2011 9th IEEE International Conference on
Control and Automation (ICCA)
Santiago, Chile, December 19-21, 2011

Integrated Online Auto-tuning and Digital Implementation of PID Controllers in Industrial Processes

Manuel Manyari-Rivera and João Carlos Basilio

Abstract— **The continued progress and growth of industry has increased the need for communication and control integration. In this regard, it would be interesting to apply simple algorithms that combine model parameter identification and tuning of PID controllers. In this work we consider the digital implementation of a PID self-tuning system formed with an industrial PLC and a PC. The PLC is used acquire field data whereas the PC is used to run a Matlab program to perform parameter identification, PID parameter tuning and control. Communication between the PLC and the PC is carried out by means of an OPC (Ole for Process Control) server. The proposed scheme has been applied to the design of a PID controller for a pressure control station of a natural gas pipeline.**

I. INTRODUCTION

In the history of control theory, many design strategies have been developed to comply with regulation and transient specifications. These include linear model-based, predictive and non-linear controllers. Among these techniques, only a few have been implemented in industry, in part because of the difficult to design the required controllers and in part due to the high computational cost the control algorithms require for their implementation.

Current industry has a large number of programmable logic controllers (PLCs) of different manufacturers installed, which implement the process control routines. The most commonly used controller to comply with IEC 61131-3 as far as regulation and control are concerned is the Proportional-Integral-Derivative (PID) controller [1]. More complex instructions are poorly implemented in PLCs of middle and even upper ranges, being therefore less advantageous than distributed control systems (DCSs) in the implementation of routines for tuning and auto tuning PIDs, predictive control, multivariable control, neural-based control, fuzzy logic control and others.

In spite of its limitation, it would be interesting to apply control algorithms in industrial applications taking advantage of PLC hardware installed, together with an application located on a programming station or station operator support. Such an application should be able to read/write data to PLC, have high capacity data processing and memory. It is therefore necessary to find technologies that provide high level communication between the PLCs and the programming station. A widely used data exchange tool for Windows environment is the OPC (Ole for Process Control) communication, which provides connectivity in both industrial automation and enterprise systems. OPC has been adopted by various manufacturers to exchange data, control variables and states of industrial processes. Through OPC, it is possible to Exchange data between different platforms for various manufacturers using a client - server philosophy. In summary, OPC allows open database interconnectivity between industrial controllers (e.g. PLCs, DCSs) and applications within Windows environment.

The objective of this paper is to propose an integrated platform formed with a PLC and a PC which is able to perform parameter model identification, auto-tune the parameter of a PID controller, set the initial conditions for real implementation, which is necessary in order to avoid undesirable transitions in closed-lopop, and execute control action. Parameter identification is carried out using recursive least-squares (RLS) and PID parameters are set for an overdamped second order plant model. Although the implementation scheme described has been developed for a specific controller, more complex design techniques, such as predictive, robust or adaptive control, can be deployed in the design stage of the controller block.

In the context of developing control strategies, several techniques for the tuning and design of PI and PID controllers are presented in [2], offering several auto-tuning techniques. The most commonly used PID parameter tuning technique in Chemical Industry is the traditional Ziegler & Nichols method [3]. More recently, Basilio & Matos [4] has proposed an easy way to tune the parameters of PI and PID controllers for plant with monotonic step response. The so-called internal mode principle has been used in [5] and [6] to obtain PID and PI controllers; the latter has been obtained by using H_∞ control theory.

This paper is structured as follows: Section II presents a brief review of the PID parameter tuning method used here; Section III describes the digital implementation of the PID self-tuning system proposed here using an industrial PLC and a PC; Section IV presents the application of the technology proposed here in the design of a PID controller for a pressure control station that is part of Peru Natural Gas Pipeline; conclusions are drawn in Section V.

The research of M. Manyari-Rivera has been partially supported by the Pipeline Department of Peru LNG Operator Company - COLP SAC. The research of J. C. Basilio has been partially supported by the Brazilian Research Council (CNPq), under grant number 306592/2010-0.

J. C. Basilio is with COPPE - Programa de Engenharia Elétrica, Universidade Federal do Rio de Janeiro, 21949-900, Rio de Janeiro, RJ, Brazil. basilio@dee.ufrj.br

M. Manyari-Rivera is with Pipeline Department of Peru LNG Operator Company - COLP SAC, Lima, Peru. mmanyari@colp.com.pe

II. DESIGN OF PID CONTROLLERS

Consider the block diagram of a linear single-input single-output feedback control system shown in Figure 1, where $C(s)$ and $G(s)$ denote, respectively, the transfer functions of the controller and of the plant to be controlled, $R(s)$, $U(s)$, and $Y(s)$ denote the Laplace transforms of the reference (set-point), control and plant output signals, respectively, and $D_i(s)$ and $D_o(s)$ denote the Laplace transforms of input and output disturbance signals, respectively. Let $e(t)$ denote the error signal between the reference input $r(t)$ and the process variable or actual output $y(t)$.

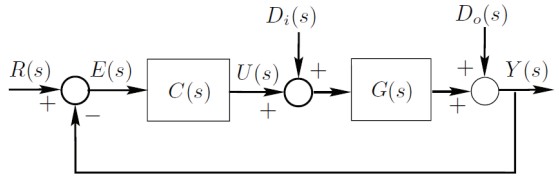

Fig. 1. Block diagram of a simple feedback control system.

A proportional-integral-derivative (PID) controller provides control signals that are proportional to the error, to the integral of the error and to the derivate of the error, namely,

$$u(t) = K_p \left[e(t) + \frac{1}{T_i} \int_0^t e(\lambda) d\lambda + T_d \frac{d}{dt} e(t) \right] \quad (1)$$

Computing the Laplace transform of each side of Equation (1), we obtain:

$$C(s) = \frac{U(s)}{E(s)} = K_p \left[1 + \frac{1}{T_i s} + T_d s \right]. \quad (2)$$

Assume now that the plant is modeled as a second ordem system (see also [7]), *i.e.*

$$G(s) = \frac{K}{(s+a)(s+b)}, |b| > |a|. \quad (3)$$

A. Design of PI controllers for second order systems

In order to design a PI controller for the second-order system described by Equation (3), we first set $T_d = 0$ in Equation (2), and, after simple algebraic manipulation, this equation can be written as:

$$C(s) = \frac{K_p(s + 1/T_i)}{s}. \quad (4)$$

Since $|b| \geq |a|$, a natural choice for T_i is [5]:

$$T_i = \frac{1}{a}. \quad (5)$$

Therefore, the closed-loop transfer function for this choice of T_i is given by:

$$T(s) = \frac{Y(s)}{R(s)} = \frac{KK_p}{s^2 + bs + KK_p}. \quad (6)$$

Comparing the denominator polynomial of $T(s)$ with the usual denominator of a second-order system $p_c(s) = s^2 + 2\zeta\omega_n s + \omega_n^2$, we see that it is possible to tune K_p so as to obtain an underdamped closed-loop response to a reference step signal. For some $zeta < 1$, it is not difficult to check that

$$K_p = \frac{b^2}{4K\zeta^2}. \quad (7)$$

B. Design of PID controllers for second order systems

The simplest way to tune de parameters K_p, T_i and T_d of the PID controller given in Equation (2 is to make cancel the poles of the plant with the zeros of the controller and, in the sequel, to choose the gain K_p with the vies to satisfying some performance specifications [5]. Proceeding in this way, we first obtain T_i and T_d, which are given as:

$$T_i = \frac{a+b}{ab}, \; T_d = \frac{1}{a+b}. \quad (8)$$

Therefore, the resulting open-loop system will have the following transfer function:

$$T(s) = \frac{Y(s)}{R(s)} = \frac{KK_p/T_d}{s + KK_p/T_d}. \quad (9)$$

Notice that $T(s)$ is a first-order transfer function whose DC gain depends on K_p and T_d. In order for the closed-loop step response to have an appropriate settling time, Rivera et al. [5] proposes the following setting for the proportional gain K_p:

$$K_p = \frac{(a+b)\sqrt{ab}}{k}. \quad (10)$$

C. Transfer function identification

The plant transfer function parameters will be determined through Weighted Recursive Least Square (WRLS) [2]. The objective is to find the parameters of a discrete-time transfer function

$$H(z) = \frac{Y(s)}{U(z)} \frac{b_n z^n + b_{n-1} z^{n-1} + \ldots + b_1 z + b_0}{z^{n+1} + a_n z^n + \ldots, + a_1 z + a_0}, \quad (11)$$

where $a_i, b_j \in \mathbb{R}$, $i, j = 0, \ldots, n$. Let T denote the sample time and assume that $u(k)$ and $y(k)$ the input and output sequences recorded at sample instantes kT, $k = 1, 2, \ldots, N$. Define the parameter vector θ as

$$\theta = \left[\begin{array}{cccccccc} a_m & a_{m-1} & \cdots & a_0 & b_m & b_{m-1} & \cdots & b_0 \end{array} \right],$$

the output-input vector $\phi(k+1)$ as

$$\phi(k+1) = \left[\begin{array}{cccccc} -y(k) & \cdots & -y(k-n+1) & u(k) & \cdots & u(k-n+1) \end{array} \right].$$

Assume that there are N observations, and define the output vector $\mathcal{Y}(N)$ as

$$\mathcal{Y}(N) = \left[\begin{array}{ccccc} y(1) & y(2) & \cdots & y(N) \end{array} \right]^T$$

and

$$\Phi(N) = \left[\begin{array}{c} \phi^T(1) \\ \phi^T(2) \\ \vdots \\ \phi^T(N) \end{array} \right].$$

Then the solution of the WRLS problem is obtained through the following recursive algorithm [2].

Algorithm 1:
STEP 1. Set $N = N_0$, such that and $\Phi^T(N_0)\Phi(N_0)$ is invertible and compute:

$$P(N) = [\Phi^T(N)\Phi(N)]^{-1},$$
$$\theta(N) = P(N)\phi^T(N)y(N),$$

and choose β and λ such that $\beta + \lambda = 1$ ($\beta = 0.1$ and $\lambda = 0.9$ in our case). STEP 2. After each new observation compute

$$L(N) = \frac{P(N)}{\lambda}\phi(N+1)\left[\frac{1}{\beta} + \frac{\phi^T(N+1)P(N)\phi(N+1)}{\lambda}\right],$$

$$P(N+1) = \frac{P(N)}{\lambda} - L(N)\phi^T(N+1)\frac{P(N)}{\lambda},$$

$$\theta(N+1) = \theta(N) + L(N)\left[y(N+1) - \phi^T(N+1)\theta(N)\right].$$

STEP 3. Set $N \leftarrow N+1$ and go back to step 2 □

This algorithm describes how to find the parameters for the discrete transfer function given in Equation (11).

However, the parameters K_p, T_i and T_d are given in terms of the parameters of the continuous-time transfer function given in Equation (3). The parameters of the continuous transfer function can be obtained replacing $s = e^{sT}$ or by using Tustin approximation (see [2], [8]). In our case, since Matlab is used to implement the self-tuning algorithm, the Matlab function `d2c` will be deployed.

III. IMPLEMENTATION OF PID TUNING BLOCKS

The continued progress and growth of the industry has increased the need for communication and control system integration. Many vendors have launched actuator elements and sensors and that interact with different communication platforms. Control technologies, on the other hand, have found problems when integrating various elements of different vendors into a unique platform. An alternative is to provide drivers for different applications and its convergence towards a main platform in order to integrate all of them within a transparent communication. A flexible digital alternative is OPC (Ole for Process Control). This application software developed for Microsoft Windows environment acts as a data communication bus in a client — server architecture. Figure 2 shows how the connection between the OPC client, the application that accesses the public data for processing and monitorin, and the OPC severs, the application that exports and publics variables of the physical elements into a data exchange platform, that may exist in the system. OPC is an open connectivity technique that supports the industry provides open standards in industrial automation systems and companies. Interoperability is ensured through the creation and maintenance of open standard specifications. OPC currently has seven standards specifications completed and is in constant improvement.

