

# Sistema de supervisión y adquisición de datos en un ambiente distribuido

Juan Francisco Novoa Colín, Luis Pastor Sánchez Fernández

Centro de Investigación en Computación del IPN  
Av. Juan de Dios Bátiz s/n esquina con Miguel othón de Mendizábal  
Unidad Profesional Adolfo López Mateos. México, D.F. 07738, México  
Juan Francisco Novoa Colín [jfnovoac@correo.cic.ipn.mx](mailto:jfnovoac@correo.cic.ipn.mx),  
Luis Pastor Sánchez Fernández [lsanchez@cic.ipn.mx](mailto:lsanchez@cic.ipn.mx)

**Abstract.** A project of a virtual Data Acquisition and Supervisory Distributed System (SCADA) for industrial processes, it was developed on a communication protocol for industrial networks called OPC; this was implemented in the programming language G, on the LabVIEW development environment. The system counts on several frontal panels to watch the behavior of the process variables, three types of input data, kept from data to a text or binary file, an algorithm that allows the flexibility to have so many sampling periods as process variables supervised, as well as digital filters, if necessary

**Resumen.** Se presenta un proyecto de un Sistema Distribuido de Supervisión y Adquisición de Datos (SCADA) virtual para procesos industriales, se desarrolla sobre un protocolo de comunicaciones para redes industriales llamado OPC; esta implementado en el lenguaje de programación G, en el ambiente de desarrollo LabVIEW. El sistema cuenta con varios paneles frontales para observar el comportamiento de las variables a monitorear, tres tipos de entrada de datos, guardado de datos a un archivo de texto o binario, un algoritmo que permite la flexibilidad de tener tantos periodos de muestreo como variables del o de los procesos supervisados, así como filtrado digital, en caso de ser necesario.

**Palabras clave.** SCADA, distribuido, OPC, periodo, muestreo, LabVIEW, variable, virtual.

## 1 Introducción

Un sistema SCADA permite conocer los diferentes estados de las variables que son parte de un proceso supervisado; su configuración se realiza mediante las aplicaciones HMI/SCADA, que son programas orientados a objetos y eventos con un amplio manejo de información. Al usar sistemas SCADA comerciales, lo único que se hace es configurar el sistema, sin realizar ningún tipo de desarrollo, y generalmente hay una falta de flexibilidad o un exceso de opciones; aunque por otro lado no se requiere de ningún conocimiento de programación para utilizarlos, además de tener un costo muy elevado. Por otro lado, al desarrollar un sistema SCADA en un entorno de

programación, se tiene la flexibilidad necesaria dependiendo de las opciones y los algoritmos que se requieran dentro del sistema sin tener exceso de opciones [1], [2], [7].

LabVIEW (Laboratory Virtual Instrumentation Engineering Workbench) es un ambiente de programación gráfico basado en el concepto de programación de flujo de datos. Este tipo de programación ha sido ampliamente adoptado por la industria, en ambientes universitarios y laboratorios de investigación [3], [6].

Ahora, la ventaja de utilizar un entorno de desarrollo gráfico como LabVIEW, en lugar de un ambiente de programación visual, es un ahorro considerable de tiempo, ya que, aunque en los ambientes visuales ya hay objetos predefinidos, muchas veces, es difícil configurarlos; mas cuando no se tiene un gran conocimiento en programación. En LabVIEW, debido a cuenta con numerosas herramientas como los son: algoritmos matemáticos, de procesamiento digital de señales, protocolos de comunicación, tanto para ambientes distribuidos como para comunicar con dispositivos de campo, manejo de tarjetas de adquisición de datos, así como la posibilidad de poder manejar código en C y archivos tipo DLL, entre otras características; hacen que este entorno de desarrollo sea muy potente, productivo y fácil en el desarrollo de sistemas SCADA [4].

## 2 Muestreo de señales y teorema de muestreo de Nyquist

Para usar técnicas de procesamiento digital de señales, primero se debe representar una señal analógica en forma digital. En la práctica, esto se hace con un convertidor analógico digital (A/D). Considere una señal analógica  $x(t)$  que esta muestreada cada  $\Delta t$  segundos. El intervalo de tiempo  $\Delta t$  es conocido como “intervalo o período de muestreo”. El recíproco de este intervalo,  $1/\Delta t$ , es conocido como “frecuencia de muestreo”, con las unidades de muestras/segundo ( $s^{-1}$ ). Cada uno de los valores discretos de  $x(t)$  cuando  $t=0, \Delta t, 2\Delta t, 3\Delta t, \dots$ , es conocido como muestra. Entonces la señal  $x(t)$  puede ser representada por el conjunto discreto de muestras:

$$\{x(0), x(\Delta t), x(2\Delta t), x(3\Delta t), \dots, x(k\Delta t), \dots\}$$

Otro parámetro importante es la tasa o frecuencia de muestreo con que se adquiere la señal. Una tasa de muestreo rápida adquiere más puntos en un tiempo dado y puede dar una mejor representación de la señal original. Una tasa de muestreo demasiado lenta, tiene como consecuencia una representación muy pobre de la misma señal, a este efecto se le llama *aliasing*.

De acuerdo al teorema de Nyquist, para evitar el *aliasing*, “la tasa de muestreo debe ser por lo menos el doble de la frecuencia de la componente más alta en frecuencia de la señal a muestrear”.

Para el diseño del sistema se propuso que cada variable tenga su propia tasa de muestreo, esto proporciona flexibilidad al sistema para poder adquirir señales con diversos rangos de frecuencia independientes, además de aligerar al sistema, ya que

no es necesario adquirir una muestra de todas las variables cada período de muestreo, aunque para cada señal, el operador debe elegir un período congruente que cumpla con lo dicho anteriormente y evitar el *aliasing*

### 3 Filtrado digital

El filtrado es el proceso mediante el cual el contenido de frecuencia de una señal es alterado. Los filtros alteran o remueven frecuencias que no son deseadas. Dependiendo del rango de frecuencia que los filtros dejen pasar o atenúen, se clasifican de la siguiente manera:

- Filtro Pasa Bajas. Deja pasar bajas frecuencias pero atenúa las altas frecuencias.
- Filtro Pasa Altas. Deja pasar las altas frecuencias pero atenúa las bajas frecuencias.
- Filtro Pasa Banda. Deja pasar una cierta banda de frecuencias.
- Filtro Supresor de Banda. Atenúa una cierta banda de frecuencias.
- 

La ganancia de un filtro ideal es uno (cero decibeles) en la banda de paso, o sea que la amplitud de la señal ni se incrementa ni decrecienta, y cero en la banda de paro ( $-\infty$  decibeles). Pero en la práctica siempre hay una región de transición finita entre las bandas de paso y transición, la ganancia del filtro en esta región cambia gradualmente de uno (cero decibeles) en la banda de paso, a cero ( $-\infty$  decibeles) en la región de paro [11].

En algunas aplicaciones, es aceptable que en la banda de paso la ganancia varíe muy suavemente de la unidad, a esta variación se le llama "rizo de pasa banda". "La atenuación en el supresor de banda", en la práctica, no puede ser infinita, por lo que se debe especificar un valor aceptable de supresión. Ambos, el rizo de pasa banda y la atenuación del supresor de banda, se miden en decibeles (dB), y están definidos por la ecuación 1.

$$dB = 20 \log_{10} [A_0(f)/A_i(f)] \quad (1)$$

Donde  $A_i(f)$  y  $A_0(f)$  son las amplitudes de una frecuencia en particular antes y después del filtrado, respectivamente.

### 4 Diseño del sistema SCADA

Tomando en cuenta lo anterior se hizo el diseño del sistema, en cual contempla varios instrumentos virtuales, tanto para la adquisición de datos y la parte de filtrado. Este sistema consta de los siguientes módulos de interfaz principales:

- Panel general del sistema.
- Panel de cambio de parámetros.
- Gráfico para visualización de señales.

- Módulo de recolección de datos.

Los instrumentos virtuales anteriores mandan llamar otros, necesarios para el funcionamiento del sistema.

#### 4.1 Panel general del sistema

En la figura 1 se muestra este panel, el cual cuenta solo con tres botones, el primero es para apagar el sistema, el segundo se utiliza para mandar llamar al panel de cambio de parámetros y el tercero es para mandar llamar el programa de visualización general de todo el sistema: los gráficos de visualización de señales, cambio de parámetros de filtrado, cambio de parámetros para la alarma predictiva y señalizaciones de alarma [6].



Fig. 1. Panel general del sistema SCADA.

## 4.2 Panel de cambio de parámetros

Este panel que se encuentra en la Fig. 2, muestra el panel donde se encuentran los parámetros generales, por los que se va a regir todo el sistema. El panel esta dividido en dos secciones principales.

