

# A Flexible Fieldbus Simulation Platform for Distributed Control Systems Laboratory Courses\*

M. PINOTTI Jnr and DENNIS BRANDÃO

*Department of Materials, Aeronautic and Automobilst Engineering, Eng. School of São Carlos, University of São Paulo, São Carlos, Brazil. E-mail: dennis@sc.usp.br*

*A PC-based fieldbus simulation platform has been designed to be used as an educational tool for distributed control systems teaching. The FBSIMU (Foundation Fieldbus Function Blocks Simulator) tool can be used to highlight via hands-on activities some fundamental concepts of discrete control and control networks. The full platform has been written using LabVIEW due to its graphical interface and rapid development time. The simulator user interface is similar to a typical industrial fieldbus system interface and its operation configuration is fully flexible, experiments can be conducted for different physical communication parameters, medium access control mechanisms and controller and acquisition algorithms. This paper describes the design and development of the FBSIMU and the incorporation of the tool in the mechatronic engineering laboratory curriculum of Eng. School of São Carlos.*

## INTRODUCTION

THE INCREASING TREND towards the use of distributed architecture for industrial control systems pushes the academic research and educational laboratories to progressively include fieldbus systems on academic settings.

In this process, some academic laboratories had their industrial pilot plants upgraded from traditional instrumentation, usually 4 to 20 mA transmitters, to smart field devices, which communicate with each other through digital fieldbus protocols [17].

Some recent academic pilot plants meanwhile were originally designed with all the instruments networked on a fieldbus control system [10]. The use of a small-scale industrial plant for research and teaching is an interesting approach with many advantages. The student's contact with updated industrial technology and equipments is highly positive [3], especially on an experimental workbench. Although, it's necessary to consider certain limitations when planning experimental classes based on pilot plants or process trainers [16]. The main concerns are about the equipment prohibitive cost, the small number of students able to access the plant at the same time, which requires large student groups and consequently a complex laboratory schedule, and finally the long setup time combined with usually slow response times makes it difficult to complete a laboratory in standard class period.

To overcome some of these limitations and to enlarge the benefits of a process control labora-

tory, recent works showed that the use of web-based remote access to physical plants is suitable for a large number of academic practices [19].

There are different approaches to configure the remote access to the fieldbus variables. It can be done, for example, by the plant supervisory system through an OPC (OLE for process control) Client interface [19] or through Internet applications (Java applets), like described by Buhler, et al. in [1].

Buhler also uses the software simulation of a real pilot plant as a first training phase, with the arguments that it is important to give the students a good idea of the plant dynamic behavior and of the action of each actuator before exposing them to costly hardware, which may cause harm to humans.

The proposal of this paper is to demonstrate that the distributed control systems teaching can take advantage of simulated networked instrumentation, and give an example of how to use this technique using a dedicate software tool. The simulation software FBSIMU (Foundation Fieldbus Function Blocks Simulator), designed on the Onboard Electronics Laboratory of Eng. School of São Carlos, has been idealized to complement the experiments executed on a pilot fieldbus plant during laboratory process control classes.

Primarily considered as a low-cost and individual educational tool, the simulation provides to students a real industrial fieldbus feeling [14], once all network induced delays, variables, configuration scenarios and operation modes are the same as in a real fieldbus. The Foundation Fieldbus (FF) protocol was adopted to this project due to its current international standard status and also because this protocol has been used by authors

\* Accepted 7 June 2005.

on pilot plants for research and teaching since its specification was released in 1998.

The use of simulated instrumentation permits the practice in a wider range of operation that would be restricted on the pilot plant. Another advantage of using simulation is that the user can choose the configuration of a simulated plant to be controlled by setting the dynamic order of the system, its dynamic coefficients, noise level, the number of inputs/outputs and the interrelations between the variables. The LabVIEW development platform was selected for implementation because the intuitive graphical programming language is engineering oriented and speeds up the development process and also permits the use of a modular and expandable software structure.

Likewise a fieldbus communication protocol, the simulator has a layer-structured core, and each communication layer can be configured independently. This feature gives the system enough flexibility to be used on experiments that evidence the influence of many digital distributed control system concepts.

### FIELDBUS FOUNDATION PROTOCOL BASICS

A fieldbus control system can be defined basically as an aggregate of networked smart transmitters (which can be either sensors or actuators), controllers, interface boards, HMI modules and configuration stations. A dedicated serial digital communication bus connects these instruments, based on a digital communication protocol suited for the target application and environment. The instrument's interoperability is guaranteed by the protocol specifications, and consequently independent instruments from different vendors should be able to work together as a distributed system. The Foundation Fieldbus specification is geared to industrial application, so its hardware characteristics and application layer objects are designed to operate on hazardous environments under hard real-time requirement.

This section describes basic aspects of the Foundation Fieldbus protocol, which are fundamental to abstracting the FBSIMU architecture.

