running clock, until the inferior level is reached. The dotted line represents the desired level and the stream line represents the current level of water in the tank.

Figure 10 shows an image where the process is currently controlled at 15 centimeters. The left side shows the actual tank and the right side shows the MATLAB working file.

5. Conclusions

After analyzing the results of the systematic review it can be concluded that the implementation of an intelligent algorithm into a PLC in order to optimize process like the control of hydraulic systems, solar panels or robotic arms, etc, can be achieved, in this research it was implemented to control the level in a cylindrical tank.

To make the above possible, there are different ways to achieve it, among which we followed the languages allowed by the PLC, specifically the ladder diagram. It is important to notice that MATLAB comes as a relevant support tool that offers many possibilities to reach the desired objective, thanks to all the modules it supports.

During the research, some problems emerged for which their solution caused to a certain point to adjust the parameters of the research itself. Among these issues, we can find the distinct limitations that need to be considered when programming in ladder diagram language for the Allen Bradley PLC, mainly the import of the ANN code that it is created in MATLAB. Other issues were more simple, like a delay when reading the water level sensors, which were caused by the limited processing power of the PC that was being used. To

solve this it was just a matter of using a more powerful PC.

It was also relevant to consider the limitations of the tanks system by Quanser since it is not very “friendly” when adding distinct process to what they are programmed for, specifically when transferring the program to the data acquisition card. It was necessary to create additional process so the system could adapt to the Quanser requirements.

It is also important to notice the wide spectrum of solution that MATLAB offers along all its modules. Using these modules made possible to create the ANN, adapting it to the Quanser tanks system and importing the code to RSLogix into the PLC and all of it in a practical manner due to its different graphical interfaces.

Acknowledgements

Thanks to the support from scholarships system by DGEST and the “Divisin de Estudios de Posgrado e Investigacin from Instituto Tecnolgico de Hermosillo”. This project was partially financed by PROMEP (Oficio PROMEP/103.5/12/4633).

References

- [1] H. Moravec, and Mind Children, The future of Robot and Human Intelligence. Harvard University Press, p. 222, 1988.
- [2] F. Lewis, J. Campos, and R. Selmic, Neuro-Fuzzy Control of Industrial Systems with Actuator Nonlinearities, Philadelphia: SIAM, p. 244, 2002.
- [3] K. Suzuki, Artificial Neural Networks Industrial and Control Engineering, India: InTech, p. 490, 2011.
- [4] D. Kolokotsa, G. S. Stavrakakis, K. Kalaitzakis, and D. Agoris, “Genetic algorithms optimized fuzzy controller for the indoor environmental management in buildings implemented using PLC and local operating networks,” *Eng. Appl. Artif. Intell.*, Vol. 15, No. 5, pp. 417428, Sep. 2002.
- [5] I. Topalova, and A. Tzokev, “Automated Texture Classification of Marble Shades with Real-Time PLC Neural Network Implementation,” *Neural Netwokrs (JCNN)*, 2010 Int. Jt. Conf., pp. 18, 2010.
- [6] I. S. Jung, B. Mulman, D. Thapa, L. Koo, J. H. Bae, S. Hong, S. Yeo, C. M. Park, S. C. Park, and G. N. Wang, “PLC Control Logic Error Monitoring and Prediction Using Neural Network,” *Fourth Int. Conf. Nat. Comput.*, pp. 484488, 2008.
- [7] C. Peshek, and M. Mellish, “Recent Developments and Future Trends in PLC Programming Languages and Programming Tools for Real-Time Control,” *Forth-Fifth Annual Conference of Electrical Engineering Problems in the Rubber and Plastics Industries*, pp. 219230, 1993.
- [8] A. A. Ghandakly, M. E. Shields, and A. M. Farhoud, “Enhancement of Exiting PLC’s with an Adaptive Control Technique,” *Industry Applications Conference, Thirtieth IAS Annual Meeting, IAS '95*, pp. 16341640, 1995.
- [9] E. M. Perez, J. M. Acevedo, C. Silva, y J. Armesto, *Automatas Programables y Sistemas de Automatizacion*, 2nd ed. Mxico, D.F., Grupo Alfa Omega, p. 1120, 2010.
- [10] N. K. Kasabov, *Foundations of Neural Networks, Fuzzy Systems, and Knowledge Engineering*, 2nd ed., Cambridge, Massachusetts: MIT Press, p 581, 1998.
- [11] J. Jantzen, *Foundations of Fuzzy Control*, Chichester, UK: John Wiley & Sons, Ltd, p. 220, 2007.
- [12] B.M. del Brio, y A. Sanz Molina, *Redes Neuronales y Sistemas Borrosos*, 3rd ed. Madrid: Alfa Omega, p. 404, 2007.
- [13] T. Munakata, *Fundamentals of the New Artificial Intelligence Neural, Evolutionary, Fuzzy and More*, 2nd Ed., Cleveland, OH: Springer, p. 266, 2008.
- [14] Q. Inc., “Q2-USB User Manual.” Canada, p. 14, 2010.
- [15] P. I. Viñuela, y I. M. G. León, *Redes Neuronales Artificiales Un Enfoque Practico*, Madrid: Prentice Hall, p. 229, 2008.