In this work we consider the digital implementation of a PID tuning scheme using PLCs. The choice of PLC devices has been motivated by the fact that these devices besides being widely used in industry, have analog interfaces through

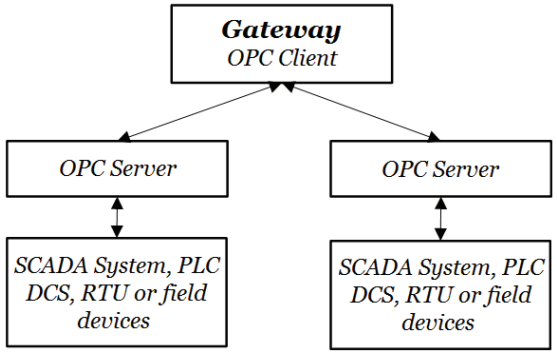

Fig. 2. Exchange scheme between several field devices and PLC via OPC.

analog inputs and outputs. They are, therefore, suitable for our application, dispensing the use of data acquisition cards or other similar extra boards. It is worth remarking that for industrial application, the implementation of the present solution is very unlike to require additional hardware. We use in this paper, Allen Bradley PLCs and its software as communication tool for the tuning procedure. Notice that the communication philosophy of Allen Bradley PLCs uses RsLinx as a platform for high-level communication, which enables transparent connectivity between external applications and Allen Bradley variables. RsLinx has an OPC server which exports items found in online controllers via OPC topics. For this reason, RSLinx is the software that allows the reading and writing of tags or signals from/to a compatible PLC.

MATLAB is a simulation and calculation software with applications for control theory. It has an specific application called Simulink. Through the Simulink OPC Toolbox, it is possible to configure an OPC client, giving the ability to process in a digital way the data available in a PLC. Figure 3 shows an example of communication between an SLC 5/04 Allen Bradley PC with Matlab via OPC Toolbox. In Figure 3 we can see the exported item ("N7: 0") through OPC, located in RsLinx for MATLAB application, using the corresponding OPC Toolbox. Notice that it is possible to read and write on the button of item "N7: 0" on topic "EJEMPLO" from the OPC client application. In the same way we can establish some communication between other families of PLC manufactured by Allen Bradley, like CompactLogix and ControlLogix via RSLinx Remote OPC Server. We can use the corresponding OPC server for other PLC brands as well.

The implementation of the control/identification scheme is carried out with using three blocks, as depicted in Figure 4: *(i)* the block labeled as $G'(z)$ that acquires the process input (manipulated variable) and output signals (process variable); *(ii)* the "Parameters identification" block that performs WRLS Algorithm 1; *(iii)* the "PID parameters calculation" block that computes the PID parameters according to Equations (8) and (10) and; *(iv)* the block labeled as $C(s)$ that sends to the PLC the controller parameters tuned in block "PID parameters calculation" block. To ensure the

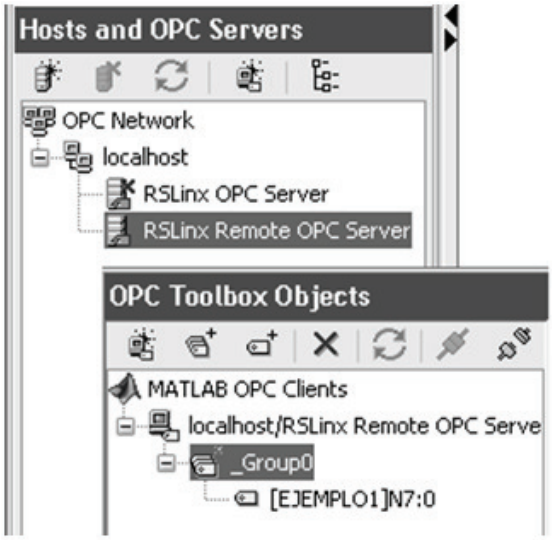

Fig. 3. OPC communication between Matlab and RsLinx.

stability of the digital control system in closed loop, the relationship between the sample time T and the shortest plant time constant τ must be $5T \leq \tau$ [9]. The plant input (actuator command or manipulated variable) and the data associated with the plant response (process variable) will be given in the Simulink diagram by the `Read` and `Write` blocks, respectively. Both variables are sampled and forwarded to the parameter identification block. After the discrete transfer function parameters have been identified with the WRLS algorithm, the next step is to find the equivalent continuous time transfer function, using the `c2d` Matlab function using the same sample time as that deployed during the identification procedure. The final stage is the tune of the PID parameters. A summary of the main steps is given in the following procedure.

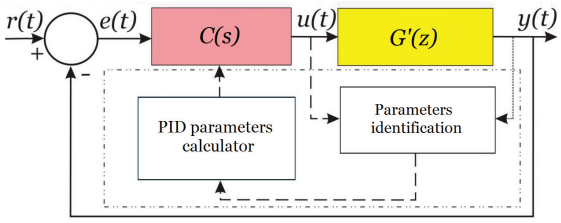

Fig. 4. Block diagram of closed-loop system including the tuning/identification blocks used in the implementation of the digital PID proposed here.

Procedure 1:

STEP 1. Using compatible analog input/output cards (4-20mA or 0-10V), acquire signal from field sensors and send commands to actuator.

STEP 2. Link the sensor/actuator signals in the PLC or stand alone controller with the Matlab in a personal computer or existent engineering station, using OPC communication servers. For that, insert the OPC configuration block in Matlab and OPC read/write blocks for the manipulated and process variable respectively. Note that this link, may be implemented locally or remotely through an Ethernet VLAN.

STEP 3. Using the signal acquired by the PLCs, perform Algorithm 1 to compute in real-time the parameters of the digital transfer function of the industrial plant or process.

STEP 4. Based on the digital transfer function computed in the previous step, find, in real-time, the equivalent continuous time transfer functions of the system.

STEP 5. Find the PID parameters for the nominal plant using Equations (8) and (10).

STEP 6. Send the parameters calculated in the previous step to the PID instruction in the PLC or stand alone controller using Write OPC blocks.

STEP 7. Evaluate the performance of the PID controller in closed-loop. □

Additional items, like delays caused by communication issues may be considered in the implementation. Typically satellite communication insert delays of a few seconds that may affect the algorithm when it is executed remotely through a VLAN.

IV. CASE STUDY

In order to illustrate the tuning procedure proposed in the paper, the design of a PID controller for a pressure control station that is part of Peru LNG Natural Gas Pipeline will now be considered. Peru LNG pipeline has a length of 408 kilometers and crosses 100 kilometers of Peru desert coast and 308 km of high mountains in the Andes Mountains, where it reaches its highest point at 4.817 meters over sea level. Peru LNG pipeline is a high pressure 34" Class 900 pipeline that consists of a inlet filtering/meter station, 14 block valve stations, scraper/launcher station and the plant receiver facility spaced at an average of 30 km. Since Peru LNG pipeline crosses the Andes Mountains, the pipeline altitudes range from 100m up to 4817m. As a consequence, Peru LNG pipeline has a variable pressure profile, which is a function of the gas density dependence on altitude — the natural gas is less dense as altitude increases because there is lower concentration of gas molecules as a result of gravity.

Peru LNG pipeline has a pressure control station (PCS) to regulate the pressure profile through a pressure regulator skid. A mounted skid has been designed to prevent the pipeline pressure in lower pipeline sections downstream the mountains from exceeding the maximum allowable operation pressure (MAOP) during line-pack conditions. The PCS includes two parallel control runs. Each control run comprises two 10" normally open control valves — pressure regulator valves (PRV) in series and normally open gas operated isolation raised face flanged valves at inlet and outlet control runs. Both pressure control runs are anticipated to operate simultaneously. The upstream control valve in each runs function as spare. If the downstream control valve fails open, the upstream control valve shall resume the control function. In normal operation, the pipeline gas flow is through a normally-open-gas operated 30" NPS pressure control system that bypasses the 30" valve located within the

PCS. The 30” valve closes when the downstream pipeline pressure reaches the primary set pressure, which is set to prevent the pipeline pressure in lower pipeline segments from exceeding the MAOP during either low or no flow conditions. After the bypass, the 30” valve closes; the pressure regulating valves begin regulating the downstream pressure to maintain the pressure at or below the primary set pressure.

A diagram of the pressure control station is shown in Figure 5. The PRVs are commanded by an analog output (4-20mA) from a local Remote Operation Controller ROC-809 (manufactured by Emerson), and the process variable (pressure downstream) is acquired by the same ROC-809 via a 4-20mA analog signal. Two PID instructions, executed by the remote operation controllers, control the pressure downstream given by the pressure transmitter PT-107. PID # 1 commands valves PRV 102 and PRV 104 simultaneously whereas PID # 2 command valves PRV 101 and PRV 103.

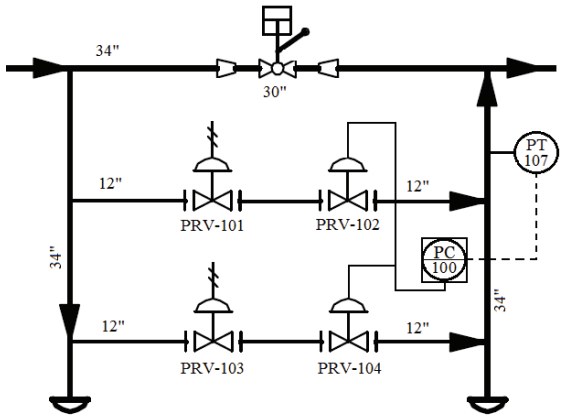

Fig. 5. Diagram of the pressure control station. This corresponding station is located at 4847 m above see level.

Input and output data were acquired by means of an Ethernet network and sent from ROC-809 to an existent ControlLogix PLC via Modbus RTU communication. An engineering station, located in the Central Control Room, 164km away from the PLNG Pressure Control Station, monitors remotely all pipeline PLCs.

For purposes of tuning the controller, we applied Algorithm 1 between the manipulated variables PRVs 102 and 104 — command (control signals for percentage opening of the valves — and the output variables — system response pressure (PT-107). The acquired signals are shown in Figure 6. It can be noticed the existence of a time delay near the instant when the step random variation has been applied, which suggests that the system can be approximated using a second-order model. The resulting second-ordem transfer function is given by:

$$G_a(s) = \frac{0.000837}{(s+0.02)(s+0.0284)}.$$

We will now design a PI controller for the system. The choice of a PI is due to the fact that the plant to be controlled consists of a pressure loop in which the pressure variation

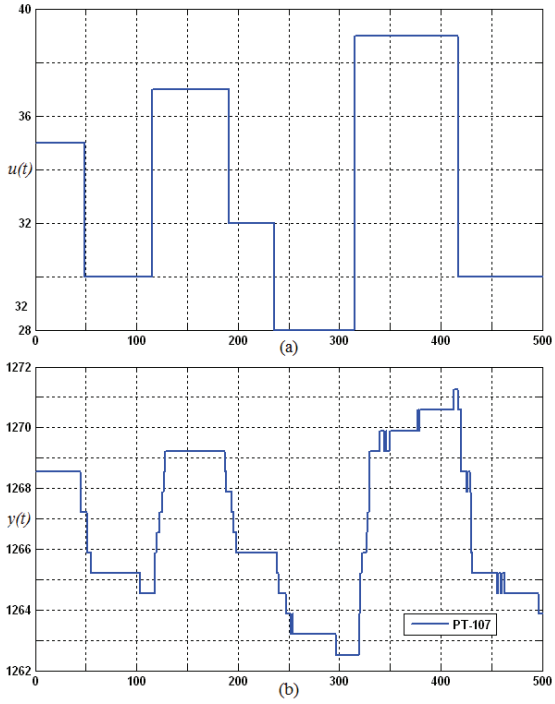

Fig. 6. (a) Random variations in the percentage opening valve; (b) temporal response of the system in open-loop.

and the instantaneous pressure itself are functions of the gas flow through the 10” valves, being therefore a fast and noisy loop. In addition, abrupt pressure variations can activate the Automatic Line Break (ALB), a mechanical device that locks the 30” valve in its close position, disabling the 30” valve to be opened from control room. For this reason, it is not advisable that the controller have derivative action. If we specify 10% as the maximum allowed overshoot, then $\zeta = 0.6$ must be used in Equation (7), and using $G_a(s)$ obtained above, the PI controller will have the following transfer function:

$$C(s) = 0.65\left(1 + \frac{1}{49.5s}\right).$$

Figures 7 and 8 show the pressure response signal when the PI controller calculated in this section has been used in the real system for instantaneous gas flows of 420MMSCF and 250MMSCF, respectively. Note that the controlled system has succeeded in regulating the pressure in the PT-107 acceptably even when changes in the set-point occurs. Notice, in addition, that the pressure control loop reaches the desired value with overshoots that are approximately equal to 10%. It is important to remark that the tuning has been calculated for pressures ranging between 1180 to 1280 psi, so it is important to maintain the integrity of the minimizing regulatory effort of the valves.

Figure 9 shows the valve percentage opening corresponding to the closed-loop response of Figure 8. Notice that there is no sudden swings in control valves and also no saturation,

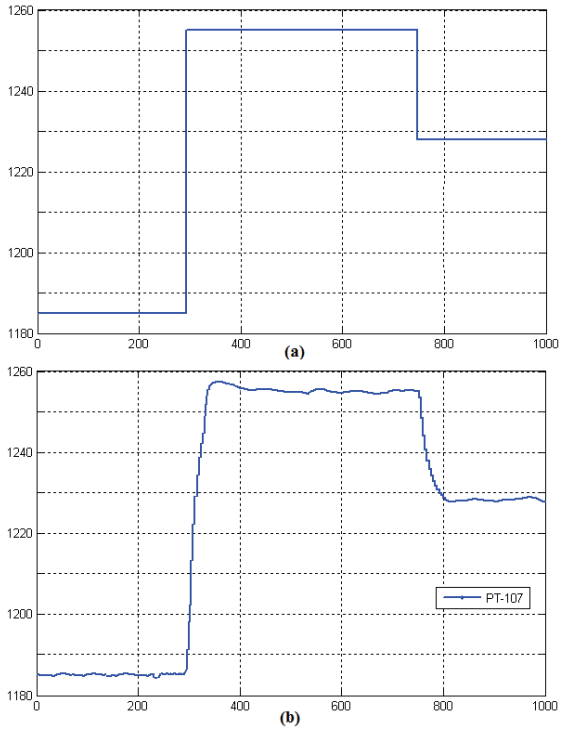

Fig. 7. Response for the PI control. Time (secs.) vs. pressure(psi): (a) Step variations in setpoint signal (b) the temporal response with K_p = 0.65 and T_i = 49.95. Instantaneous gas flow: 420MMSCF.

which shows the accuracy of the digital implementation scheme proposed here and of the PI tuning strategy adopted.

V. CONCLUSION

This paper has proposed an integrated platform formed with a PLC and a PC which is able to perform parameter model identification, auto-tune the parameters of a PID controller, set the initial conditions for real implementation, and execute control action. The tuning procedure has been applied in the design of a PI controller for a pressure control station belonging to Peru Natural Gas Pipeline. The practical results have confirm the efficiency of the implementation procedure proposed in the paper.

Although the implementation scheme described here has used a PID controller tuned for an overdamped second order model, more complex design techniques, such as, predictive, robust or adaptive control, can be applied in the design stage of the controller block.

REFERENCES

[1] S. Bennett, "The past of PID controllers," in *Proceedings of the IFAC Workshop on Digital Control: Past, Present and Future of PID Control*, 2000, pp. 1–11.

[2] K. J. Astrom and B. Wittenmark, *Computer-controlled Systems: Theory and Design*, Prentice-Hall, Englewood Cliffs, USA, 2nd edition, 1990.

[3] J. G. Ziegler and N. B. Nichols, "Optimal settings for automatic controllers," *Transactions of the ASME*, vol. 64, pp. 759–768, 1942.

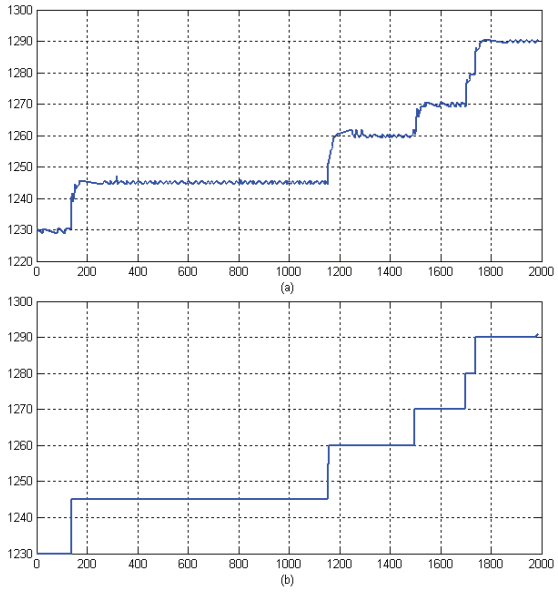

Fig. 8. Response for the PI control. Time (secs.) vs. pressure (psi): (a) the temporal response with K_p = 0.65 and T_i = 49.95, (b) Step variations in the set-point signal. Instantaneous gas flow: 250MMSCF.

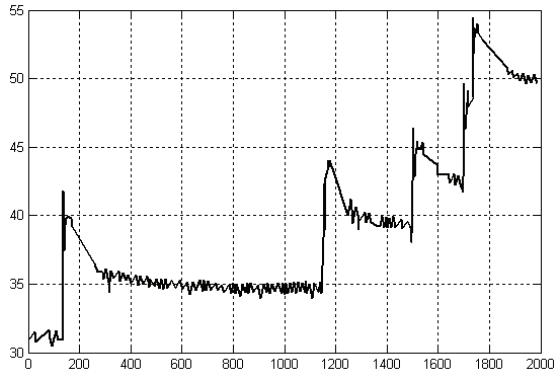

Fig. 9. Response for the PI control. Time (secs.) vs. valve opening (%).

[4] J. C. Basilio and S. R. Matos, "Design of PI and PID controllers with transient performance specification," *IEEE Transactions on Education*, vol. 45, no. 4, pp. 364–370, 2002.

[5] D. E. Rivera, M. Morari, and S. Skogestad, "Internal model control. PID controller design," *Ind. Eng. Chem. Process Design Development*, vol. 25, no. 4, pp. 252–265, 1986.

[6] J. C. Basilio, J. A. Silva Jr., L. G. B. Rolim, and M. V. Moreira, "H_∞ design of rotor flux-oriented current-controlled induction motor drives: speed control, noise attenuation and stability robustness," *IET Control Theory and Applications*, vol. 4, pp. 2491–2505, 2010.

[7] M. Manyari-Rivera, "Tuning of controllers based on model and pole location," in *Proceedings of the 14th IEEE International Conference in Electrical, Electronic and systems engineering — INTERCON 2007*, 2007.

[8] K. Ogata, *System Dynamics*, Prentice Hall, Upper Saddle River, NJ, USA, 3rd edition, 1998.

[9] G. F. Franklin and J. D. Powell, *Digital Control of Dynamic Systems*, Prentice-Hall, Englewood Cliffs, USA, 3rd edition, 1997.