Real-Time experimentation environment for digital controllers applied to industrial processes.

R. BÁRCENA and A. ETXEBARRIA
Departamento de Electrónica y Telecomunicaciones
Universidad del País Vasco
E. U. Ingeniería Tca. Industrial. Plaza de La Casilla, 3. Apto. 48012 de Bilbao
SPAIN

Abstract: - This paper presents a Hardware-In-the-Loop experimentation environment for digital controllers applied to industrial processes using the pole location design method. This environment is formed by PC architecture equipment, multifunction I/O cards and the xPC-Target software tool of Mathworks. The main goal is to be capable of real-time experimenting, in its development stage, with controllers to be applied to industrial processes. Correct performance verification of the environment is done by the application of pole placement classical control theory to a typical plant of an industrial process, identified with the Areas method.

Key-Words: - Real-time control, Rapid prototyping, Hardware-in-the-loop simulation, digital control, autotuning controllers, pole placement.

1 Introduction.

Automation of manufacturing processes is growing up in the present days. The fast development of electronic hardware has allowed a rapid substitution of analogical controllers for digital ones. Distributed control, digital control and field buses are usual terms in the current industry. This change has been possible by introducing more sophisticated control equipments and with many possibilities of communication among them, for instance, PLC’s (Programmable Logic Controllers). However, the actual trend is to change these devices for systems based on the open PC architecture.

At first, digital control techniques were applied to industry by means of PLC’s. But these devices started to present some limitations as to memory capacity, manufacturer dependency and so on. This is why that, in spite of its initial rejection, industrial PC’s started to be used so as to carry on modern digital control theories. These systems, which present better characteristics such as more calculation capacity, less manufacturing dependency, an easy integration on communication systems and reduction of systems faults, in addition to an improvement on equipment robustness, are becoming more popular in the industrial environment [1],[2].

A similar development suffered by control hardware has happened to the control techniques. This is to say, the improvement of control hardware is involving the application of advanced control techniques (see, for example, [3]-[5]). On the other hand, controllers based on pole placement design method [6], which is not highly advanced, is in use on process control applications with, for instance, identifications methods for an automatic design of controller.

A gradual introduction of advanced control in industry has been produced because of the more strict specifications in the control of processes and development of microelectronic and hardware (DSP’s, Industrial PC’s, …). The use of advanced control methods is necessary, because recent studies show that 80% of controllers in process control applications are poorly adjusted [7]. In order to find a solution to this problem, advanced control techniques have been developed like adaptive or autotuning techniques. These techniques adjust control loops automatically by only testing plant or process without external help.

This paper describes a complete environment for a realistic experimentation of advanced controllers in order to facilitate its study and posterior implementation in productive processes. The experimentation environment is formed by an industrial PC that acts as real time controller and other PC or real actuators like process plants. Hardware-in-the-loop (HIL) experimentation provides a very useful tool for testing systems or subsystems at early stages of the development process. This can reduce the debugging process of the system, in addition to eliminate project risks, because hardware prototype is not necessary before carrying out testing and system integration. HIL experimentations work in real time and carry out input and output operations as a part of the real
system in their operative environment. This enables the system or subsystem to be tested under work nominal conditions as well as at its intended operational boundaries [8],[9].

Experimentation results described here are based on the using of the digital controller designed with the pole placement control theory. The pole placement design method simply attempts to find a controller that gives desired closed-loop poles, that is, the response of closed-loop can be determined. On the other hand, this experimentation environment is valid for any other type of advanced control.

2 System Description.

The experimentation laboratory is formed by an industrial PC, which is the controller, with a Pentium II processor to 700 Mhz and compact-PCI bus for connection with peripherals. A multifunction I/O board from National Instruments (PXI-6070E model) is allocated to acquire and send data, which enables to close the control loop on the plant. There is, as well, a process plant that can be a real plant or a model of the plant. In this case, A PC is used as emulator of the plant. Model emulator implemented is downloaded to the model PC so that emulates the plant behaviour. The PC used as model of the plant consists of a Pentium II MMX processor to 350 Mhz and a multifunction input-output board from National Instruments (PCI-6024E model) to receive control signal and send the output signal or process variable of the plant. Both, the industrial PC and the plant model PC, work as targets supervised by a desktop PC (host) where Matlab program, with toolboxes Simulink, Real Time Workshop and xPC-Target included, have been installed.

