Received: 11 March 2016 | Revised: 19 April 2016 | Accepted: 28 April 2016

DOI: 10.37917/ijeee.12.1.6 Vol. 12| Issue 1 | June 2016

Open Access

Iraqi Journal for Electrical and Electronic Engineering

Original Article

Fuzzy-Neural Petri Net Distributed Control System Using Hybrid Wireless Sensor Network and CAN Fieldbus

Ali A. Abed — Computer Engineering, Basra University, dralia.a.iq@ieee.org

Abduladhem A. Ali — Computer Engineering, Basra University, abduladhem@ieee.org

Nauman Aslam — Computer Science & Digital Techniques, Northumbria Univ. nauman.aslam@northumbria.ac.uk

Ali F. Marhoon — Electrical Engineering, Basra University, alifm60@gmail.com

Abstract *The reluctance of industry to allow wireless paths to be incorporated in process control loops has limited the potential applications and benefits of wireless systems. The challenge is to maintain the performance of a control loop, which is degraded by slow data rates and delays in a wireless path. To overcome these challenges, this paper presents an application–level design for a wireless sensor/actuator network (WSAN) based on the "automated architecture". The resulting WSAN system is used in the developing of a wireless distributed control system (WDCS). The implementation of our wireless system involves the building of a wireless sensor network (WSN) for data acquisition and controller area network (CAN) protocol fieldbus system for plant actuation. The sensor/actuator system is controlled by an intelligent digital control algorithm that involves a controller developed with velocity PID-like Fuzzy Neural Petri Net (FNPN) system. This control system satisfies two important real-time requirements: bumpless transfer and anti-windup, which are needed when manual/auto operating aspect is adopted in the system. The intelligent controller is learned by a learning algorithm based on back-propagation. The concept of petri net is used in the development of FNN to get a correlation between the error at the input of the controller and the number of rules of the fuzzy-neural controller leading to a reduction in the number of active rules. The resultant controller is called robust fuzzy neural petri net (RFNPN) controller which is created as a software model developed with MATLAB. The developed concepts were evaluated through simulations as well validated by real-time experiments that used a plant system with a water bath to satisfy a temperature control. The effect of disturbance is also studied to prove the system's robustness.*

***Index Terms*— Fuzzy-Neural Network, Petri Net, Robust Fuzzy-Neural Petri Net, Wireless DCS, CAN**

I. INTRODUCTION

The use of DCS is becoming popular and useful to control large and complex industrial processes that involve multiple variables and control loops. The continued development of these DCS systems results in good advantages for the users such as [1]: flexible hardware architecture and software templates, robust communication systems among hardware components (field devices), good management system for alarms and abnormal events, ability to archive data for the sake of trending, data logging or reporting, etc. The concept of wireless networked control is universal and provides a replacement for the point-to-point wire or UHF radio that uses an antenna system for remote supervision [2]. Information among distributed sensors, controllers and actuators is exchanged over a communication network to perform a specified control task. Building a DCS system supported by a wireless network (i.e. WDCS) is a challenging task because it requires a new design approach for both systems. Control systems and communication networks are designed using different principles. Traditional control loops require the feedback data to be accurate, synchronized and lossless [3]. In contrast, random delay and packet loss are accepted in communication networks. In DCS systems, the network design should optimize the control performance. From the control point of view, the more the controller knows about the system, the better the control performance. This can be done by increasing the number of sensor devices that in turn lead to the congestion in the network. The congestion causes longer delays and packet loss, which leads to a degradation in the control performance of the overall system. Hence, a related design of the wireless network and controller is required. WSANs appear as a new generation of sensor networks [4] that may overcome the above problems and satisfy the related design of the wireless network and controller. WSANs will enable a good degree of distributed control. However, the unreliability of the wireless communication and real-time requirements of control loops are big challenges for WSAN design. In the control field, most of the commercial DCS systems are organized as many periodic processes, with nearly the same period, but without common clock. These processes communicate with each other by means of shared memory through: Serial links or Fieldbus. The fieldbus communication method has been adopted in this work. Basically, the concept of automatic control includes accomplishing two major operations: transmission of signals (information flow) back and forth, and calculation of control actions (decision making). Carrying out these two operations in real plants requires a set of hardware, instrumentation, and suitable software. There are also a number of major issues that must be considered in the design of a DCS system such as: the adopted communication protocol (CAN protocol in this work), the adopted network topology (bus topology in this work), the Interoperability concept that means: Using of nodes from different vendors, satisfying Quality-of-Service (QoS) of the communication protocol, system throughput, system performance, fast response time, and some real time requirements such as sampling time. These design principles will be satisfied in the design of the WDCS system presented in this paper. A new methodology via the use of the WSAN principles that uses a WSN network for data acquisition and a fieldbus system for plant actuation is proposed. The work is the first trial to merge between WSN and fieldbus system in the design of a DCS to get a WDCS. It presents a good trial that serves the industry very much. Therefore, this paper presents an application-level design method for a WSAN suitable for the use in DCS systems. The method is generic because it is independent of the embedded platforms, environment, plant, or controller design. Beside all the above, the paper proposed a new control algorithm for the closed control loop of the WSAN. This control algorithm depends on a control structure built as a Fuzzy Neural Petri net (FNPN) network learned (trained) by a back-propagation online algorithm. The algorithm tries to increase the real-time on-line adaptation by decreasing the convergence time via reducing the number of rules.

There are many publications that deal with fieldbus systems, CAN communication protocol, DCS systems, WSN, WSAN, and intelligent control systems but there is no publication that put them together. To the best of our knowledge, the work presented in this paper is the first trial to design a WSAN system with this collection of subjects leading to a WDCS. In the following discussion, we present related works that made contributions to the different parts of the overall proposed system.

In 2002, Almeida L., *et al* presented a response-time-based scheduling analysis for the real-time traffic. The analysis considered both types of traffic in an integrated way according to their priorities. A fixed-priorities-based policy had been used to schedule the periodic traffic that is important when using planning scheduler instead of offline static scheduler [5]. In 2008, Felipe G.C. *et al* proposed and evaluated the possibility of extending CAN fieldbus with delay-tolerant networks based on multi-hop paradigm [6]. In 2009, Qingfeng L. *et al.* investigated the delays associated with the use of Foundation Fieldbus (FF) H1 networks within control loops. Analytical and experimental evaluations were performed with a test loop using hardwired analog channel. They concluded that significant delays could be introduced if the traditional analog channels of a DCS are replaced by an FF H1 network [7]. Ajay J., *et al* presented an analytical view of a WSN architecture design along with its objectives and implementation challenges [8]. In 2009, Tik L.B. *et al.* also presented a temperature monitoring system for environmental (not industrial) application [9]. In 2009, Shih-Lun, *et al*. presented four-level hierarchical wireless body sensor network (WBSN) system for biometrics and healthcare applications. It also separates pathways for communication and control [10]. WSANs are a relatively new topic that extends the wireless networking principles for actuation purposes. Most of the publications in this field have been focused on the communication protocols e.g. in 2003, Sinopoli B. *et al.* presented a mixed model for design, analysis and synthesis of control algorithms within sensor networks [11]. In 2006, Li prototyped a light monitoring and control application as a case study of a WSAN [12]. In 2007, Feng X. *et al.* presented a design methodology for WSANs in mobile control applications in terms of link quality and its relation with packet loss [4]. In 2009, Anderzej P. *et al.* described how greenhouse climate control can be represented as an event-based system in combination with WSNs. Analysis is given by means of simulation results [13]. In 2010, Hazem M.S. developed a wireless networked control system coordination agent, as part of an intelligent supervisory control system, which grants the WSN gateway as much latitude in meeting its objective with maintaining the control loop performance [14]. In relation to soft computing techniques, the following are some related publications. In 1999, Lin C.T. *et al.* proposed a neural fuzzy inference network for temperature control and proved good learning ability but the inclusion of too many input and output variables increased the network size and decreased the learning speed [15]. In 2002, Mastorocostas P.A. *et al.* introduced a dynamic fuzzy neural network in which, the recurrence is achieved by including external feedback. To apply this network to temperature control problems, it still need to know the order of both control input and network output to participate in the recurrent model [16]. In 2006, Jingwei X., proposed a novel method called PID design through numerical optimization for the design of adaptive fuzzy PID controllers to achieve optimal control performance. He applied numerical optimization to transfer the fuzzy PID design problem to a numerical optimization problem [17]. In 2009, Rong J.W. *et al.* succeeded in the addition of petri net to the fuzzy neural to get a recurrent and dynamic online algorithm to path tracking control of a mobile robot [18]. In 2014, Shahram M-B presented an artificial neural network and adaptive neuro-fuzzy inference system (ANFIS) for overall oil flow rate prediction of the wells on the basis of available temperature and pressure measurements of lines [19]. There is limited work in the field of DCS or WDCS system design. In 2005, Suphan G., *et al* presented network device groups distributed through digital and analog Remote Terminal Units (RTUs). These devices were centralized by computer control and communicated through serial bus [20]. In 2009, Ivan C. *et al.* presented the main advantages of using an industrial DCS in the operation of a distillation column which is used in an undergraduate unit operations laboratory course [1]. In 2010, Amir F. *et al.* focused on an innovative and intelligent DCS for tank gauging control system for tank farm oil terminal [21]. In 2010, Amir F. developed a multi-networking for a computer-based (DCS) tank gauging system for large tank farm [22].

The rest of this paper is organized as follows: Section 2 displays the basic fundamentals and design of the CAN fieldbus. In Section 3, the fundamentals and practical setup of the WSN are explained. The proposed RFNPN intelligent controller is outlined in Section 4 while Sections 5 is concerned to the identifiers and control system design. The overall wireless control system (WCS) system is shown in Section 6. Validation results for the offline and online identification as well as the real-time experimental results are presented in Section 7. Conclusions are summarized in Section 8.

II. CAN FIELDBUS SYSTEM DESIGN

The evolution of process control systems due to increasing complexity of industrial processes has led to a big increase in the number of field devices and the quantity of data. The classical point-to-point communication between the site (plant) and the control room i.e. the 4~20mA system appears to be unsuitable for these complex systems. This led to the development of serial communication systems for process control applications. One of the most famous serial communications is the fieldbus. It is a digital communication network that is used in industry to replace the existing 4~20mA classical analog system. Serial transmission has many advantages as compared with other transmissions. For example, it reduces the number of physical wirings over longer distances than that of the point-to-point or the parallel transmissions. Since network communication is very critical function, selecting the right fieldbus is very important and suits the operation and cost. There are two major reasons in favoring the fieldbus system: The data transmission is done in a standard form, and the data exchange on the bus is seen by all nodes at the same instant. That means there is no need for extra cabling to connect the various nodes all together. One of the most important protocols (that is adopted in this work) is the CAN which has specifications suitable for industry e.g. cost effective and high baud rate (1Mbit/s). The T89C51CC01 is the first member of the CAN family of 8-bit microcontrollers dedicated to CAN network applications [23]. This controller is used in the design presented in this paper. The MCP2551 is a high speed CAN transceiver that works as an interface between a CAN controller and the physical bus. It provides a differential transmit and receive capability for CAN controllers. Each node in a CAN system must have a device to convert the digital TTL signals generated by the CAN controller to differential output (CANH and CANL). A typical CAN fieldbus system (master-slave mode) is shown in Figure 1.

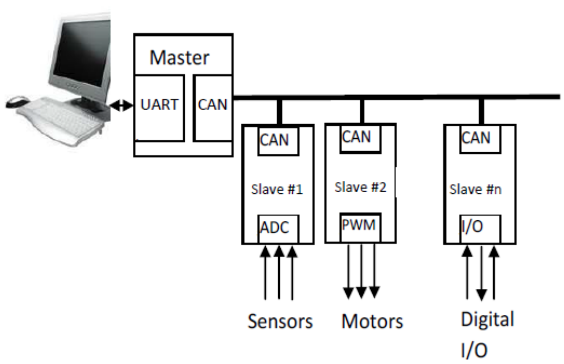


Fig.1. Overall CAN fieldbus (master-slave mode)

Each node acting as slave of the fieldbus system consists of a CAN microcontroller and a CAN transceiver. The microcontroller deals with the sensors and actuators of the plant, while the transceiver transforms the digital data comes from the microcontroller to a differential signal. The slaves communicate with each other via the master controller, which also transforms the data received from the computer by the universal asynchronous transmit receive (UART) protocol of the RS232 serial port to CANH and CANL signals sent to the slaves via the CAN bus and vice versa. Therefore, each microcontroller should be attached to a CAN transceiver. The practical electronic implementation of the master controllers is shown in Figure 2. WCS system uses the slave controller for sending the 4~20mA analog control signal to the actuator, which is an SCR (Silicon Controlled Rectifier). Digital output of port 1 (P1) of the slave controller is a byte that should be converted to an analog signal using a digital-to-analog (DAC) converter. The DAC used is the AD7528, which is a dual 8-bit digital-to-analog converter. The output analog signal is a low voltage value (0~ -0.6V) obtained at pins (OUT A or OUT B) and then fed to LM324 signal conditioner circuit in order to get the standard range of 1~5V analog signal. The output of the signal conditioning circuit should be converted to analog current 4~20mA signal suitable for sending, without drop and more immunity against noise, to the actuator (SCR). Hence, a V-to-I converter circuit is designed for this purpose. The complete electronic diagram for the analog unit (DAC, signal conditioner, and VI converter) is shown in Figure 3. The SCR power controller used in this work is designed to regulate AC power to electrical heating process. The SCR receives a 4~20mA analog signal fed by a temperature controller via the analog handling unit. The combination of the temperature controller and the SCR provides very accurate automatic temperature control. Images for the designed master controller, slave controller, and the analog handling unit are shown in Figures 4 and 5.

III. Wireless Sensor Network for Data Acquisition

A WSN consists of a number of interconnected sensor nodes (sometimes called motes) that interact with the physical environment for collection and dissemination of data useful in areas where ordinary networks are unsuitable. Each mote may carry different types of sensors suitable for desired measurement such as: temperature, pressure, flow, humidity, etc. All these physical quantities are required in industry.

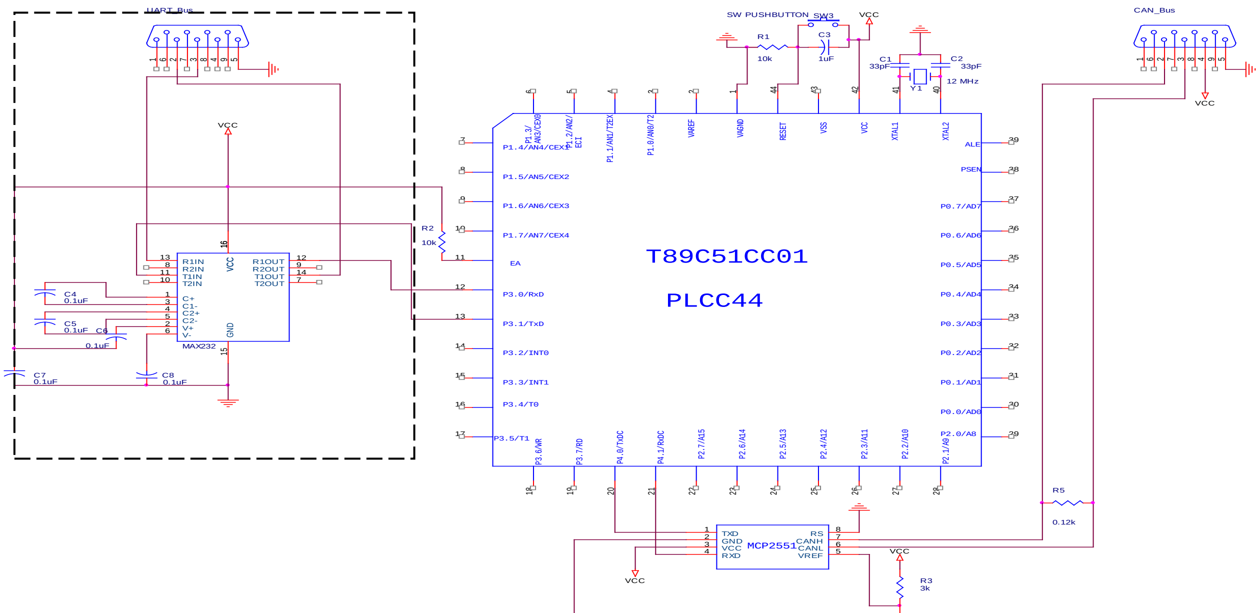


Fig.2. Master Controller of the fieldbus system

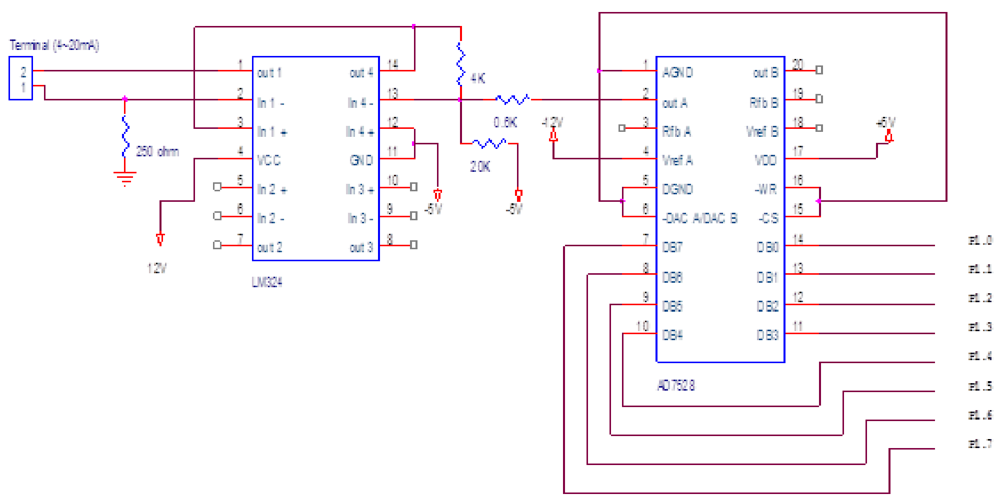
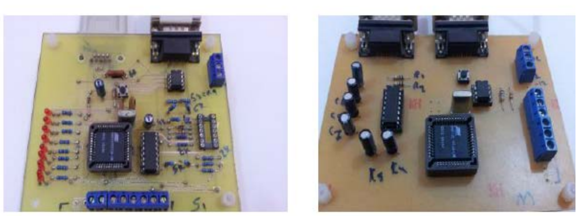


Fig.3. Electronic diagram of the analog unit



(a) (b)

Fig.4. Real photo for the master (b) & slave (a) controllers of the fieldbus

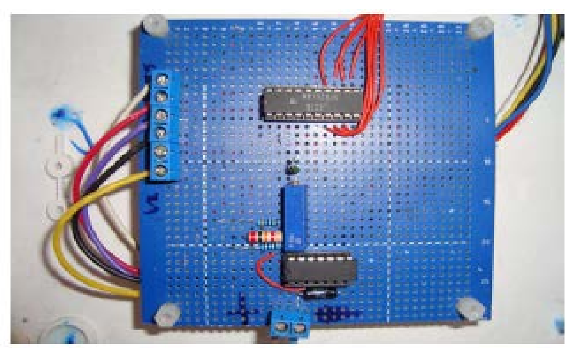


Fig.5. Real photo for the analog handling unit

The use of wireless network for monitoring will not only reduce the overall system--cost in terms of facilities setup and labor cost, but will also provide flexibility in terms of distance or location, scalability and reliability [24]. The use of automation in monitoring task will reduce the dependency on human thus reducing errors and cost. Low power radio platforms with well-designed communication protocol allow a generalized wireless communication among the network nodes, rather than the point-to-point communication. The computing and networking capabilities allow sensor networks to be reprogrammed or re-tasked after deployment in the field. This paper presents an implementation of a general WSN network suitable for the use in a wireless DCS system. The results of the monitoring system are displayed both numerically and graphically. The sensing unit is a sensor connected to an analog-to-digital channel (ADC) embedded in the microcontroller. The microcontroller used here is ATmega1281 which has an 8 ADC channels. The communication unit is an RF transceiver suitable for converting the TTL signals of the microcontroller to an RF signal for the sake of wireless transmission of sensed data to the gateway. The gateway is connected to a server computer for the purpose of monitoring. The RF unit presented in this paper contains a 2.4 GHz IEEE 802.15.4 wireless transceiver designed for the embedded sensor network and used for data acquisition with baud rate of 250 Kbit/s and maximum distance of 500 m. The MIB520 USB interface board- attached with an RF mote is used for two main functions: mote programming; and as a gateway. As the gateway needs to continuously receive data from the field sensor nodes, it remains powered at all times via the USB port of the server computer. The schematic diagram of the RF unit used in this paper is shown in Figure 6.

The hardware components of the overall WSN designed for temperature monitoring is displayed in Figure 7.

Many software packages are needed in the development of a WSN application. One of the most popular software is the Tiny Operating System (TinyOS). TinyOS is a lightweight operating system specifically designed for small, low power microcontrollers used in wireless sensor nodes [25].

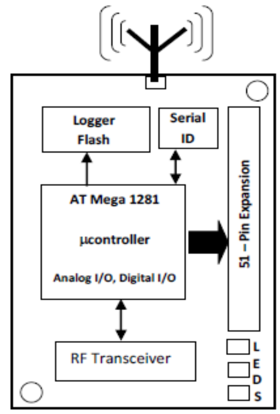


Fig.6. Schematic diagram of the RF unit

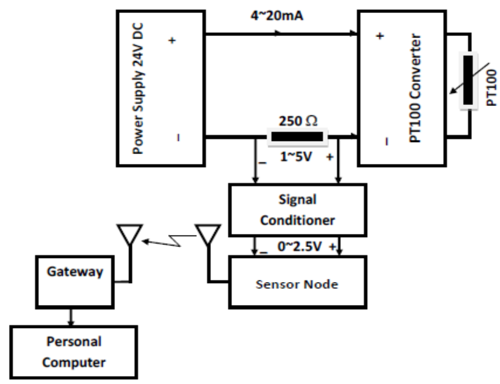


Fig.7. Overall WSN for water bath temperature monitoring

TinyOS and its applications and systems are written in the nesC (Network Embedded System C) language which is a C dialect with extra features. TinyOS has a set of application program interfaces (APIs) for common functionality such as sending packets, reading sensors, and responding to events. Server software called *XServe* is used for the required services such as parsing, transforming and processing data at run time [26]. It is required to save the desired data (temperature sensor reading) in a *.csv* Excel file i.e. making the server program to work as a producer that produces the items (data) and put it in a First-In-First-Out (FIFO) buffer (the Excel file) in order to consume it by a consumer. All the required data will be saved automatically to the *Excel .csv* file. The consumer must be another program that read the data from the buffer and displays it on a display screen. This data will be accessed by an m-file and routed to the display for the purpose of monitoring. To get a complete

WSAN system (closed loop), we will use this data as a measured feedback values that should be compared with the software generated set points to produce an error signals supplied to the designed intelligent controller, that in turns produces a control signal for the actuator of the plant leading to a closed control loop. It is clear that the overall problem is a producer-consumer problem. It is known from the principles of operating systems that this problem needs some form of synchronization (timing) between producer and consumer to overcome some problems such as the buffer empty and buffer full. Also, the actions of the control algorithm have to be synchronized with the requirement of the process to be controlled i.e. the control action is done at a specified sampling rate. There are many methods for timing such as: polling, external interrupt, ballast coding, and real-time clock [27]. The adopted method in this work is the real-time clock. The code written for this real-time requirement is as follows:

```
tic
for k=1:L
        ttt=toc;
        while (ttt < Ts)
            ttt=toc;
      end
      tic
      {Control Algorithm}
  end
```

Where "*tic-toc*" are instructions in MATLAB language. "*tic*" starts ticking and "*toc*" calculates the elapsed time, T_s is the sample time, and L is the number of samples. This code segment, which runs in the beginning of each control cycle, uses a "busy waiting" for a constant time of T_s seconds. It should be noted that the sampling time for the consumer must be greater than the sampling time of the producer in order to satisfy exact synchronization. Hence, it is insured that the consumer never read the buffer if it is empty or not yet produces a new item. The estimated value for T_s is found to be 2 seconds while that for the sensor node is investigated to be 0.7 seconds. These values are compatible with the real time ranges of the commercial DCS systems that have an industrial range of 0.5~60 seconds.

IV. THE PROPOSED RFNPN INTELLIGENT CONTROLLER

Recently, hybrid systems such as: fuzzy-neural, neuro-fuzzy, etc., have proven to be very effective in designing intelligent control systems. These soft computing techniques have been widely used in identification and control of dynamic control applications. In this paper, we propose a novel hybrid control system that combines three soft computing techniques: fuzzy, neural, and petri net leading to a Fuzzy-Neural Petri Net (FNPN) controller that is trained and used with temperature control plant to verify its operation and performance. Many features are added to this controller to increase its robustness against plant disturbance such as bumpless transfer and anti-windup. The presented RFNPN controller for temperature control system is shown in Figure 8.

