

Aplicación cliente-servidor MatLab[®] -WinCC[®] empleando comunicación OPC

D. Sabin, A.I. González

Instituto Superior Politécnico José Antonio Echeverría, Departamento de Automática y Computación, Ciudad de La Habana, Cuba

RESUMEN / ABSTRACT

Se presenta una aplicación desarrollada con el sistema SCADA WinCC[®]. La aplicación permite el control proporcional integral de una red RC de primer orden, la identificación de un motor de corriente continua para la detección de fallos y el análisis de sensibilidad de las condiciones iniciales en un reactor semi-batch. La principal ventaja de la aplicación demostrativa realizada es mostrar las potencialidades que pueden adquirir las aplicaciones SCADAs, implantadas actualmente en la industria, cuando interactúan a través de la comunicación OPC con otras aplicaciones específicas de mayor potencial en materia de cálculo numérico como las desarrolladas en el Simulink de MatLab[®].

Palabras claves: Comunicación OPC, control, detección de fallos, identificación, MatLab[®], WinCC[®]

An application developed with SCADA WinCC[®] system is presented in this paper. This application allows the proportional integral control of a first order RC network, the identification of a DC engine for failure detection and a fed-batch reactor initial conditions sensibility analysis. The main advantage of implemented application is to show the potentialities of the SCADAs applications that may be implemented in the industry at present, when interacting through the OPC communication with other specified applications with a higher potential in numerical calculation as Simulink-MatLab programs.

Key words: Control, failure detection, identification, MatLab[®], OPC communication, WinCC[®]

INTRODUCCIÓN

La industria de procesos demanda soluciones rápidas y eficaces para el trabajo con procesos multivariables y altamente no lineales desde hace varias décadas. En el entorno industrial actual existen ya implantados sistemas de monitorización y control de procesos, comúnmente reconocidos como SCADAs (*Supervisory Control and Data Acquisition*)¹.

Los SCADAs son aplicaciones informáticas que gestionan los procesos desde los niveles superiores a través de una computadora personal (PC) o una red de computadoras. En la actualidad, estos programas informáticos no llegan a ser totalmente sistemas de supervisión pues sólo implementan una etapa primaria de la detección de fallos a través del tratamiento de las alarmas de procesos y prescinden de las etapas de diagnóstico del fallo y de reconfiguración del sistema.

En muchas ocasiones dichos sistemas informáticos se operan como simples sistemas de monitoreo y no se explotan en toda su potencialidad. Estas aplicaciones *software* pueden llegar a influir en la operación de los procesos, propiciando soluciones de alta significación económica y donde se garantiza la productividad con calidad, seguridad y fiabilidad en las operaciones.

Las aplicaciones SCADAs requieren de la adecuada configuración de las comunicaciones digitales para una implantación exitosa.

El surgimiento, desarrollo y algunos aspectos de interés del estándar de comunicación *OLE for Process Control (OPC)*²⁻⁶ son necesarios para comprender el motivo de su estudio y el por qué es utilizado en el presente trabajo como protocolo para la comunicación.

Microsoft junto a un grupo de líderes en automatización, hardware y software del mundo se propusieron como meta crear la fundación OPC. Actualmente la fundación posee más de 300 miembros de todo el mundo donde se encuentran los mayores proveedores de instrumentación y sistemas de control de procesos.

La fundación OPC creó un protocolo que se soporta en las tecnologías *OLE*, *COM*, *DCOM* y *ActiveX* confeccionadas por *Microsoft*. Con el establecimiento del primer estándar OPC se constató que a través de la comunicación con otros tipos de datos se podían lograr estandarizaciones. Por ello se generaron estándares para las alarmas, eventos, datos históricos, procesos por lotes, entre otros.

Además de los estándares OPC obtenidos se continuó trabajando para concebir un sólo estándar. En Junio de 2006 surge la Unified Architecture (UA) y de esta forma se crea el

OLE for Process Control-Unified Architecture (OPC-UA)^{7,8}. Con este nuevo grupo de especificaciones se adquiere una interoperabilidad global, lo que posibilita la interrelación eficiente entre aplicaciones, lenguajes de programación y hasta de sistemas operativos.

Hoy día se pueden observar gran cantidad de aplicaciones donde se emplea la comunicación OPC⁹⁻¹³. En procesos de la industria del petróleo, en la generación de energía eléctrica y hasta en tecnologías de alta precisión ya existen reportadas soluciones que emplean este estándar. Algunas notas de aplicación de proyectos dirigidos por la empresa alemana Softing¹⁴, líder en automatización industrial y miembro de la fundación OPC, pueden verse en¹⁵⁻²¹.

En investigaciones previas² a este trabajo se instalaron y analizaron un grupo significativo de entornos comerciales de desarrollo de aplicaciones SCADAs. WinCC[®] v5.1²²⁻²⁵, InTouch[®] v8.0^{26,27}, Citect[®] v6.0²⁸, Wizcon[®] v7.6²⁹ y Lookout[®] v3.8³⁰, constituyeron los entornos de desarrollo analizados. A raíz de esa investigación se arribó a la siguiente conclusión:

- *Los protocolos propietarios han cedido terreno a nuevos estándares de comunicaciones no propietarios. OPC se ha convertido en un estándar incorporado por los SCADAs que permite el establecimiento de la comunicación entre dispositivos y aplicaciones. Todos los SCADAs estudiados presentaban la posibilidad de comunicación OPC.*

Estas herramientas de desarrollo de sistemas de monitorización y supervisión constan de módulos^{9,31} que agilizan la realización de aplicaciones pero *no vienen en su mayoría, respaldadas por herramientas de cálculo numérico (librerías de optimización, integración numérica, etc.) que contribuyan a formalizar y sintonizar con rapidez controladores avanzados, diagnosticadores de fallos, etc.*

La importancia del resultado de investigación que aquí se expone radica en lograr dentro del desarrollo de SCADAs actuales un mecanismo de comunicación con otras herramientas existentes que faciliten la implementación de aplicaciones en los entornos industriales para el control avanzado, la identificación y la DDF (Detección y Diagnóstico de Fallos).

El diseño de una aplicación demostrativa soportada en el entorno de desarrollo comercial WinCC[®] se lleva a cabo como una alternativa de solución. La aplicación emplea la comunicación OPC con programas realizados en el lenguaje de bloques Simulink de MatLab[®] y potencia el desarrollo del sistema de supervisión. Se debe recordar que la supervisión va más allá de una mera monitorización de las variables de proceso.

El trabajo se encuentra organizado en cinco secciones fundamentales. La primera sección aborda la aplicación principal SCADA. Las tres secciones siguientes centran su atención en los procesos estudiados y en las aplicaciones logradas con ellos. Además, existe una última sección destinada a las conclusiones generales del trabajo.

APLICACIÓN PRINCIPAL SCADA

Es una generalidad que las aplicaciones SCADAs posean una ventana principal, desde la cual se tiene acceso de forma más

específica a cada uno de los procesos o subprocesos que se vigilan y controlan en la industria en cuestión. A través del establecimiento de usuarios independientes con contraseñas, estos programas informáticos configuran y establecen los privilegios del personal que trabaja con estos sistemas en la industria.

En la figura 1 se muestra la ventana principal de la aplicación desarrollada desde donde se accede a los tres procesos estudiados a través del botón de etiqueta "Entrar" correspondiente. A través del botón con la etiqueta "Salir" se abandona la aplicación.

Desde el punto de vista gráfico la ventana es sencilla ya que además de los botones mencionados contiene algunos textos estáticos y las tres imágenes que representan a los procesos de interés.

Otro aspecto importante a destacar es que el programa es una aplicación cliente-servidor que corre sobre el sistema operativo Windows, un sistema operativo que no es de tiempo real. Esto implica que no se puede garantizar una respuesta en un tiempo prefijado. Sin embargo, si se trabaja con un tiempo de intercambio mayor y alejado del tiempo de comunicación, se obtienen resultados satisfactorios. Hay que precisar también que el tiempo de comunicación varía en dependencia de las características de hardware de la PC en la que se esté ejecutando el programa.

Desde la aplicación se realizaron varias pruebas para determinar el tiempo de comunicación. En general, el promedio de comunicación entre cliente y servidor, incluyendo los dos sentidos, fue de 91 milisegundos. Considerando que existe un pequeño código que demora el resultado y que la comunicación se realiza en los dos sentidos, se valora este tiempo como un tiempo adecuado para el intercambio de información relacionada con el control, debido a que son pocos los procesos que necesitan una respuesta tan rápida.

En las siguientes secciones se explican las principales características del resto de las ventanas de la aplicación que permiten el acceso a los tres procesos.

VENTANA SISTEMA RED RC

Modelo del proceso

La red RC es uno de los procesos a los que se accede desde la aplicación principal. Es un sistema eléctrico muy común que se encuentra compuesto por una resistencia en serie con un capacitor. Como se observa en el esquema de la figura 2 se trata de un sistema muy sencillo donde se obtiene su función de transferencia $G(s)$ como muestra la ecuación (1)³².

$$G(s) = \frac{Vc(s)}{Vin(s)} = \frac{1}{RCs + 1} \quad (1)$$

donde: $G(s)$ es la función en el dominio de Laplace

Vin es el voltaje a la entrada,

Vc es el voltaje de salida,

R es la resistencia y

C es la capacitancia.

Para la red RC se asumen los valores de capacitancia y resistencia iguales a 100 μf y 100K Ω respectivamente y se parametriza la función transferencial como expresa la ecuación (2).

$$G(s) = \frac{1}{10s + 1} \quad (2)$$

donde: $G(s)$ es la función en el dominio de Laplace y s es el operador de la transformada.

El sistema que representa la ecuación (2) está escrito en forma de función de transferencia en el dominio de Laplace, considerando a la red RC como un sistema lineal e invariante en el tiempo. En este caso la variable controlada es el voltaje en el capacitor V_c y la variable manipulada es el voltaje de entrada V_{in} .

La función de transferencia en el dominio continuo (2) se discretiza con un tiempo de muestreo de 1 segundo con la ayuda del MatLab[®]. La función de transferencia del proceso en transformada Z es reflejada en la ecuación (3).

$$G(z) = \frac{0.09516}{z - 0.9048} \quad (3)$$

donde: $G(z)$ es la función en el dominio de Z y z es el operador de la transformada.

Aplicación SCADA para la red RC

La comunicación OPC y las variables involucradas para el control de la red RC desde la aplicación WinCC[®] se presentan de forma gráfica en la figura 3.

El voltaje del capacitor V_c es la variable medida en el esquema de comunicación y por ende es leída desde el cliente OPC, es decir, desde el programa Simulink del MatLab[®]. En este caso el controlador sintonizado calcula una salida que se corresponde con el voltaje de entrada a la red V_{in} . Este valor es enviado a la aplicación WinCC[®] para que lo aplique al sistema "real". De esta forma se cierra el lazo de control donde el controlador se encuentra en el programa MatLab[®] tomando las decisiones para que el sistema se regule correctamente.

En la figura 4 se indica la ventana de la aplicación WinCC[®] para el caso de estudio de la red RC. En dicha ventana se muestra el voltaje del capacitor cuando se le aplica a la entrada de la red una señal paso escalón unitario a lazo abierto. También se muestra el esquema del circuito RC, el gráfico de la variable controlada (voltaje del capacitor), los valores de la referencia, el tiempo, el voltaje de entrada y botones que permiten el intercambio con las demás ventanas de procesos.

Para lograr la conexión OPC es necesario en el entorno de bloques Simulink³³ establecer la configuración de varios bloques del *ToolBox OPC*. Entre estos bloques se encuentran el bloque de configuración donde se establece el servidor a conectarse, dos bloques de lectura para leer la variable de salida y la referencia y el bloque de escritura para el valor del voltaje de entrada de la red. Todos los bloques se configuran con un tiempo de muestreo de 1 segundo.

Resultados de las simulaciones del control del sistema red RC

Para la red RC se sintonizó un controlador discreto de tipo Proporcional-Integral (PI). Con este fin, se aplicaron técnicas de ajuste en lazo abierto^{34,35}, utilizando las típicas tablas de Ziegler y Nichols. Los valores ajustados del regulador PI fueron una ganancia $K=10$ y un tiempo integral $T_i=2$.

Como se trabaja en una PC, se discretiza el controlador PI sintonizado y se obtiene la función de transferencia en Z indicada en la ecuación (4).

$$G_c(z) = \frac{10z - 8}{z - 1} \quad (4)$$

donde: $G_c(z)$ es la función en el dominio de Z del controlador y z es el operador de la transformada.

De esta forma se obtuvo un controlador eficiente para establecer el ajuste del proceso. En la figura 4 se muestra desde la aplicación WinCC[®] la salida del sistema en lazo cerrado. Al sistema se le aplicó a la entrada una secuencia de paso escalón de amplitud 1-3-1-2 que el sistema controlado siguió adecuadamente.

VENTANA SISTEMA MOTOR DE CC

Modelo del proceso

La aplicación con el motor de corriente continua (CC) fue desarrollada empleando un motor real. Dicho motor CC se encuentra instalado en el laboratorio del Instituto de Cibernética Matemática y Física (ICIMAF), perteneciente al CITMA de Cuba^{36,37}.

El proceso real lleva asociado como elemento de medición un tacómetro con el cual se puede obtener el valor de la velocidad que alcanza el eje del motor. La señal de alimentación es de corriente continua y puede ser modificada variando el ancho del pulso.

Para este proceso se realizó un proceso de identificación^{38,39,40} tomando en consideración el voltaje de entrada al motor como variable de entrada y la velocidad de rotación que alcanza el motor como variable de salida.

El motor es del modelo PIVT 6-25/3A de la firma búlgara Dynamo Sliven. El elemento de medición es un tacogenerador, también de la firma Dynamo Sliven, incorporado dentro del motor.

Los valores para la identificación del modelo del proceso se obtuvieron a través de experimentos en la planta real y se almacenaron en ficheros para un tratamiento posterior.

Se manipularon un total de 300 valores en la investigación realizada. Para estimular al proceso se trabajó con una señal PRBS de ciclo útil (CU) entre 10 y 90. La salida obtenida es la velocidad a la que gira el motor. Esta salida en lazo abierto, en correspondencia con la entrada, es también una señal PRBS.

Aplicación SCADA del motor de CC

En este caso se establece una comunicación entre la aplicación WinCC[®] y el programa desarrollado en MatLab[®] con el

objetivo de realizar una comparación de valores reales del motor con los valores obtenidos a partir de la resolución del modelo que se identifica para el propio motor. En un primer paso, se lleva a cabo una identificación del proceso de la que se comentará en el siguiente epígrafe.

La ventana correspondiente a este sistema se presenta en la figura 6 y contiene una foto real del motor de CC, un gráfico de la velocidad real y la velocidad estimada por el modelo y botones que entrelazan las aplicaciones.

El gráfico permite al operador detectar por inspección visual cualquier mal funcionamiento del sistema.

El programa desarrollado en el lenguaje de bloques Simulink del MatLab® posee el característico bloque de configuración de la conexión OPC. Además tiene un bloque de lectura, que lee el voltaje de entrada al motor desde WinCC®, y tiene un bloque de escritura gracias al cual se exporta al WinCC® la velocidad estimada.

Resultados de la identificación del motor de CC

La identificación realizada para el motor CC obtuvo modelos con estructura ARX y ARMAX utilizando el *Toolbox Identi*³⁸ del MatLab®. A través de esta herramienta se obtuvieron varios modelos³⁹ después de realizar la metodología correspondiente. Se acometieron análisis de correlación y de aproximación a los valores reales según criterios del propio *toolbox* para cada modelo.

La ecuación (5) ejemplifica una de las estructuras obtenidas del modelo, la ARX:

$$A(q)Y(t) = B(q)u(t) + e(t) \quad (5)$$

donde: $A(q)$ es el polinomio de salida,

$B(q)$ es el polinomio de entrada,

$Y(t)$ es la variable de salida del motor,

$u(t)$ es la variable de entrada del motor y

$e(t)$ es la variable de error.

Los valores obtenidos de los órdenes de los polinomios $A(q)$ y $B(q)$ fueron $n_a=4$, $n_b=4$ y $n_k=1$, es decir, un modelo ARX441. En las ecuaciones (6) y (7) se señalan los valores numéricos determinados en la identificación.

$$A(q) = 1 - 0.5301q^{-1} - 0.2896q^{-2} - 0.179q^{-3} + 0.00815q^{-4} \quad (6)$$

$$B(q) = 7.435q^{-1} - 3.465q^{-2} - 2.578q^{-3} - 1.316q^{-4} \quad (7)$$

En esta aplicación se simula la obtención de los datos ya que otros valores de entrada y salida se introdujeron en la aplicación WinCC® mostrándolos en pantalla cada 1 segundo como si se estuviera obteniendo de forma real.

En la figura 6 se muestra el gráfico de la aplicación WinCC® revelando los valores reales derivados del propio WinCC® y los valores producidos después de aplicársele la misma entrada al modelo ARX441 que se encuentra implementado en el MatLab®.

El operario puede percibir la no correspondencia entre el modelo y los valores reales y por consiguiente puede detectar un fallo antes de que se produzcan consecuencias grandes.

Este simple ejemplo es el primer paso para el DDF pudiéndose trazar estrategias para lograr una supervisión completa.

De forma general valiéndose de la herramienta del MatLab® Opctool^{41,42} se puede instaurar una comunicación con el servidor OPC que esté activo. A partir de este momento los valores deseados del proceso real pueden ser adquiridos y se pueden realizar estudios, identificaciones, simulaciones y modelaciones. Con lo anteriormente expuesto pudieran obtenerse modelos en línea para la aplicación de control avanzado como el control adaptativo.

El modelo ARX referido para el motor de corriente directa resultó aproximado al proceso real. Se ajustó también un controlador PI para el proceso. No resulta necesario aplicarle un control derivativo al proceso por la oscilación nula que presenta.

VENTANA SISTEMA REACTOR FED-BATCH

Modelo del proceso

En la revisión bibliográfica se detectó un resultado¹² previo que presenta similitudes con respecto a la propuesta que en este trabajo se plantea. En dicha investigación se simula el control de una planta piloto en un laboratorio. En dicho trabajo se emplea la comunicación OPC para el conocimiento a priori de los valores de las variables más importantes.

En este trabajo se aborda específicamente otro proceso que es un reactor semi-batch en el que se desarrolla un proceso de obtención del ácido maleico. La fase de preparación del reactor se inicia a partir de un reactivo denominado en las ecuaciones como **(a)** con un volumen inicial *V_{ro}*. Una vez que la temperatura del reactivo alcanza los 328K se pasa a la fase de operación y el reactor comienza a alimentarse con un flujo de otro reactivo denominado como **(b)**. El flujo del componente **(b)** se mantiene constante hasta concluir con una alimentación de 1.2 Kmol de **(b)** por cada 1 Kmol de reactivo **(a)** que existía en un inicio. Como resultado final el componente **(a)** se va convirtiendo en **(b)**.

En este proceso existe un compromiso entre la velocidad de reacción y la temperatura máxima pues a mayor velocidad de alimentación aumenta más la temperatura. La temperatura no debe sobrepasar los 373 K por un problema de seguridad y también para que se logre la sustancia final esperada.

Un valor importante de las variables es el tiempo requerido para duplicar el volumen originalmente presente y denominado como **(A)**. Las simulaciones realizadas⁴³ son para distintos valores del tiempo **(A)**. Otro valor peculiar es la velocidad de reacción **(k)** que depende exponencialmente de la temperatura de la reacción. En el presente trabajo la velocidad de la reacción se consideró constante por su influencia mínima en la dinámica.

Las ecuaciones correspondientes⁴³ a la síntesis del ácido maleico mono-éster en un reactor semi-batch expresan las variaciones de las concentraciones de los elementos **(a)** y **(b)** y la variación de la temperatura, factor muy importante en estos procesos. Las expresiones se plantean en las ecuaciones (8), (9) y (10).

$$\frac{dCa}{dt} = -\frac{Ca}{A+t} - kCaCb \quad (8)$$

$$\frac{dCb}{dt} = \frac{CbI - Cb}{A+t} - kCaCb \quad (9)$$

$$\frac{dT}{dt} = BkCaCb - \frac{T - Tb}{A+t} - C(T - 328) \quad (10)$$

La descripción de las constantes y de las condiciones iniciales se indica en la tabla 1. En este caso el objetivo es lograr la resolución de las ecuaciones diferenciales que representan al modelo del proceso.

El control clásico para estos sistemas se hace muy difícil por la alta no linealidad del sistema y la gran cantidad de variables de entrada y salida. El proceso se puede regular con lazos independientes utilizando control clásico pero para algunas de las variables de estos sistemas no es eficiente.

El control que se le aplica a estos sistemas es avanzado. En estos sistemas suele aplicarse un control adaptativo debido a las características variantes del proceso. En este regulador adaptativo se cambia la ley de control de acuerdo a la evolución del proceso. De igual forma, podemos encontrar variantes del control predictivo como el caso del Predictor de Smith. Todas estas estrategias requieren de la resolución del modelo del proceso.

Aplicación SCADA del reactor semi-batch

En este caso las condiciones iniciales de la fase de operación del reactor son enviadas, mediante el protocolo OPC, al programa MatLab® para que este resuelva las ecuaciones diferenciales con las herramientas que cuenta. Una vez que esta aplicación cliente calcula los valores de las concentraciones de sustancias (**a**), (**b**) y la temperatura (**T**) se le envían dichos resultados a la aplicación WinCC® y se muestran en un gráfico. El esquema que ejemplifica el intercambio de datos se muestra en la figura 7.

Este subsistema de la aplicación es el más complejo de los procesos implementados y cuenta por ello con dos ventanas. Uno de las ventanas muestra los valores de las variables y el otro lo modela en forma gráfica. La ventana de la figura 8 muestra un esquema del reactor, las válvulas, el autómatas y otros elementos importantes del sistema como se representan en la realidad así como los valores de las condiciones iniciales, las variables y los botones de intercambio con los otros procesos.

Resultados con el reactor semi-batch

En la figura 9 se muestran los resultados de la simulación de una conexión activa entre el programa MatLab® y la aplicación WinCC®. Los resultados obtenidos son resultados lógicos y de acuerdo a lo que debe pasar en la reacción. Como se aprecia en el primer gráfico, la concentración del componente (**a**) disminuye pues se está transformando en el componente (**b**). La concentración del componente (**b**) aumenta como se indica en el segundo gráfico y la temperatura de la reacción aumenta y luego disminuye como se constata en el tercer gráfico.

A través de esta aplicación se puede analizar la sensibilidad de las variables del proceso ante variabilidad o perturbaciones en los valores de las condiciones iniciales. En este tipo de proceso una simple desviación de las condiciones iniciales, necesarias para comenzar la fase de operación, puede ocasionar resultados inesperados en cuanto a productos finales no deseados.

CONCLUSIONES

En el trabajo se presenta una aplicación demostrativa desarrollada con el entorno WinCC® a través de la cual se interacciona vía el protocolo de comunicación OPC con tres procesos de diferentes características. Dicha aplicación es sencilla y simula los entornos informáticos reales industriales donde a partir de una ventana principal se monitorea el comportamiento de diferentes subprocesos o subsistemas.

Lo importante de la aplicación mostrada es la potencialidad que se le incorpora a la misma a partir de su intercambio con otros programas desarrollados en el lenguaje en bloques Simulink de MatLab®.

Aunque en cada una de las secciones se fueron revelando los resultados del trabajo, a continuación se enuncian las conclusiones generales a las que se arribó:

- De forma general se demuestra que el protocolo OPC puede ser utilizado como soporte de aplicaciones para el control, la identificación y el diagnóstico de fallos. Las aplicaciones diseñadas garantizan la comunicación estable entre las aplicaciones de WinCC® y el MatLab®.
- En el caso de estudio del sistema de primer orden (Red RC) se logra desde el MatLab®, mediante la comunicación OPC, el control PI del proceso. En este caso la aplicación en MatLab® hace función de controlador, la cual recibe la salida del proceso y devuelve el valor adecuado a la entrada del proceso.
- El caso del motor de CC demuestra que la comunicación OPC puede ser utilizada para la detección de fallos. El ejemplo trabaja sólo la etapa de la detección del fallo, sin embargo, se pueden realizar las otras dos etapas de la supervisión apoyándose en el MatLab®.
- El reactor semi-batch es el proceso de mayor complejidad estudiado por su característica no lineal y su difícil control. Se demuestra con la comunicación OPC lograda que se pueden resolver sistemas de ecuaciones no lineales en herramientas adicionales al WinCC®. Con ello, se sientan las bases para implementar técnicas de control avanzado y potenciar el control de procesos no lineales.
- La aplicación se soporta en un sistema operativo que no es de tiempo real y, por tanto, no se garantizan soluciones en un tiempo prefijado. Los procesos que requieran respuestas rápidas deben ser analizados detalladamente para aplicar esta técnica.

Las futuras investigaciones estarán dirigidas a aplicar esta técnica a sistemas reales donde se pueda influir directamente en los procesos industriales desde los niveles superiores de la

pirámide de control. En estos niveles altos de la pirámide, se podrán brindar otras soluciones con significación económica.

RECONOCIMIENTOS

Este trabajo no hubiera sido posible sin la ayuda de los investigadores del ICIMAF que dieron su aporte, así como a los investigadores y profesores del departamento de Automática del ISPJAE (Instituto Superior Politécnico José Antonio Echeverría).

REFERENCIAS

1. **DANEELS A., SALTER W.:** "What is a SCADA?" presented in the *International Conference On accelerator and Large Experimental Physics control systems*, Trieste, Italy, 1999.
2. **SABIN D.** "Los entornos comerciales actuales de desarrollo de aplicaciones SCADAs: tendencias, bondades y limitaciones", *Trabajo de diploma en Ingeniería Automática*, Departamento de Automática y Computación, Facultad Ingeniería Eléctrica, ISPJAE, 2007.
3. **OPC Foundation.** "OPC Data Access Automation Specification", Version 2.01-January 6, 1999, disponible en el sitio Web de la Fundación OPC, <http://www.opcfoundation.org>. Fecha de acceso: febrero, 2007.
4. **OPC Foundation.** Sitio Web de la Fundación OPC: <http://www.opcfoundation.org>, 2009
5. **AYZA, J.** "Software de Adquisición, Supervisión y Control", *Automática e Instrumentación*, No 299, página 117, 1999.
6. **GONZALEZ A.I., MORENO V.** *Introducción a la automatización de los sistemas productivos*, (Libro en proceso de revisión e impresión), Editorial Félix Varela, 2009.
7. **SOFTING.** "Free choice of Platform, The next generation", 2006 disponible en el sitio Web de la Empresa Softing: <http://www.softing.com>. Fecha de acceso: junio-2009.
8. **LANGE J.** "Quo Vadis OPC? From Data Access to unified Architecture". Director de OPC Softing, (2006) member of the European Steering Committee of the OPC Foundation, disponible en el sitio Web de la Empresa Softing: <http://www.softing.com>. Fecha de acceso: junio-2009.
9. **ZAMARENO J., CRISTEA S., RUEDA A., AREF R.** "Módulos OPC para el control y supervisión de entornos industriales" presentado en las *XXVI Jornadas de Automática*, Alicante-Elche, 2005.
10. **ROCA L., YEBRA M. BERENGUEL, D.** "Obtención de modelos para plantas desaladoras basadas en energía solar" presentado en las *XXVI Jornadas de Automática*, Alicante-Elche, 2005.
11. **REYES C., CEPEDA A., PORTES B., ALVARADO I., CAMACHO E.** "Control de la planta de los cuatro tanques mediante la realización de una pasarela MATLAB[®]-HTTP-OPC" presentado en las *XXVI Jornadas de Automática*, Alicante-Elche, 2005.
12. **CASTRILLEJO D., OLIVEIRA F., ALVES R.** "Entorno de ayuda a la operación de una planta piloto de laboratorio" presentado en las *XXVI Jornadas de Automática*, Alicante-Elche, 2005.
13. **IWANITZ F.** "OPC without Windows, Web Services make it possible", Project Manager for OPC (2007), disponible en el sitio Web de la Empresa Softing: <http://www.softing.com>. Fecha de acceso: junio-2009.
14. **SOFTING SA.** Sitio Web de la Empresa Softing: <http://www.softing.com>, 2009.
15. **SUSS G.** "Driverless Transport in Hospital"(2008), disponible en el sitio Web de la Empresa Softing: <http://www.softing.com>. Fecha de acceso: junio-2009.
16. **SOFTING SA.** "Application News, OPCTunnel vs Dcom Headache" (2007), disponible en el sitio Web de la Empresa Softing: <http://www.softing.com>. Fecha de acceso: junio-2009.
17. **SOFTING SA.** "OPC technology in oilfield operations (2007), disponible en el sitio Web de la Empresa Softing: <http://www.softing.com>. Fecha de acceso: junio-2009.
18. **SOFTING SA.** "OPC XML-DA Cliente implementation - A case study (2007), disponible en el sitio Web de la Empresa Softing: <http://www.softing.com>. Fecha de acceso: junio-2009.
19. **SOFTING SA.** "Buhter's Grain Milling Solution Relies on Softing's OPC technology". (2008), disponible en el sitio Web de la Empresa Softing: <http://www.softing.com>. Fecha de acceso: junio-2009.
20. **SOFTING SA.** "Clear power management with Softing Modbus/TCP OPC Server and Siemens PAC 3200". (2008), disponible en el sitio Web de la Empresa Softing: <http://www.softing.com>. Fecha de acceso: junio-2009.
21. **SOFTING SA.** "Lenze's new L-Force Product Line Transparently. Employ Integrated OPCTechnology". (2008), disponible en el sitio Web de la Empresa Softing: <http://www.softing.com>. Fecha de acceso: junio-2009.
22. **SIEMENS.** Sitio Web de la compañía Siemens: <http://www.siemens.com>, 2009.
23. **SIEMENS.** "SIMATIC WinCC[®], Add-ons for Individual Expansions Overview", (2001) Siemens AG, disponible en el sitio Web de la Empresa Siemens: <http://www.siemens.com>. Fecha de acceso: Mayo-2007.
24. **SIEMENS.** "SIMATIC hmi WinCC[®], Visualización de proceso y plataforma para IT Business Integration". (2003), disponible en el sitio Web de la Empresa Siemens: <http://www.siemens.com>. Fecha de acceso: Mayo-2007.
25. **SIEMENS.** "HTML Help", Paquete de ayuda de la versión 6.0 de WinCC, (2004), disponible en el sitio Web de la Empresa Siemens: <http://www.siemens.com>. Fecha de acceso: Mayo-2007.
26. **WONDERWARE.** Sitio web de la compañía Wonderware: <http://www.wonderware.com>, 2009.
27. **WONDERWARE.** "SPCPro User's Guide", Revision A, Invensys Systems Inc. FactorySuite (2002), disponible en el sitio Web de la Empresa Wonderware <http://www.wonderware.com>. Fecha de acceso: mayo-2007.

28. **CI TECHNOLOGIES.** Sitio Web de la compañía Ci Technologies: <http://www.citect.com>, 2009.
29. **WIZCON.** Sitio Web de la compañía Wizcon: <http://www.wizcon.com>, 2009.
30. **NI.** Sitio Web de la compañía National Instruments: <http://www.ni.com>, 2009
31. **AYZA J.** “Software de supervisión y control”, *Automática e Instrumentación*, No. 344, página 78, 2003.
32. **OGATA K.** “*Ingeniería de control moderna*”; Ed. Prentice-Hall, pp-75-158, 1987.
33. **MATLAB.** MatLab® 7.4.0 (R2007a), “Ayuda de Simulink”. Paquete de ayuda, 2007.
34. **MORRILLA F.** “Controladores PID: ajuste de parámetros”, *Automática e Instrumentación*, No 207, 1990.
35. **MORRILLA F.** “Controladores PID: Algoritmos y estructuras”, *Automática e Instrumentación*, No 204, 1990.
36. **ALVAREZ G.** et al. “Módulos de supervisión y control empleando microcontroladores PIC”, reporte de investigación, ICIMAF, 2003.
37. **FERNANDEZ L. M.** et al. “Laboratorio virtual para el entrenamiento del personal de la industria” Reporte de investigación, ICIMAF, 2003.
38. **MATLAB.** MatLab® 7.4.0(R2007a), “Ayuda de System Identification Toolbox”, versión 7.0. Paquete de ayuda, 2007.
39. **AGUADO A.** “*Temas de identificación y control adaptable*”, Editorial PALCIEN, Instituto de Cibernética, Matemática y Física, Cuba, 2000.
40. **LJUNG L., GLAC T.,** “*Modeling of Dynamic Systems*”, Ed. Prentice-Hall, 1994.
41. **FERNANDEZ Y., TRANK S.** “Aplicación del OPC en los laboratorios docentes”, *Trabajo de diploma en Ingeniería Automática*, Departamento de Automática y Computación, Facultad Ingeniería Eléctrica, ISPJAE, 2007.
42. **MATLAB** MatLab® 7.4.0(R2007a), “Ayuda de OPC Toolbox”, versión 2.0.4. Paquete de ayuda, 2007.
43. **BHAT J. Bhat, MADHAVAN K. P. CHIDAMBARAM M.;** “Multivariable global Input/Output Linearized Internal Model Control”, *Industrial Chemistry Res*, Vol 30, Number 7, 1991.

Ana Isabel González Santos. Graduada de Ingeniería en máquinas computadoras en 1994. Máster en Automática en 1997. Doctora en Ciencias Técnicas desde el 2000. Profesora Titular del Departamento de Automática. Realizó su doctorado en control predictivo. Actualmente se dedica al control predictivo y a los sistemas SCADAs. Ha tenido responsabilidades como jefa del Departamento de Automática y Decana de la Facultad de Eléctrica.

Correo electrónico: anita@electronica.cujae.edu.cu

AUTORES

Daniel Sabin Díaz. Graduado de Ingeniero Automático en el 2007. Profesor Instructor del Departamento de Automática de la Facultad de Ingeniería Eléctrica. Termina los estudios en la maestría de informática industrial y automatización. Se dedica al estudio de los sistemas SCADAs, a los módulos específicos para el control, la identificación y la detección de fallos.

Correo electrónico: sabin@electronica.cujae.edu.cu

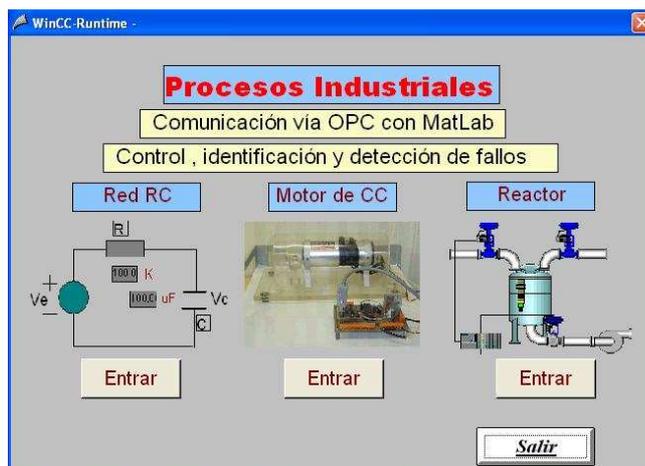


Figura 1 Ventana principal de la aplicación SCADA

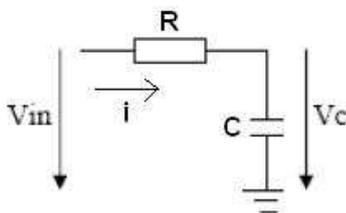


Figura 2 Sistema de primer orden (Red RC)

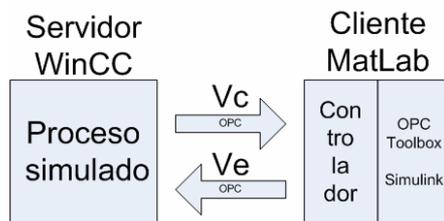


Figura 3 Esquema de la comunicación OPC entre MatLab®-WinCC® para la red RC

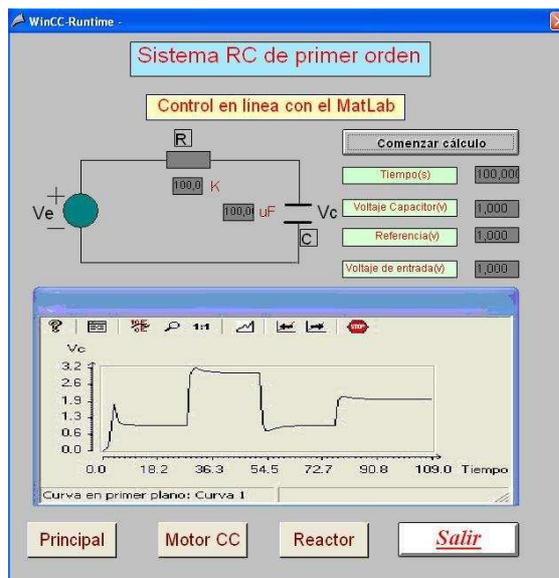


Figura 4 Ventana en WinCC® de la red RC

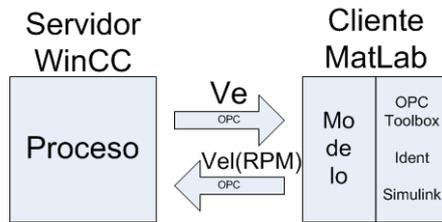


Figura 5 Esquema de la comunicación OPC entre MatLab[®]-WinCC[®] del motor CC



Figura 6 Ventana en WinCC[®] del motor de CC

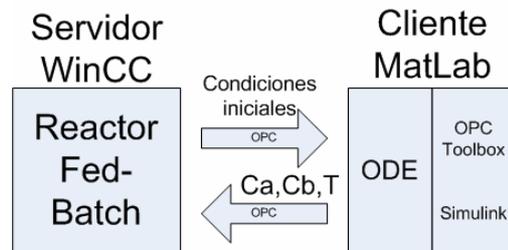


Figura 7 Esquema de la comunicación OPC entre MatLab[®]-WinCC[®] del reactor semi-batch

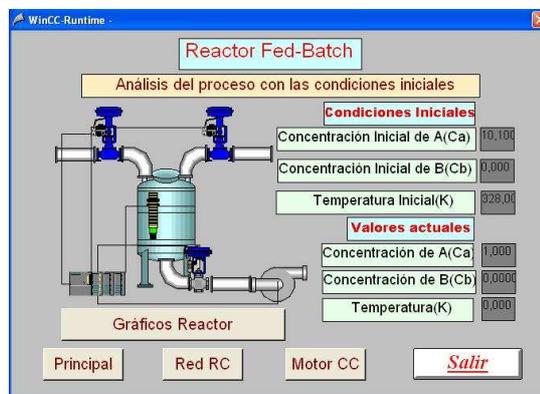


Figura 8 Ventana en WinCC[®] del reactor semi-batch

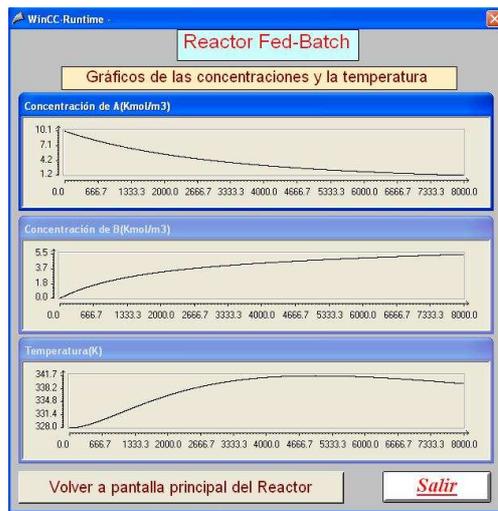


Figura 9 Gráficos de las variables del reactor semi-batch en una conexión OPC activa entre MatLab® y WinCC®

Tabla 1 Valores de las constantes del Reactor Fed-Batch

Constante	Significado	Valor	Unidades
C_{bl}	Concentración inicial de entrada de (b)	9.7	$KmolesB / m^3$
T_b	Temperatura a la cual es introducido (b)	328	K
$A = \frac{V_{ro}}{\phi b}$	Tiempo requerido para duplicar el volumen inicial	6197	s
B	Velocidad de adición de (b)	16.92	$m^3 K / Kmol$
C	Parámetro de enfriamiento	0.253×10^{-3}	s^{-1}
K	Velocidad de reacción	2.608×10^{-5}	$m^3 / Kmols$
Condiciones iniciales	Significado	Valor	Unidades
C_{ao}	Concentración inicial del compuesto (a)	10.1	$Kmol / m^3$
C_{bo}	Concentración inicial del compuesto (b)	0	$Kmol / m^3$
t	Tiempo inicial	0	s