In fig.1 the experimentation environment of laboratory is depicted. In fig.2 the feedback control loop is shown, where the controller, target 1, sends the control signals or controlled variable through their I/O multifunction board. At the same time, the plant, target 2, collects the control signals and returns the output signal or process variable to the controller. The host is attended to the matter of downloading applications in each one of the targets and supervising its performance. The host also can change some parameters of experimentation environment, previously selected and prepared for this, for instance, reference signals or controller parameters.

A Matlab tool, called xPC-Target, is the software used for implementation of HIL experimentation. xPC-Target is a solution for deploying, testing and prototyping real time systems using standard PC hardware [10]. It is an environment that uses a target PC, separate from a host PC, for running real time applications. xPC Target allows for adding I/O blocks to the model, and then use the host PC with Real-Time Workshop and a C/C++ compiler to create executable code.

In the host, with the help of Simulink tool of Mathworks, the plant model to be controlled is created. This model is compiled by a C compiler to use it in the Real-Time Kernel and then is downloaded to the target PC used for emulation.

Controller is created, in the same way, by the host in Simulink, but now using S-functions in language C and specific driver blocks for multifunction boards, that is allocated on the industrial PC. In fig.3 the Simulink model implemented for initial experimentations is shown. The model is formed by
different Simulink blocks: controller block, named Pole-Placement Controller, is based on Matlab functions (S-functions) programmed in C language, I/O blocks of the National Instruments card, which are drivers to manage the card and signal display blocks, named Target Scope for target and Host Scope for host, present selected data for monitoring on graphic format.

The Simulink model used to experiment in real time is shown in fig. 3. The process of adjusting the controller, that want to be carried out, is based on three consecutive stages (see fig. 4). Process plant identification is realized to calculate an approximated model close to the real plant model. This process can be carried out in open or closed loop and as a result of that, identified model parameters are obtained. Afterwards, controller parameters are calculated from the identified model to obtain a desired dynamics. Finally, when the controller is adjusted, the feedback control is applied to the plant [11].

![Simulink model of Pole placement controller.](image)

**Figure 3.** Simulink model of Pole placement controller.

Host PC is connected to target PC, where an operative Real-Time system kernel of xPC-Target is loaded. The kernel communicates the host and the target through TCP/IP, and executes in Real-Time code generated by host. In this way, real time plant or PC model control is carried out. At the same time, it is possible to display different signals (control, output signal, intermediate variables, reference signals, etc.) from target and monitor them from host.

### 3 Experimentation Example.

The aim of the results presented here is verify the correct performance of the experimenting environment and show the complete process of create a digital controller. In this way, only one identification and the pole placement design method used in the aforementioned toolbox have been implemented. Moreover, plant HIL experimentation, is a PC model that allows verify the performance of experimental environment with more accuracy.

![Diagram](image)

**Figure 4.** Stages of the autotuning process.

The first order plus dead time (FOPDT) plant for experimentation (1) is selected for two reasons. First, the response of FOPDT describes many industry chemical processes [12], such as, feed heat exchange, chemical reactors or product separation. On the other hand, the identification method obtains a FOPDT model, that’s why, the parameters obtained can be compared with real parameters of the plant implemented.

\[
G(s) = \frac{1.2}{(10s + 1)} e^{-0.5s} 
\]  

(1)

Controller programming is carried out with two S-functions in C Language and Simulink blocks. The structure of the digital controller is implemented with direct programming (2) -see [6]-.

\[
G_b(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} 
\]  

(2)

where \(b_0\), \(b_1\) and \(b_2\) are the coefficients of polynomial, that define the location of the zeros, and the parameters \(a_1\) and \(a_2\) are the coefficients of the controller characteristic equation. The controller is:

\[
y(k) = b_0 u(k) + b_1 u(k-1) + b_2 u(k-2) - a_1 y(k-1) - a_2 y(k-2) 
\]  

(3)

This controller is programmed in a S-function according to the algorithmic description given by expression (3).

### 3.1 Identification method.

The identification method applied to recognize plant dynamics is the Areas method [13]. By means of Areas method using plant or process step response can be obtained a first order lag plus dead time model (4). Obtained model values are three: \(k\) (gain), \(T\) (time constant) and \(L\) (dead time). These values are utilized
for calculate the discrete transfer function digital of the plant.

\[ G(s) = \frac{K}{1 + sT} e^{-sL} \]  

(4)

This method is based on the calculation of upper and lower areas (see fig. 5) and, from these values, calculate model parameters according to equations (5).

\[ T = \frac{e A}{K}, \quad L = \frac{A_0 - e A}{K} \]  

(5)

Figure 5. Areas method.

The plant, which this identification method is applied, needs to fulfill that its step response is monotonically increasing or asymptotically stable. The industrial process plant selected (1) verifies this condition; therefore, this identification method is valid to the process chosen for this experimentation.

3.2 Design method.

Pole placement design method has been selected to the experimentation process [6]. This method calculates the transfer function coefficients of the digital controller to obtain the desired pole location of the system characteristic equation in closed-loop. The pole location is selected to determine the response of closed-loop system. The designer uses the continuous-time parameters of the characteristic polynomial in terms of factors of first and second order. From these parameters is a simple task to compute the discrete-time parameters.

The design procedure to obtain the pole placement desired uses the discrete-time transfer function of the plant model calculated and the characteristic equation desired. Solving the diophantine equation (6), the digital controller parameters are calculated.

\[ A(z)S(z) + B(z)R(z) = A_d(z) \]  

(6)

where B/A is the discretized plant model, R/S is the controller, T is a precompensator and A_d is the characteristic equation desired of the closed-loop system (fig. 6).

Figure 6. Pole-placement scheme.

3 Results.

Results of the experimentation example presented are divided into three main parts: the first one is a step response, the second an identification of the model and, at last, control response already adjusted.

In the identification stage, the control signal is set to a step of 5 units of amplitude. Supervisor program collects samples of the response (see fig. 7) until the signal arrives to the steady state. When all samples are stored in memory, the program calculates the identified model parameters, according to the Areas method.

Figure 7. Functional flow diagram of the autotuning supervisor program.

Parameters of the identified model of first experimentation are shown in Table I. From these
values, first order lag plus dead time model step response has been simulated through Simulink. Two responses (experimentation plant and identification model response) are shown in fig. 8. superimposed for a better comparative. Identified model response follows to the system and there is minimum errors in identifying.

![Figure 8. Open loop step response of real process and identification model comparative.](image)

After identifying, plant is discretized to including it in the diophantine equation function. Solving the diophantine equation, the parameters of digital controller are obtained. In table II, calculated controller parameters by means of pole placement design method are shown. The characteristic equation desired for the closed loop is shown in expression (7).

\[ A_{cl} = z^2 - 0.6z + 0.05 \]  

\[ (7) \]

<table>
<thead>
<tr>
<th>TABLE I</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gain</td>
</tr>
<tr>
<td>1,206055</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TABLE II</th>
</tr>
</thead>
<tbody>
<tr>
<td>R/S</td>
</tr>
<tr>
<td>2.905631z</td>
</tr>
<tr>
<td>z + 0.150552</td>
</tr>
</tbody>
</table>

The pole placement desired is selected to decrease rise time and no overshoot.

![Figure 9. Closed loop step response of controlled process.](image)

Identification and calculation of digital controller stages are finished when calculated parameters have been loaded into the controller. Then supervision program block finishes its function and lets the digital controller block start to work. In this moment, plant is controlled in closed loop by digital controller. With the aim of verifying a correct performance, a step of 5 units on signal reference is introduced. In figure 9, controlled system response is shown before a change in the input signal. Control signal of the controlled plant before a step on input reference is shown in figure 10.

![Figure 10. Signal control of controlled process.](image)

4 Conclusions.

A HIL experimentation environment for advanced controllers has been developed to experiment, in a realistic way, control loops in real time. The main objective of the environment is carrying out HIL experimentations in real-time to check a correct performance of controllers to be applied to productive processes, without using an expensive plant prototype.
In order to check the performance of this experimentation environment, digital controllers has been applied to a typical plant of the industrial environment. The methods used for the automatic adjustment of the controller are specifically the Areas identification method and the Pole Placement design method. After observing responses obtained in the experimentation, a correct performance of such a environment has been verified.

This system is a complete laboratory that allows experimentations with all types of digital controllers. By means of this environment, the implementation of digital controllers is very easy by using Simulink blocks. Moreover, other types of advanced controllers that need other requirements not supplied by Simulink blocks can be developed. The designer may program blocks in a high level language like C language.

The experimentations can also be applied to real plants, with actuators of the real process, as well as to emulated plants, by means of the connection to one or several PC’s that act like emulators. The system allows us to minimize the development time and the costs of creating prototypes. In the development stages, the experimentation can be carried out with real process, when possible, or with models, when is very expensive to get a real process into the lab. Nowadays, this HIL environment is in use in order to experiment with real positioning systems of industrial processes.

ACKNOWLEDGMENT
The authors would thank the University of The Basque Country (UPV/EHU) for partially supporting this work through (Projects UE03/A06 and I/UPV0147-363-E-15384).

References: