12

Remote-Tuning – Case Study of PI Controller for the First-Order-Plus-Dead-Time Systems

Dennis Brandão, Nunzio Torrisi and Renato F. Fernandes Jr
University of Sao Paulo/Sao Carlos
Federal University of ABC/Sao Paulo
Brazil

1. Introduction

Control loop tuning today is the key factor for quality improvement and optimization of production costs. Simply replacing old systems with modern technology and networked equipment with high-processing capacity does not necessarily imply an improvement in product quality or an increase in plant productivity. However, according to researches conducted in the Industrial sector, most control loops in automatic operation have tuning problems (Harris et al., 1999; Ruel, 2003; Yu, 2006).

Nowadays, systems that aid automatic tuning for control loops can be typically located at two levels in the automation systems hierarchy: they can be installed in the workstations, running together with software control systems and data acquisition or SCADA (Supervisory Control and Data Acquisition) systems; or they can be embedded within a distributed control equipment in the field, for example, in PLCs (Programmable Logic Controllers), smart transmitters, or DCS (Distributed Control System).

When embedded, automated tuning systems are usually operated by an adaptive control where the controller's parameters are continuously adjusted to accommodate changes and process disturbances. Tuning systems already installed in workstations are more advantageous than embedded systems due to their superior processing power and information storage. This characteristic leads to the development of more sophisticated algorithms, and provides additional resources, such as simulation and graphical analysis (Aström & Hägglund, 1995), (Ang et al., 2005).

Remote access systems that use the Internet as the means of communication have become widespread in recent years, both in academic researches and industrial applications. Studies such as (Avoy et al., 2004) show great potential to growth and diversification of remote applications mainly in industrial environments.

Among the advantages of remote access via the Internet in industrial applications, it is important to highlight: enterprises with distributed units can access, share, analyze and process plant floor information in real time and faster; and technical or specialized administrative services can be outsourced with a higher level of interaction between partners, thereby avoiding the need for experts in the staff.

Studies in the Literature report proposals and systems for automatic tuning and dynamic control that use the communication via the Internet in different ways (Yu et al. 2006) (Yang

et al., 2007). An important application registered in those work is related to learning and research centers that provide experiments in robotics, manufacturing control and process control for remote access over the Internet.
However, when considering the use of the Internet directly on the shop floor, observe that the nature of production and automation systems demands certain requirements that must be guaranteed, such as multiple access management, communication and control system security, maximum time interval for process data update, and the integration of different computing platforms and equipments from several technologies.
The aim of this paper is to propose and verify the technical feasibility of a computer architecture in order to achieve remote tuning of control systems on the Internet using open industry communications standard, with requirements satisfactory of performance and security.
The main contributions of this chapter is to propose the use of a Internet link to connect a automation system to a PID tuning tool and evaluate the impact of the communication non-determinism into the final performance of the controller, when compared to a local PID tuning.
The next section will study in detail the main features of the SCADA system communicating remotely with the factory system.

2. Remote SCADA systems

In remote monitoring, SCADA systems are network clients remotely connected to the control system of the shop floor. Typically, control centers, servers and the shop floor are located within the plant, while remote stations, which access data from these servers, are geographically distributed from each other. Remote connections between clients and servers are based mainly on the physical Ethernet, connected remotely via the Internet through the host server. The following figure shows an example of a typical network installation in the industrial environment.

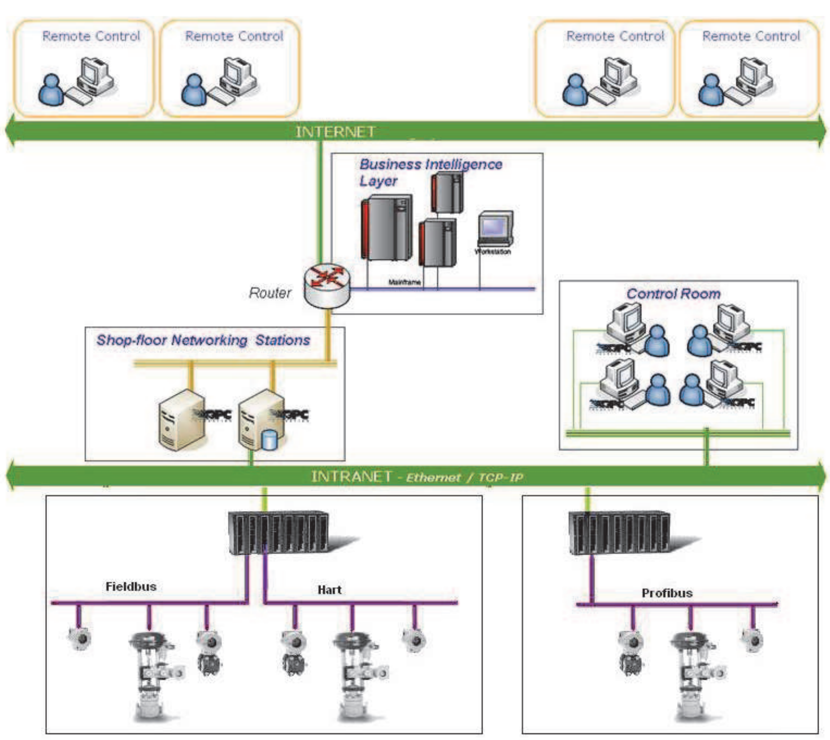


Fig. 1. Example of industrial applications using remote communication.

It is necessary to develop mechanisms for communication networks that can provide services with differentiated quality for real-time applications and multicast, since these applications demand minimum quality in terms of temporal parameters (such as delay and jitter) and effective transmission capacity (such as bandwidth). Several protocols for different network layers attempt to improve quality of service and determinism, for example, RSVP (Reservation Protocol), which only handles the reservation of resources along the route between network nodes. RTP (Real-time Transport Protocol) provides synchronization services, multiplexing, and security for data transfer, and these later two features focus mainly on image processing and voice using the Internet (Hanssen & Jansen, 2003).

Market solutions and academic researches aiming to facilitate the "open" system integration scenario for a shop floor over the Internet, make use of object-oriented technologies such as OPC (OLE for Process Control) through DCOM(Distributed Component Object Model), and DAIS (Data Acquisition from Industrial System) through CORBA (Common Object Request Broker Architecture), that is, all web-based services.

For application layers from the OSI model, there are several technologies to access data from industrial processes, such as: ASP (Active Server Pages) used together with ActiveX objects, and PHP (Hypertext Preprocessor) accessing process database servers available in SQL (Structured Query Language) (Zeilmann et al., 2003). OPC DCOM can be used to access distributed applications, as well as open standard technology such as XML (Extensible Markup Language) or JSON (JavaScript Object Notation), which are currently in widespread use for communication over the Internet.

The OPC Foundation has been developing a new OPC standard based on XML (OPC-XML 1.0 Spec.) since 2003, and included this new standard in a multiple protocol profile specification called OPC UA (Unified Architecture). The OPC UA aims to integrate the various existing OPC specifications (AD, AE, HDA, DX, etc.) into a single database, facilitating the development of applications (OPC Foundation, 2006). In addition, OPC UA offers support to portability and, therefore, can be integrated to any platform. However, this technology is still under approval and there are few commercially launched devices based on this standard.

The new OPC UA specifications show the path to open technologies, like XML, as a major trend in industrial systems interactivity over the Internet (Torrisi & Oliveira, 2007).

3. Current supervision and control researches over the Internet

The World Wide Web has provided opportunities for development and analysis of control systems over the Internet, according to studies by (Yu et al. 2006). Several papers propose the use of the Internet in control systems with different architectures.

Remote access architectures may be implemented at different levels in the manufacturing control hierarchy: at process level, at supervisory level, and at system optimization level.

The works of (Overstreet & Tzes, 1999) and (Yang et al., 2007) include remote control at the process level. In this case, the conventional discrete control structure must be changed to meet the diverging times of the Internet. (Luo & Chen, 2000) analyzed the network delay over the Internet using process control, concluding that the time interval for reading and writing over the Internet increases with the distance, depending on the number of nodes and the occupation of the network.

At the supervisory level, the concern is related to the quality of service. The work of (Kunes & Sauter, 2001) is based on SNMP (Simple Network Management Protocol) in fieldbus

technology systems. This architecture works well for read and write operations and asynchronous notifications, such as alarms. However, most firewalls do not allow UDP (User Datagram Protocol) traffic, and SNMP has low security levels.
Remote Internet protocol solutions, such as OPC-UA and NISV (National Instruments Shared Variables), adopt Client/Server architectures where remote control client applications receive periodic data refresh from Server values and send aperiodic sets of data.
A common practice is grouping the values of items of interest from the Server, with similar change rates, and assigning them to a group in order to allow the remote client to later retrieve all updated values from the group simply requesting the group name identifier.
It is common to schedule periodic updates of data values sent from the Server to the Client using a mechanism called Subscription Polling Mechanism. Although this mechanism speeds up the refresh rate and minimizes the number of update requests to the Server, not all data values might be of interest to the remote control client because some values may not change enough to be relevant for the client application.
In order to minimize the amount of data related to changed values of interest sent from the Server to the Client, the Client can specify a parameter called *DeadBand* for each group, which determines the percentage of range that an item value must change prior to the value being of interest to the Client. Changes in values that are not interested to the Client are not sent to the Client, therefore reducing the amount of data delivered over the network (Torrisi 2011).
(Yang et al., 2004) proposes a remote control at the supervision level for services that do not dependent on the Internet delay, which would be restricted to acyclic services such as SP alteration and tuning parameters of a PID block. The (Yang et al., 2004) studies present a virtual supervisory parameters control. This work shows the control would be invoked only when alterations for parameters such as setpoint (SP) and PID tuning parameters were requested, and then data would be sent to the control. In this context, multiple concurrent accesses are allowed by solving possible conflicts. Also, the security for the whole process is guaranteed since it is possible to provide redundancy and failure diagnostics in remote communication. Another approach at the supervisory level would be remote executing identification and tuning.
(Qin & Wang, 2007) studied the admission control to a web server, which accepts or rejects requests for the system. A Linear Parameter Varying (LPV) method is proposed to identify and control a web server, because the LPV approach tunes the model by specifying the loading conditions of the Internet, allowing the system to adapt to variations in load and operating conditions.
Companies currently offer some programmable logic controllers (PLCs) solutions with embedded web servers, but these solutions have limitations when applied to complex industrial plants (Calvo et al., 2006). For example, the work of (Batur et al., 2000) shows the architecture for remote monitoring and tuning using an SLC 500 Allen Bradley Company. The proposed system uses the measurement variables with the respective sampling times to ensure more determinism in the network. A mechanism for access control is also described, but the disadvantage of the system is that it consists of a proprietary solution, fully based on enterprise software to achieve monitoring and tuning for the controller.
(Yang et al., 2007) presents the architecture for processes control maintenance based on the Internet. The studied characteristics include industrial system performance indices, and failures and successes detection in the degraded control performance. The proposal monitors the system performance index locally, and if any noticeable change occurs in the

index, it will be identified in the system and, then, analysis and tuning will be executed for the stations. In the proposed architecture, the work considered as "heavy", such as the performance index calculation and model identification, is divided and processed locally. Work considered as "light", such as performance test results and process model, is sent to remote analysis. Thus, data analysis would be undertaken by experts who would propose tuning.

Several institutes and companies have conducted researches and provided control and distance learning applications for control systems over the Internet. These works are basically divided in two levels of interaction: the concept of virtual laboratory that brings together a developed physical structure and its subsequent release on the Internet; and distance learning courses that also offer a high level of interactivity, enabling, in some cases, simulation of physical phenomena. These virtual labs allow the user to tune control plants remotely, either through the simulated plant or through a real plant (Ko et al., 2005), (Zeilmann et al., 2003).

4. Common problems for Internet-based supervision and control

The Internet and web services have some obstacles related to their use for industrial control systems, such as delay in communication, data security and latency of web services.

Delay in communication – Over the Internet, a data packet suffers from several types of delays throughout the path from the source to the destination. The main types of delays are: processing delay, queuing delay, transmission delay and propagation delay, for each network node. Processing delay refers to the internal software processing to scan the message and determine where to send it, or check for errors in the message. The queuing delay happens while the message is waiting for queuing during transmission. Transmission delay refers to the time taken to get to the equipment and then be transmitted over the network. Finally, propagation delay refers to the time interval to spread the message on the line. According to (Han et al., 2001), the delay time T_a from the Internet at the time k can be described by:

$$T_a(k) = \sum_{i=0}^{n} \left[\frac{l_i}{C} + v_i^R + \frac{Q}{r_i} + v_i^L(k) \right] = d_N + d_L(k) \tag{1}$$

Where l_i is the distance to the n^{th} link on the network, C is the speed of light coming and the speed of the n^{th} router, Q is the amount of data, r_i is the bandwidth of the n^{th} link and $T_a\ (k)$ is the delay caused by the load of the n^{th} node.

Separating the terms that are dependent and independent of time, there will be a d_N part of time-independent terms and a d_L part of time-dependent terms.

The contribution of each delay component can vary significantly. For example, the propagation time is negligible for the communication between two routers located in the same laboratory; however, it may vary significantly for equipment connected by a satellite link and be the dominant term in the total time delay (Kurose and Ross, 2006).

According to a study by (Luo and Chen, 2000), the performance associated with time delay and data loss shows a large spatial and temporal variation. The average delay of messages increases linearly with the increased traffic, according to (Boggs et al., 1988).

Non-determinism of the network - the Internet network is composed of multiple subnets and multiple routers between the source and destination station. The routers are responsible to select the most appropriate route for the message traffic between those two points. The routing algorithm varies with changes in stability of hardware and software throughout the network. The decision of the best route to be used should be taken for each data packet received. Consequently, there is no guarantee to the determinism of the network (Kurose, Ross, 2006).

However, many techniques have been developed to support real-time traffic, in particular, for the Ethernet. The work of (Wang et al., 2000) proposed management of collisions on the Ethernet. (Loeser & Haertig, 2004) proposing the joint use of intelligent switches and traffic management. (Gao et al., 2005) propose a real-time optimal smoothing scheduling algorithm with the variable network bandwidth and packet loss for data streaming.

The work of (Yang et al., 2004) reports that including the Internet to the levels of industrial control systems would not be practical because the Internet is highly non-deterministic with substantial delays, and the determinism is required on the network.

Data security on the network – It is fundamental to data traffic that distributed control systems meet the requirements for secure communication. In this case, it is necessary to fulfill the following security properties: confidentiality, authentication and message integrity. The confidentiality expects that only the sender and the recipient involved in the connection should understand the message content. Authentication requires that both the source and the destination confirm the identity of the other party involved in the communication. Integrity is required to ensure that the content of the message is not altered during transmission (Kurose and Ross, 2006).

Latency of web services - Despite the advantage of high interoperability, since all SOA (Service Oriented Architecture) entities use common languages for service descriptions, messages and records of services, the use of SOA causes problems of latency and memory space related to the use of web services, and according to (Pham and Gehlen, 2005) these features can be critical depending on the application. For industrial applications where asynchronous and synchronous communication is necessary, jitter effects– in terms of delay variation between successive data packets - may occur due to high internal processing (Torrisi & Oliveira, 2007).

Figure 2 shows the steps for exchanging data using web services. The Application layer represents the boundary between the OPC DCOM client and OPC DCOM Server located locally or remotely.

An application request is made by the remote Internet client through a call to the web server. This web server will receive the request and transfer it to the HTTP-SOAP (Hypertext Transfer Protocol- Simple Object Access Protocol) processor server (this process is shown in Figure 2 as Step 1). In this step, the SOAP/XML request is parsed to recognize commands and parameters, and it could have been binary decoded previously if it were an OPC-UA SOAP/XML request. Then, the corresponding API is invoked (Step 2), to forward the corresponding requested function to the application server (Step 3). The application server requests the message to be handled by the client on the OPC DCOM protocol. After that, the message is passed to an OPC server. Finally, the OPC server will request the data from a field device that will respond, and then the cycle is reversed and the whole process is executed until returning to the Http layer again (Step 6).

According to (Torrisi, 2011), this solution was not developed to meet the requirements and performance standards that are required for the industrial environment.

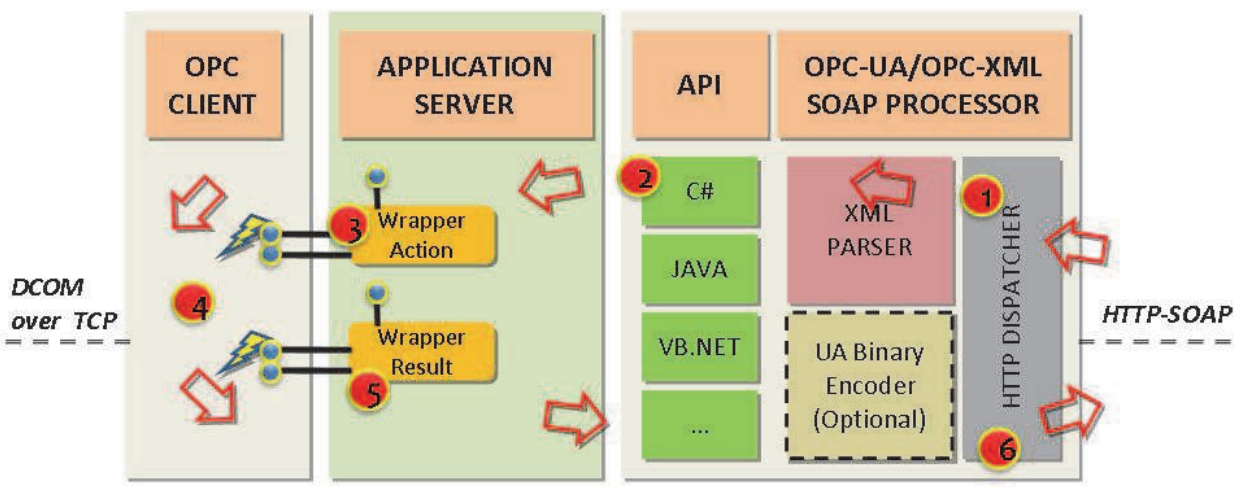


Fig. 2. Data processing path within OPC profiles using web server.

(Torrisi & Oliveira, 2007) proposed a new form of remote communication without the use of web services, called CyberOPC. The CyberOPC solution has a new process data transport protocol with the following characteristics: reduce transport delays for critical time data; ensure security for the communication channel used; ensure integrity and confidentiality for transmitted messages. In order to obtain maximum interoperability with existing shop floor technologies, open standard technologies were used, such as OPC DCOM. CyberOPC communication foresees the use of a gateway station called *CyberOPC gateway* that processes messages sent to the OPC through the public network, and vice versa. Due to the simplicity and short number of CyberOPC commands, the "Parser" containing the rules to recognize these commands is simpler than any XML parser for SOAP messages. Therefore, the OPC commands are executed quickly and, in the case of a periodic request, it is possible to increase the response time using a dedicated cache shared by the OPC client and OPC HTTP Broker.

A quick OPC data cache can be written asynchronously by the OPC client to all periodic data request from the remote Internet client, as shown in Figure 3.

A client application request is received by the gateway (Step 1), which now has the SOAP processor block. Introducing the OPC cache strongly reduces the time taken to call the OPC client. Tests conducted by (Torrisi & Oliveira, 2007) showed a significant reduction for posting time optimization when compared to the gateway-based web services, such as OPC-XML and OPC-UA SOAP/XML. Steps 2, 3, and 4 represent the interaction between the CyberOPC library and the OPC layer.

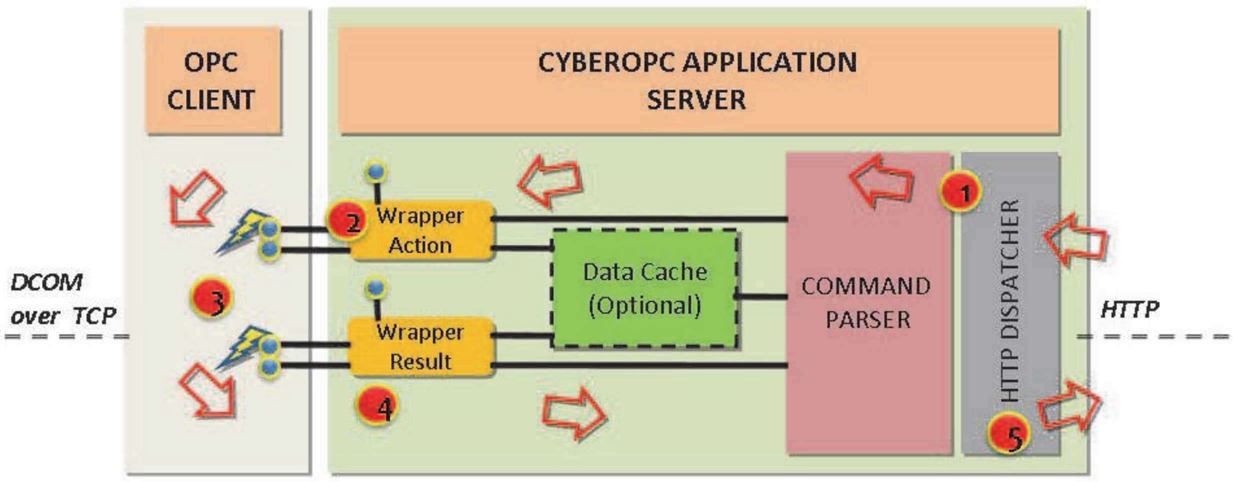


Fig. 3. Data processing path within OPC profiles using CyberOPC web server.

5. Architecture for remote tele-tuning

According to (Zeilmann et al., 2003), software for remote monitoring and acquisition must have a generic framework for data acquisition via the Internet to try to meet the vast majority of industrial automation systems. For such structure to be met, the following characteristics are desirable:

- Remote access to industrial automation system data for clients;
- Network data acquisition performance, specified in the refresh rate for data and maximum data delivery delay;
- Ensuring security in the communication channel to prevent unauthorized access;
- Ensuring integrity, confidentiality and reliability of transmitted messages;
- Open communication interface between software components, as a requirement for system scalability;
- Independence from field devices and protocols in operation;
- Independence from the platform of the remote client and the server, from the industrial automation system.

This section describes a tele-tuning architecture based on the interconnection of modules contained in three different contexts: the industrial plant, the server, and client, as shown in Figure 4. The architecture is based on the client-server application cooperation model, consisting of separated modules that are interconnected in order to provide process and configuration variables from the plant to the remote client. The entire HTTP communication is secured using SSL (Secure Sockets Layer) and, for such reason, HTTPS (HyperText Transfer Protocol Secure) will be cited instead of HTTP.

The Industrial Plant. Nowadays, there are several communication protocols for devices that meet specific applications in industrial environments, for example, process control and manufacturing control. The physical means of communication between these devices also differ from each other, either on the possible topologies, cable types, presence or absence of feeding overlapped communication, adaptation to usage requirements for hazardous areas, among others issues.

The tele-tuning architecture provides a communication channel between the field controller and the device driver for data acquisition, the latter being installed in a computer. Since it is

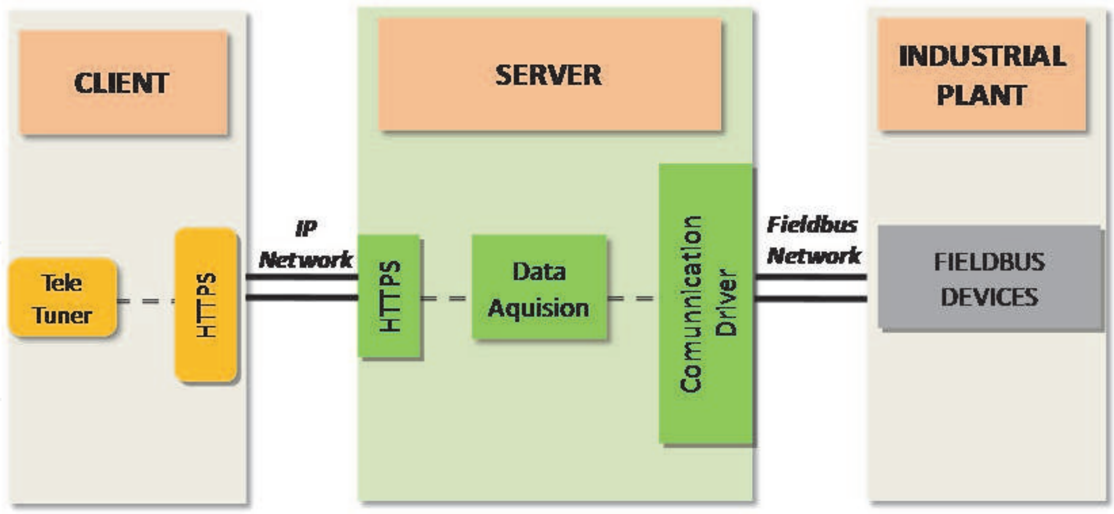


Fig. 4. Tele-Tuning architecture.

assumed the controller or the field devices are generic, the communication protocol between these devices and the related device driver is defined according to the equipment or system in use. In this case, a proprietary or open communication protocol can be used. However, the device driver needs to have an open software interface that can be easily integrated to any software component.

The **network server** is responsible for the interface between the plant and the remote clients. The server consists of several communication modules, as shown in Figure 4: the device driver for communicating to the controller and field devices (OPC Client), the Data Acquisition System, and also, the data web server for remote clients.

The **client** is the remote monitoring and tuning unit, as indicated in Figure 4. The requirement for the client is being an OPC DCOM client that enables communication to several equipment networks. In this architecture, an OPC DCOM client and CyberOPC client were used, which facilitates the implementation of the communication in both local and remote environment. The following section describes more details about the client side.

5.1 Monitoring and tuning system

The control system tuning is typically composed of the following phases: plant data acquisition, system identification, model validation, plant dynamics simulation tuned for verification purposes, control loop tuning, and data effectiveness in the plant (Ljung, 1999).

The proposed tele-tuning called Cybertune, is composed of four main operational modules: data acquisition module, the system identification module, the auto-regressive exogenous (ARX) model to open loop transformation module, and the tuning module. Figure 5 illustrates the relationship between these modules.

The **Data acquisition module** consists of an OPC client or CyberOPC, according to the OPC DCOM specifications (OPC Foundation, 2006) or CyberOPC specifications (Torrisi & Oliveira, 2007). The interface component has the same data access philosophy, consisting of an OPC DCOM library record, groups and items added to the database, and acyclic communication per event, when the client is notified in the occurrence of a new Data event issued by the server.

The **System identification module** is responsible for determining the system transfer function. In this work, ARX model was used due to good results for first and second order linear systems, and it is well-known in the consulted Literature (Aguirre, 2004).

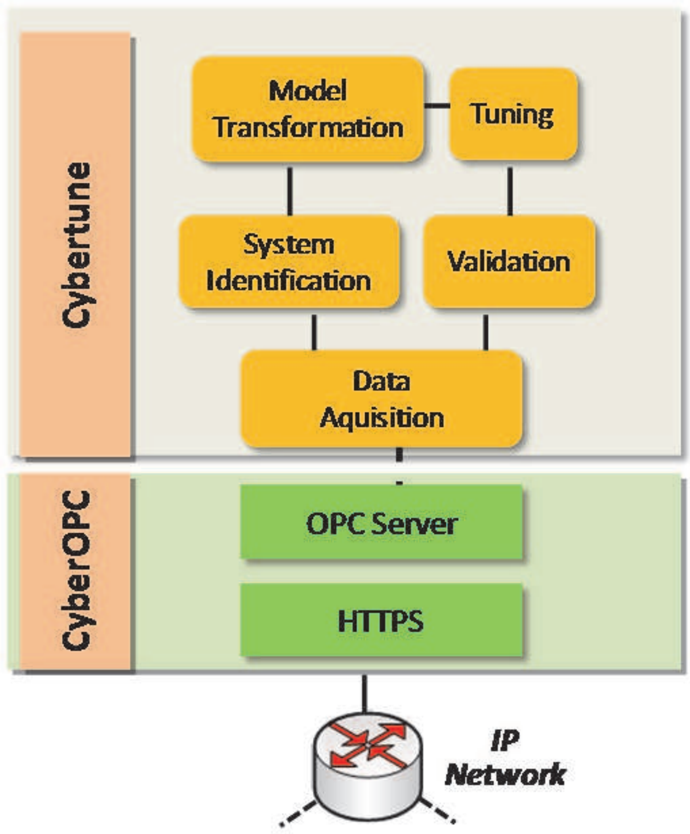


Fig. 5. CyberOPC client Tele-Tuning Schematic.

Because this project aims to validate the architecture for online identification and tuning, in a fair and reliable manner, the main purpose of the identifier is receiving process data and automatically processing the identification, through online identification. It is also important to process the identification offline, where an initial data collection is processed and recorded on the database for later identification and tuning.
The remote online identification presumes that communication delays and sending failures occur. This paper proposes the following solution to prevent these occurrences.
First, all samples collected by CyberOPC are recorded with the timestamp when the gateway acquired the data. Second, since the ARX model requires continuous sampling and CyberOPC sends data in a streaming optimized way (on data change), it is necessary to reconstruct the process signal at a constant sampling rate. To solve this matter, the **pre-identification module** was included. This module is responsible for receiving data from the acquisition module queue and sampling the data to the data identification queue at a constant sampling rate. In order to connect two sampling points, a first-order interpolation is used. Figure 6 show an example for this architecture.
The Cross Test method presented by (Aguirre, 2004) is executed in order to validate the identified model. This method compares the response generated by the identified model and the actual system response, for the same input signal. During the validation, the mean squared error and the percentage rate of the output variation is calculated as a performance measure and method validation measure.

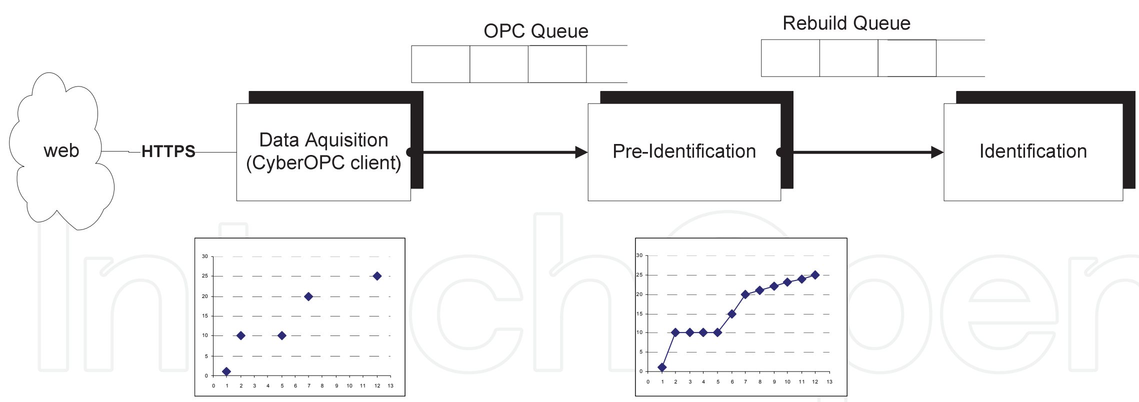


Fig. 6. Processing scheme for data received by OPC and CyberOPC.

In order to obtain the tuning of the model estimated using the model-based methods, it is necessary to obtain the transfer function of the system that in the architecture proposed is obtained by **the model transformation module**. In the identification module was obtained the differences equation (ARX) from the data collected. Thus, it is still necessary to convert the ARX model to the transfer function. The first order plus dead time (FOPDT) transfer function is showed in (2), where K_p it is the static gain, τ the time constant and θ is the system dead time.

$$G(s) = \frac{K_p}{\tau s + 1} e^{-\theta s} \tag{2}$$

In the work (Fernandes & Brandão, 2008) described a mathematical formulation to convert the ARX model in a transfer function (2). Below is described the equations for open loop.
Consider a block diagram of a classic feedback controller system as shown in the figure 7.

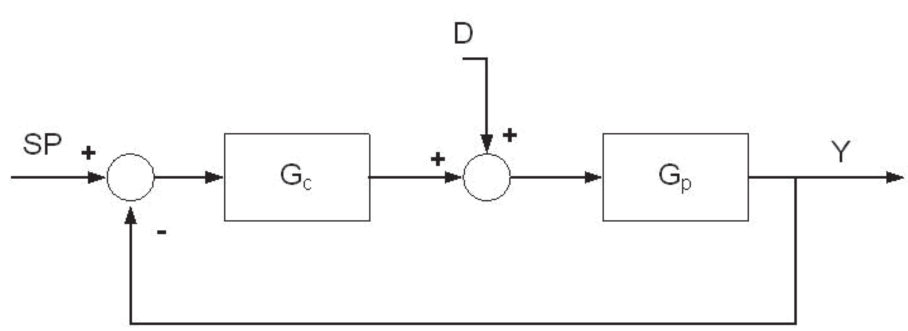


Fig. 7. Classic feedback controller, where G_c is a controller and G_p the plant. The SP is the setpoint , D the load variation, and the Y the system output.

The specification of tuning a controller can be classified as variations due to changes in setpoint (SP) or load variation (D). The control process should act to minimize these disturbances.

$$\frac{Y}{SP} = \frac{G_c G_p}{1 + G_c G_p} \tag{3}$$

$$\frac{Y}{D} = \frac{G_p}{1 + G_c G_p}$$

Consider a PID controller ISA standard, as described in (4) where K_c is the proportional gain , T_i integral time constant and T_d the derivative time constant of the controller.

$$G_c = \frac{Y(s)}{U(s)} = K_c\left(1 + \frac{1}{T_i s} + T_d s\right) \quad (4)$$

The open loop system is accomplished by placing the controller on manual. The real system can be described by equations in open loop showed in (5), where G_p the transfer function of the process, Y the system output, and D disturbance of excitation system.

$$\frac{Y}{D} = G_p \quad (5)$$

For the first-order system shown in (5) and Paddé approach yields the equation (6) in the continuous domain.

$$\frac{Y}{D}(s) = \frac{(-0.5K_p\theta)s + K_p}{(0.5\tau\theta)s^2 + (0.5\theta + \tau)s + 1} \quad (6)$$

The equation (7) is the conversion from continuous time to discrete using backward difference approximation.

$$\frac{Y}{D}(z) = \frac{(bz_1)z + (bz_2)}{(az_1)z^2 + (az_2)z + (az_3)} \quad (7)$$

Where the arguments of equation (7) are showed in (8). T_0 is the system sample rate.

$$\begin{aligned} az_1 &= 0.5\tau\theta + 0.5\theta T_0 + \tau T_0 + T_0^2 \\ az_2 &= -\tau\theta - 0.5\theta T_0 - \tau T_0 \\ az_3 &= 0.5\tau\theta \\ bz_1 &= K_p T_0^2 - 0.5K_p\theta T_0 \\ bz_2 &= 0.5K_p\theta T_0 \end{aligned} \quad (8)$$

The equation (8) is equivalent to differences equation from ARX model showed in (9).

$$\frac{B}{A}(z) = \frac{B_1 z + B_2}{z^2 + A_1 z + A_2} \quad (9)$$

Then, equating (9) and (7) obtain (10).

$$B_1 = \frac{bz_1}{az_1} \quad ; \quad A_2 = \frac{az_2}{az_1} \quad (10)$$

Finally, applying (8) and (9) in (10) and making some adjustments are obtained the values of Kp, τ and θ from the FOPDT model as (2).

$$\begin{cases} K_p = \dfrac{(B_1 + B_2)\left[(0.5\tau\theta) + (0.5\theta T_0) + (\tau T_0) + (T_0^2)\right]}{T_0^2} \\ \tau = \dfrac{-\left[(0.5A_2\theta T_0) + (A_2 T_0^2) + (0.5\theta T_0)\right]}{\left[\theta + T_0 + (0.5A_2\theta) + (A_2 T_0)\right]} \\ \left[(-0.25A_2T_0) - (0.25T_0) - (0.25A_3T_0)\right]\theta^2 + \\ +\left[(-0.5A_2T_0^2) - (A_3T_0^2)\right]\theta - (A_3T_0^3) = 0 \end{cases} \tag{11}$$

For the solution of equation (11) is necessary to solve a second-order polynomial where there are two possible solutions to the system. To determine the solution that best fits the system is compared to identify the model obtained with each solution obtained from the open-loop system. For comparison we used the criterion integral of absolute error (ITAE).
Using the same concept we obtain the equations of a closed loop system.
The **Tuning module** is responsible for applying the tuning method to the model obtained by the identification module. Among several tuning methods discussed in the literature, here will be considered the methods that fulfill the requirements related to good performance in first-order systems. This work does not intend to validate tuning methods; therefore only the integral optimization methods of square error and absolute error (ITSE and ITAE) and internal model control (IMC) method will be used in practical experiments for the proposed system, due to their good response to FOPDT systems. However, other developed methods can be feasibly gathered to the tuning knowledge base simply by meeting the requirements of identification systems.
The methods used in this work are based on methods already studied in the literature. First of all, is assumed the controller has original tune based on the classical Ziegler-Nichols (ZN). Then, it is suggested other methods to improve the tuning that are the methods based on performance criterion error integral and by internal model control (IMC). The advantages of these methods include a low overshoot and good settling time (Lipták, 2003), (Seborg et al., 2004), (Zhuang & Atherton, 1993). Below is showed the parameterization of each method used for tuning proposes.
The parameters for ZN method for open loop are showed in (12).

$$K_c = \frac{0.9}{(K_p / \tau)\theta} \quad ; \quad T_i = 2\theta \tag{12}$$

For methods based on integral error criterion ITAE and ITSE, the relationship between the tuning of the controller and the integral criteria is based on the relationship θ/τ (the ratio of dead time and time constant of the system) and expressed in the equation (13), where X is a parameter of the controller (such as Proportional, Integral and Derivative) and m and n constants.

$$X = m\left(\frac{\theta}{\tau}\right)^n \tag{13}$$

The parameters of ITSE and ITAE for PI controller for disturbance due to load change are showed in the table 1.

	P		I	
Method	**m**	**n**	**m**	**n**
ITAE	0,859	-0,977	0,674	-0,680
ITSE (θ/τ 0.1 - 1.0)	1.053	-0.930	0.736	-0.126
ITSE (θ/τ 1.1 – 2.0)	1.120	-0.625	0.720	0.114

Table 1. Parameters of a PI controller for load change using the methods ITAE and ITSE.

The IMC method which name "internal model control" comes from the fact that the controller contains an internal process model. For a system FOPDT as in (2) the IMC controller equations is showed in (14).

$$G_c(s) = \frac{(1+(\theta/2)\,s)(\tau s+1)}{K_p(\lambda+(\theta/2))s} \tag{14}$$

Where λ is the tuning parameter specified by the user. The choice of design parameter λ is a key decision in more conservative or not controller. In this work is used λ=θ. Arranging (14) for a PI controller as in (4) became the equations (15).

$$K_c = \frac{1}{K_p}\frac{2(\tau/\theta)+1}{2(\lambda/\theta)+1} \quad ; \quad T_i = \theta/2+\tau \tag{15}$$

5.2 Remote tuning use case

Identification tests were performed to validate the proposed architecture simulating FOPDT systems with equation (2) using local and remote identification in a corporate network. The tests use the Cybertune software that allows the monitoring and updating remote data control systems in industrial environments using OPC and CyberOPC technologies.

Tests were conducted using the Fieldbus Plant Simulator (FBSIMU) (Pinotti & Brandão, 2005), which simulates the industrial plant and the fieldbus control logic. (Pinotti & Brandão, 2005) showed that FBSIMU has a good approach to the real system.

Figure 8 illustrates the tests scenarios for local and remote tuning. Local tests involved Cybertune communicating to FBSIMU in the same station using OPC DCOM communication. Remote tests consisted of Cybertune communicating to FBSIMU inside the campus intranet network, using CyberOPC protocol.

Simulation tests were performed using six systems with different characteristics, and considered the relation θ/τ varying from 0.1 to 2.5. The relation is used as a comparison among different types of FOPDT systems, as verified in (Seborg et al., 2004).

To validate the tests, the ITAE performance index and the correlation index (FIT) showed in (16) were used in relation to the real signal and the identified signal. For a higher correlation index, identification is considered good (Ljung, 1999).

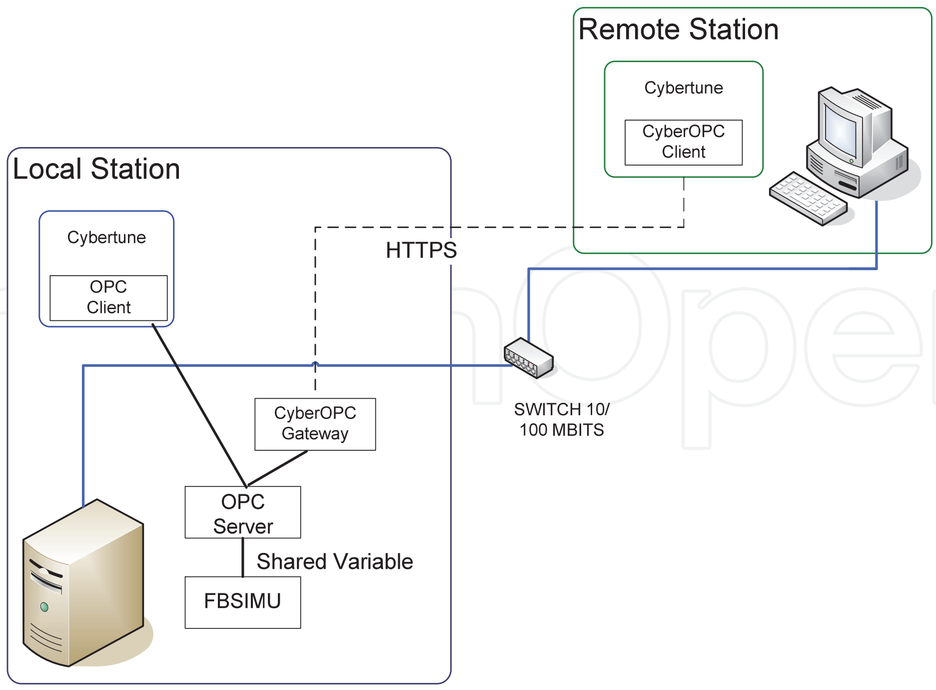


Fig. 8. System architecture for local and remote communication between Cybertune and FBSIMU.

$$FIT = \left(1 - \frac{\sum_{k=1}^{N} Norm\left(y(k) - \hat{y}(k) \right)}{\sum_{k=1}^{N} Norm\left(y(k) - \bar{y}(k) \right)} \right) 100 \quad and \quad Norm(V) = \sum_{k=1}^{N} \sqrt{ABS(V)^2} \tag{16}$$

Where, $y(k)$ is the actual process output in the k instant, $\hat{y}(k)$ is the estimated output and $\bar{y}(k)$ is the average of samples throughout the identification.
Consider the FOPDT example from (2) showed in (17):

$$G_p = \frac{2}{100\,s+1} e^{(-50\,s)} \tag{17}$$

Initially, in the model identification phase, to achieve the identification of noisy signals using the parametric models, it is necessary to use a model to identify the highest order to obtain a good approximation of the model. And after the identification of high-order model, the poles and zeros used to describe the noise signal can be canceled (Ljung, 1999). One way of reducing the order of the model is to use only the dominant poles. The equation (9) of the open loop algorithm requires a second-order equation.
Regarding the local identification test, identification is estimated according to approximation using a fourth-order ARX model and sampling rate (To = 1.0 sec). The open-loop transfer function (OPTF) model shown in (18) was obtained. The FIT obtained was 98.50%.

$$OPTF_{LOCAL}(z) = \frac{-0.0170z^3 + 0.0060z^2 + 0.0105z + 0.0023}{z^4 - 1.0030z^3 - 0.4280z^2 - 0.0129z + 0.4448} \tag{18}$$

Using equation (18) was performed a reduction of the model for a second-order equation as in (9) and then applied the equation (11) for the model transformation. The result was the equation (19) which represents the G_p estimated locally.

$$G_{pEST\_LOCAL} = \frac{2.00}{102.05\,s+1} e^{(-46.62\,s)} \tag{19}$$

The graph in Figure 9 compares the real system and the system identified locally. The final solution has FIT equals to 98.09%.

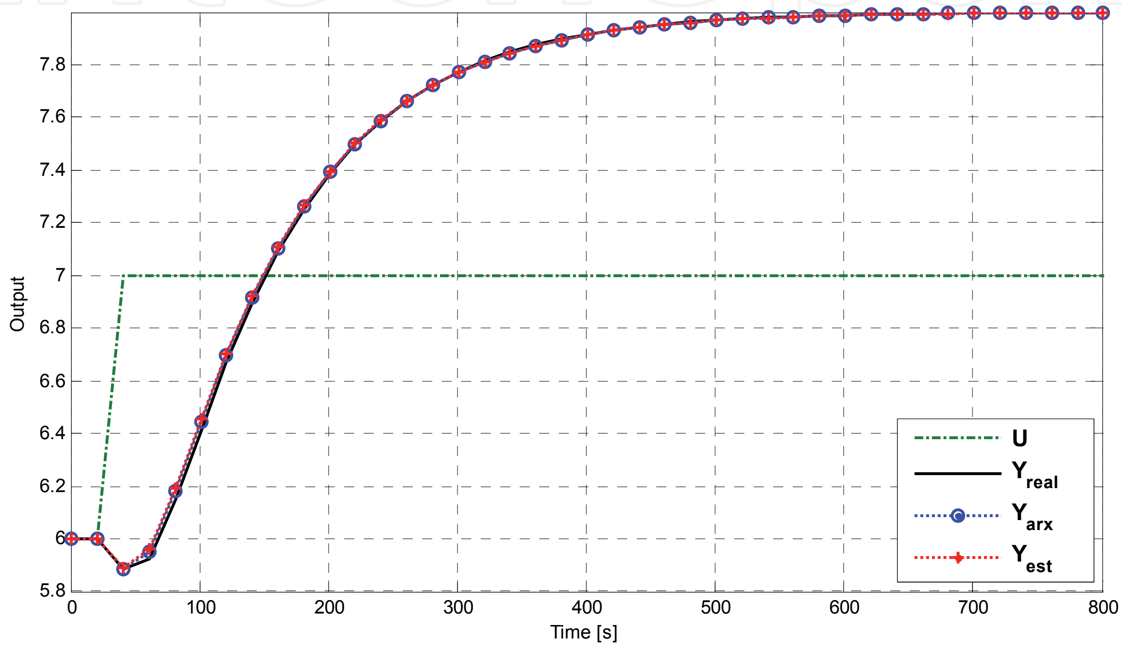


Fig. 9. Graphic comparison of system responses to a step change in an open loop. It shows the original signal (Y_{real}), the 4th order ARX signal (Y_{arx}) and the identified open loop system (Y_{est}).

The remote test with the same system (17) obtained the fourth-order ARX model with sampling rate (To = 2 sec), the estimated model (OPTF) resulted in equation (20) with FIT=95.59%.

$$OPTF_{REMOTE}(z) = \frac{-0.0398z^3 + 0.0065z^2 + 0.0220z + 0.0190}{z^4 - 0.9003z^3 - 0.478z^2 - 0.0808z + 0.4630} \tag{20}$$

After transforming the ARX model into the open loop model, the model approximation is obtained in (21), which represents the G_p from (17) estimated remotely:

$$G_{pEST\_REMOTE} = \frac{2.00}{100.56\,s+1} e^{(-45.36\,s)} \tag{21}$$

The graph in Figure 10 compares the real system and the system identified remotely. The final solution has FIT equals to 93.63%.

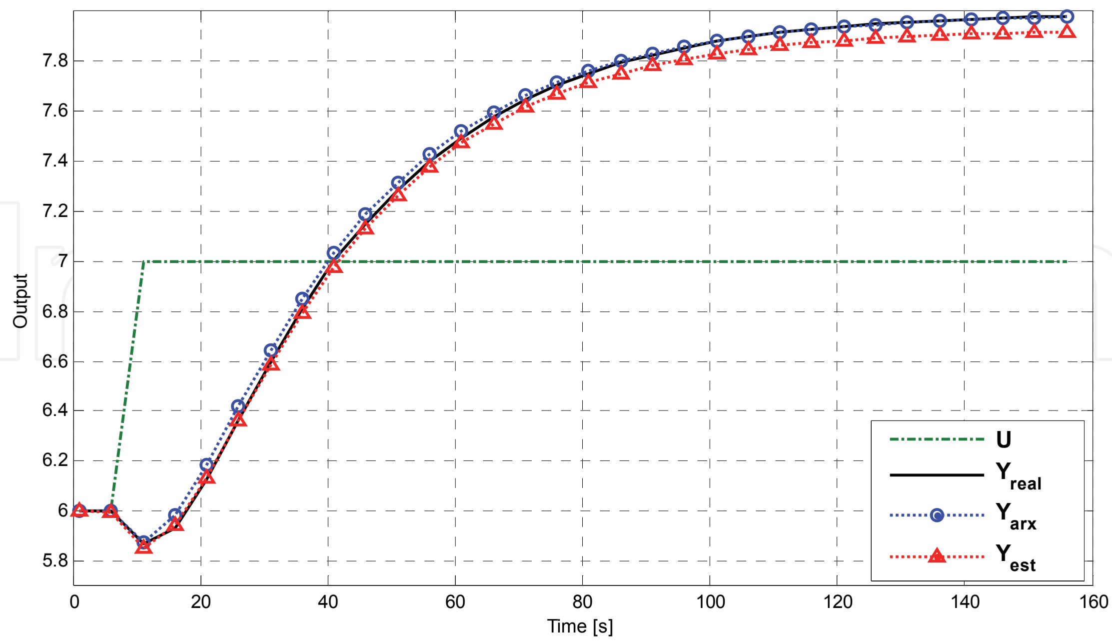


Fig. 10. Graphic comparison of system responses to a step change in an open loop. It shows the original signal (Y_{real}), the 4th order ARX signal (Y_{arx}) and the identified open loop system (Y_{est}).

Table 2 and Figure 11 summarize the results from remote and local ARX identification for each of the six systems. The "Local" identification was realized with the Cybertune running in the same station as showed in the Local Station in the Figure 8. The "Remote" identification was executed with the Cybertune running in a remote station connected to the local station using an internet link of 1 Mbyte.

System	θ/τ	Type Identification	To [s]	FIT [%]	ITAE
1	0.13	Local	1	98.17	1.22E+03
		Remote	2	97.91	1.25E+03
2	0.50	Local	1	98.50	7.92E+02
		Remote	2	95.59	3.42E+02
3	1.14	Local	1	95.24	4.60E+03
		Remote	1	94.82	8.34E+02
4	1.53	Local	1	93.77	3.11E+03
		Remote	2	93.44	2.51E+03
5	1.90	Local	1	98.02	1.80E+03
		Remote	5	96.38	2.02E+03
6	2.33	Local	1	93.77	3.11E+03
		Remote	5	91.60	1.78E+03

Table 2. Local and Remote Identification Results.

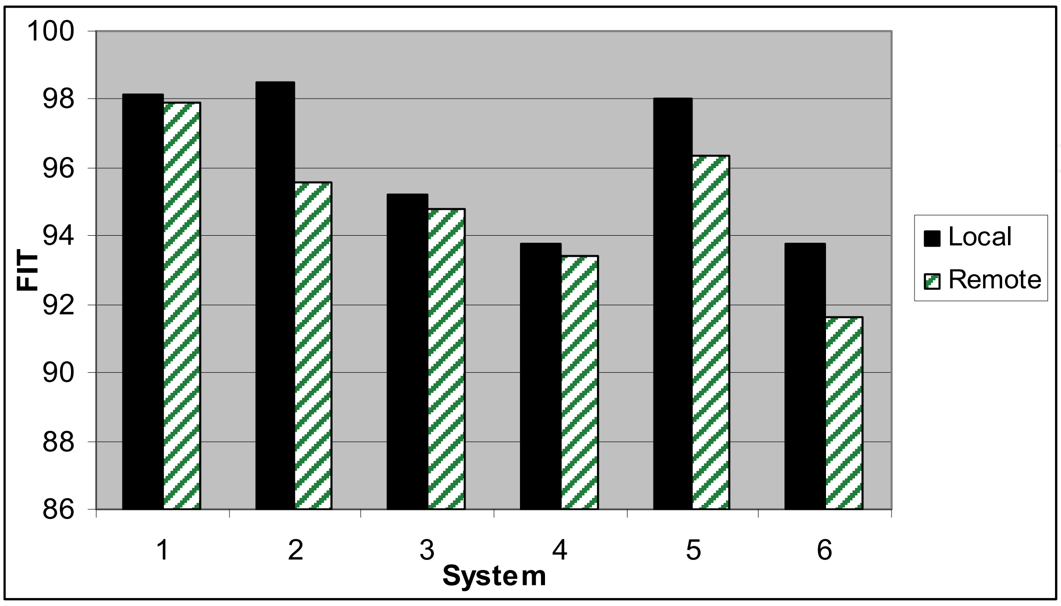


Fig. 11. Comparing local and remote tests for different systems.

The open loop model obtained from the identification phase is used in the tuning phase. This example assumes the controller (Gc) it is PID-ISA as showed in (4).
The tuning parameters of the PI controller for this work, the parameters of a PI controller (K_c and T_i) of current tuning using Ziegler-Nichols (ZN) method and three suggest methods ITSE, ITAE and internal model control (IMC) are summarized in the table 3. The table shows the column "error ITAE" for each method to help in choose the best tuning method.

Method	K_c	T_i	Error ITAE
ZN	0.90	100	1.06E5
ITSE	1.35	298.733	4.05E5
ITAE	0.93	87.1519	8.57E4
IMC	0.8903	146.1284	1.84E5

Table 3. Tuning Results for methods where K_c and T_i are the Parameters of PI controller and ITAE is the error for a step response.

The figure 12 shows the step response for load variation (D) and the system response for the suggest methods ITSE, ITAE e internal model control (IMC).

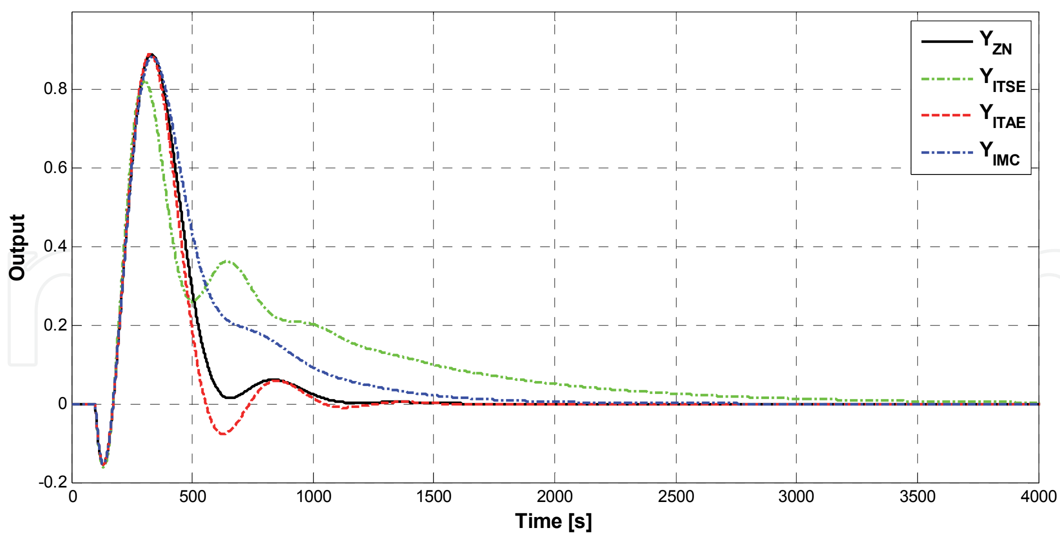


Fig. 12. Results of tuning for the model obtained in the remote station. The step response of the load variation of original tuning ZN (Y_{ZN}) and some common methods based on model (Y_{ITSE}, Y_{ITAE} and Y_{IMC}).

6. Conclusion

The Tele-tuning architecture executes remote tuning for industrial control systems using the Internet, fulfilling acceptable security and performance requirements. In order to validate the architecture, a software application, called CyberTune, using CyberOPC was presented. Validation consisted of a model-based identification and tuning of six FOPDT systems with diferentes θ/τ. This model is a typical class of industrial systems, but the architecture can be extended to other configurations.

The analysis of table 1 demonstrated that remote model identification is really near to local identification and the original system, which validates the architecture for identification and subsequent tuning, implemented with model-based methods.

For remote identification or noise signal, it is necessary to pre-filter the signal and use the highest order of the ARX model to obtain a good approximation of the model as showed in (20).

For remote tuning, the proposed architecture using CyberOPC and reconstruction of the data showed satisfactory results as shown in Fig 12.

Remote monitoring and tuning of control system might be a good solution for process plant companies with multiple sites in remote locations in order to provide the central support for their geographically dispersed control systems. By using this remote monitoring and maintenance system control software suppliers can monitor and maintain their control software products remotely over the Internet.

Nowadays the Ethernet and the Internet are increasing the speed quickly. Industries are beginning to implement networked control systems through this high speed communication. The speed of the next generation Internet might be sufficiently fast to be able to dramatically reduce the transmission delay and data loss. Therefore, it is possible that Internet latency and data loss might become less important issues in future Internet applications. But questions about the security of the Internet shall be continuing existing, because of the public nature of the Internet.

7. Acknowledgment

The authors gratefully acknowledge the Brazilian agency FAPESP for the financial support received, and the academic support and research structure of the Engineering School of Sao Carlos - University of Sao Paulo.

8. References

Aguirre, L. A., 2004. Introdução a Identificação de Sistemas, Técnicas Lineares e Não Lineares aplicadas a sistemas reais. 2ª edição, 2004, Editora UFMG.

Aström, K.J.; Hägglund, T., (1995), Pid Controllers: theory, design and tuning. 2a Edição, 1995, Editora Instrument Society of America (ISA).

Avoy, T. M.; Jounela, S.L.J.; Patton, R.; Perrier, M.; Weber, H.; Georgakis, C., 2004. Milestone report for area 7 industrial applications. In: Control Engineering Practice 12 (2004) 113-119.

Ang, K. H.; Chong, G.; Li, Y. (2005). PID Control System Analysis, Design, and Technology. In: IEEE Transaction on Control Systems Technology, Vol 13, No 4, July 2005.

Batur, C.; Ma, Q.; Larson, K., Kettenbauer, N., 2000. Remote tuning of a PID position controller via internet. In: American Control Conference, Chicago, 2000.

Calvo, I.; Marcos, M.; Orive, D.; Sarachaga, I., 2006. A methodology based on distributed object-oriented technologies for providing remote access to industrial plants. In:Control Engineering Practice 14 (2006), pp. 975-990.

Fernandes, R. F.; Brandão, D. (2008). Online identification method for industrial systems with closed-loop PI controllers, In: INDUSCON, Poços de caldas, August 2008.

Gao, K.; Gao,W.;He,S.; Zhang,Y. (2005). Real-time smoothing for network adaptive video streaming. Journal of Visual Communication and Image Representation Volume 16, Issues 4-5, August-October 2005, Pages 512-526.

Han, K. H.; Kim, S.; Kim, Y. J.; & Kim, J. H. (2001). Internet control architecture for Internet-based personal robot. Autonomous Robots, 10, 135–147.

Hanssen, F. T. Y.; Jansen, P. G. (2003). Real-time communication protocols: an overview. Technical Report TR-CTIT-03-49, Centre for Telematics and Information Technology,Univ. of Twente, The Netherlands, Oct. 2003.

Harris, T.J.; Seppala,C.T.; Desborough, L.D. (1999). A review of performance monitoring and assessment techniques for univariate and multivariate control systems. Journal of Process Control 9 (1999) 1-17.

Hjalmarsson, H., 2005. From experiment design to closed loop control. Automatica, 41 (2005) pp. 393-438.

ITU Recommendation, 1997. X.509 version 3. Information Technology - Open Systems Interconnection - The Directory Authentication Framework, August 1997.

Ko, C.C.; Chen B.M.; Chen, J.; Zhang, J.; Tan, K.C. (2005). A web-based laboratory on Control of two degrees of freedom helicopter. In: International Journal of Engineering, Vol 21, n 23.

Kurose, J. F.; Ross, K. W., 2006. Redes de computadores e a Internet. 3ª Edição, 2006, Editora Addison Wesley.

Kunes, M. & Sauter, T. ,2001. Fieldbus-Internet connectivity: The SNMP approach. In: IEEE Transactions on Industrial Electronics, 48(6), 1248–1256.

Lipták, B. G., 2003. Instrument Engineers' Handbook - Process Control and Optimization, 4nd Edition, 2003, Chilton Book Company.

Li Zheng and Nakagawa, H., OPC (OLE for process control) specification and its developments, SICE 2002 – Proceedings of the 41st SICE Annual Conference, pp. 917-920 vol. 2, 2002.

Loeser, J. E Haertig, H. (2004). Low-latency hard real-time communication over switched Ethernet. In: 16th Euromicro Conference on Real-Time Systems, Catania, Italy, 2004.

Luo, R.C.; Chen, T.M., 2000. Development of a multibehavior-based mobile robot for remote supervisory control through the Internet. IEEE/ASME Transactions on mechatronics, vol. 5, no. 4, December 2000.

Ljung, L., (1999), System Identification – Theory for the User. Second Edition, 1999, Englewood, Prentice Hall.

OPC Foundation, 2006. OPC Standard. In: <www.opcfoundation.org>. Access em: 26 jul. 2007.

Overstreet, J. W.; & Tzes, A. (1999). Internet-based client/server virtual instrument designs for real-time remote-access control engineering laboratory. In Proceedings of the American control conference, Vol. 2, pp. 1472–1476.

Pham, L., Gehlen, G. (2005), Realization and Performance Analysis of a SOAP Server for Móbile Devices, Proceedings of the 11th European Wireless Conference 2005, Vol. 2, Nicosia, Cyprus, April 2005, pp. 791-797.

Qin, W.& Wang, Q., 2007. An LPV approximation for admission control of an internet web server: Identification and control. In: Control Engineering Practice, Feb 2007.

Torrisi, N. M. ; Oliveira, J. F. G. (2007), Remote control of CNC machines using the CyberOPC communication system over public networks. International Journal of Advanced Manufacturing Technology, v. 2007, p. 001, 2007.

Torrisi, N. M. (2011), Monitoring Services for Industrial Applications Based on Reverse AJAX Technologies, IEEE Industrial Electronics Magazine, ISSN: 1932-4529, 2011, Volume 5, Issue 1.

Pinotti Jr, Mario ; Brandão, D. (2005). A flexible fieldbus simulation platform for distributed control systems laboratory courses.. The International Journal Of Engineering Education, Dublin, v. 21, n. 6, p. 1050-1058, 2005.

Ruel, M. (2003). The conductor directs this orchestra –Instrument Society of America. Intech, November 2003, vol 50, no 11, pp. 20-22.

Seborg, D. E., Edgar, T. F., Mellichamp, D.A., 2004 - Process Dynamics and Control, Second Edition, John Wiley & Sons, inc.

Visioli, A.(2001). Optimal tuning of PID controllers for integral and unstable processes. Control Theory and Applications, IEE Proceedings - , vol.148, no.2, pp.180-184, Mar 2001.

Yang, S.H., Zuo,X., Yang,L., 2004. Control System Design for Internet-Enabled Arm Robots. In Innovations in Applied Artificial Intelligence, Volume 3029, Springer Berlin, ISBN 978-3-540-22007-7.

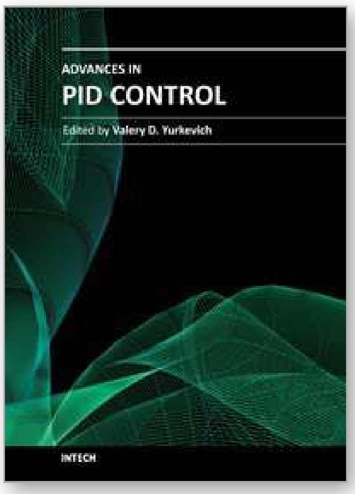
Yang, S. H., Dai, C., Knott, R.P., 2007. Remote Maintenance of Control System performance over the Internet, Control Engineering Practice, Volume 15, Issue 5, May 2007, Pages 533-544.

Yu, C. C., 2006. Autotuning of PID Controllers: A Relay Feedback Approach. 2nd Edition, Springer, 2006.

Wang, Z.P., Xiong, G.Z., Luo, J., Lai, M.Z., Zhou, W. (2000). A hard real-time communication control protocol based on the Ethernet. In Proceedings 7th Australian Conference on Parallel and Real-Time Systems, pages 161-170, Sydney, Australia, Nov. 2000. Springer-Verlag. ISBN 962-430-134-4.

Zeilmann, R.; Gomes Jr.; J.M., Bazanella; A., And Pereira, C. E. (2003). Web-based Control Experiments on a Foundation Fieldbus Plant, 5th IFAC International Conference on Fieldbus Systems and their Applications, Aveiro, Portugal, July 2003, pp. 325-330.

Zhuang, M.; Atherton, D.P. (1993). Automatic Tuning of Optimum PID controllers. In: IEEE Proceeding-D, 140, no 3, pp 216-224.



Advances in PID Control

Edited by Dr. Valery D. Yurkevich

ISBN 978-953-307-267-8
Hard cover, 274 pages
Publisher InTech
Published online 06, September, 2011
Published in print edition September, 2011

Since the foundation and up to the current state-of-the-art in control engineering, the problems of PID control steadily attract great attention of numerous researchers and remain inexhaustible source of new ideas for process of control system design and industrial applications. PID control effectiveness is usually caused by the nature of dynamical processes, conditioned that the majority of the industrial dynamical processes are well described by simple dynamic model of the first or second order. The efficacy of PID controllers vastly falls in case of complicated dynamics, nonlinearities, and varying parameters of the plant. This gives a pulse to further researches in the field of PID control. Consequently, the problems of advanced PID control system design methodologies, rules of adaptive PID control, self-tuning procedures, and particularly robustness and transient performance for nonlinear systems, still remain as the areas of the lively interests for many scientists and researchers at the present time. The recent research results presented in this book provide new ideas for improved performance of PID control applications.