Debido a que el número de variables que puede adquirir el sistema diseñado es en teoría ilimitado; con la única limitante del sistema de cómputo y el tipo de dato del campo (entero sin signo) que define el número de variables que el sistema manejará en total. La primera sección corresponde a las "VARIABLES CONECTADAS", aquí se define el número total de variables que manejará el sistema, para esto se tiene el control numérico de la parte superior izquierda, este caso, se muestra el número "6". En la parte inferior a este control numérico se encuentran un arreglo que contiene una serie de conectores, lo cual permite conectar o desconectar dicha o dichas variables, los campos se habilitan dependiendo del número de variables establecidos. A la derecha de estos conectores se puede definir el nombre o tag de la variable correspondiente. Por último, en la parte inferior de esta sección se puede observar un indicador que señala la última variable que se muestra en el panel, por ejemplo, si se llena el panel que se visualiza, de la variable 0 a la 11, se mostrará de la variable 12 a la 23.

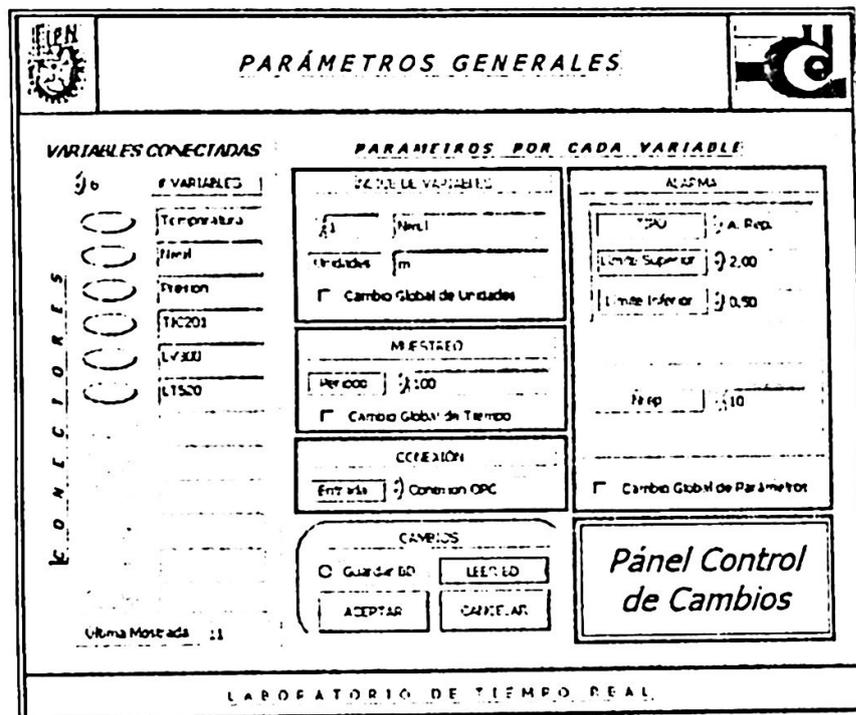


Fig. 2. Panel de cambio de parámetros.

En la segunda sección se muestran los "PARÁMETROS GENERALES POR CADA VARIABLE", y se enuncian como sigue:

#### **4.2.1 ÍNDICE DE VARIABLES**

En la sección anterior se muestran todas las variables conectadas, éste como su nombre lo dice, es un índice que muestra la variable actual, nombre y unidades que representa. Por otro lado, hay una opción que hace que las unidades sean las mismas para todas las variables.

#### **4.2.2 “MUESTREO”**

En este apartado se puede determinar el período a la que la variable que corresponde al número de índice, va a ser muestreada, esto es independiente por cada variable, pero también se tiene la opción de tener el mismo período de muestreo para todas las variables existentes en el sistema.

#### **4.2.3 “CONEXIÓN”**

Este sistema selecciona tres tipos de fuente de datos, puede ser vía conexión ethernet por OPC, mediante reproducción de datos grabados en un archivo de texto o binario, o por la generación de señales; se pueden generar 5 tipos de funciones: senoidal, exponencial de 1<sup>er</sup> orden, exponencial de 2<sup>do</sup> orden en sus tres tipos: subamortiguado, críticamente amortiguado y sobreamortiguado. La fuente más importante de datos es por red, mediante OPC; la opción de reproducción por un archivo binario o de texto sirve como datos históricos para ver sucesos importantes en un determinado tiempo. Y la opción de generación de señales es para hacer pruebas de adquisición de datos y como trabajo futuro.

#### **4.2.4 “CAMBIOS”**

Aquí se tienen tres botones y un campo de selección. Con el botón “LeerBD” se pueden leer los parámetros almacenados previamente en un archivo de texto o binario. Con el botón “Aceptar” se aceptan los cambios realizados con redundancia, ya que, el sistema pregunta si realmente se quieren hacer los cambios. Con el botón “Cancelar” se aborta la operación de cambio de parámetros, y con el campo de selección “Guardar BD” se almacenan los parámetros de los cambios realizados, después de dar click en el botón “Aceptar”; aparece un cuadro de diálogo para crear el archivo donde se van a almacenar los datos de la base, y se almacenan en un archivo de texto o binario.

Por otro lado cuando se trabaja con generación de señales, aparece un cuadro de diálogo para seleccionar el tipo de función que se va a generar. Esto se hace por cada variable.

También, se tienen varios subVI adicionales que permiten guardar en un archivo binario o de texto todos los parámetros contenidos en la base de datos, así como, poder cargar esos mismos datos guardados.

#### 4.2.5 “ALARMA”

Con los algoritmos de alarma se pueden detectar cuando las variables de un proceso han tomado valores extremos que pueden resultar peligrosos o que deben ser señalizados pues resultan importantes en el funcionamiento del sistema.

En un proceso tecnológico las posibles tendencias que pueden derivarse a partir del estado inicial de la variable y llegar a un límite superior o inferior de alarma (LSA, LIA respectivamente). Por arriba del LSA la región se denomina GAMA ALTA; por abajo del LIA la región se denomina GAMA BAJA, y a la región entre estos dos límites se le nombra GAMA NORMAL. Y a su vez, la variable puede tomar una tendencia con una pendiente positiva, negativa o tener pendiente cero.

Se implementan tres tipos de alarma: por límite simple, por límite con repeticiones y por límite con histéresis.

#### 4.2.6 Algoritmo para obtener el máximo común divisor (MCD)

Como se comentó anteriormente, para evitar que el sistema se cargue innecesariamente, cada variable puede tener un período de muestreo diferente, por lo que se tiene que tener un criterio para saber cuando tiene que ser leída cada variable; es por esto que se necesita obtener el máximo común divisor (MCD), y una vez que se han hecho los cambios pertinentes en el Panel de Cambios y se da click en el botón “ACEPTAR”, se calcula el MCD del arreglo donde se encuentran los períodos de muestreo de todas las variables. Una vez que se obtiene, se almacena en una base de datos global para que pueda ser utilizado por el módulo de recolección de datos. El algoritmo que se sigue para obtener el MCD es el llamado algoritmo de Euclides, el cual consiste en hacer divisiones sucesivas como se enuncia en las ecuaciones (2) a (7).

Sean **a** y **b** los números de los que se quiere calcular el MCD. Se hacen las siguientes divisiones hasta que el residuo sea cero:

$$\begin{array}{r} q \\ b \overline{)a} \\ r \end{array} \quad (2)$$

$$a = bq_1 + r_1 \quad (3)$$

$$b = r_1q_2 + r_2 \quad (4)$$

$$r_1 = r_2q_3 + r_3 \quad (5)$$

$$r_2 = r_3q_4 + r_4 \quad (6)$$

$$r_{n-1} = r_n q_{n+1} + 0 \quad (7)$$

Donde  $q_1, q_2, \dots$  son los cocientes y  $r_1, r_2, \dots$  son los residuos. El MCD es  $r_n$ .

Para calcular el MCD de varios números  $(a_1, a_2, \dots, a_n)$  con el algoritmo de Euclides se procede de la siguiente forma: se calcula el MCD de los números  $a_1$  y  $a_2$ . Sea este número  $M_1$ . Después calculamos el MCD de los números  $M_1$  y  $a_3$ . Sea este número  $M_2$ . Repetimos el proceso hasta el último elemento. El MCD de los números  $M_{n-1}, a_n$ , será el MCD de  $a_1, a_2, \dots, a_n$ .

#### 4.2.7 Gráfico para visualización de señales

En este gráfico se puede apreciar el comportamiento de las variables que se están adquiriendo, la apariencia del diseño asemeja a un osciloscopio.

El control numérico con el nombre de “Índice” permite cambiar la variable que se esta visualizando. El siguiente control numérico permite variar la velocidad de visualización en el gráfico, este toma por nombre “Actualización”. Y por último el botón que dice “Parar” detiene el modulo de visualización de variables.

Se encuentra otro control en forma de pestañas con los nombres “Rango” y “Generación”. En “Rango”, que son los controles parecidos a las perillas de un osciloscopio, se ajustan las escalas de los ejes X y Y del gráfico. El botón “Guardar” da la posibilidad de almacenar los datos adquiridos en un archivo binario o de texto. Y por último se tienen los gráficos que muestra el comportamiento de la variable, en la Fig. 3 se muestra la opción de filtrado de señales, se usan dos tipos de filtro digital, un IIR y uno de Media con todos sus parametros. Dentro del gráfico se pueden comparara las dos señales, la filtrada y la no filtrada, y también la respuesta del filtro, tanto en magnitud como en fase en el dominio de la frecuencia.

La pestaña de “Generación” se usa solo cuando se ha escogido la opción de generar las señales a visualizar y no se están adquiriendo; las funciones que se pueden generar son: senoidal, exponencial de 1er orden, exponencial de 2<sup>do</sup> orden en sus tres casos principales: subamortiguado, críticamente amortiguado y sobreamortiguado.

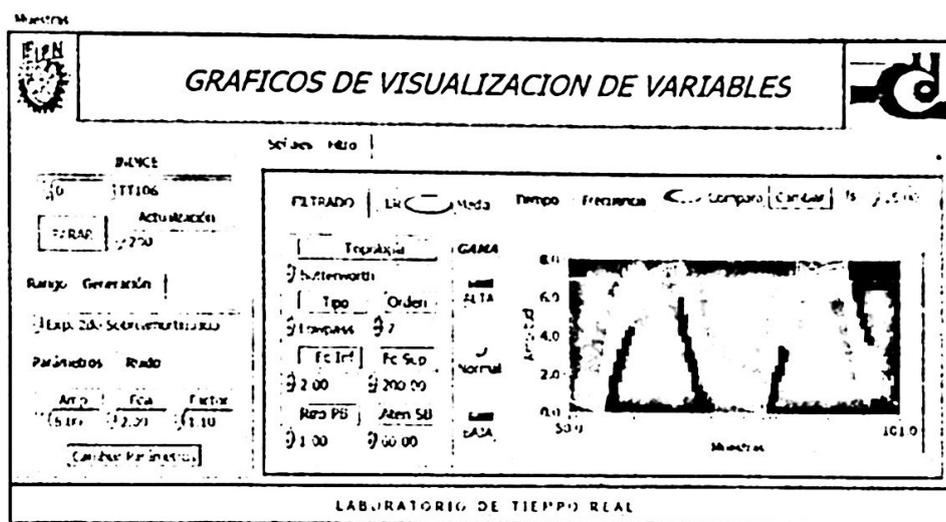


Fig. 3. Gráficos de visualización de variables con opción de filtrado.

Por otro lado también se pueden guardar las muestras que se van adquiriendo en un archivo binario o de texto, y a su vez, se pueden cargar los datos guardados previamente.

#### 4.2.8 Recolección de datos

Este módulo se puede decir que es el más importante del sistema, ya que es el que se encarga de leer los datos, ya sea, por una conexión OPC, por reproducción de datos de un archivo binario o de texto, o por la generación de los mismos.

El funcionamiento de este módulo consiste en obtener las muestras de las variables conectadas en los períodos de tiempo especificados; como ya se había comentado, este sistema no requiere que la frecuencia de muestreo sea igual para todas las variables, por lo que se requiere de un valor que sea común para todas las frecuencias de muestreo, este valor es el MCD (máximo común divisor) que se calculó en el módulo del panel de cambios. Lo que hace el algoritmo, es obtener el índice de las variables que se van a leer en ese instante, y esto se hace restando el MCD del período de cada variable constantemente, cada que llega a cero es cuando se manda a leer dicha variable. Por último, una vez que se adquiere el dato solo resta pasarlo por un filtro digital, en caso de ser necesario.

Cabe señalar que en todos los accesos a la base de datos global se usaron semáforos para evitar que en algún momento se pierdan datos por conflictos de sincronización.

#### 4.3 Protocolo de comunicaciones para redes industriales OPC

“OLE para control de procesos” (OPC) está diseñado para permitir a aplicaciones cliente tener acceso a los datos del piso de una planta de una manera consistente. Con la amplia aceptación de la industria, OPC dará varios beneficios [9], [10], [12]:

- Fabricantes de hardware sólo tienen que hacer un conjunto de componentes de software, para sus compradores y utilizarlos en sus aplicaciones.

- Los diseñadores de software no tienen que reescribir drivers a partir de cambios de características y adiciones a nuevo hardware.
- Compradores tendrán más opciones para desarrollar sistemas de manufactura integrados.

#### **4.3.1 Condiciones necesarias para la puesta en marcha del modelo cliente-servidor mediante OPC**

Para que un modelo cliente servidor OPC pueda funcionar se tienen que tomar ciertas medidas, las que se enumeran a continuación [8]:

- Asegurarse que tanto las máquinas cliente como los servidores estén bajo un mismo dominio.
- Configurar adecuadamente de la tecnología DCOM en los servidores como en máquinas cliente.
- Configuración del acceso al registro de windows en caso de tener sistemas operativos windows 2000 o XP, o instalación del acceso al registro de windows para los sistemas operativos windows 95/98/ME..

#### **4.3.2 DataSocket e implementación de la comunicación**

Dentro del modelo OSI, la capa 6 o de transporte, no solo es una capa más. Es el corazón de la jerarquía completa de protocolos, la tarea de esta capa es proporcionar un transporte de datos confiable y económico de la máquina de origen a la máquina destino, independientemente de la red o redes físicas en uso. DataSocket funciona en esta capa, y está construido usando tecnología TCP/IP y activeX/COM [6].

En el caso de este proyecto, como la conexión OPC se basa en la operación de DataSockets. Dentro de LabVIEW ya existen subVI's prediseñados para utilizar estas herramientas.

Estos VI's abren o cierran conexiones. La función <<Open>>, abrirá la comunicación; que está basada en una entrada de URL. Esta entrada puede tener los protocolos HTTP, FTP, DSTP u OPC, una vez que esta URL entra, esta función es una referencia DataSocket, la cual se usa de la misma manera que una conexión refnum. Los demás VI's usan esta referencia para realizar otras operaciones.

Para leer datos del servidor, es necesario determinar si el servidor es local o remoto. Una vez que el nombre del servidor está resuelto y se tiene la conexión abierta con el atributo de leer, se necesitará usar el VI <<Update>> si se quieren leer nuevos datos.

## **5 Conclusiones**

1. La supervisión de procesos mediante técnicas digitales es una tecnología muy difundida actualmente.

2. El técnica de adquisición de datos que permite tener un periodo de muestreo diferente para cada variable, permite al sistema ser flexible y robusto.
3. En la actualidad en el control industrial el uso de redes de comunicaciones se han vuelto una parte fundamental para su desempeño, es por eso que se han estandarizado un gran número de protocolos de redes industriales basados en TCP/IP

## Referencias

1. Jacob, M. (1993). *Industrial Control Electronics*. USA, New Jersey, Prentice-Hall International, Inc., 584p.
2. Johnson, C. (1993). *Process Control Instrumentation Technology*. Fourth Edition, USA, New Jersey, Prentice Hall Inc., 592p.
3. Franklin, G.F., J.D. Powell and M.L. Workman, (1990.). *Digital Control of Dynamic Systems*, USA, New Jersey, Prentice-Hall.
4. Aström, K.J., and B. Wittenmark. (1984). *Computer Controlled System, Theory and design*. USA, New Jersey, Prentice Hall Inc., Englewood Cliffs.
5. PACHECO Xavier; *Borland DELPHI 4 Developer's Guide*; Ed. SAMS; 1998; p.p. 653-659.
6. National Instruments; *LabVIEW QuickStart Guide*; 1999.
7. SAUCEDO Flores S; *Sistemas de Control Distribuido*; Escuela Superior de Ingeniería Mecánica y Eléctrica IPN; 1998.
8. SIMMONS Curt; *Guía Avanzada Microsoft Windows Server Configuración*; Ed. Prentice Hall Iberia; Madrid; 2000; p.p. 149-162.
9. NI Developer Zone; *General DCOM Configurations for Windows 95/98/Me*.
10. NI Developer Zone; *Setting Proper Access for Remote OPC Server When Using National Instruments OPC Clients*
11. OPPENHEIM A.V, R.W. Schafer y J.R Buck, *Tratamiento de señales en tiempo discreto*; 2ª edición; Ed. Prentice-Hall, Inc., Madrid, España, 2000, pp. 873.
12. OPC Technical Overview. Parts of this OPC Overview were extracted from a special OPC Supplement of Startmagazin published in their January/February 1998. The entire OPC Supplement can be viewed at: <http://www.startmagazinc.com/v2n1opc0.html>