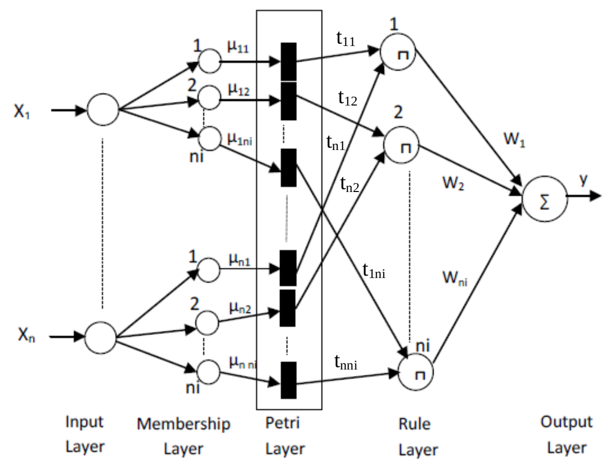


Fig.8. Framework of the RFNPN

The operation of each layer of this RFNPN framework is introduced as follows:

1. Input Layer

   Each node in this layer transmits the input crisp variable x_i *(i=1,2,3,…n)* to next layer.

2. Membership Layer

   This layer represents the fuzzification stage for the RFNPN in which the crisp inputs x_j are transformed to fuzzy inputs via the adopted Gaussian membership function given by:

$$\mu_{ij} = exp\left(-\frac{1}{2}\frac{(x_j - C_{ij})^2}{S_{ij}^2}\right) \tag{1}$$

Where c_{ij} and s_{ij} are the center and width of the membership function.

3. Petri Net Layer

It is used to produce tokens that make use of competition laws for node firing as follows:

$$t_{ij} = \begin{cases} 1 \text{ if } \mu_{ij} \geq d_{th} \\ 0 \text{ if } \mu_{ij} < d_{th} \end{cases} \quad (2)$$

Where t_{ij} is the transition and d_{th} is the dynamic threshold that varies with error and can be tuned by the following equation [18]:

$$d_{th} = \frac{\alpha \exp(-\beta E)}{1 + \exp(-\beta E)} \quad (3)$$

Where α and β are positive constants that can be chosen randomly. It is clear that the larger the error, the smaller the threshold. If the error becomes large, the threshold values will be decreased to fire more rules for the current situation.

4. Rule layer

The output of each node is the product of its inputs after passing from petri layer and it is given by:

$$\emptyset_j = \prod_i^n \mu_{ij}\, t_{ij}$$

Or, it can be written as:

$$\emptyset_j = \begin{cases} \prod_i^n \mu_{ij}; & \text{if } t_{ij} = 1 \\ 0; & \text{if } t_{ij} = 0 \end{cases} \quad (4)$$

Where $\emptyset_j$ is the output of the j^{th} node of the rule layer; n is the number of crisp inputs.

5. Output Layer

Output node calculates the total output y as a summation of the input signals as follows:

$$y = \sum_j^{ni} w_j\, \emptyset_j \quad (5)$$

Where the connection weights w_j is the output action strength associated with the j^{th} rule; ni is the number of rules.

In the structure learning, the number of fuzzy rules, initial number of membership functions, and initial consequent parameters are chosen. The parameter learning is used to tune the free parameters of the constructed network to its optimal values. To explain the learning algorithm of RFNPN using the supervised back-propagation method, the error function is defined as:

$$E = \frac{1}{2}(y - y_p)^2 \quad (6)$$

Where y is the output of RFNPN network and y_p is the output of the plant.

The update laws for w_j, c_{ij}, and s_{ij} are:

$$w_j(k+1) = w_j(k) - \eta_w \frac{\partial E}{\partial w_j} \quad (7)$$

$$c_{ij}(k+1) = c_{ij}(k) - \eta_c \frac{\partial E}{\partial c_{ij}} \quad (8)$$

$$s_{ij}(k+1) = s_{ij}(k) - \eta_s \frac{\partial E}{\partial S_{ij}} \quad (9)$$

Where η_w is the learning rate for the weights of rule layer, η_c , η_s are the learning rates for the center and width of the Gaussian membership function respectively. Choosing suitable values for these learning rates is very important during training process. The three partial derivatives in the above three equations are derived and given by:

$$\frac{\partial E}{\partial w_j} = \begin{cases} (y - y_p)\emptyset_j & \text{if } t_{ij} = 1 \\ 0 & \text{if } t_{ij} = 0 \end{cases} \quad (10)$$

$$\frac{\partial E}{\partial c_{ij}} = \begin{cases} (y - y_p) w_j \emptyset_j \frac{(x_i - c_{ij})}{s_{ij}^2}, & \text{if } t_{ij} = 1 \\ 0, & \text{if } t_{ij} = 0 \end{cases} \quad (11)$$

$$\frac{\partial E}{\partial s_{ij}} = \begin{cases} (y - y_p) w_j \emptyset_j \frac{(x_i - c_{ij})^2}{s_{ij}^3}, & \text{if } t_{ij} = 1 \\ 0, & \text{if } t_{ij} = 0 \end{cases} \quad (12)$$

V. Digital Control System Design

Before proceeding with the proposed controller design, a mathematical model of the temperature plant is required. The water bath is chosen as a temperature controlled object adopted as a case study. It can be described by a continuous time delay, first order, type 0 approximate model of transfer function [27] shown in Figure 9 and has the following difference equation:

$$y_p(k) = \frac{\tau}{T_s+\tau} y(k-1) + \frac{KT_s}{T_s+\tau} u\left(k - \frac{\tau_d}{T_s}\right) \qquad (13)$$

Where K is the plant gain, τ is the plant time constant, τ_d is the plant delay time, and (τ_d/T_s) is the delay order of the plant input *u(k)*. These parameters should be found empirically in order to use them in the simulation software. Experimentally, these parameters are found to be: K=0.4, τ=130 seconds, τ_d=20 seconds, T_s=2 seconds, and τ_d/T_s=10.

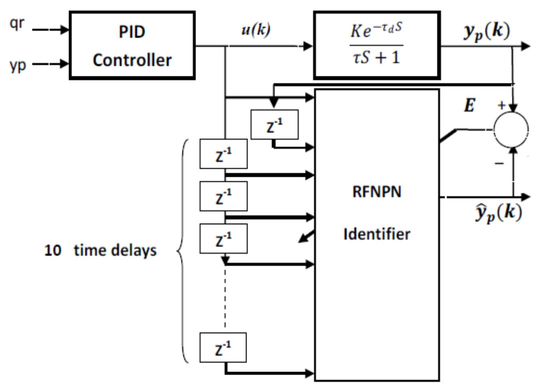


Fig.9. Offline identification model

As a second step for the controller design process, an identification model is required. Identification is the process of constructing a mathematical model of a dynamic system using experimental data measured directly from the system. There are two types of identification methods namely, forward dynamics- and inverse dynamics identification. Also, there are two representations for identification models namely, parallel- and series-parallel models. In this work, the forward control with forward series-parallel identification model is used because it is simpler and more accurate than the inverse control. Since the value of (τ_d/T_S) is 10, that means there are *u(k), u(k-1), u(k-2),... u(k-10)* inputs to the identifier. The total number of delays (z^{-1}) is 11 (10 for delaying the input, and 1 for delaying the output of the plant) as shown in Figure 9. Offline identifier is utilized for two main purposes: The results obtained from the offline identification are used as initial weights and parameters for the online control system. First this initial data will speed up the learning process of the online control system and hence the convergence will be fast, second the identifier is used in the online forward control system to measure the sensitivity S of the plant and provides it to the FNPN controller for the sake of adaptation.

The PID controllers were used to control industrial temperature systems and some of these systems are still used today. Adaptive control is introduced to deal with complex systems that could not be controlled by conventional methods such as less-defined plants or time varying parameters plants. Adaptive controller is a controller that can modify its behavior in response to changes in the dynamics of the process and the characteristics of disturbance. In this type of control systems, plant response should follow the response of a defined system such as Model Reference Adaptive Control system (MRAC). Fuzzy logic was used for many years to realize adaptive control. Then conventional PID controllers are realized with fuzzy logic to get PID-like fuzzy controllers. Even though fuzzy controllers often produce superior results as compared to those of classical controllers, there is a difficulty in accessing the fuzzy controller because: firstly, its design is not straightforward due to heuristic of the rules and membership functions; secondly, tuning of its parameters is complex. The generation of membership function is a challenging problem in fuzzy systems. Hence, this leads to the emerging of fuzzy neural and neuro-fuzzy systems in which the neural network is adopted to learn the membership functions of the fuzzy systems. The MRAC system is an important adaptive controller by which the desired performance of the overall closed loop system is obtained if it is made to track the performance of a stable reference model. The proposed control system attempts to guide the plant output y_p to match the reference model output y_r asymptotically i.e.

$$\|y_r - y_p\| \leq \varepsilon \qquad (4)$$

for some specified $\varepsilon \geq 0$, which is used for the controller network training. The reference model is designed as a PID system tuned to get the desired response and added to the online control structure of our RFNPN controller. The proposed RFNPN forward (online) control system is shown in Figure 10. This developed control system structure, which is based on MRAC, performs two tasks: system identification and control. The initial values for parameters w_j, c_{ij}, and s_{ij} of the identifier are taken from the offline learning that should be achieved already. The identification process should overcome the changes that may occur in the system parameters due to disturbance. Also, the control process requires the sensitivity parameter (S) of the plant to be introduced by the identifier to the main controller as shown in Figure 10.

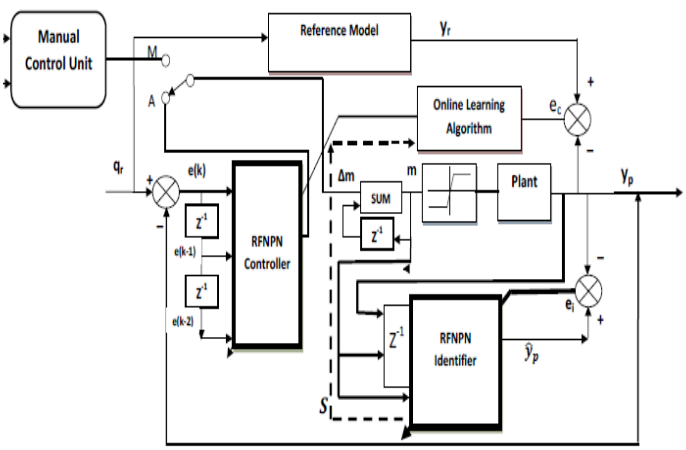


Fig.10. Developed RFNPN online controller operating with two modes: Manual and Automatic

The RFNPN for the controller and the identifier has the same structure with a small difference in the number and type of the inputs and outputs. The update training laws for the RFNPN is the same as the previously mentioned equations but the error function will be:

$$E_c = \frac{1}{2}e_c^2 = \frac{1}{2}(y_r - y_p)^2 \tag{15}$$

Now, the partial derivatives $\frac{\partial E_c}{\partial W_{cj}}, \frac{\partial E_c}{\partial C_{cij}}$ and $\frac{\partial E_c}{\partial S_{cij}}$ of the updating laws should be derived according to the chain rule as follows:

$$\frac{\partial E_c}{\partial W_{cj}} = \frac{\partial E_c}{\partial e_c}\frac{\partial e_c}{\partial y_p}\frac{\partial y_p}{\partial m}\frac{\partial m}{\partial W_{cj}} \tag{16}$$

Where m is the plant input or the controller output. The term $S = \frac{\partial y_p}{\partial m}$ represents the system sensitivity that should be found because the convergence of the RFNPN controller cannot be insured if S is unknown. Since the RFNPN identifier of Figure 10 is used to provide the S function and since it has the same plant input and output, i.e.

$$\frac{\partial y_p}{\partial m(k)} = \frac{\partial \hat{y}_p}{\partial m(k)} \tag{17}$$

Then:

$$\frac{\partial \hat{y}_p}{\partial m(k)} = S = \frac{\partial \hat{y}_p}{\partial \emptyset_j}\frac{\partial \emptyset_j}{\partial \mu_{ij}}\frac{\partial \mu_{ij}}{\partial m(k)} \tag{18}$$

Hence, the S function is derived to be:

$$S = \begin{cases} -\sum_{i=1}^{n_i} w_j \emptyset_j \frac{m(k) - C_{ik}}{S_{ik}^2}, & if\ t_{ij} = 1 \\ 0, & if\ t_{ij} = 0 \end{cases} \tag{19}$$

Therefore, we can summarize the updating laws for RFNPN controller by:

$$\frac{\partial E_c}{\partial W_{cj}} = ec\, S\, \emptyset_{cj} \tag{20}$$

$$\frac{\partial E_c}{\partial C_{cij}} = \begin{cases} \emptyset_{cj}(X_{ci} - C_{cij})\frac{W_{cj} S\, ec}{S_{cij}^2} & if\ t_{ij} = 1 \\ 0 & if\ t_{ij} = 0 \end{cases} \tag{21}$$

$$\frac{\partial E_c}{\partial S_{cij}} = \begin{cases} \emptyset_{cj}(X_{ci} - C_{cij}^2)\frac{W_{cj} S\, ec}{S_{cij}^3} & if\ t_{ij} = 1 \\ 0 & if\ t_{ij} = 0 \end{cases} \tag{22}$$

As seen from Figure 10, the designed control system is operated with two modes: manual and automatic. In manual mode, the controller output is manipulated by the operator using pushbuttons. When the mode is changed to automatic, it is important to avoid switching transients. This can be satisfied with a concept called "bumpless transfer" [27]. The summation in Figure 10 is provided to deal with increments. Because the switching only influences increments, there will not be any large transients. Bumpless transfer can be satisfied by several methods. One of the most widely used methods is the velocity algorithm, which gives the change in the value of the controller output (Δm) for each sample rather than the absolute value of the controller output (m). In this case the inputs to the RFNPN controller must be as shown in Figure 10. If the error is large and the integral term of the controller saturates the actuator, the feedback loop will have no effect because the actuator remains saturated even if the process output changes. This means that the integrator is being unstable and may integrate up to a very large value. This effect is called integral windup. Again the velocity algorithm is the solution to that problem. It has an inherent integral limit that prevents the buildup of error leading to avoidance of saturation. Hence, velocity algorithm method is used for the design of our PID-like FNPN controller to satisfy bumpless transfer and anti-windup.

VI. OVERALL WIRELESS CONTROL SYSTEM

The structure of the designed wireless control system is shown in Figure 11.

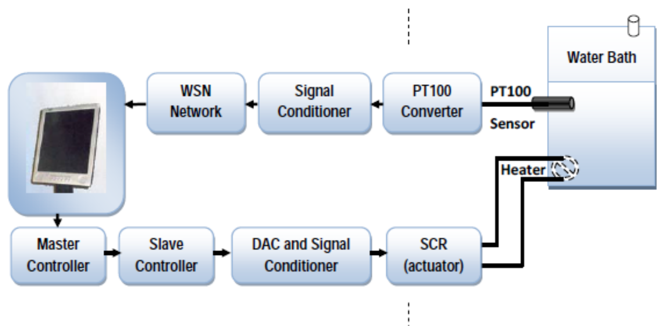


Fig.11. Schematic diagram of the control rig of the WSAN

The sensing part is a wireless system while the actuation part is the CAN fieldbus wire system. The control algorithm for this wireless control loop is a FNPN intelligent algorithm developed with MATLAB. Generally, it is possible to use any number of such wireless control loops with attention to some real time requirements such as the sampling time. The relationship among the various software and hardware of the total system is shown in Figure 12. The input devices (WSN components) plus its software provide the information to create the 'input image' of the plant. This input image is a snapshot of the plant status, which is renewed at specified time intervals. The 'output image' represents the current set of outputs generated by the controller. This output image is updated periodically by the controller. The job of the output task is to transfer the output image to the plant via the output devices (fieldbus controllers). The communication between the operator and the plant is done via the GUI interface designed as a management system.

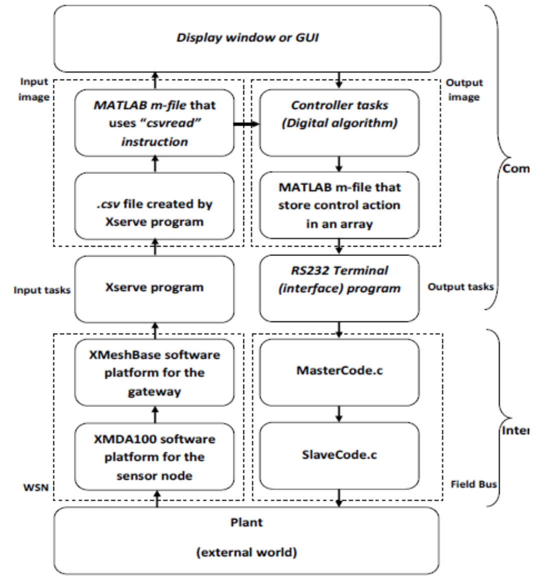


Fig.12. Overall computer control WSAN distributed system showing hardware and software interface

VII. VALIDATION RESULTS

A. Simulation Results for Offline Identification

The results of the offline identification were obtained using a simulation model built in MATLAB. Figure 13 shows the trained response after few epochs while Figure 14 shows the trained response after 8000 epochs. Figure 15 shows the mean squared error (MSE), which displays an acceptable value of 0.14. For all these tests, the number of inputs of the identifier is 12, number of membership functions for the three network parameters (w, c, s) is 6, learning rates $\eta_w=0.5$, $\eta_c=0.5$, $\eta_s=0.5$, set point=70 ^oC and d_{th} is 0.1. To test the robustness of the identifier against disturbance, the gain value was changed 0.4 to 0.2 in order to emulate the disturbance conditions. It can be clearly observed from Figure 16 that identifier tracked plant output in an efficient and fast manner.

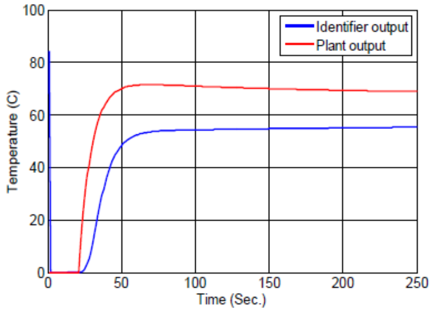


Fig.13. Response of the offline identification after few epochs

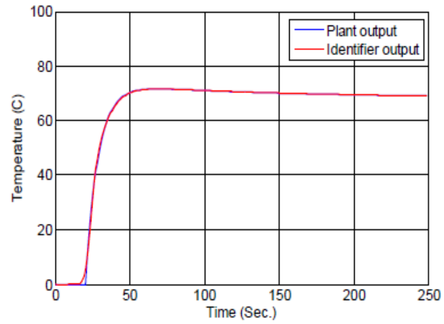


Fig.14. Response of the offline identification after 8000 epochs

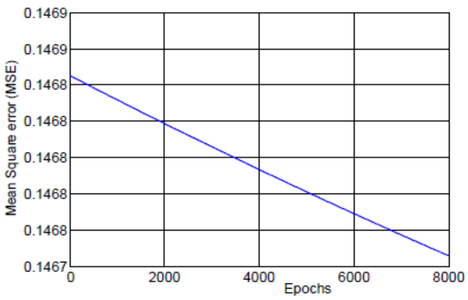


Fig.15. MSE of the offline identification after 8000 epochs

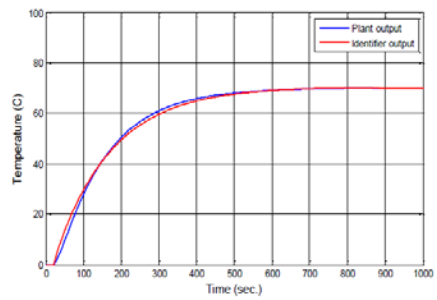


Fig.16. Response of the offline identification after 200 epochs when some disturbance occurs (K=0.2)

B. Simulation Results for Online Identification and Control

The online forward controller of Figure 10 is implemented in another model as a digital control algorithm. The forward identifier, which is used to estimate the plant sensitivity, will take its initial parameters from the final values of the offline identification system. All initial values for parameters such as: learning rates, number of rules, number of membership functions, and constants of the dynamic threshold, are chosen by trial and error for the online and offline identification. Figure 17 shows the trained response after few epochs while Figure 18 shows the result after 8000 epochs. The mean squared error (MSE) during the 8000 training epochs is shown in Figure 19. From Figure 19, an average value for error approximately equals to 0.6 is observed, which is very good and acceptable in industrial applications. If a disturbance causes a change in the value of the time constant of the plant from 130 to 150 seconds, then the response after little iterations is shown in Figure 20. After about 200 epochs it will return to the steady state as shown in Figure 21. Hence, it is clear that the robustness of this system against disturbance is good.

C. Real-Time Offline Identification Results

All the information mentioned for simulation results are used here but with real time software model that receives its data from the PT100 sensor via the WSN and starts with real-time identification. The system is trained with two set points: 65^o C and 35^o C. Figure 22 shows the response of the system at the beginning of learning while Figure 23 shows the response of the system after 100 epochs. The error of this system is presented in Figure 24 which is settled to an average value of approximately 4. The identification system is working well with disturbance, e.g. internal disturbance in which a power change in the actuator input is done from 220VAC to 170VAC (by Variac). This disturbance is created at time 50 seconds continues till 750 seconds.

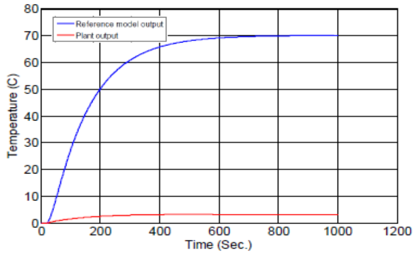


Fig.17. Response of the online control system after few epochs

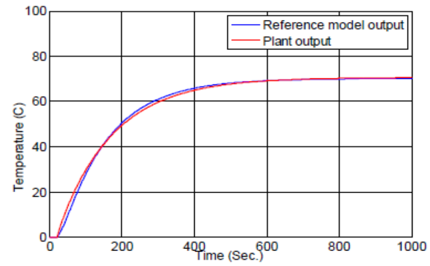


Fig.18. Response of the online control system after 8000 epochs

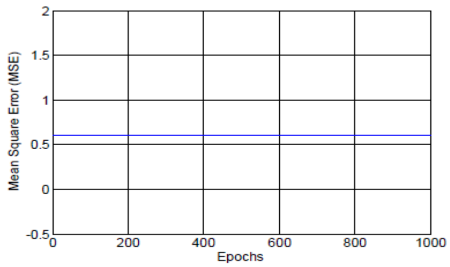


Fig.19. MSE of the online control system after 8000 epochs

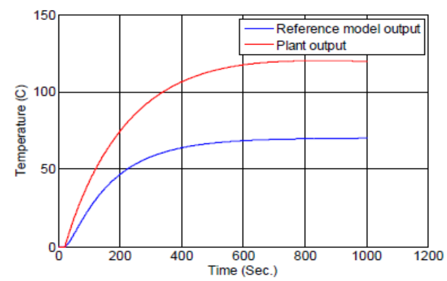


Fig.20. Response of the online control system after few epochs when some disturbance occurs (τ=150)

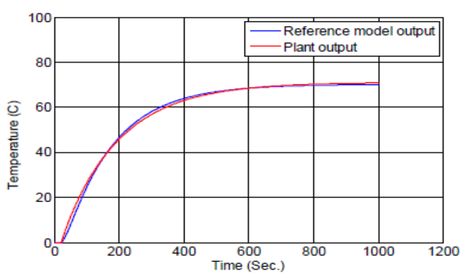


Fig.21. Response of the online control system after 200 epochs when some disturbance occurs (τ=150)

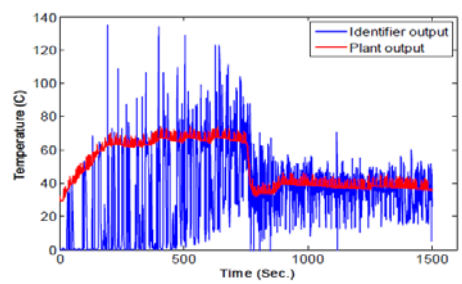


Fig.22. Response of the offline identification system at the beginning of learning

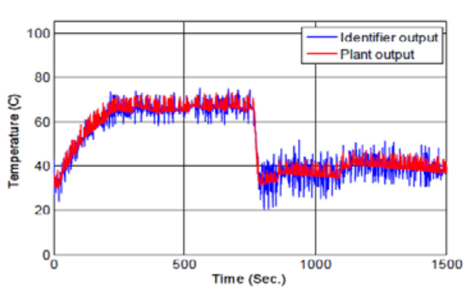


Fig.23. Response of the offline identification system after 100 epochs from beginning of learning

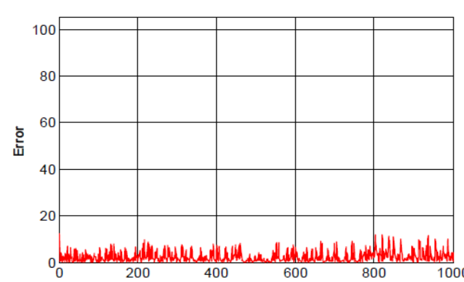


Fig.24. Error of the offline identification system after about 100 epochs

The obtained response is shown in Figure 25. This curve proves the robustness of the identifier against internal disturbance.

D. Real-Time Online Identification and Control Results

The online forward controller of Figure 10 is implemented by a real-time model as a digital control algorithm. The real-time system is started and after few epochs the obtained response is shown in Figure 26. After 100 epochs, the obtained response of the system is shown in Figure 27. The system is trained with set points of 65° C and 35° C. The error of this control system is shown in Figure 28.

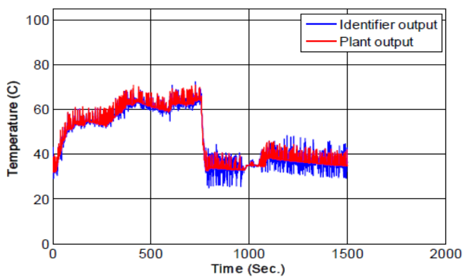


Fig.25. Response of the offline identification system with internal disturbance done at k=50 sec.

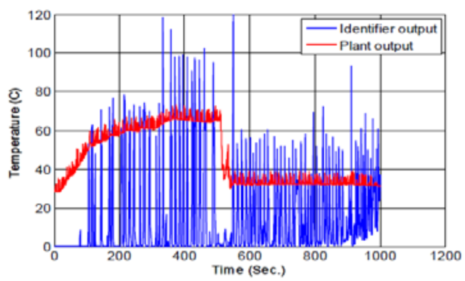


Fig.26. Response of the online wireless RFNPN control system after few epochs.

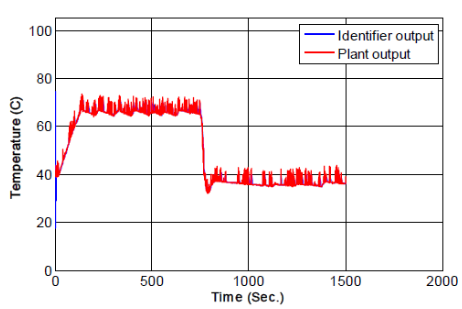


Fig.27. Response of the online wireless RFNPN control system after about 100 epochs

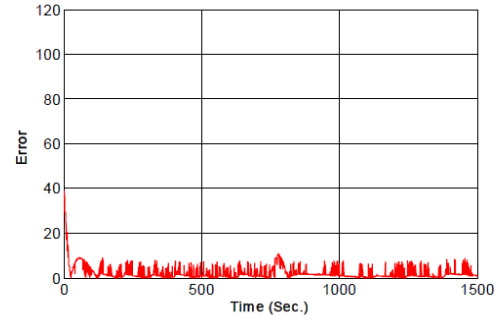


Fig.28. Error of the online wireless RFNPN control system after about 100 epochs

The average value of the error is approximately 5. The same internal disturbance done with the offline system is repeated here at time of 50 seconds from beginning. The effect of this disturbance is increasing the settling time of the system by approximately twice as shown in Figure 29. Comparing Figure 27 and Figure 29, the settling time is increased from 125 s to 250 s and then returns to its steady-state, which proves the robustness of the system against disturbance.

E. Real-Time PID Control System Results

The FNPN intelligent control algorithm is replaced by a conventional PID control algorithm to make a comparison of results between the two controller algorithms. The PID controller is designed and empirically fine-tuned to control our temperature control system. The obtained tuning parameters are: K_p=2.49, K_i=0.000005, K_d=330. The obtained response for the wireless PID control system is shown in Figure 30. If the same disturbance that applied to the intelligent control system is again applied to the PID system, the effect is increasing the settling time by more than 500 seconds relative to the disturbed RFNPN controller as shown in Figure 31. The settling time of the RFNPN system was 125 seconds, while for the PID system is approximately 750 s.

F. Real-Time Manual Control Unit Results

The manual/auto mode is added to our event-driven software of the wireless control system. The adjustment of the controlling value is done with pushbuttons one for increasing and the other for decreasing. The bumpless transfer and the anti-windup are satisfied using the velocity algorithm for implementing the fuzzy neural PID control algorithm.

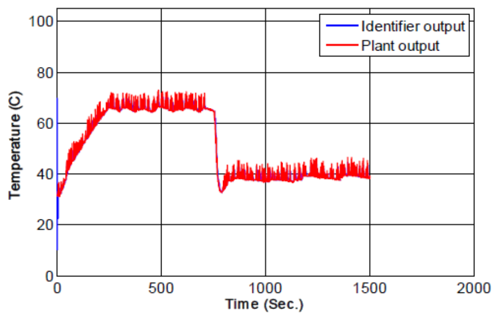


Fig.29. Response of the online wireless RFNPN control system with internal disturbance done at k=50 sec.

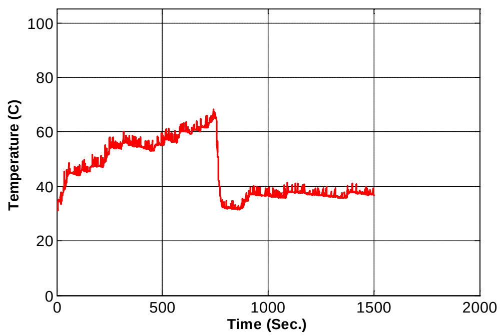


Fig.30. Response of the wireless PID control system

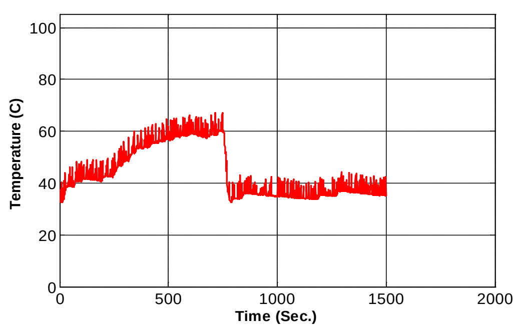


Fig.31. Response of the wireless PID control system with internal disturbance done at k=50 sec.

The system is started manually and continued until reaching its steady state value as shown in Figure 32. At time 500 seconds, the system is switched to auto mode. Figure 33 shows the response after switching to auto mode. It is clear that there is no change in response during or after switching, which proves that the velocity algorithm has satisfied its major function. The same real-time measurement is done for the PID control system as shown in Figures 34 and 35. The aforementioned real-time results proved that the RFNPN technique is applied successfully for identification and control of process industrial systems especially temperature control. They proved a good stability in error, fast training convergence, and short settling time as compared to the conventional PID systems. The results proved that the adaptive RFNPN reaches to the steady state faster than PID system.

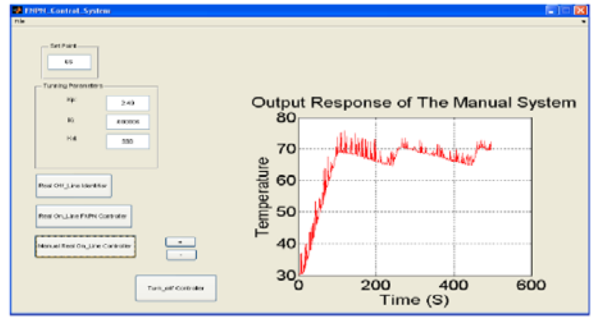


Fig.32. Response of the manual mode of the RFNPN wireless control system

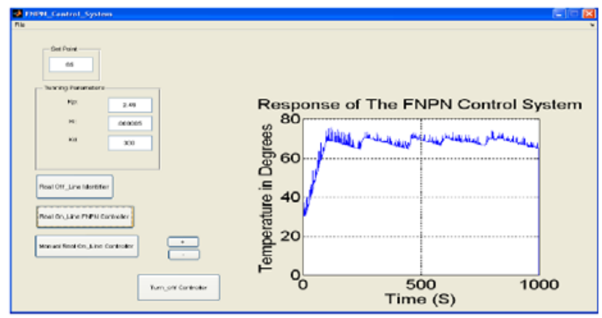


Fig.33. Response of the RFNPN wireless control system immediately after switching from manual to automatic mode at time= 500 sec.

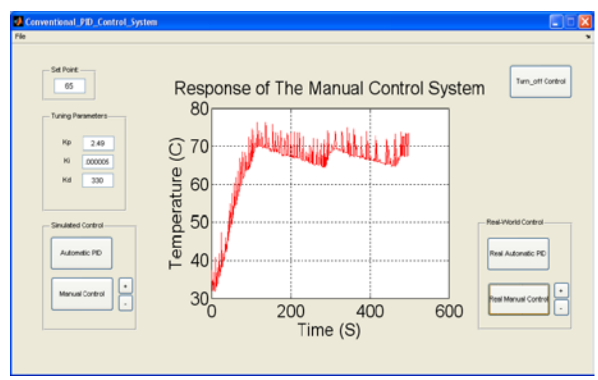


Fig.34. Response of the manual mode of the PID wireless control system

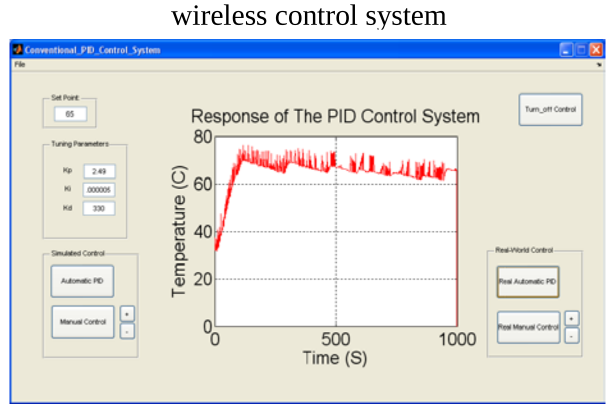


Fig.35. Response of the PID wireless control system immediately after switching from manual to automatic mode at time=500 sec.

VIII. Conclusion

The development of the DCS systems and the concept of networked control results in good advantages for the users that are explained in detail. Wireless networks have become increasingly important in distributed control systems because they have many advantages explained previously. Control system and communication network are designed using different principles with tradeoff between communication and control performance. A related design of the wireless network and controller is presented. WSANs are found to satisfy the related design of the wireless network and controller. Therefore, it has been used for the design of DCS systems to get a WDCS. The design of the actuation part proved that the CAN protocol is a high reliable protocol and therefore can be adopted and highly recommended in industrial applications. The use of WSN for data collection and monitoring reduced the congestion on the fieldbus network i.e. the fieldbus has been used only for actuation. The paper also developed a control algorithm for the closed control loop of the WSAN. This control algorithm was based on a control structure that built as a fuzzy neural petri net algorithm learned with a back-propagation algorithm. The real-time experimental results proved that the RFNPN technique is very good for identification and control of any process systems especially temperature control. The RFNPN proved its capability to reduce the number of fired (active) rules to two rules only if the error is decreased since it increases the dynamic threshold. Also, it proved a good stability in error and fast training convergence. A manual/auto mode has been added to the control system structure with bumpless transfer and anti-windup real-time features, which are insured by using the velocity algorithm method for the PID-like fuzzy neural controller. The overall wireless distributed control system has been tested with a real water bath temperature control to satisfy its effectiveness and robustness.

References

[1] Ivan C. and Thomas F., Using distributed control system (DCS) for distillation column control in an undergraduate unit operations laboratory, 2009 American Control Conference, Hyatt Regency Riverfront, St. Louis, MO, USA, 2009.

[2] Santos M.R., de Olivera B.G.M, de Melo M.T., de Olivera A.J.B., Santos E.A.B., Freitas R.D., Serra R., Santos M.M.L., A smart antenna system for remote supervision, International Journal of Applied Electromagnetics and Mechanics 45 (2014) 293-299.

[3] Xiangheng L., and Andrea G., Wireless network design for distributed control, Stanford University, USA, Supported by ABB through the Stanford Networking Research Center, 2004.

[4] Feng X., Yu-Chu Tian, Yanjun Li and Youxian Sun, Wireless sensor/actuator network design for mobile control applications, Sensors 2007, 7, 2157-2173.

[5] Almeida, L. ; Tovar, E. ; Fonseca, J.A.G. ; Vasques, F., Schedulability analysis of real-time traffic in WorldFIP networks: an integrated approach, IEEE Transactions on Industrial Electronics, Vol. 49, Issue 5, 2002, 1165 – 1174.

[6] Felipe G., Gonzalez-Castano F.J., Franck, L., Extending vehicular CAN field buses with delay tolerant networks, IEEE Trans. On Industrial Electronics, Vol. 55, No.9, Sept. 2008.

[7] Qingfeng L., *et al.*, Evaluation of delays induced by foundation field bus H1 networks, IEEE Transaction on Instrumentation and Measurement, Vol. 58, No.10, October 2009.

[8] Ajay J., *et al*, Wireless Sensor Network (WSN): Architectural Design issues and Challenges, (IJCSE) International Journal on Computer Science and Engineering, Vol. 02, No. 09, 2010, 3089-3094.

[9] Tik L., *et al.*, Monitoring of an aeroponic greenhouse with a sensor network, IJCSNS, Vol. 9, No.3, 2009.

[10] Shih-Lun C., *et al*, Wireless Body Sensor Network with Adaptive Low-Power Design

for Biometrics and Healthcare Applications, IEEE system Journal, Vol. 3, No. 4, 2009

[11] Sinopoli B., *et al.*, Distributed control applications within sensor networks, proceeding of the IEEE 2003, Vol.91, Issue 8, P. 1235-1246.

[12] Li S., Wireless sensor actuator network for light monitoring and control application, proceeding 3rd IEEE Consumer Communications and Networking Conference, Vol.2, 2006, P.974-978.

[13] Andrzej P., *et al.*, Simulation of greenhouse climate monitoring and control with wireless sensor network and event-based control, Sensors, 2009, *9*(1), 232-252.

[14] Hazem M., Wireless networked control system coordination agent, M.Sc. Thesis, University of New Brunswick, Canada, 2010.

[15] Lin C. *et al.*, Temperature control with a neural fuzzy inference network, IEEE Transactions on Systems, Man., and Cyber. Part C: Applications and Reviews, Vol. 29, No.3, 1999, P.440-451.

[16] Mastorocostas P., and Theocharis J. A recurrent fuzzy-neural model for dynamic system identification, IEEE Transactions on Syst., Man., and Cyber, Part B: Cybernetics, Vol. 32, No.2, 2002.

[17] Jingwei X., Fuzzy PID control through optimization: A new method for PID control, PhD Thesis, Marquette University, Milwaukee, Wisconsin, 2006.

[18] Rong J., and Chia M., Design of dynamic Petri recurrent fuzzy neural network and its application to path-tracking control of nonholonomic mobile robot, IEEE Transactions on Industrial Electronics, Vol. 56, No. 7, 2009.

[19] Shahram M-B, A committee machine based soft sensor as an alternative to multiphase flow meter for oil flow rate prediction of the wells, Journal of Intelligent & Fuzzy Systems 26 (2014) 2719-2729.

[20] Suphan G., *et al.*, Distributed control of network devices through remote terminal, KINTEX, Gyeonggi-Do, Korea, 2005.

[21] Amir F., and Li M. Tank gauging control system monitoring based on innovative DCS, IEEE Region 8 SIBIRCON-2010, Russia, July 2010.

[22] Amir F., High speed redundant multi-network DCS-based in innovative tank gauging control system, 2010 IEEE.

[23] Enhanced 8-bit microcontroller with CAN controller and flash memory, Atmel Corporation, 2008.

[24] Victoria J. Hodge, Simon O’Keefe, Michael Weeks, and Anthony Moulds, Wireless Sensor Networks for Condition Monitoring in the Railway Industry: A Survey, IEEE Transactions on Intelligent Transportation Systems, Vol.16, No.3, PP. 1088-1106, 2015

[25] Philip L., David G., TinyOS programming, Cambridge University Press, UK, 2009.

[26] Crossbow, Xserve user's manual, Revision E, PN: 7430-0111-01, 2007.

[27] Stuart B., Real-time computer control: An introduction, 2nd edition, 1994 Prentice Hall International (UK) Ltd.