Implementing Intelligent Functionality into Embedded System

Isabele Morais Costa
Eloi Cagni Júnior
Felipe César Couto
Leonardo Sávio Guanabara
Diego Domingos
José Alberto Nicolau de Oliveira
Jorge Dantas de Melo
Adrião Duarte Dória Neto
Luiz Affonso Guedes

Departamento de Engenharia de Computação e Automação
Universidade Federal do Rio Grande do Norte
Natal, RN, Brazil
{isabele,eloi}@dca.ufrn.br
{lipecouto,lguanabara}@gmail.com
diegodomingosf@yahoo.com.br
nicolau@ufrnet.br
{jdmelo,adriao,affonso}@dca.ufrn.br

Abstract— **In the last years it was observed that there has been a great evolution related to industrial automation as well as the evolution regarding DSP (Digital Signal Processor) and industrial networks. The use of DSP, related to artificial intelligence techniques and neural network, permits gradually the incorporation of intelligence onto terminal control units such as sensors and actuators. This paper presents the development of a intelligent functionality system to process sensor signals through remote parameterization of a neural network in a DSP board, enabling the increase in field device processing capability (intelligence).**

I. Introduction

For a long time production systems have suffered significant changes either related to the growing competition amongst companies or due to the major development of systems that embed microprocessors and microcontrollers in their basic control units. Not long ago the only device that had process capability was the PLC (Programmable Logic Controller). Currently, it is possible to verify the increase in performance requirement, necessary for automated demand systems; this is done considering a more sophisticated hardware infra-structure. Moreover, this infrastructure generally has to be implemented in a distributive manner in order to increase the solution´s trustworthiness and robustness. This leads to the interconnection among field device (controllers, sensors and actuators) through networks. This means to associate these elements to a dedicated hardware which contains the communication protocol and their inherent functions, both implemented in software.

With the great price decrease related to digital technology, communication protocols designed for industrial network are now more sophisticated, permitting the transmission of byte packages (Fieldbus protocols) rather than the communication of few bytes (Sensorbus protocol). In the same way, the availability of embedded hardware related to field device has also lead to the investigation of new ways to local process variables as well as the establishment of more efficient computational manners to treat this problem. This is the case of Functional Blocks existent in Foundation Fieldbus protocols [1]. It is resulted to the sprouting of softwares sensor, they allow implementation of new functionalities in the devices, and they are capable of making it possible to have indirect knowledge of variables that are difficult to measure. It is also possible to enhance procedures related to local diagnosis of sensor performance and treatment of distributive detection of failure and the local implementation of higher control strategies.

Experiences detailed in previous works have shown that the use of functional blocks and dedicated hardware allied to artificial intelligence techniques and neural network, has lead to a gradual incorporation of intelligence to terminal control units such as sensors and actuators [2], [3], [4].

Some difficulties occurs in the analysis of intelligence strategies; it is related to absence of a dedicated platform that permits integration of different sensors in an industrial network environment, as well as the implementation of new functions based on intelligent of information processing. This paper presents an integrated system that allows sensor insertion with interconnections into levels of current (4-20mA) an industrial Modbus network environment. At the same time, it is to implement communication between sensors and the supervisor system, reproducing an existent situation in industrial plants. Moreover, the dedicated hardware based in DSP allows the integrated system to implement different functional tests.

This work is organized as follows: In Section 2 there is a brief explanation about the Modbus protocol. Section 3 relates the details of the proposed system, its disposition, intercommunication and blocks configuration. Section 4 deals with the experiment of the proposed system's validation and the results. In Section 5 the conclusions are presented.

II. Modbus Communication Protocol

The sensor network architecture has experimented constant changes. In a recent past, sensors were accessed individually by wire and provided a 4-20 mA transmission standard signal. These sensors were then connected to a central processing unit with wires, this made costs high and flexibility low.

Currently, there is a partial decentralization where sensors are connected to a communication central which task is link them to the main computer. The rapid growth of intelligent sensors and network field technologies have made intelligent transducers a very economical solution as well as an attractive for measurement and control applications. However, the existence of various network protocols that are incompatible to the number of sensor interfaces and the great deal of effort needed to work with networks are huge, either for producers or sensor users. This is due to the fact that the personalized network interface of a sensor does not work for a particular network.

This shows the choice of communication protocol importance for the system development. Not always it is possible to integrate components of the same provided into a project. Then, one possible solution is obtained common interface among devices.

There are several communication protocols available to interconnect instruments. In our solution was choose the Modbus protocol [5], because it has specifications of public domain and is widely used in industrial solutions, such as softwares, supervisors and PLCs (Programmable Logic Controllers). Another important characteristic of this protocol is its simplicity and flexibility; this makes a good cost/benefit relationship for industrial applications.

The Modbus protocol basically defines a message structure byte oriented. This is done considering that communication happens through a master-slave-technique, where a device (master) can initialize communication by query. The slaves respond sending the master´s requested data. Figure 1 shows this communication by block diagram.

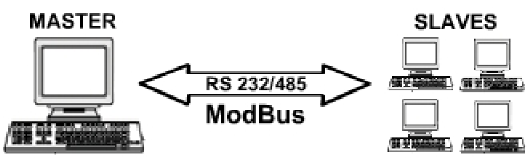

Fig. 1. Communication between master and slave.

This protocol should be used to realize communication between devices that are capable of data transfer in the ASCII or RTU mode. These can perform either in RS232 or RS485. In these conditions the Modbus performs along with these devices, it does not depend on electrical means.

During communication, the Modbus protocol determines how each device can identify its network address; it recognizes messages that were sent, determines which kind of action it will perform and obtains all the necessary information to execute it. When it is necessary to provide an answer for the received command, the device puts together an answer and sends it, even though it demonstrates a communication error.

Modbus establishes the query messages sent format and defines the slave's address, the code function, this indicates the action that must be executed by the slave, the parameters or relevant data, the defined function and the checksum field enabling to check the sent message's integrity.

The slave's answer is generated obeying the format that corresponds to the received function by the master; this basically defines confirmation of the realized function, the parameters or pertinent data and the requested function. Thus, as in the master device it is done in the checksum field. However, in order to send a message, there must be beginning and end indicators. The RTU mode does not have bytes that indicate beginning and end of frame. In order to identify these fields, there should not be a data transmission for a minimal period, which is equal to 3.5 times the data word length (silence). The Figure 2 shows a typical RTU mode frame.

Start Framming	Slave Address	Modbus Function	Data sent to Slave	Checksum	End Framming
Start T	1 char	1 char	N chars	CRC -/+	End T

Fig. 2. A typical RTU Mode Frame.

As it was specified before, the CRC (Checksum) is responsible for the integrity of the sent message, that is, it informs if the message was sent correctly. This is done through a function that aims to compare using XOR (Exclusive OR), the origin message and its destination, If the result is zero, it means that the message was sent without any errors, on the contrary, it indicates that there was communication error.

The Modbus protocol implements only the communication between devices, not allowing the instantiation and configuration of other software objects in the device. For this, one another layer, called in this work Application Layer, must be added to the communication into devices. This layer will be discussed in details in the next Section.

III. Proposed System

As mentioned before, artificial neural network can be an excellent intelligence insertion device for commercial sensors. Many kind of features for improvement related to measurement or control strategies can be added to them. This can be performed either from the use of functional blocks in a Foundation Fieldbus network or with the use of dedicated hardware.

This research proposes a neural network insertion, through remote parameterization, to a DSP (Digital Signal Processor) board in a way that it processes the signal originated from a sensor or a sensors network, increasing the processing capability (intelligence) of the field devices.

The DSP devices can be considered as being a type of microcontroller adapted to applications of digital signals processing in the sense that they possess an optimized architecture for applications that require an intensive computation, besides the devices with specialized functions integrated to the chip. There are, on the market, a great number of manufacturers of these devices, the choice of the best model coming to the user, based on the characteristics of its application.

The module used for the development of this research was SPECTRUM DIGITAL TMS320lf2407 EVM, that incorporates as a central processing unit the 16 bits DSP TMS320lf2407, from Texas Instruments, Inc. Among the principal characteristics of this family of DSPs the following stand out: maximum operation frequency of 40 MHz, SRAM memory of 128 KWords, EEPROM/Flash memory, Analog/Digital converter with 10 bits resolution and 16 input channels, and the capacity to execute conversions in an approximate 500 ns.

The choose for a DSP system presents many advantages related to information processing, once some of the functions are already implemented in the development module, for example the digital/analog 16 bit converter. This makes the implementation process easier and faster.

Fixed point or floating point data treatment units is available as well as optimized architecture for intensive computation, the special addressing modes for efficient support and operations related to real time signal processing, such as circular addressing for filter treatment.

The architecture is composed of many modules, where each one has a specific function: converter circuit of the sensor signal, dedicated hardware for protocol implementation of the intelligent and supervisor functions that should control and monitoring process variables. In Figure 3, the system architecture is observed.

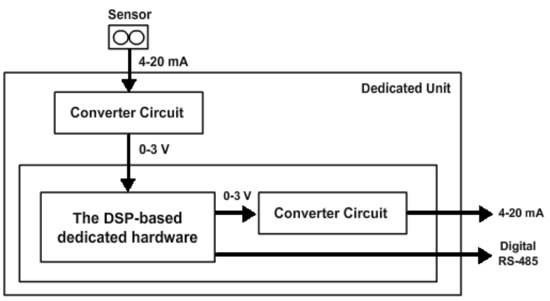

Fig. 3. System Architecture.

The converter circuit exists because the connection of the hardware to a sensor is not easy. The sensor signal is connected to the converter circuit, capable of converting levels of current (4 to 20mA) into levels of tension (0-3.3V), so that can be processes by the DSP.

The signals that were processed by the converter circuit are digitalized by the analog/digital converters, inserted into the DSP board, and manipulated. Signal processing is then realized by the DSP board through a neural network, off-line trained, and the weights are remotely spread by the supervisor.

For trustworthy communication between the supervisor and the DSP board, a serial asynchronous communication Modbus protocol was used. It is specified here that the protocol must transfer data in a RTU mode through RS232 interface. Data transmission mode basically defines how the data are "packaged" in the message. In the RTU mode, for each message data word, only one hexadecimal standard character is sent.

In Figure 4 the software architecture is presented, where master is the supervisor and it has two distinct behaviors: initially as a messages transmitter towards the slave unit and after, it is the receptor of slave messages. The implementation of the supervisor was done by the licensed program, Ellipse PRO Master Full.

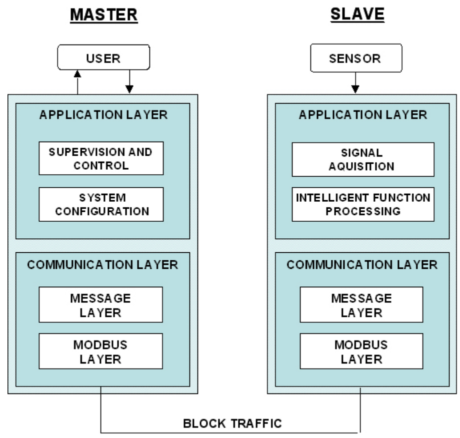

Fig. 4. Software Architecture.

The slave, in turn, will be the system based on dedicated hardware using DSP board, and it will only transmit messages to the master when they be requested.

With the objective to guarantee a trustworthy traffic for message frames, in the operations of reading and writing, a Communication Layer was implemented, as much in the supervisor as DSP environment.

This layer is divided in Modbus Layer and Message Layer. The Modbus Layer is responsible for the sending and receiving of message frames, and Message Layer mounts the message frames to be sent, obeying the Modbus protocol, and interprets the received message frames, guaranteeing that the communication, between master and slave, is in a common language.

The Message Layer was implemented into DSP board, once the protocol Modbus is not native of this hardware. In the reading and writing operations the Message Layer adds addressing data that allow to differentiate the received structures and to allocate them in the appropriate variable for the correct execution of the implemented algorithms.

The Application Layer is responsible for managing the data proceeding from Communication Layer and external environment, guaranteeing the correct functioning of applications.

The Application Layer functions implemented in the supervisor comprehend:

- **System Configuration -** it is responsible for loading and sending all the parameters of a trained neural network. The parameters that were sent are values of weights, bias, architectures and activate function codes of each neural network layer. These datas are loaded through a *.xls* file or manually in the supervisor screen as shown in Figure 6.

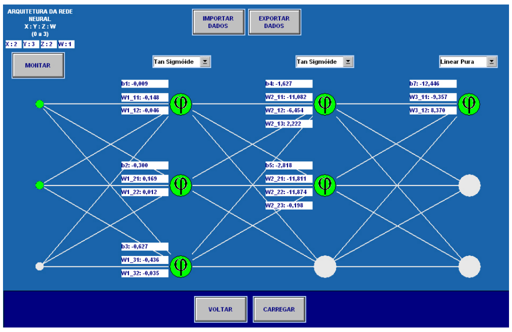

Fig. 5. Supervisor screen where neural network parameterization is specified.

- **Supervision and Control -** it acquires the process variable, controlling the communication, modifies some parameter of neural network instantiated in the DSP board and emits reports and/or alarms.

The Application Layer functions implemented in the DSP comprehend:

- **Signal Acquisition -** its function is the sensor's signals digitalization, through analog/digital converters, and dispose them for the application.
- **Intelligent Function Processing -** it submits the sensor datas to the intelligent routines implemented by the neural network, once all their parameters had been instantiated.

IV. Application

For validation propose, in this section will be presented two application examples based on neural networks. In both experiments, the neural networks architecture employed were the MLP (Multilayer Perceptron)[6].

The first application chosen to illustrate the proposed system operation was a neural network, which the aim is to learn and generalize data set in a way to present similar behavior to the following square function:

$$y = -x_1^2 - x_2^2 \tag{1}$$

This application can specifically represent a problem of sensor merger, which is achieved by two sensor information combination or different information sources to produce the desired information, assuming that no sensor individually is ideally adjusted to do the job.

The neural network architecture used in this experiment was 2-3-2-1, which means that the network has two input neurons, three neurons in the second hidden layer, both presenting the sigmoid tangent as activation function, and one neuron in the output layer which has a linear activation function. Figure 6 illustrates this neural network architecture.

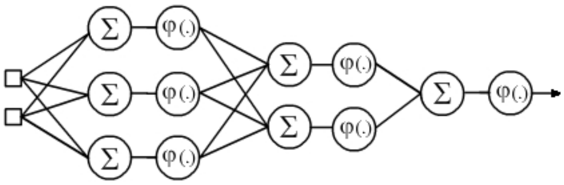

Fig. 6. Neural Network Architecture.

Network training occurred in a off-line mode in a PC computer (no approximation) and the data set, used in training, was formed by input values varied between -1 and 1, with 0.1 variation. In accordance with the equation (1), the desired outputs are shown in the Figure 7.

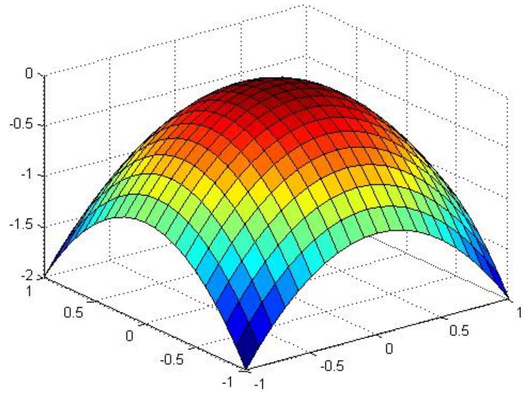

Fig. 7. Set of three dimensional training data.

With the neural network training and its convergence, the parameters such as architecture, activation function codes, weights and bias, were transferred by the supervisor to DSP in a remote form, by the Modbus protocol.

Once the neural network instantiated in the DSP, a data set that contains random values between -1 and 1 for both input is generated internally in the DSP, sent to a neural routine that presented the similar behavior to the desired function. In Figure 8 is observed the neural network reply, captured by the supervisor, to the input data set.

For one better evaluation of the results, was calculated the medium square error between the neural network output vector in environment DSP and the reply of the simulated function. The joined error is 1.6×10^{-2} and its graph is presented in the Figure 9.

Another example mentions the classical linearization technique to improve the measurements of sensors [7]. Then, by processing the sensor output a linear behavior is achieved. This sensor preprocessing can be carried out by programming software functions on computer-based data acquisition systems.

For illustration purposes, in this paper is used a sinusoid as original signal seen in the equation 2. This signal

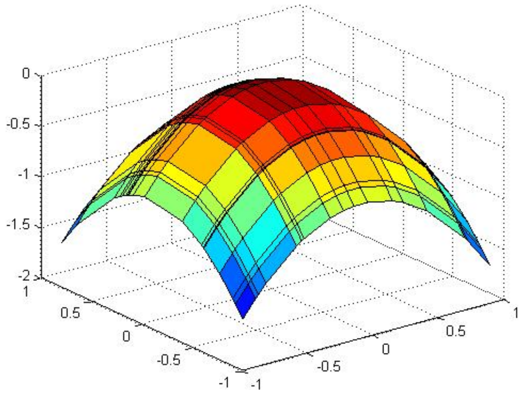

Fig. 8. Neural network response at DSP board.

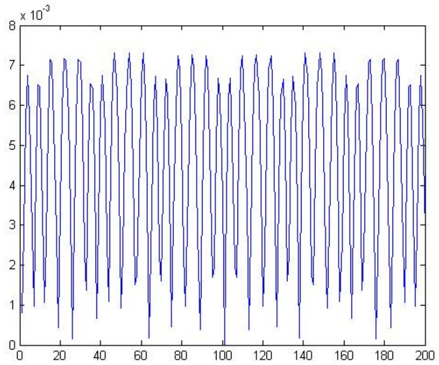

Fig. 9. Error between neural network output and function reply.

is processed by non-linear function seen in the equation 3, that simulates a non-linear sensor. Then, the output sensor will be the input to neural network. The desired neural network output should rebuild the original signal. The implemented sequence for this application can be seen in the Figure 10.

$$r = 5 * sin(t) + 2 \tag{2}$$

$$x = 2/(1 + exp(-2 * r)) - 1 \tag{3}$$

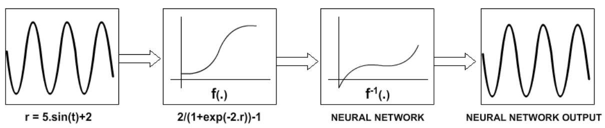

Fig. 10. Application sequence implemented to linearization process.

The training was carried through with the architecture 1-3-1 (Figure 12) and the activation functions for hidden and output layers are hyperbolic tangent and linear, respectively.

After training, to execute the application the neural network parameters are instantiated into DSP board by supervisor. In the execution phase, the neural network gets the sensor signal and applies a linearization process. The obtained result is shown in the Figure 12.

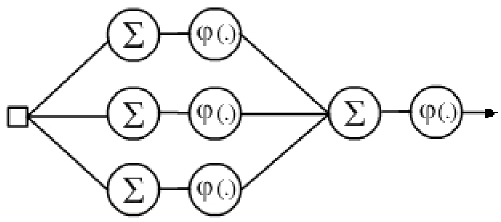

Fig. 11. Neural network architecture.

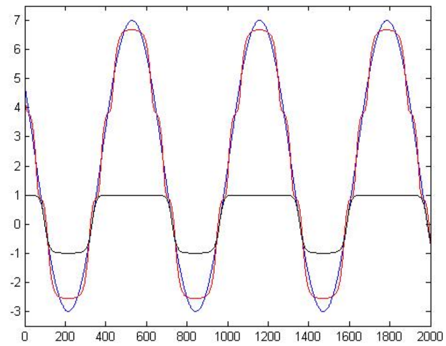

Fig. 12. Original signal (blue), sensor signal (black) and neural network output (red).

It is observed that the neural networks have performed successfully their roles not only in the simulated environment as well as the real one, proving their learning and generalization capacity. The success in parameters transferring and the role of the DSP can be observed as well.

V. Conclusion

The work presented a technical viability study related to the use of artificial neural network in DSP boards, through remote parameter configuration realized by a supervisor using Modbus protocol and RS232 interface, showing that implementation process is possible.

The tests presented in this paper showed satisfactory results, because the neural networks had their parameters correctly carried into the DSP and presented good performance in validation experiments.

Other aspect that must be highlighted is that the Application Layer proposed in this paper allowed the development of application in flexible and fast way.

Finally, notice that actual hardware technology already supports implementations based on intelligent techniques into devices. Thus, there is a wide area to use these techniques in industrial applications.

References

[1] J. Berge, "Fieldbuses for process control: Engineering, operation and maintenance, the instrumentation, systems and automation," *Society-ISA*, 2001.

[2] I. M. Costa, A. D. Doria, J. D. Melo, and J. A. N. Oliveira, "Embedded fastica algorithm applied to the sensor noise extraction problem of foundation fieldbus network," *IJCNN'2005*, 2005.

[3] D. R. C. Silva, A. D. Doria, J. D. Melo, and L. A. H. Guedes, "Implementation of neural networks in foundation fieldbus environment using standard function blocks," *ETFA'2005*, 2005.

[4] D. R. V. Pereira, A. D. Doria, J. D. Melo, and E. C. Júnior, "The implementation of the self-calibration, self-compensation and self-validation algorithms for foundation fieldbus sensors using standard function blocks," *CIMSA'2005*, 2005.

[5] Modbus-IDA, "Modbus application protocol specification v1.1a," 2004.

[6] S. Haykin, *Neural Networks: A Comprehensive Foundation.* Prentice-Hall, 1998.

[7] N. Medrano-Marqués and B. M. del Brío, "Implementing neural networks onto low-cost microcontrollers for sensor processing," *IEEE Transactions on Circuits and Systems*, 2005.