

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

3,700

Open access books available

108,500

International authors and editors

1.7 M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Equipment Based on the Hardware in the Loop (HIL) Concept to Test Automation Equipment Using Plant Simulation

Eduardo Moreira¹, Rodrigo Pantoni^{1,2} and Dennis Brandão¹

¹*University of São Paulo,*

²*Federal Institute of São Paulo,
Brazil*

1. Introduction

Considering the difficulties to test a real control system at a learning laboratory, related to equipment acquisition or physical installation, it is necessary to develop a simulation system that will act as parts of the real system.

In this scenario, through the use of recent computational resources in hardware and software, it is possible to explore the concept named Hardware in the Loop (HIL), which consists on well known technique mostly employed on the development and testing of electronic, mechanical and mechatronic real equipments by the use a data and signal interface between the equipments and computational real-time systems.

This work proposes a HIL-based system where a Foundation Fieldbus control system manages the simulation for a generic plant in an industrial process. The simulation software is executed in a PC, and its purpose is the didactic use for engineering students to learn to control a process similar to the real system.

The plant is simulated in a computer, implemented in LabVIEW and part of the fieldbus network simulator named FBSIMU (Mossin et al., 2008). Foundation Fieldbus physical devices are configured to operate in a control strategy and interact with the simulated environment.

Results demonstrate it is a feasible technique that can be extended to more complex and elaborate control strategies.

2. Works related to hardware in loop for industrial networks

HIL concept is well-known and used in the research area. For this reason, this section cites the most recent works related to HIL applied to industrial networks.

(Godoy & Porto, 2008) suggest the use of HIL technique to develop Networked Control Systems (NCS) with CAN (Controller Area Network) networks, presenting the structure for HIL usage and evaluating the benefits of using this tool to develop NCS with CAN.

Huang & Tan (2010) proposed a HIL simulator that provides an efficient development and test platform for real-time systems used in assembling lines of automation factories.

Fennibay et al. (2010) introduced the HIL technique for hardware and software shared in embedded systems. This technique reduces the need for developing models for existing hardware platforms and increases system accuracy and performance.

Li & Jiang (2010) performed a study using a physical system with steam generation together with HIL simulation, inside a training simulator of an industrial nuclear plant. The purpose is the study of fieldbus networks for steam systems. Results showed that some existent delays could be eliminated, according to the authors' suggestions.

The next chapters describes some important details of the Foundation Fieldbus protocol as well as its simulator named FBSIMU, which includes a simulation module for the dynamics of industrial plants, developed over LabVIEW and used in an HIL experiment with a real fieldbus network.

3. The foundation fieldbus protocol

The term FOUNDATION Fieldbus indicates the protocol specified by the Fieldbus Foundation. It is a digital, serial, bidirectional, and distributed protocol which interconnects field devices such as sensors, actuators and controllers. Basically, this protocol can be classified as a LAN (Local Area Network) for instruments used in process and industrial automation, with the ability to distribute the control application through the network.

This protocol is based on the ISO/OSI (International Organization for Standardization/Open System Interconnection) seven-layer reference model (International Organization For Standardization, 1994). Although being based on the ISO/OSI model, the FF does not use the network layer, the transport layer, the session layer, neither the presentation layer, because it is restricted to local applications. The entire network structure of the FF concentrates on the physical layer, the data link layer (DLL) and the application layer. Besides these three implemented layers, the protocol defines an additional layer called User Application Layer.

The FF Physical Layer, named H1, uses shielded twisted pair cable as a communication medium. The H1 specifies a 31.25 KBit/s baud rate with Manchester bit codification over a bus powered channel. The network topology configuration is flexible: it is typically configured with a trunk and several spurs, attending certain physical and electrical limitations regarding maximum lengths and number of transmitters.

The DLL carries the transmission control of all messages on the fieldbus and its protocol grants to the FF network temporal determinism. The communication is based on a master-slave model with a central communication scheduler (master), named Link Active Scheduler or LAS. This node performs the medium access control (MAC).

Two types of DLL layer are standardized: Basic and Link Master. A Basic DLL transmitter does not have LAS capabilities, it operates passively as a communication slave. A Link Master DLL transmitter, on the other hand, can execute LAS functions and thus, if the active LAS node fails, become the LAS node.

The FF Data Link Layer supports two transmission policies: one addressed to scheduled cyclic data and another to sporadic (unscheduled) background data. These two communication policies share the physical bus but they are respectively segmented in cyclic time slots or periods. In the scheduled communication period, most process variables

generated by periodic processes are transmitted cyclically according to a static global schedule table loaded on the LAS node. This cyclic transmission mode has higher priority over acyclic transmission modes.

A periodic process can be defined as a process initiated at predetermined points in time, also called a time-triggered process. The period for this class of process is typically some milliseconds, and it is mandatory to consider that the generated data must be delivered before the next data is available. This type of periodic data is usually related to measurement and control variables (Cavalieri et al., 1993).

The sporadic or unscheduled communication is used to transmit non periodic, or aperiodic, data generated by sporadic processes not directly related to the control loop cycles, but to configuration actions and data supervision efforts. The unscheduled transmissions are dispatched under a token pass scheme. A token that circulates among all active nodes on the bus is used in FF protocol. Once a transmitter receives the token, it has granted the right to send pending aperiodic messages with a minimum priority for a specific time period.

Non periodic (or event-triggered) processes are initiated as soon as specific events are noted (Pop et al., 2002). The event-triggered processes are unpredictable and usually related to alarm notifications, configuration data and user commands as cited before. Although the acyclic traffic is less frequent than the cyclic one, the acyclic data should be delivered also prior to a certain time deadline, according to the system requirements.

For a description of the MAC operation on both cyclic and acyclic phases, refer to (Hong & Ko, 2001) (Wang et al., 2002) (Petalidis & Gill, 1998).

The FF User Layer is directly related to the process automation tasks themselves and it is based on distributed control or monitoring strategies of Function Blocks. Function Blocks (FBs) are User Layer elements that encapsulate basic automation functions and consequently make the configuration of a distributed industrial application modular and simplified (Chen et al., 2002). Distributed among the transmitters, the FBs have their inputs and outputs linked to other blocks in order to perform distributed closed control loop schemes. When blocks from different transmitters are linked together, a remote link is configured and mapped to a cyclic message. Considering that all cyclic messages should be released in a predetermined instant defined on a schedule table, and that they carry data generated by the FBs, it is adequate to synchronize the execution of the FB set on the system with the referred cyclic transmissions schedule table. This solution leads to the concept of joint scheduling (Ferreiro et al., 1997).

The Foundation Fieldbus standardized a set of ten basic function blocks (Fieldbus Foundation, 1999a), a complementary set of eleven advanced control blocks (Fieldbus Foundation, 1999b), and a special flexible function block intended to be fully configurable, i.e., internal logic and parameter, by the user (Fieldbus Foundation, 1999c). The standard and advanced block sets provide mathematical and engineering calculations necessary to configure typical industrial control loops strategies, while the flexible function block can be applied to custom or advanced controls or to complex interlocking logics based on ladder nets. It is important to state, however, that the standard is open at this point, permitting the integration of "user-defined" custom function blocks in order to enhance the capabilities of FF control system and make the integration of novel control techniques possible.

4. FBSIMU architecture

The basic concept of the FBSIMU architecture is to map each Function Block, as well as the plant, in an independent LabVIEW application, also named Virtual Instrument (VI). The configuration of the whole system is centralized in the FBSIMU.CONF module. This module's front panel (GUI) is inspired by commercial fieldbus configuration tools.

As mentioned before, the FBSIMU is focused on the function block application layer and it is composed exclusively of software according to a modular and extensible architecture. The simulator was developed in LabVIEW using the G graphical programming language, "native" language in this environment. Each FBSIMU module or software unit simulates an element or a structure of a real FOUNDATION Fieldbus system (Mossin et al., 2008).

4.1 Function block simulation

The Function Block modules are programmed into the FBSIMU according to the FF specifications directions and, consequently, the usage and configuration of a simulated control loop on the FBSIMU environment is identical to a real FF system.

A VI library has been developed (Pinotti & Brandão, 2005) to provide a range of typical Foundation Fieldbus control and acquisition functions according to the standards. Another VI functionality facilitates the development and integration of standard and custom FBs to the system. These functions encapsulate different FF calculations and data type manipulations necessary to build Function Blocks, configuring a "LabVIEW Foundation Fieldbus Tool Kit". A Function Block seed module is also used to facilitate the process of developing and integrating new projects. The seed has the whole FB module structure (an empty structure) and directions to proceed with a FB project from the design to the final test procedures.

Each FB module is built as two different versions that share the same FB core: stand-alone and process. The stand-alone FBs are executed by user commands and controlled by its front panel. Its execution can be performed independently of any other module, so the user is able to test the FB and simulate its operation under a controlled condition of inputs and outputs. The graphical user interface is intuitive and enables the user to execute the FB continually or in a step-by-step mode.

The process version of a FB, on the other hand, is controlled remotely likewise real FBs. Each process FB has a unique identification and its operation is controlled by the user through the following commands:

- FB_Read: this service allows the value associated with a block parameter to be read.
- FB_Write: this confirmed service allows the value associated with a block parameter to be written.
- FB_Exec: this service triggers the block algorithm to be executed.
- FB_Reset: this service allows default values associated with all block parameters to be written.

Process FBs do not have front panels; they are instantiated by the FBSIMU.CONF in each simulation process. The communications between process FBs and the FBSIMU.CONF are performed programmatically and dynamically by the LabVIEW function "Call by Reference Node". It is important to note that the industrial transmitters are not considered in the FBSIMU architecture, i.e., function blocks are instantiated on the simulation without being

allocated in specific “virtual” transmitters. The FBSIMU.CONF module front panel for fieldbus configuration is shown in Figure 1.

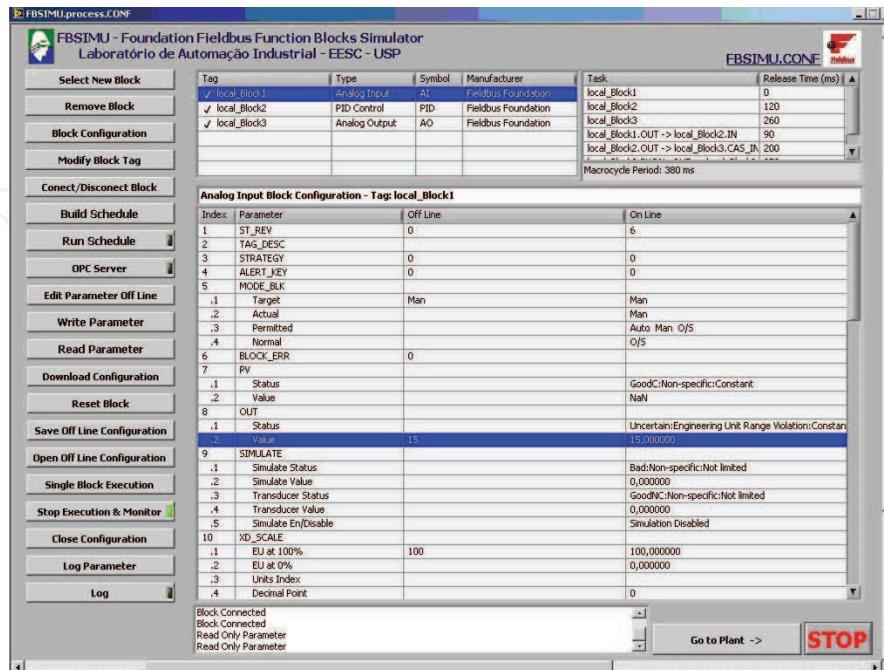


Fig. 1. FBSIMU front panel for fieldbus configuration

4.2 Physical plant simulations

The plant module cyclically executes a discrete single variable (SISO) linear ARX (Auto-Regressive with Exogenous Inputs) mathematical structure (Ljung, 1999), according to equation 1. This module is configured by the FBSIMU.CONF and simulates the controlled plant. The adopted ARX structure is represented by equation (1), where k is the discrete time instant, Y is the output vector, U is the input vector, i is the number of MIMO plant inputs and outputs, na is the number of output regressors, and nb is the number of input regressors. In the current version, i is set to 1 (one) to reflect a SISO model.

$$Y_{ix1}(k) = \sum_{s=1}^{na} A_{s_{ixi}} \otimes Y_{ix1}(k-s) + \sum_{s=1}^{nb} B_{s_{ixi}} \otimes U_{ix1}(k-s) \quad (1)$$

The simulated plant dynamic behavior is modeled on the dynamic matrixes A and B . It must be observed that the number of regressors limits the model dynamic order - in the actual version it is limited to third order systems - and that all regressors must be initialized prior to starting the simulation. A white noise generator function adds a simulated acquisition noise to each plant output bounded by user configurable amplitude.

As the user chooses the plant order (1st, 2nd or 3rd) and dynamics (gain for 1st and 2nd order systems, damping ratio, natural frequency and time constant), the selected plants' Bode Magnitude Chart, Pole-Zero Map, Root Locus Graph and the Step Response are instantly presented on the front panel. The third order system is composed of a first order system in series with a second order one, both adjustable by the user, as stated.

A white noise signal can be also introduced with configurable absolute amplitude over the plant output. Figure 2 shows the FBSIMU.CONF module front panel for plant configuration.

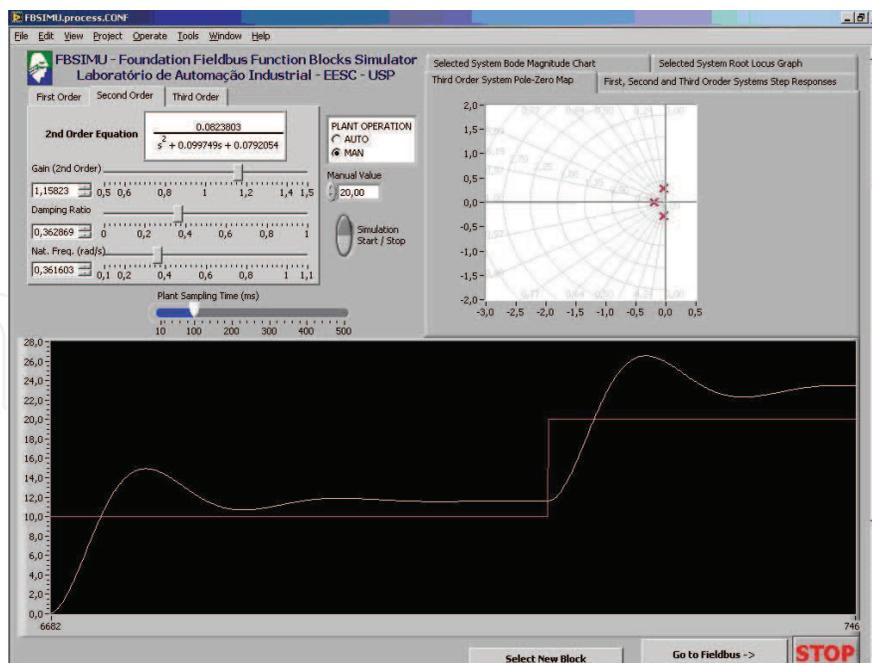


Fig. 2. FBSIMU front panel for plant configuration

4.3 Simulation architecture

The proposed execution model for the fieldbus simulation on FBSIMU can be considered hybrid, because some tasks are event-driven while other tasks are time-triggered. All tasks related to the user interface are event-driven, they are executed after a user action such as selecting a new block, configuring schedule table, saving a configuration or starting the execution.

On the other hand, the tasks related to executing FBs according to a schedule table, plant simulation, and online monitoring of FBs are time-triggered.

Due to the fact that all tasks are performed on a single microprocessor they are, naturally, concurrent. The proposed solution for preventing unexpected delays of time-triggered tasks (considered critical) due to executing event-driven tasks (considered non-critical) is adopting priority levels for each task and preemptive execution mode.

In the preemptive execution mode, a higher priority task that is ready to execute preempts all lower priority tasks, which are also ready to execute or actually during execution.

Table 1 summarizes the FBSIMU task set and its timing and execution characteristics.

Module	Priority	Execution	Timeout	Determinism
GUI & User commands	Low-Low	Event driven	1 sec.	No
FB Schedule	High-High	Time triggered according to the schedule table	No	Yes
Plant Execution	High	Periodic with configurable period	No	Yes
Online FB Parameters Monitoring	Low	Periodic with period = 500ms	No	Yes

Table 1. FBSIMU task set

5. Proposed hardware in loop

This section presents the topics related to developing a process automation system based on the HIL concept: the architecture, variables and equipments dimensioning, materials used, and application tools used. The goal is to prove the efficiency of using the HIL concept to create a process automation didactic kit with analog measuring for a simulated plant.

5.1 Architecture

The simplest architecture of a device using the HIL concept consists of two components: the system simulator and the related control. It is important to consider the communication interface of the devices, since the interaction between them is not really common from the point of view of the applications they were designed for.

The project development is divided based on these components, because from the HIL applications viewpoint these areas are modular and, apart from few restrictions, their components can be replaced without compromising the quality of the tests. However, this is just one path to start researching the devices to be used.

To reach the final architecture, intermediate steps decide the types of components necessary for the proper operation of the device. Available resources were analyzed at each step, considering some restrictions, such as: device availability, architecture requirements and simplicity (only one input and one output).

The diagram, illustrated in Figure 3, was defined after the final step, and it contains all types of devices necessary for proper operation.

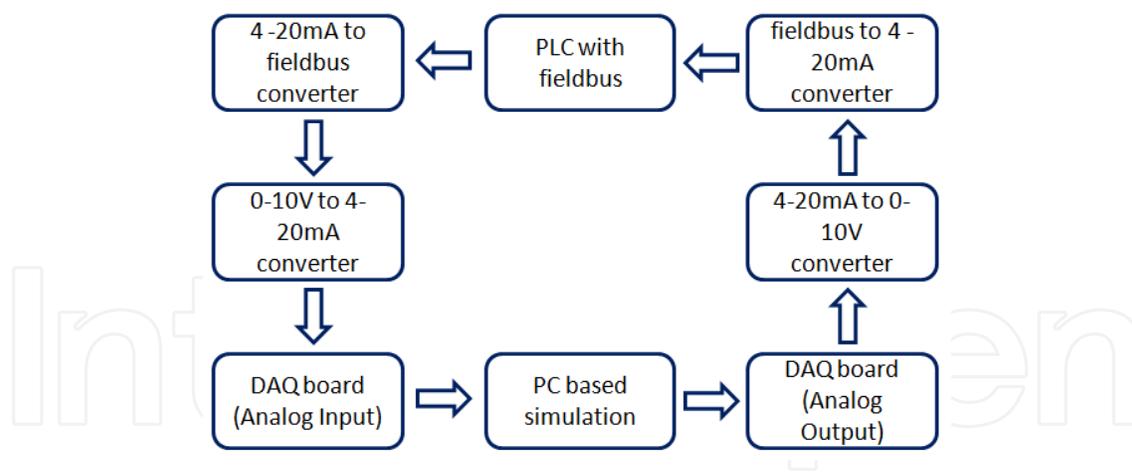


Fig. 3. View of the architecture

5.2 Materials and dimensioning

After defining the final architecture, it is possible to select the components available for the kit. Table 2 lists all physical components used in the project as well as the corresponding description, including the device model and manufacturer. Components with the description “---” are accessories to the device operation, but they do not perform a role related to control/simulation.

Role in Architecture	Device Model	Manufacturer
Bridge	DF51 1x10 Mbps, 4xH1 - FF	SMAR
---	PS302 - DC Power Supply for DF51	SMAR
---	DF53 - Power Supply Impedance for FF - H1 network	SMAR
---	BT302 Bus Terminator	SMAR
Converter FF -> 4-20mA	FI302 - Triple Channel Fieldbus to Current Converter	SMAR
Converter 4-20mA -> FF	IF302 - Triple Channel Current to Fieldbus Converter	SMAR
---	BT302 - Bus Terminator	SMAR
Converter 4-20mA -> 0-10V	TCA1100 - 1-Channel Current to Voltage Converter	TECNATRON
Converter 0-10V -> 4-20mA	TCA1100 - 1-Channel Voltage to Current Converter	TECNATRON
---	24 V DC Power Supply for TCA1100	TECNATRON
Data Acquisition Boards	NI-DAQmx 6221	NATIONAL INSTRUMENTS
PC	Personal Microcomputer with two NIC	Not available
Software Programs	LabVIEW®8.2 - for simulation Syscon 7 - for bridge configuration	NATIONAL INSTRUMENTS & SMAR

Table 2. Physical components

Devices were installed according to the corresponding datasheets and manuals. Figure 4 shows the electrical installation schema among the devices listed in Table 2.

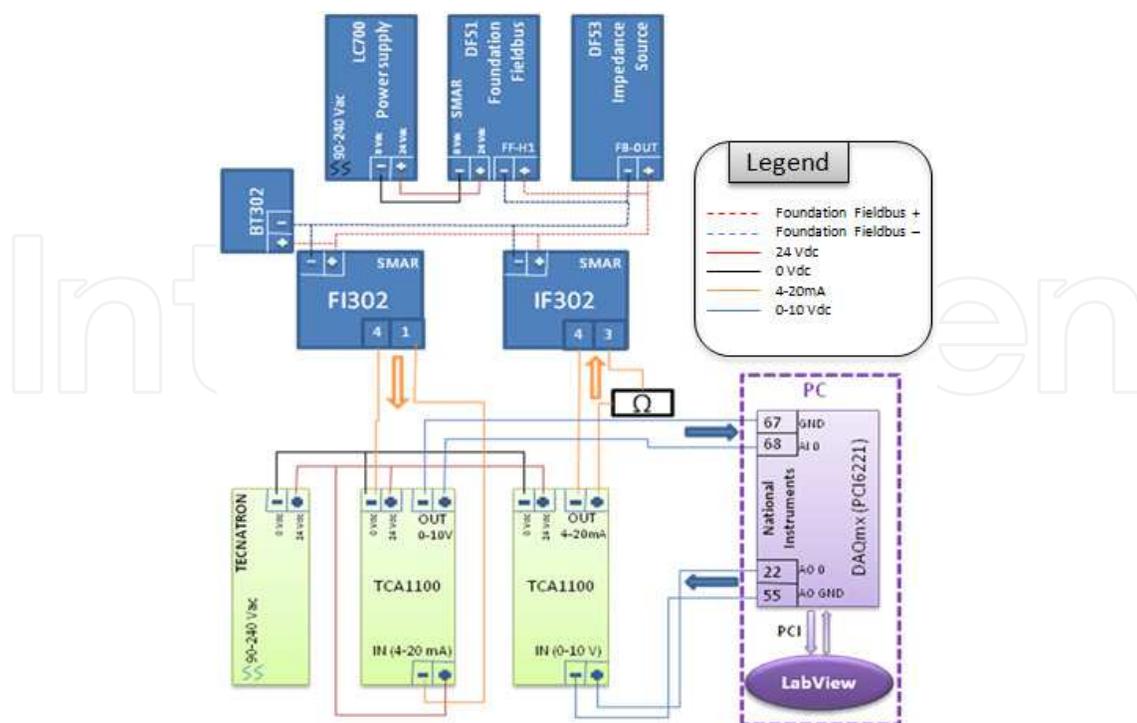


Fig. 4. Electrical installation among components

5.2.1 Communication interface

Automation systems are developed to operate with real systems and not computer simulations, therefore it is necessary to develop an interface that packs the signals and assures a coherent signal conversion.

The interface is composed by devices that connect simulation to control instruments. The interface is used when conforming input and output signal from the virtual plant created, so the field devices are able to read those signals.

National Instruments Data Acquisition board - NI-DAQmx PCI 6221 (National Instruments, 2003) - was chosen as the communication interface to assure compatibility and signal quality, because the virtual plant is configured by LabVIEW, software product from the same manufacturer. However, since this board generates signals by varying the 0-10 volts voltage, it was necessary to add transducers because field devices read and write 4-20 mA standard signals.

Figure 5 shows the electrical installation schema for the interface to pack signals. The box on the right represents the Analog Digital Converter NI-DAQmx PCI 6221 connected to the PCI slot in the computer executing simulation. The other boxes are components for the 0-10 volts to 4-20 mA standard converter (s).

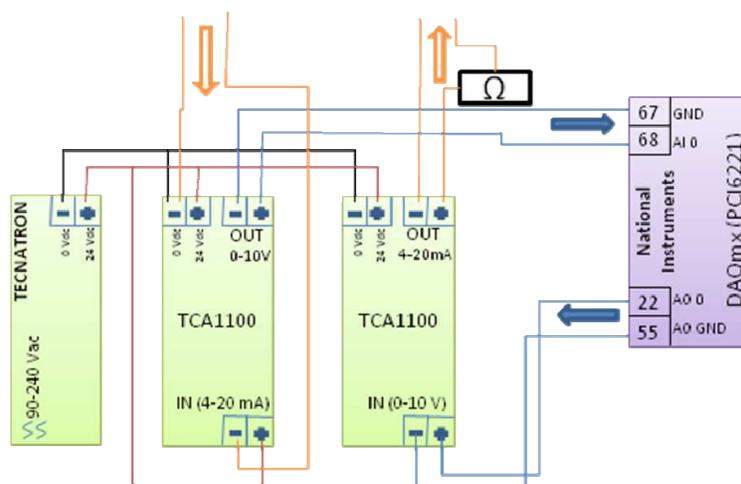


Fig. 5. Interface electrical installation

The interface operation is very simple. The control signal from the FI302 transmitter, from SMAR, is sent to the 4-20 mA converter input, and then converted to the 0-10 volts standard, so the National Instruments data acquisition board can read and send the signal to the simulator.

On the other way around, the simulator sends the signal converted to the 0-10 volts standard by DAQ 6221, straight to the transducer to be then converted to a 4-20 mA signal. After that, the signal is read by IF302.

A system that reads many variables and operates several instruments is clearly more complex because it requires a large amount of transducers, but, it would be only necessary to repeat basic transducer units describe here. A boiler, for instance, may request reading from the internal pressure, temperature, or internal liquid flow. Control operation for this system may involve liquid input and output flow control, pressure valves, etc.

5.3 Software

After installing the physical instruments of the project, simulation and FF instruments configuration are implemented. LabVIEW® is used as a development environment to

configure the simulator, and Syscon (Smar Syscon, 2011), configures and monitors the Foundation Fieldbus network.

5.3.1 LabVIEW®

In this project, the FBSIMU simulation environment is a software tool developed using LabVIEW®, from National Instruments. The advantage of this environment is the fact that it is a platform for creating virtual instruments (VIs), with emphasis to the external signal acquisition.

The Simulator was developed to be intuitive and simple to use. Simply define the parameters intrinsic to the transfer function that represents an industrial process, and the user will view the response to this function in a graphical screen. This functionality was designed considering three main areas: the response chart, the fundamental analysis chart, and parameter inputs. In order to manually control the input, a scroll button was implemented to enable the user to visualize the plant reaction (transfer function) according to the input alteration.

Function $H(s)$ may represent first-order, second-order, and third-order functions. Parameters can be altered according to the user's need on the same tab showing the function. There are three tabs: the first-order, second-order, and third-order. Besides showing the parameters to be configured, these tabs also show Function $H(s)$ - response to the unitary impulse - related to the selected tab, and the user can understand how altering the parameters can affect the function.

Select the tab related to the function to be used to activate the function, and then, execute the function.

While the parameters values of the transfer function vary, the user can analyze the function in several ways. For instance, when the fundamental frequency bar varies, the impulse response chart indicates the function behavior according to each unitary impulse on the function.

For a real time system, input and output information must be updated constantly. During implementation, it was necessary to discretize the transfer function, then it would be possible to add a parameter (sampling time) to reduce processing costs. This fact would contribute to the veracity of the simulation, if the measuring devices were digital, which is not the case in this project, based on analog measuring.

After the discretization of the transfer function, the function output is calculated and sent to the data acquisition board output. This result is also plotted on a chart, in the software screen.

Input and output processing requires three steps to be executed:

- I. Read the value the controller is sending to the input function, directly from the data acquisition board;
- II. Process the input value according to the applied function - discretization and output calculation;
- III. Write the value directly to the data acquisition board so the controller can read it and make a decision.

Similarly, this process can also be applied to other analysis charts in the simulator. Therefore, system designers, as well as control theory students, will have a clear view, quickly calculated, of several graphical analyses of the simulated transfer function.

The NI-DAQmx 6221 data acquisition board has VIs (or Virtual Instruments, which represent several function blocks) for control and communication ready to be used in LabVIEW®. These functionalities enable the simulator to connect directly to the world outside the computer. Input and output acquisition blocks were connected direct to the input and output of the simulator's transfer function. Thus the information cycle is closed, as indicated in Figure 6.

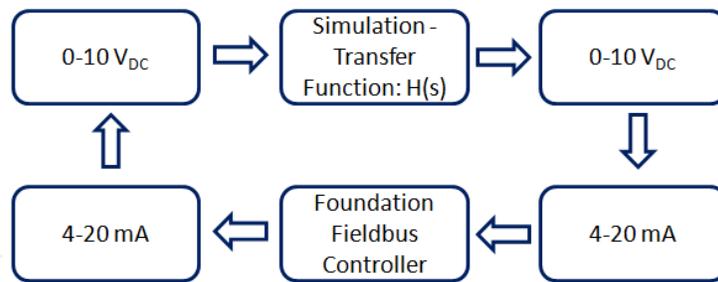


Fig. 6. Connection between simulation and devices

LabVIEW programming is implemented with blocks connected through “lines” where the information travels. Each line can carry information in different formats, from a single Boolean variable to a status matrix. There also loops for cyclic processes, created by the programmer. Figure 7 shows the logical project for the FBSIMU Fieldbus network simulator.

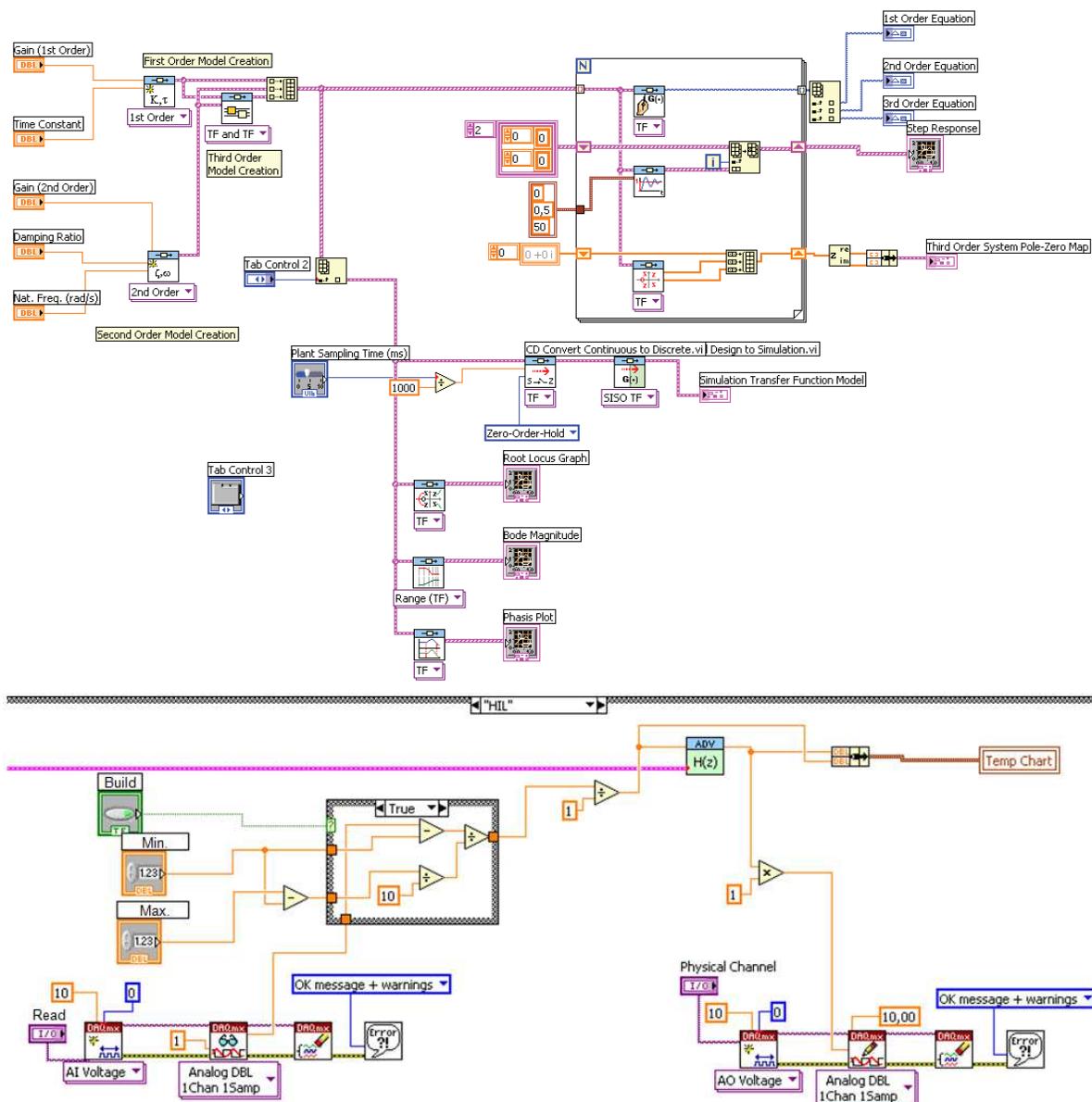


Fig. 7. Instruments and simulation connections

5.3.2 SYSCON

Syscon, a software tool developed by Smar, configures and maintain FF Networks.

In this project, Syscon is responsible for configuring and maintaining the following devices:

- SMAR DF51 – Bridge Foundation Fieldbus
- SMAR FI302 – Foundation Fieldbus to 4-20mA Transducer
- SMAR IF302 – 4-20mA to Foundation Fieldbus Transducer

Considering the list of all blocks available, the most convenient blocks are selected for the application. In this project, the following blocks are configured:

- Analog Input – IF302 function block that converts the analog current signal to Foundation Fieldbus standard selected during configuration;
- Analog Output – FI302 function block that converts the Foundation Fieldbus standard signal to current;
- PID Control – FI302 function block that executes Proportional Integral Derivative control.

The Tests and Results section describes the blocks involved in the control strategies and their corresponding parameterizations, on which the tests and results from this work are based.

6. Tests and results

This section describes how the simulator and its control operation were started, which tests were performed, and results from the tests. Information generated in this phase will be a base to conclude the project. Two tests were performed: Loop test and PID Control test.

6.1 Loop test

The loop test measures all process variables and may indicate how the system behaves, as for response time or error range, for example. In addition, this test usually calibrates instruments. The following paragraphs reports the steps to perform this test.

Initially, the Fieldbus network must be created using Syscon. The bridge (DF51) is added to this network, and the slave devices for measuring and reading (IF302 e FI302) are added to this bridge.

Since no processing is really executed in the devices during this test, only the input and output blocks need to be configured for the IF302 and FI302 devices. Therefore, the next step is to configure parameters for the devices.

The Analog Input block is added to IF302, because the device's internal transducer will only have to convert the 4-20mA signal to the Foundation Fieldbus standard. Figure 8 illustrates Syscon window showing the devices and the blocks configured for IF302. TR, RS, DSP and AI blocks are configured according the parameters and related values listed in Table 3.

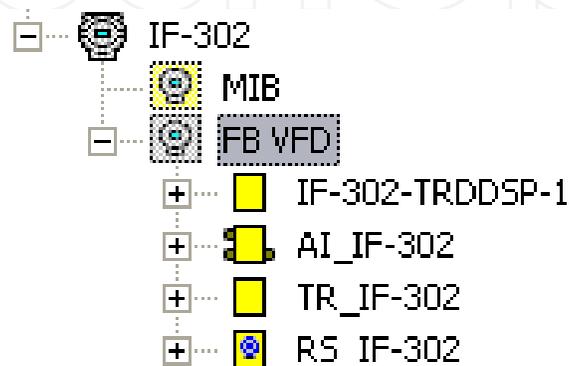


Fig. 8. Syscon window and IF302

Block	Parameter	Value	
TR	MODE_BLK TARGET	AUTO	
	TERMINAL_NUMBER	3	
RS	MODE_BLK TARGET	AUTO	
DSP	MODE_BLK TARGET	AUTO	
	BLOCK_TAG_PARAM_1	AI_IF-302	
	INDEX_RELATIVE_1	8	
	MNEMONIC_1	CUR_FIELD	
	ACCESS_1	MONITORING	
	ALPHA_NUM_1	MNEMONIC	
	DISPLAY_REFRESH	UPDATE DISPLAY	
AI	MODE_BLK TARGET	AUTO	
	XD_SCALE	EU_100	20
		EU_0	4
		UNITS_INDEX	mA
	OUT_SCALE	EU_100	100
		EU_0	0
		UNITS_INDEX	%
	CHANNEL	3	
	L_TYPE	INDIRECT	

Table 3. Function block configuration for IF302

FI302 contains the Analog Output block, because this device will have to convert Foundation Fieldbus standard signals to 4-20 mA signals. Figure 9 illustrates Syscon window showing the devices and the blocks configured for FI302. TR, RS, DSP and AO blocks are configured according the parameters and corresponding values listed in Table 4.

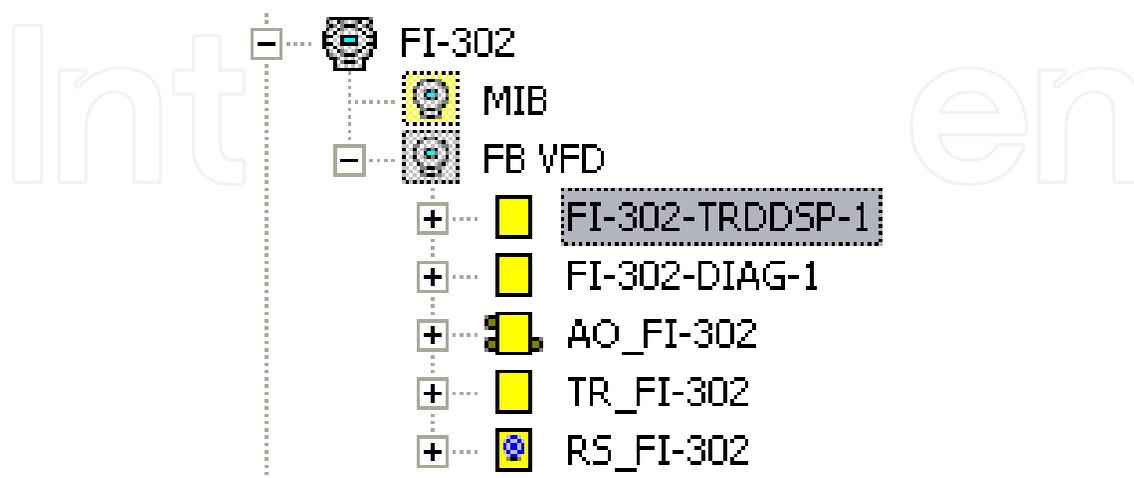


Fig. 9. Syscon window and FI302

After configuring the devices, it is necessary to create the control strategy. In loop test, data travels in loop with no alterations, and the control strategy simply connects an input block to an output block, as indicated in Figure 10.

Block	Parameter	Value	
TR	MODE_BLK TARGET	AUTO	
	TERMINAL_NUMBER	1	
RS	MODE_BLK TARGET	AUTO	
DSP	MODE_BLK TARGET	AUTO	
	BLOCK_TAG_PARAM_1	AO_FI-302	
	INDEX_RELATIVE_1	9	
	MNEMONIC_1	CUR_FIELD	
	ACCESS_1	MONITORING	
	ALPHA_NUM_1	MNEMONIC	
	DISPLAY_REFRESH	UPDATE DISPLAY	
AO	MODE_BLK	TARGET	CAS_AUTO
		NORMAL	CAS_AUTO
	PV_SCALE	EU_100	100
		EU_0	0
		UNITS_INDEX	%
	XD_SCALE	EU_100	20
		EU_0	4
		UNITS_INDEX	mA
	CHANNEL		1

Table 4. Function block configuration for FI302

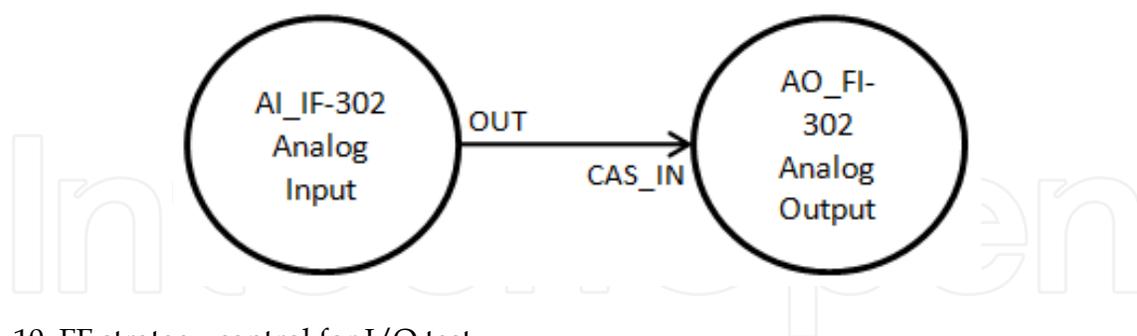


Fig. 10. FF strategy control for I/O test

Once devices are configured and the control strategy is implemented, the plant must be commissioned, and after that, the configuration and the strategy must be downloaded. Finally, the physical control plant is ready to operate.

The next step is using a software tool to complete the prototype data loop. Therefore, a new VI is configured for the test. In this VI, the user can set a value - using the scroll bar - to the loop. If the implementation is correct, this value should appear on a second bar with a slight variation. For example, if value "7" is set on the scroll bar, a sequence close to this value should appear during the loop execution, as indicated in Figure 11.

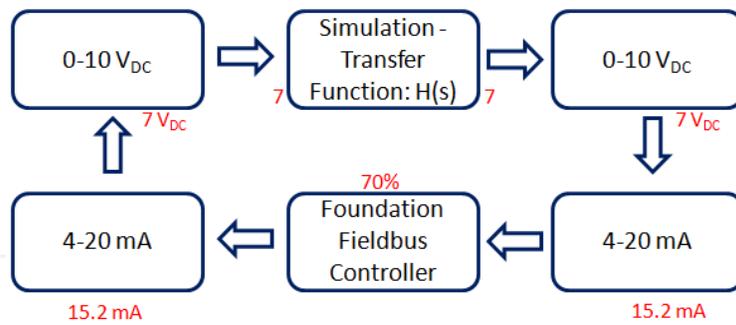


Fig. 11. Loop values for the first test

Figure 12 exemplify the VI specially developed to execute the test, with value “7” being written and read.

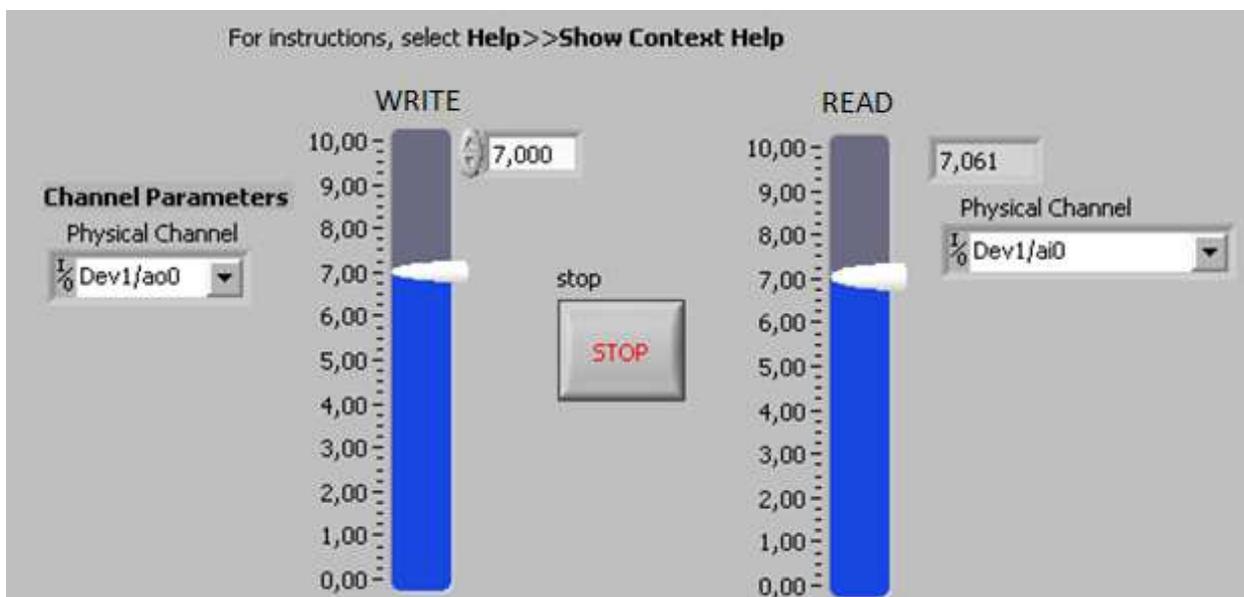


Fig. 12. Screen capture for the VI developed for the test

Figure 13 represents the logic applied to develop the VI from Figure 9.

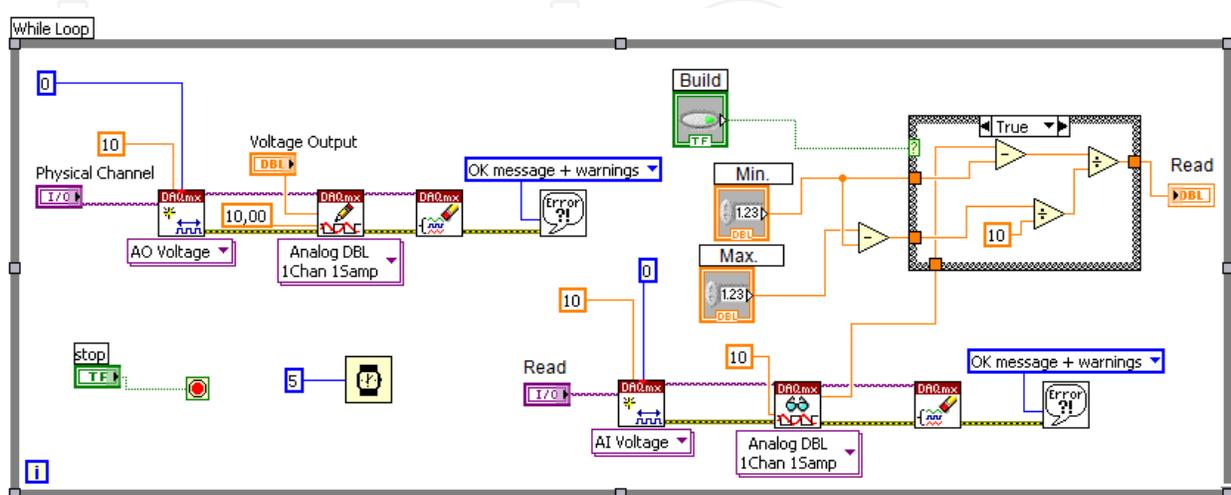


Fig. 13. Logic developed on the VI for the test

During the cycle tests, a difference on the voltage values the controller sends to the simulated plant was detected. Several tests were performed to evaluate this problem and the conclusion is that somehow the analog input from NI-DAQmx 6221 was sampling values different from the values measured externally (with multimeters, for example). Other analog inputs were used, but they all showed the same problem. Several attempts to calibrate the board were made, but none was successful.

Since a more complex calibration would demand more time, it was decided to adopt a mathematical solution by changing the coordinates, so there would be a better signal adequacy. Differences in values are according to the chart in Figure 14.

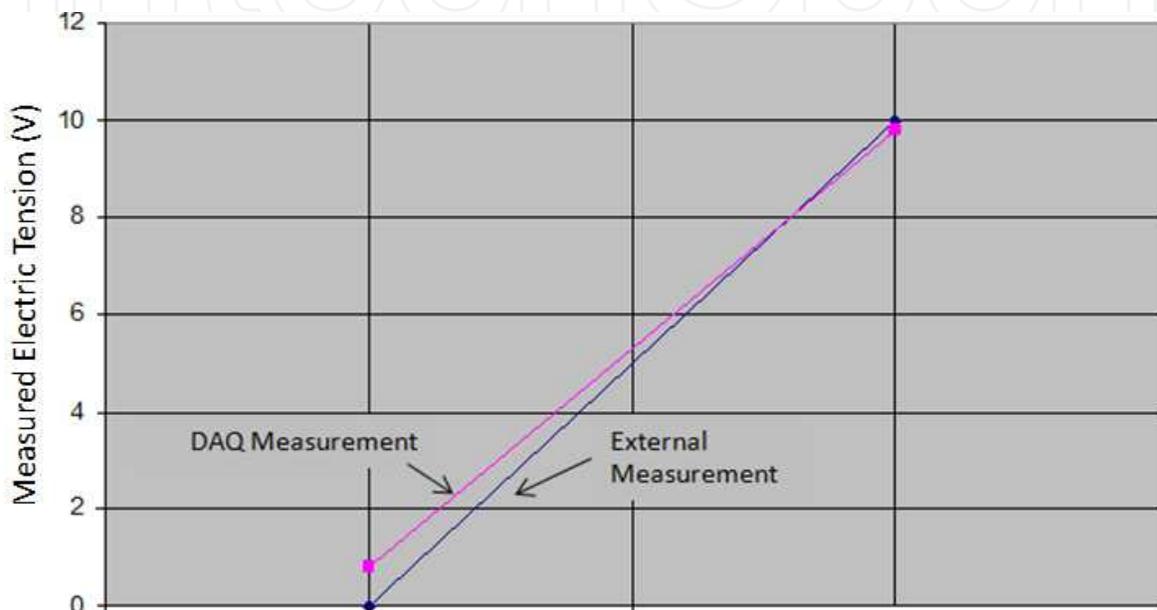


Fig. 14. Difference in values read and corrected

The next step was measuring the delay time for the data to travel the entire loop. A maximum value (10) was set on the scroll bar, and the time the second bar took to get to a very close value was then measured. The results indicated in Table 5 were obtained.