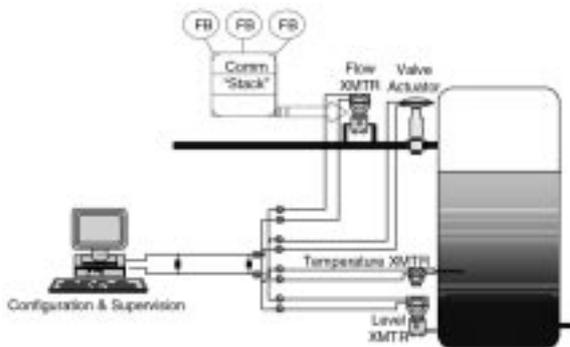


Fig. 1. Fieldbus control system.

Based on the reduced ISO/OSI reference model for open communication protocols, the Foundation Fieldbus communication stack is divided into three layers: Physical Layer, Data Link Layer and the Application Layer which is composed of a Fieldbus Access Sublayer and a Message Sublayer. On the top of the communication stack, standardized User Layer is added to the protocol to support a range of industrial automation functions, calculations and structures.

#### Physical Layer (Phy)

The Physical Layer receives messages from the communication stack and converts them into physical signals on the fieldbus transmission medium and vice-versa. The FF Physical Layer uses shielded twisted pair as communication medium. There are three specification profiles [9] for this Layer: H1, H2 and HSE. The simulation proposed in this paper is based on the H1 specification profile.

The H1 specifies a 31.25 KBit/s twisted pair bus powered channel. The network topology configuration is very flexible, attending certain physical and electrical limitations.

#### Data Link Layer (DLL)

The DLL carries the transmission control of all messages on the fieldbus; this feature grants to the FF protocol temporal determinism. The communication is based on a master/slave model with a central communication scheduler (master) in each link: 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 transmitter with Basic DLL does not have LAS capabilities; it operates passively as a communication slave. A transmitter with Link Master DLL, on the other hand, can execute the LAS functions and thus, if the active LAS node fails, becomes the LAS node.

The FF Data Link Layer supports two transmission politics: one just to scheduled cyclic data and another for sporadic (unscheduled) background data. These two communication politics share the physical bus but are 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 high priority over acyclic transmission modes.

A periodic process can be defined as a process

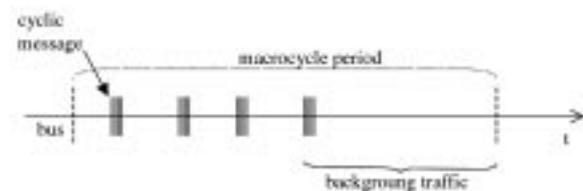


Fig. 2. FF communication cycle.

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 [2].

The sporadic or unscheduled communication is used to transmit aperiodic data generated by sporadic processes not directly related to the control loop cycles. 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.

Aperiodic (or event-triggered) processes are initiated as soon as specific events are noted [15]. The event-triggered processes are unpredictable and are usually related to alarm notifications, configuration data and user commands for example. Although the acyclic traffic is less frequent than the cyclic one, the acyclic data should be delivered as well prior to a certain time deadline, according to system requirements.

For a description of the MAC operation on both cyclic and acyclic phases refer to [11, 13, 18].

#### *Fieldbus Access Sublayer (FAS)*

The FAS accesses the scheduled and unscheduled transmissions of DLL to provide communication cooperation model services for the local Fieldbus Message Specification. The standard specifies three cooperation models to be used by the local application: Client/Server (one to one communication), Report Distribution (one to many communication) and Publisher/Subscriber (one to many communication for buffered data).

#### *Fieldbus message Specification (FMS)*

FMS services allow the local user application send messages to remote user applications across the fieldbus using a standard set of message formats. The FMS layer describes and formats the communication services and protocol behavior needed to process user application messages.

#### *User Layer*

The FF User Layer is based on distributed control or monitoring strategies of Function Blocks. The Function Blocks (FBs) are User Layer elements that encapsulate basic automation functions and consequently make the configuration of a distributed industrial application easy [4]. Distributed among the transmitters, the FBs have their inputs and outputs linked to other blocks to perform closed control loop. 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

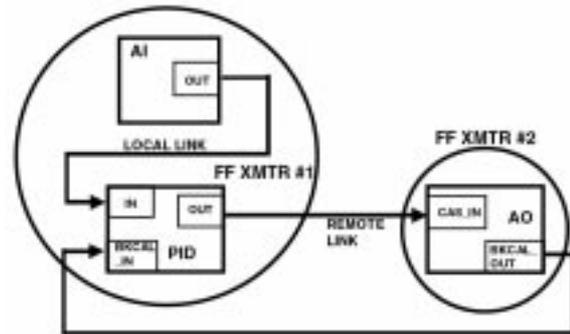


Fig. 3. PID controller control loop configured on FF function blocks strategy.

by the FBs, it is adequate to synchronize the execution of the active FB set on the system to the referred cyclic transmissions schedule table. This solution leads to the concept of joint scheduling [5].

The Foundation Fieldbus standardized a set of ten basic function blocks [8] and a complementary set of eleven advanced control blocks [7]. The standard and advanced blocks provide most mathematical and engineering tools necessary to configure all typical industrial control loop strategies. However 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.

## FBSIMU ARCHITECTURE

The basic concept behind the FBSIMU architecture is to map each instrument or transmitter, as well as the plant, as an independent LabVIEW application or Virtual Instrument (VI) and provide them a common communication channel that simulates the physical bus. An inter-application communication VI is used for this finality.

#### *Basic transmitter simulation*

The transmitter modules have a set of FBs and a communication interface composed of the Phy, DLL and FMS layer simulation modules. A unique ID number, likewise in the real transmitters identifies the unit.

'Transmitter independent modules architecture' is a good solution to simulate the behavior of a system with a variable number of transmitters on the bus. The user can set also some timing characteristics on each transmitter like the acquisition rate and messages reply delay.

A real-time finite state machine (RT KERNEL) calls synchronized real-time tasks in the core of each transmitter module, controlling this way of operation of the module, similarly to microcontrollers of real smart transmitters.

Each transmitter task set has temporal requirements to meet, which can be either fixed or

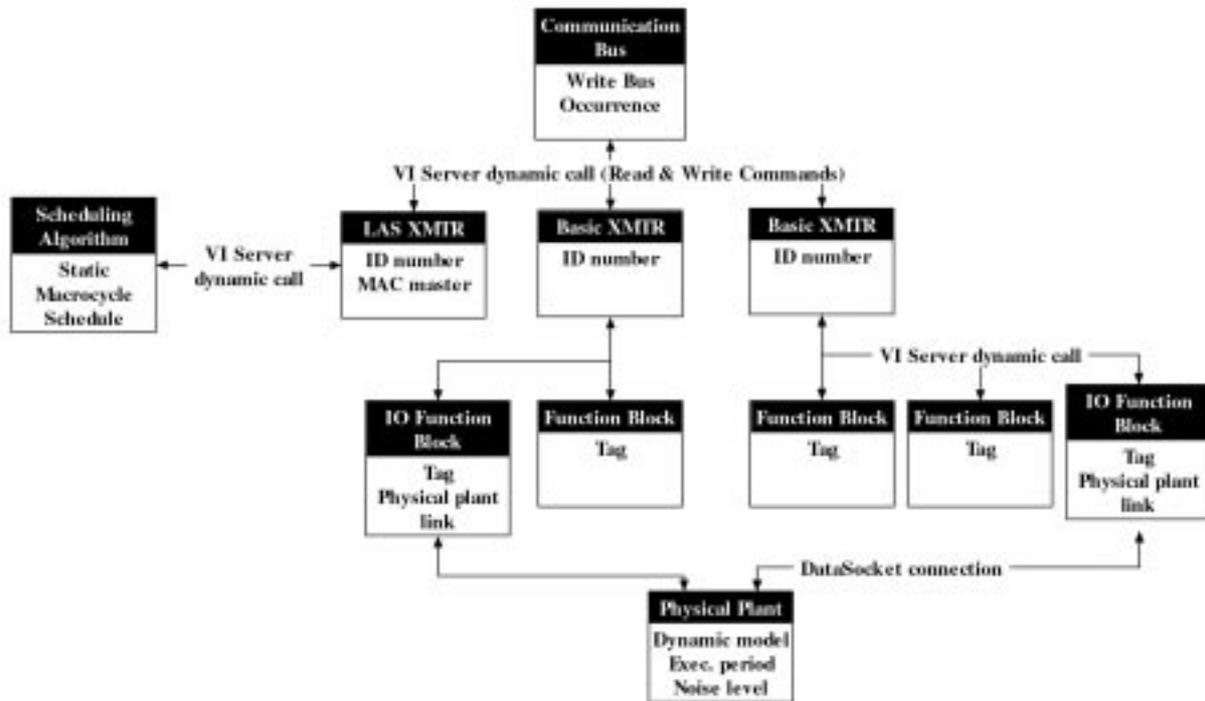


Fig. 4. FBSIMU Architecture.

dynamic. Dedicated arbitration logic is used in the RT KERNEL task to dynamically dispatch active tasks based on a fixed priority scheme. The FBSIMU transmitter modules have also connection channels used to link the transmitters to simulated communication buses and to dynamic system modules that simulate physical plants. A basic type FBSIMU transmitter processes the following states or tasks:

- INIT: initial condition, all variables are initialized and the non-volatile parameters are loaded.
- FACT INIT: factory initial condition activated by a switch, this condition causes default values override to all transmitters variables.
- RT KERNEL: real time kernel of the transmitter. All transmitters tasks are dispatched from this state based on a dynamic execution plan.
- POLL FRONT PANEL: periodic task that verifies if the user set any transmitter panel control.
- POLL TRANSDUCER: periodic task that samples the transducer value.
- FB SCHEDULE: deterministic cyclic task that starts the function blocks execution according a FB Schedule Table.
- TREAT MESSAGE: task triggered by the detection of an incoming message. This task verifies the integrity and destination address of the incoming message and a proper action is invoked.
- SEND MESSAGE: this task is executed whenever the transmitter shall transmit a message.
- IDLE: stand by state that is invoked when no task should be executed in that moment. This

state liberates the computer CPU from processing this VI until another communication is received or a task execution time is met.

*Link master transmitter simulation*

On a Link Master type FBSIMU transmitter, besides all the described tasks, it is necessary to process the LAS functions. In this case, another task set is added to the basic type real time kernel just described:

- TIME DISTRIBUTION: broadcast of the application time on the link, by sending periodical time distribution messages.
- SINC SCHEDULE: this task sends compel data messages to transmitters based on the function block schedule table.
- PROBE NODE: this task probes inactive link addresses to search for new transmitters on the link.

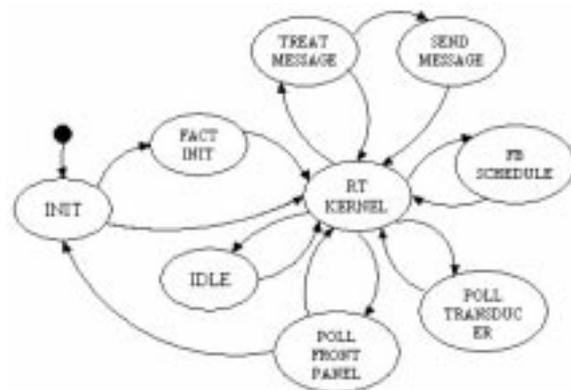


Fig. 5. Finite state machine for basic transmitter modules

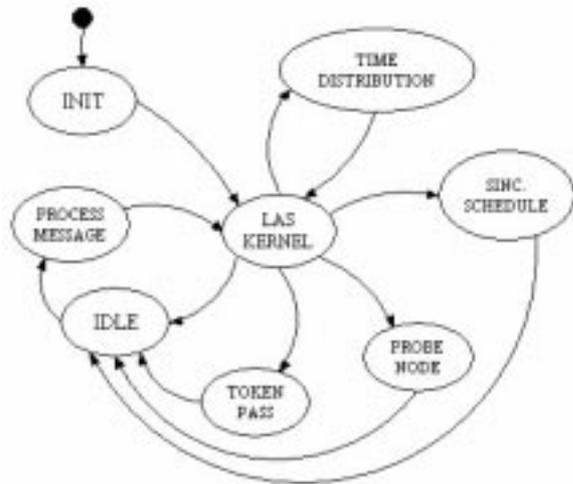


Fig. 6. Finite state machine for LAS transmitter modules.

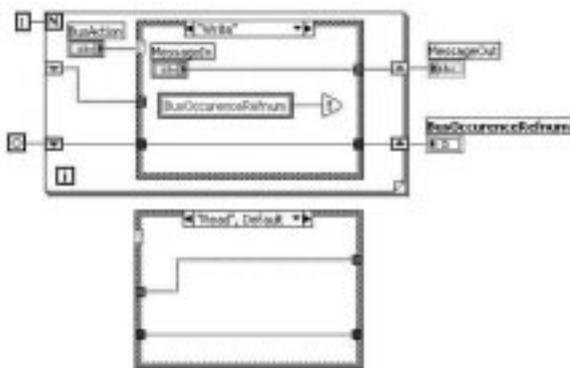


Fig. 7. Communication bus VI diagram.

- **TOKEN PASS:** this task promotes and controls the token circulation through the active transmitters on the link.

The medium access control (MAC) strategy is configurable in the communication master transmitter. Static communication and FB scheduling algorithms can be implemented in modules and dynamically accessed by the LAS node whenever required via 'Call by Reference Node' function inherent of the LabVIEW VI Server. This feature enables the comparison study of different communication scheduling policies as well as its impact on the control system stability and precision.

#### Communication bus simulation

An event-driven functional global variable VI simulates the communication bus. Each transmitter can independently 'Read' or 'Write' the simulated communication bus, and at each 'Write' action, an event (LabVIEW occurrence) named 'BusOccurrence' is triggered. All active transmitters linked to the communication bus detect this event and should process the new message as soon as possible:

#### Function block simulation

The Function Blocks modules are programmed into the FBSIMU according to the FF specifications directions and, consequently, the usage and

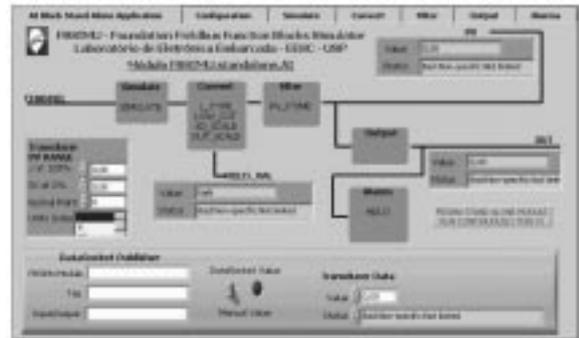


Fig. 8. Analog input standalone function block module front panel.

configuration of a simulated control loop on the FBSIMU environment is identical as it is on a real FF system.

A VI library has been developed [14] to provide a range of typical Foundation Fieldbus control and acquisition functions according to the standards and facilitate the development and integration of 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 development and integration on new projects. The seed has the whole FB module structure and directions on it to proceed with a FB project from the design to the final test procedures.

Each FB module is built onto two versions: standalone and process. The standalone FBs are executed by user commands and controlled by its front panel. The 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 interface (front panel) is intuitive and enables the user to execute the FB continually or in a step-by-step mode.

The process version, on the other hand, is controlled remotely, likewise real FBs. The user controls its operation 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 execute.
- **FB\_Reset:** this service allows default values associated with all block parameters to be written.

The process FBs don't have front panels and are loaded in the user layer of simulated transmitters modules. A FB configuration tool was also developed to configure process FBs without the presence of transmitter modules. This tool is useful for tests and verification of process FBs

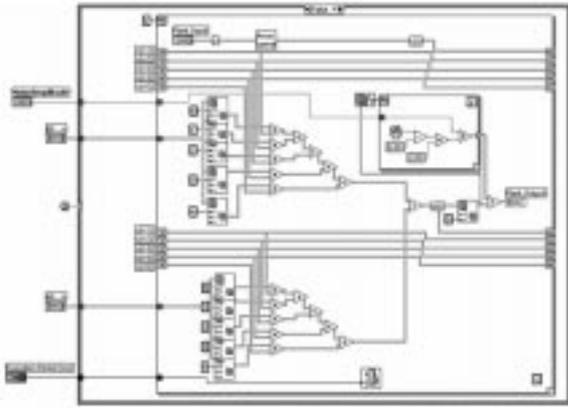


Fig. 9. MIMO plant module VI diagram.

prior the FB integration on the transmitter modules. The communication of process FBs to the FB configuration tool or to the hosting transmitter is done programmatically and dynamically by the ‘Call by Reference Node’ function.

*Physical plant simulation*

A FBSIMU plant module that cyclically executes a discrete multivariable (MIMO) linear ARX (Auto-Regressive with Exogenous Inputs) mathematical structure [12] does the simulation of the controlled plant. The adopted ARX structure is represented in Equation (1), where  $k$  is the discrete time instant,  $Y$  is the output vector,  $U$  is the input vector,  $i$  is the number of the MIMO plant inputs and outputs,  $na$  is the number of output regressors and  $nb$  the number of input regressors:

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

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 and that all regressors must be initialized prior to the simulation start. A white noise generator function adds a simulated acquisition noise to each plant output bounded by user configurable amplitude.

The proposed simulation scheme uses the LabVIEW DataSocket Server to publish and subscribe the plant output and input variables respectively, making it possible for the user to link the plant input and output variables to I/O function blocks.

**EDUCATIONAL APPLICATION**

The flexible fieldbus simulation provided by the FBSIMU platform opens a wide range of experiments where students can exploit the effect of most

communication parameters found in industrial fieldbus systems. Practical experiments consist of comparisons of given simulated fieldbus system performances over different operational conditions. The results can be analyzed via log files or graphically on charts.

On the plant simulation module, the students can introduce an independent white noise signal with configurable absolute amplitude over each plant output. The plant model is introduced on the module by the dynamic matrixes  $A$  and  $B$  as defined above. An additional parameter sets the plant module execution period in milliseconds.

The FB control strategy should be designed and distributed among the active transmitters. This operation consists in specifying in each transmitter its FB set and the correspondent FB schedule table. The parameters of all FBs should be configured to support the strategy and is linked to other FBs or to a plant simulation module input(s) and output(s).

On each transmitter module it is possible to configure the communication characteristics, by setting the parameters given in Table 1.

On the LAS transmitter module particularly, it is also necessary to specify additional DLL parameters that coordinate the MAC on the communication link, through the parameters given in Table 2.

By the described parameters, the FF operation scenario can be completely configurable and different sequences of practice exercises can be defined to embrace fundamental concepts of fieldbus control systems. This characteristic is very important because most traditional pilot plants equipped with fieldbus instrumentation offer just one scenario on which a full sequence of practice lessons should be based. Thus the use of a simulated fieldbus system enables the study of each commun-

Table 1. Configuration parameters on simulated transmitters

Parameter	Description
DelayToRT	Delay in ms used by the transmitter to process a Pass Token message
DelayToDT	Delay in ms used by the transmitter to process a Compel Data message
DeviceAddress	Configurable device address on the link
InitPeriod	Time (ms) used by the transmitter on the initialization routine
PollFPPeriod	Poll Front Panel commands period in ms
PollTransducerPeriod	Poll Transducer module period in ms
InitFactPeriod	Time (ms) used by the transmitter on the factory initialization (default) routine
FBCapabilityTable	Parameter that limits the maximum number of active FBs and maximum number of instances of a same FB on the transmitter
DisplayUpdateRate	Multiple of PollFPPeriod that commands the transmitter display refresh rate

Table 2. Configuration parameters on simulated LAS transmitters

Parameter	Description
TCL <sup>1</sup>	Token Circulation List—List of active transmitters on the bus able to receive the token
MacrocyclePeriod <sup>1</sup>	Macrocycle period in ms (cyclic plus acyclic phases)
TTRT <sup>1</sup>	Target Token Rotation Time—time in ms used as a reference for the token circulation process, used on the calculation of token priority
ScheduleTable	Relative execution time (ms) in the macrocycle for each active FB on the control strategy
MTHA <sup>1</sup>	Maximum Token Holding Time Array—array of maximum token holding time for each transmitter (ms)
DMDT <sup>1</sup>	Default Minimum Token Delegation Time—Default minimum token holding time for all basic transmitters (ms)
MID <sup>1</sup>	Minimum Inter DLPDU Delay—minimum necessary time for the LAS between two transmissions (ms)
TDP <sup>1</sup>	Time Distribution Period—period between LS-time broadcasts on the link (ms)
DTHT <sup>1</sup>	Default Token Holding Time—Default token holding time for all basic transmitters (ms)
ProbeNodePeriod	Period between consecutive Probe Node messages (ms)
FUN <sup>1</sup>	First Unpolled Node—first node address to be polled
NUN <sup>1</sup>	Number of Consecutive Unpolled Nodes—number of polled addresses starting on FUN
MRD <sup>1</sup>	Maximum Response Delay—maximum time delay (ms) permissible for basic transmitters to reply to messages
PNWaitAfter	Maximum time delay (ms) that LAS waits after a Probe Node message for a transmitter reply

Note. <sup>1</sup> DLL parameter specified on [6]

ication structure contribution and effect on the overall system performance—an impossible goal considering pilot plants with real instrumentation, due to the fact that all variables described in Tables 1 and 2 are fixed and in most cases inaccessible to end users.

### CASE STUDY

A sequence of lessons with simulated instrumentation developed for the transducers discipline from the graduation course of mechatronics engineering from the Engineering School of São Carlos which will be described. The complete module goes from an introduction to fieldbus systems to the tuning of a PID control loop on the simulated platform using the Ziegler-Nichols method, according the sequence given in Table 3:

Table 3. Transducer discipline with FBSIMU platform

Lesson	Program
1	Introduction to fieldbus systems and components
2	Configuration and start up of a control loop
3	Theoretical modeling of a physical plant and configuration of respective plant module
4	Experimental tests on the plant for parametric identification
5	Experimental plant modeling and configuration of respective plant module
6	PID controller project for theoretical and experimental models
7	PID implementation on the FBSIMU (part 1)
8	PID implementation on the FBSIMU (part 2)
9	Open loop Ziegler Nichols (ZN) test
10	PID tuning with open loop ZN gains
11	Closed loop Ziegler Nichols test
12	PID tuning with closed loop ZN gains
13	Final presentation



Fig. 10. FBSIMU configuration for open loop Ziegler-Nichols method.

The lesson 9 from Table 3 will be used as an example. The open loop Ziegler-Nichols test consists of exciting the plant with a step input. The plant reaction should be graphically measured and the proper PID gains obtained.

An analog output FB on manual mode, generating the manipulated variable, applies the step input to the simulated plant. The SISO plant module with a valid model processes the input signal and an analog input FB module is used to acquire the plant module output.

Both FB modules are scheduled to execute with a time period of 500 ms, a typical FF PID control loop macrocycle duration. The continuous transfer

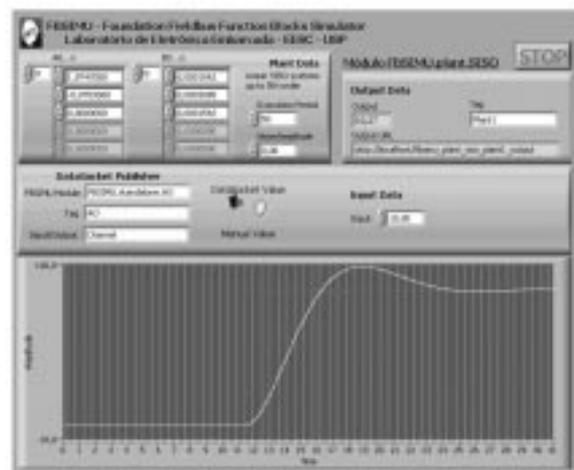


Fig. 11. Step response on the FBSIMU plant module front panel.

function  $Y(s)/U(s)$  to be controlled corresponds to a second-order linear system with  $\xi = 0,5$  (damping factor),  $\omega_n = 0,5$  (natural frequency) and  $t_s = 25s$  (settling time):

$$\frac{Y(s)}{U(s)} = \frac{0,25}{s^2 + 0,5s + 0,25} \quad (2)$$

The plant module must be configured with the ARX discrete model relative to  $Y(s)/U(s)$ . The model sampling time must be sufficiently lower than the FBs execution period. A sampling time of 50 ms will be used in this example, leading to the  $A$  and  $B$  ARX matrixes:

$$A = [1,9747 \quad -0,9753 \quad 0]$$

$$B = [0,0001542 \quad 0,0003085 \quad 0,0001542] \quad (3)$$

The DataSocket connections on each module configuration are set by the students through front panel controls; however the input variables can be loaded manually. Figure 11 shows the step response obtained for the given plant, operated remotely by the FBSIMU analog output FB module:

The step response curve is ready to be used by the students to calculate the PID controller gains proposed by the Ziegler-Nichols method.

Additionally to the curve analysis, it is possible to access the module's execution timing information registered by log files. The timing data (below) can be analyzed for each module independently or for the complete module set. The first and second columns refers to the a relative module start time and finish time (ms), the third column registers the module tag followed by the input and output values of each module when applicable:

```
25642565 25642565 Plant1 5,000000 5,138890
25642615 25642615 Plant1 5,000000 5,138932
25642665 25642665 Plant1 5,000000 5,138974
25642742 25642742 Plant1 5,000000 5,139016
25642796 25642796 Plant1 5,000000 5,139058
25642846 25642846 Plant1 5,000000 5,139100
25642896 25642896 Plant1 5,000000 5,139142
25642946 25642946 Plant1 5,000000 5,139184
25642979 25642980 AO_Tag 7,000000
25642996 25642996 Plant1 7,000000 5,139226
25643017 25643034 AI_Tag 5,139226
25643046 25643046 Plant1 7,000000 5,139576
25643096 25643096 Plant1 7,000000 5,140844
25643146 25643146 Plant1 7,000000 5,143315
25643196 25643196 Plant1 7,000000 5,146956
25643273 25643273 Plant1 7,000000 5,151738
25643349 25643349 Plant1 7,000000 5,157629
25643399 25643399 Plant1 7,000000 5,164598
25643449 25643449 Plant1 7,000000 5,172615
25643489 25643489 AO_Tag 7,000000
25643499 25643499 Plant1 7,000000 5,181648
25643534 25643538 AI_Tag 5,181648
25643549 25643549 Plant1 7,000000 5,191667
25643599 25643599 Plant1 7,000000 5,202643
25643649 25643649 Plant1 7,000000 5,214544
25643699 25643699 Plant1 7,000000 5,227340
25643749 25643749 Plant1 7,000000 5,241003
25643799 25643799 Plant1 7,000000 5,255501
```

## FINAL REMARKS

The described experience with simulated fieldbus in the laboratory of control systems confirmed the actual tendency of taking advantage of Internet and virtual instrumentation resources to complement and expand the use of traditional academic laboratory workbenches.

The flexibility of the plant simulation module enabled hands-on activities with a given control strategy to be easily set with first- second- and third-order plants.

Besides the inherent flexibility of the FBSIMU, the main advantages observed with this experience were (1) the cost of the software if compared to the cost of a pilot plant, (2) the time saving on the classes, due to the absence of equipment setup time and warm-up period of slow dynamic processes and (3) the elimination of accident risks caused by electrical and heating devices present on pilot plants.

Additional features have been integrated with the purpose to expand the FBSIMU application range. A function block development tool is currently under development. The aim of the tool is to automate the integration of standard and custom FBs to the system. Preliminary results showed that the integration of new FB to the FBSIMU is simple and consequently it can be also used on the development of industrial software as a function block test-bed.

Research is being conducted on the data link layer to make the FBSIMU an open test platform to messages and FB scheduling algorithm and MAC schemes.

The LabVIEW platform provides simple communication methods of applications hosted on different computers. Based on this capability, a distributed FBSIMU application will in future be evaluated for specific didactic use.

The current system in use on the control laboratory curriculum is based on individual experiments. The students have contact with concepts of instrumentation and control in an integrated experiment. The practice with industrial standard technology also motivates the students on the simulations.

The classes on the FBSIMU platform are conducted before and after the classes on the pilot plant. On the first classes, the students felt more confident to use a simulator instead of the real system; after an introduction to the fieldbus concepts via simulator, the experiences on pilot plants get more productive.

Advance laboratory classes about control networks and distributed digital control systems, which demands experiments beyond the pilot plant possibilities, can also be carried out on the FBSIMU to complement the traditional control classes.

## REFERENCES

1. D. Buhler, W. Kuchlin, G. Gruhler and G. Nusser, The virtual automation lab-web-based teaching of automation engineering concepts, *7th IEEE Int. Conf. and Workshop on the Engineering of Computer Based Systems*, Edinburgh, UK, 2000, pp. 156–164.
2. S. Cavalieri, A. Di Stefano and O. Mirabella, Optimization of acyclic bandwidth allocation exploiting the priority mechanism in the fieldbus data link layer, *IEEE Trans. Industrial Electronics*, **40**(3), 1993, pp. 297–306.
3. L. A. Chemane, A. F. Nunes Jnr and G. P. Hancke, Intelligent instrumentation and the fieldbus in the educational environment, *IEEE Instrumentation and Measurement Technology Conf.*, St. Paul, USA, 1998, pp. 270–275.
4. J. Chen, Z. Wang and Y. Sun, How to improve control system performance using FF function blocks, *IEEE Int. Conf. Control Application*, Glasgow, Scotland, 2002.
5. Garcia, R. Ferreira, J. Vidal Paz, X. C. Pardo Martinez and J. Coego Botana, Fieldbus: preliminary design approach to optimal network management, *IEEE Int. Workshop on Factory Communication Systems*, Barcelona, Spain, 1997.
6. Fieldbus Foundation, *Foundation Specification Data Link Protocol Specification: FF-822-1.4*, USA, (1999).
7. Fieldbus Foundation, *Foundation Specification Function Block Application Process Part 3: FF-892—FS1.4*, USA, (1999).
8. Fieldbus Foundation, *Foundation Specification Function Block Application Process Part 2: FF-891—I*, USA, (1997).
9. Fieldbus Foundation, *Foundation Specification System Architecture: FF-800-1.3*, USA, (1998).
10. Z. Hanzalek and T. Pacha, Use of the fieldbus systems in academic setting, *3rd Real-Time Systems Education*, Poznan, Poland, 1999, pp. 93–97.
11. S. H. Hong and S. J. Ko, A simulation study on the performance analysis of the data link layer of IEC/ISA fieldbus, *Simulation*, **76**(2), 2001, pp. 109–118.
12. L. Ljung, *System Identification- Theory for the User*, Prentice-Hall, Englewood Cliffs (1999).
13. N. Petalidis and D. S. Gill, The formal specification of the fieldbus foundation link scheduler in E-LOTOS, *Int. Conf. Formal Engineering Methods*, Brisbane, Australia, 1998.
14. M. P. Pinotti Jr., D. Brandão, M. J. Cunha and B. A. Gomez, Simulação operacional de bloco funcional foundation fieldbus, *3rd ISA Int. Congress of Automation, Systems and Instrumentation*, São Paulo, Brazil, 2003.
15. T. Pop, P. Eles and Z. Peng, Holistic scheduling and analysis of mixed time/event-triggered distributed embedded systems, *10th Int. Symp., Hardware/Software Codesign*, Estes Park, USA, 2002.
16. J. A. Regh, Design of a low cost trainer for flow control, *28th Annual Frontiers in Education Conf.*, Tempe, USA, 1998, Vol. 3, pp. 1063–1067.
17. J. A. Regh, W. H. Swain, B. P. Yangula and S. Wheatman, Fieldbus in the process control laboratory—it's time has come, *29th ASEE/IEEE Frontiers in Education Conf.*, San Juan, Puerto Rico, 1999, Vol. 3, pp. 13b4/12–17.
18. Z. Wang, Z. Yue, J. Chen, Y. Song and Y. Sun, Realtime characteristic of FF like centralized control fieldbus and it's state-of-art, *IEEE Int. Symp. Industrial Electronics*, L'Aquila, Italy, 2002.
19. R. P. Zeilmann, J. M. Gomes da Silva Jr., A. S. Bazanella and C. E. Pereira, Web-based control experiments of a foundation fieldbus pilot plant, *5th IFAC Int. Conf. Fieldbus and their Applications*, Aveiro, Portugal, 2003.

**Mário Pinotti Júnior** is currently a Professor at the Department of Materials, Aeronautic and Automobilst Engineering, Eng. School of São Carlos, University of São Paulo. His research interests are in the areas of instrumentation, process control, industrial automation and building automation.

**Dennis Brandão** received the Eng. degree from the Department of Mechanical Engineering, Eng. School of São Carlos, University of São Paulo in 1998. He is currently a Researcher in the Department of Materials, Aeronautic and Automobilst Engineering, Onboard Electronics Lab., University of São Paulo.