Procedure	No. of Measurements	Average Time
Maximum ascent variation (0 → 10)	10	2,33 s
Maximum descent variation (10 → 0)	10	2,70 s

Table 5. Average delay time for the complete loop

These results are just qualitative and do not accurately represent the reality of the system. However, they give an idea of the delay that occurs while actuating the plant. Delays are caused by processing and conversions the signal undergoes during the cycle.

6.2 PID control test

After the loop test was finished successfully, the plant PID control test is executed. The simulator can be finally executed to test external control. Only the alterations made on the previous item (the changes in coordinates) are included to the program for proper operation. The control diagram is according to the representation in Figure 15.

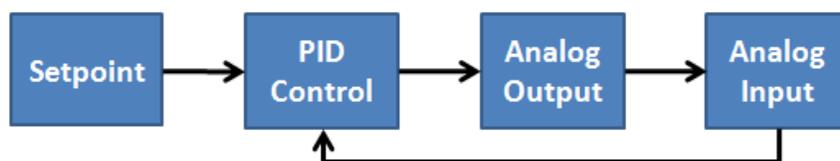


Fig. 15. Control diagram

Unlike the Pure I/O test, the PID Control test requires a control block, therefore the “PID” block was added to the strategy used in the previous test, between the AI and AO blocks, creating a cascade control strategy as illustrate in Figure 16.

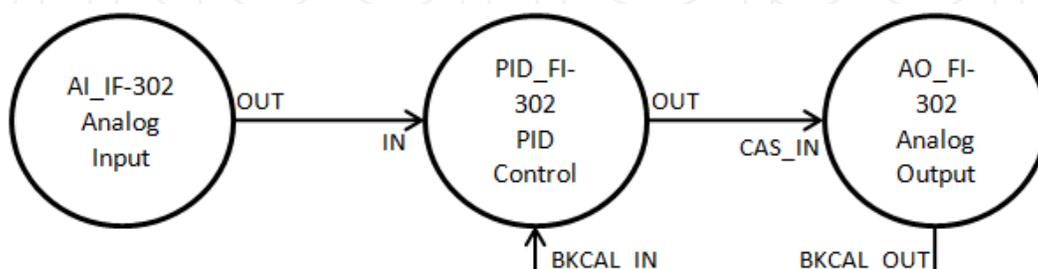


Fig. 16. FF control strategy for PID test

Since the IF302 and FI302 devices were already configured for the previous test, only the new PID control block - from FI302 - must be configure for the operation. Table 6 indicates the configuration for the PID block from FI302.

Block	Parameter		Value
PID	MODE_BLK.	TARGET	AUTO
	PV_SCALE	EU_100	100
		EU_0	0
		UNITS_INDEX	%
	OUT_SCALE	EU_100	100
		EU_0	0
		UNITS_INDEX	%
	GAIN		0.5
	RESET		1
RATE		0	

Table 6. PID Function block configuration

Parameters GAIN, RESET and RATE are Kp, Ki and Kd, respectively.

The control can finally actuate the simulated plant on LabVIEW once the configuration is ready. A first-order function is selected on the simulation environment because it is visually easier to understand the control actuation with this function. Equation (2) is used in domain “s” for the experiment.

$$H(s) = \frac{1}{4s + 1} \tag{2}$$

On the first attempt to control via PID, only the Proportional parameter was selected, because the response curve should be very close to the step response curve of the function above. The expected response is a reaction of the plant to the input that varies around the PID set point. The chart in Figure 17 shows the excitation and response curves related to the function, for a proportional gain equals to 0.5.

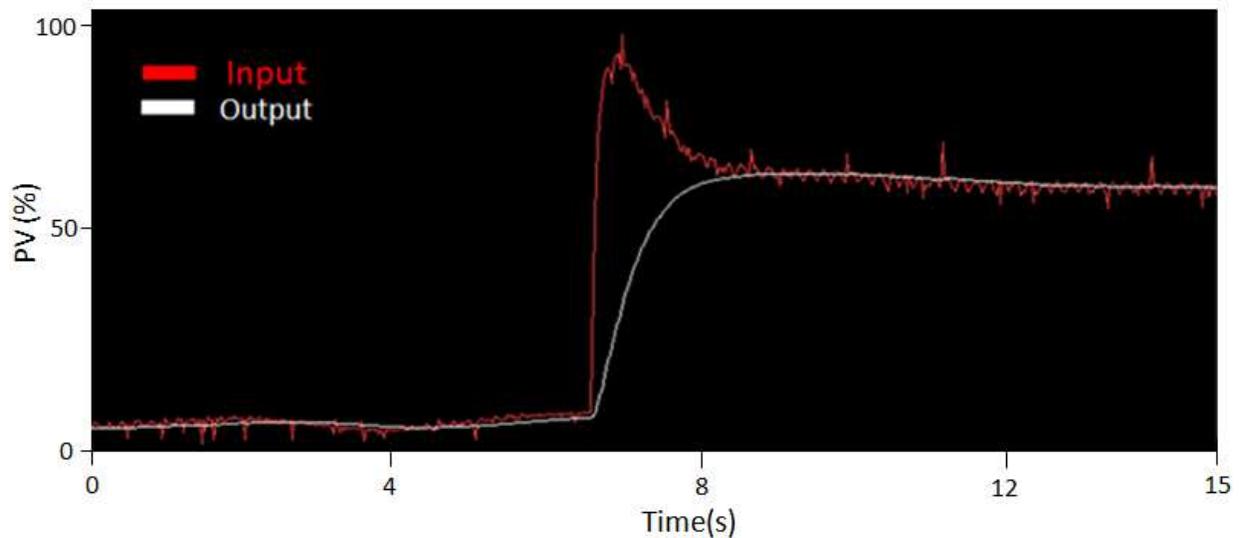


Fig. 17. Step response chart for $K_p = 0.5$

The control system actuation can be clearly noted in Figure 17, increasing the input value so the function can quickly reach the stationary value configured on the PID set point.

The Proportional control actuation was successful, and then the Integral parameter (K_i) was added with value set to 1 on the control loop of the simulated plant, keeping the proportional gain set to 0.5. Figure 18 shows the chart with the result from this experiment, a chart in response to the set point variation induced on the plant. The set point was initially at 70%, then it was altered to 10% and, right after that, it was altered to 80%. On the chart, these values are proportional to 7, 1, and 8, respectively. Observe quite an undulation due to the Integral parameter actuation mainly, which adds one magnitude to the function.

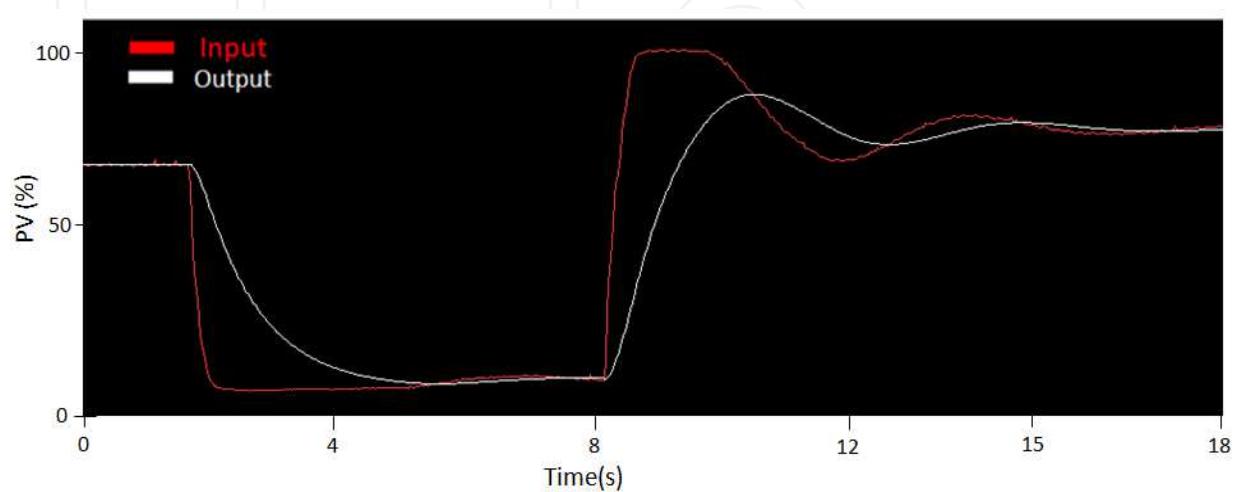


Fig. 18. Step response chart for $K_p = 0.5$ and $K_i = 1$

7. Conclusions

Considering the experiments described in the previous section, all tests for the project were performed successfully. Even with the problem related to the signal reading on the data acquisition board, the procedure for signal adequacy was successful as well, allowing the tests to continue.

An interesting fact occurred during the PID controller test: contrary to the pure I/O test, it was not necessary to correct the signal reading probably because, due to the use of a feedback control, the controller assumes the error is part of the plant's characteristics. Therefore, the signal is controlled according to the specified set point value for the PID controller.

This project intended to develop a device that could provide control and automation students the access to a variety of case studies for process automation in a learning laboratory that does not have access to a real process plant, and also provide engineers the possibility to test control strategy configurations before actuating a real plant.

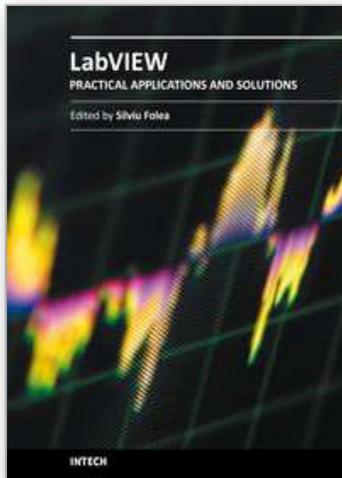
Tests demonstrated the proper operation of the controllers and the anticipated functionality. Control systems are totally independent from the simulator and can be replaced, when needed.

Fine-tuning for devices and control strategies is not on the scope of the project, since all tests had a qualitative aspect. System and equipments hereby described will be used in automation classes, and students will be able to program and tune strategies by themselves. In conclusion, the project fulfils its purpose, because even considering the signal acquisition problem on the NI-PCI 6221, the PID tests were successful.

8. References

- Cavalieri, S.; Di Stefano, A.; Mirabella, O. (1993). Optimization of Acyclic Bandwidth Allocation Exploiting the Priority Mechanism in the Fieldbus Data Link Layer. *IEEE Transactions on Industrial Electronics*, Vol. 40, No. 3, 1993, pp. 297-306.
- Chen, J.; Wang, Z.; Sun, Y. (2002). How to Improve Control System Performance Using FF Function Blocks, Proceedings of IEEE International Conference on Control Application, Glasgow, Scotland, 2002.
- Ferreiro Garcia, R.; Vidal Paz, J.; Pardo Martinez, X. C.; Coego Botana, J. (1997). Fieldbus: preliminary design approach to optimal network management, Proceedings of IEEE International Workshop on Factory Communication Systems, Barcelona, Spain, 1997.
- Fieldbus Foundation (1999a), *Foundation Specification Function Block Application Process Part 1: FF-890-1.3*. Austin, USA, 1999.
- Fieldbus Foundation (1999b). *Foundation Specification Function Block Application Process Part 3: FF-892 - FS1.4*. Austin, USA, 1999.
- Fieldbus Foundation (1999c). *Foundation Specification Function Block Application Process Part 5: FF-894 - DPS0.95*. Austin, USA, 1999.
- Fennibay, D.; Yurdakul, A.; Sen, A. (2010). Introducing Hardware-in-Loop Concept to the Hardware/Software Co-design of Real-time Embedded Systems, Proceedings of IEEE IEEE 10th International Conference on Computer and Information Technology (CIT), 2010.

- Godoy, E. P. ; Porto. A.J.P. (2008). Proposal of hardware-in-the loop in Control Systems Using CAN Networks. Proceedings of IEEE International Conference on Industry Applications, 2008.
- Hong, S. H.; Ko, S. J. (2001). A Simulation Study on the Performance Analysis of the Data Link Layer of IEC/ISA Fieldbus, *Simulation*, 2001, pp. 109-118.
- Huang, S.; Tan, K.K. (2010). Hardware-in-the-Loop Simulation for the Development of an Experimental Linear Drive. *IEEE Transactions on Industrial Electronics*, Vol. 57, No.4, 2010, pp. 1167-1174, ISSN: 0278-0046.
- International Organization For Standardization (1994). *ISO/IEC 7498-1: Information technology – open systems interconnection – basic reference model: the basic model*. Switzerland. CD-ROM.
- Li, Q.; Jiang, J. (2010). Evaluation of Foundation Fieldbus H1 Networks for Steam Generator Level Control. *IEEE Transactions on Control Systems Technology*, Vol. 99, september 2010, pp. 1-12 ISSN: 1063-6536.
- Ljung, L. (1999). *System Identification- Theory for the User*, Prentice Hall, 1999, Englewood Cliffs.
- Mossin, E ; Pantoni, R.P. ; Brandão, D. (2008). A fieldbus simulator for training purposes. *ISA Transactions*, Vol. 48, p. 132-141, 2008.
- National Instruments (2003). LabVIEW™ Basics I e II Introduction: Course Manual Course Software Version 7.0, june 2003.
- Pinotti Jr, M. ; Brandão, D. (2005) . A flexible fieldbus simulation platform for distributed control systems laboratory courses, *The International Journal Of Engineering Education*, 2005, Dublin, Vol. 21, No. 6, p. 1050-1058.
- Petalidis, N.; Gill, D.S. (1998) The formal specification of the fieldbus foundation link scheduler in E-LOTOS. Proceedings of International Conference on Formal Engineering Methods, Brisbane, Australia, 1998.
- Pop, T.; Eles, P.; Peng, Z. (2002). Holistic Scheduling and Analysis of Mixed Time/Event-Triggered Distributed Embedded Systems, Proceedings of 10th international symposium on Hardware/software codesign, Estes Park, USA, 2002.
- Smar Syscon (2011). Smar International Corporation, In: *Smar International Cooperation WebSite*, 01 abril 2011, Available from: <<http://www.smar.com/products/syscon.asp>>
- Wang, Z.; Yue, Z.; Chen, J.; Song, Y.; Sun, Y. (2002). Realtime characteristic of FF like centralized control fieldbus and it's state-of-art, Proceedings of IEEE International Symposium On Industrial Electronics, L'Aquila, Italy, 2002.



Practical Applications and Solutions Using LabVIEW™ Software

Edited by Dr. Silviu Folea

ISBN 978-953-307-650-8

Hard cover, 472 pages

Publisher InTech

Published online 01, August, 2011

Published in print edition August, 2011

The book consists of 21 chapters which present interesting applications implemented using the LabVIEW environment, belonging to several distinct fields such as engineering, fault diagnosis, medicine, remote access laboratory, internet communications, chemistry, physics, etc. The virtual instruments designed and implemented in LabVIEW provide the advantages of being more intuitive, of reducing the implementation time and of being portable. The audience for this book includes PhD students, researchers, engineers and professionals who are interested in finding out new tools developed using LabVIEW. Some chapters present interesting ideas and very detailed solutions which offer the immediate possibility of making fast innovations and of generating better products for the market. The effort made by all the scientists who contributed to editing this book was significant and as a result new and viable applications were presented.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Eduardo Moreira, Rodrigo Pantoni and Dennis Brandão (2011). Equipment Based on the Hardware in the Loop (HIL) Concept to Test Automation Equipment Using Plant Simulation, Practical Applications and Solutions Using LabVIEW™ Software, Dr. Silviu Folea (Ed.), ISBN: 978-953-307-650-8, InTech, Available from: <http://www.intechopen.com/books/practical-applications-and-solutions-using-labview-software/equipment-based-on-the-hardware-in-the-loop-hil-concept-to-test-automation-equipment-using-plant-sim>

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2011 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](#), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen