

SIMULACIÓN Y CONTROL DE PROCESOS FÍSICOS DE FORMA REMOTA

A.Valera, M. Vallés, J.L. Díez

*Dpto. Ingeniería de Sistemas y Automática
Universidad Politécnica de Valencia
Camino de Vera 14, 46022 Valencia (Spain)
Fax: +34 96 387 95 79*

E-mail: giuprog@isa.upv.es, mvalles@isa.upv.es, jldiez@isa.upv.es

Resumen: Este trabajo propone una solución, mediante aplicaciones basadas en Matlab[®] para el desarrollo de laboratorios virtuales y remotos, a los problemas que se suelen plantear con las sesiones de prácticas de laboratorio en la docencia a distancia de las materias relacionadas con la Ingeniería de Sistemas y el Control de Procesos.

Además, se presenta el concepto del proceso virtual, conjunto de rutinas y funciones software que permiten la simulación animada de la respuesta de procesos físicos en el control remoto, que elimina el problema de que varias personas no pueden conectarse simultáneamente al mismo proceso real. *Copyright © 2005 CEA-IFAC*

Palabras clave: educación a distancia, laboratorios virtuales, laboratorios remotos, aplicaciones digitales de computación, control por computador, adquisición de datos.

1. INTRODUCCIÓN

Las tendencias en educación (Poindexter, 1999) en general y en ingeniería de control en particular (Dormido, 2002; Bernstein, 2004) están incorporando Tecnologías de la Información en el proceso de aprendizaje. Este se debe no sólo al incremento del número de hogares con computadores conectados a Internet, sino también al menor número de horas que se imparten en los nuevos programas de estudio de las universidades. Todo ello provoca la necesidad de optimizar el proceso educativo.

Si además tenemos en cuenta que el uso de recursos de Internet proporcionan un incremento del tiempo y de la flexibilidad del espacio en el proceso educativo, podemos comprobar que si bien la educación a distancia obtiene más beneficios de las nuevas tecnologías que la educación tradicional, esta última las puede utilizar como un recurso de apoyo muy efectivo (Paprock, 1998).

Basado en Internet se pueden encontrar dos opciones diferentes para el desarrollo de laboratorios: los laboratorios virtuales y los laboratorios remotos. Un laboratorio virtual permite, por ejemplo, el acceso a

procesos simulados en un computador. Así, la interacción remota entre el estudiante y el proceso virtual es posible en este tipo de laboratorio.

A mitad de camino entre los laboratorios tradicionales y los virtuales nos encontramos con los laboratorios remotos, laboratorios que ofrecen experimentos reales a usuarios remotos. La incorporación de cámaras web permite a los estudiantes observar, además de las variables del sistema, la evolución de sistemas reales. La mayoría del equipamiento necesario para el desarrollo de laboratorios virtuales y remotos está disponible en los laboratorios tradicionales, el único elemento nuevo que se necesita es una interfaz entre la aplicación local y el servidor web.

El desarrollo de este trabajo está basado en aplicaciones de Matlab, puesto que éste es uno de los paquetes de diseño asistido por computador de sistemas de control más utilizado a nivel mundial.

Matlab tiene un herramienta muy potente: Matlab Web Server[®] (MWS). MWS es un paquete con el que se pueden desarrollar de una forma muy simple laboratorios virtuales (Valera, *et al.*, 2001; Vallés, *et al.*, 2001; Carreira, *et al.*, 2004). Sin embargo, con este

software no es posible desarrollar directamente laboratorios remotos para el control de procesos físicos (Díez, *et al.*, 2001, Puerto, *et al.*, 2004). Para hacerlo, *MWS* se debe complementar con varias funciones y aplicaciones software (que se presentan en este trabajo), o con paquetes de software comerciales, como por ejemplo *WinCon*[®], de Quanser Consulting[®].

El artículo, después de mostrar las configuraciones basadas en Matlab hardware y software necesarias para el acceso remoto, presenta el desarrollo de laboratorios virtuales y remotos, ilustrándose con diferentes ejemplos.

La configuración hardware se basa en un computador personal y tarjetas de adquisición de datos comerciales. Las funciones software permiten el acceso a bajo nivel desde el entorno de Matlab.

Sin embargo, a pesar que los laboratorios remotos pueden permitir a los usuarios establecer el control de procesos reales, aparece una limitación adicional: únicamente un usuario (o grupo de trabajo) puede acceder al proceso real, disminuyendo así la efectividad de estos laboratorios.

Para evitar los inconvenientes de esta limitación se presenta un nuevo concepto en el artículo: los *procesos virtuales*. Un proceso virtual se podría definir como un conjunto de funciones y procedimientos software que simulan la respuesta del proceso real mediante ficheros de vídeo *avi*. Así, si las ecuaciones del proceso virtual son muy parecidas a las ecuaciones físicas del sistema, la respuesta del sistema real y del virtual será idéntica.

2. CONFIGURACIONES BASADAS EN MATLAB PARA EL CONTROL DE PROCESOS VIRTUALES Y REMOTOS

En este apartado se va a presentar 2 aplicaciones basadas en Matlab: el software Matlab Web Server y el software *WinCon*. Con estas aplicaciones no sólo se pueden desarrollar laboratorios virtuales y remotos de una forma muy simple, sino que además también permite reutilizar todo el trabajo que se ha podido desarrollar anteriormente, ya que es casi directo generar aplicaciones para el acceso desde Internet a partir de funciones de Matlab o esquemas de Simulink.

2.1 Configuración de Matlab Web Server

MWS es un software integrado que permite a los programadores de Matlab crear aplicaciones que utilizan las facilidades de la *World Wide Web* para enviar datos a Matlab, mostrando posteriormente los resultados de los cálculos también vía web.

El conjunto de programas que permiten el acceso remoto a aplicaciones de Matlab es el siguiente (The MathWorks, 1999):

- *matlabserver*: permite la comunicación entre la aplicación web y el Matlab. Matlabserver es un servidor multihilo TCP/IP que ejecuta los programas de Matlab (ficheros .M) especificados en documentos tipo HTML. Además, también se especifica el número máximo de simulaciones

simultáneas que se permiten y el número de puerto a utilizar.

- *matweb.exe*: ubicado en el servidor HTML es un servidor cliente TCP/IP del matlabserver que usa la Common Gateway Interface (CGI) para extraer los datos desde el documento HTML y transferirlos a matlabserver.
- *matweb.m*: ejecuta el fichero .M que la aplicación web desea utilizar.
- *matweb.conf*: es un fichero de configuración que matlab necesita para conectarse con matlabserver.

Para una explicación más exhaustiva de las principales características de *MWS* o de los comandos utilizados para de desarrollo de aplicaciones básicas ver (Vallés, *et al.*, 2001). En este artículo se muestra con detalle y ejemplos la configuración y el equipamiento para el desarrollo de laboratorios remotos.

El proceso de desarrollo de aplicaciones *MWS* suele necesitar una combinación de modelos Simulink, ficheros de Matlab y código y gráficos HTML. Por ello se tendrán que generar los ficheros siguientes:

- Un documento de entrada de datos HTML para suministrarle a Matlab los parámetros de entrada del control virtual/remoto
- Un modelo Simulink o un fichero de Matlab para procesar los datos de entrada y calcular los resultados
- Un documento HTML de salida para mostrar los cálculos de Matlab
- Un fichero de prueba para validar el código antes de la distribución de la aplicación en la web

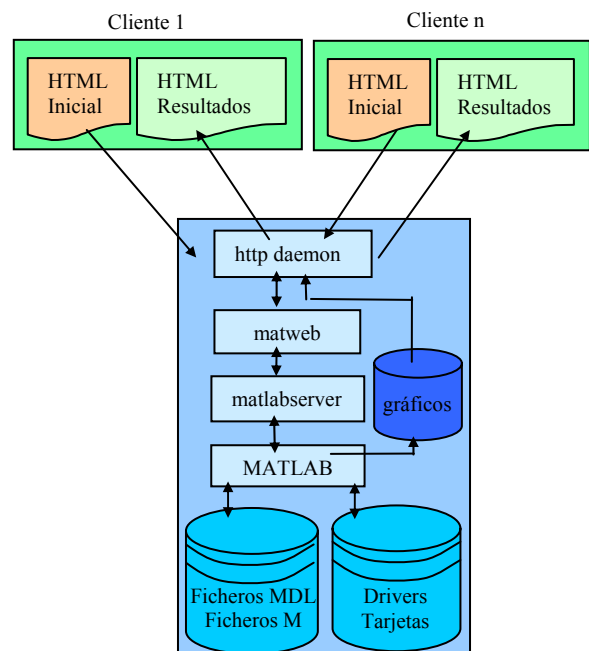


Fig. 1. Estructura de ficheros de *MWS*

Cabe destacar que *MWS* no permite el acceso a bajo nivel para programar el hardware del sistema. Por ello, a priori, no es posible desarrollar laboratorios remotos o establecer el control de procesos industriales remoto. Para evitar esta limitación utilizando *MWS* se ha generado una configuración adicional con varias tarjetas de adquisición de datos, programándose además un conjunto de rutinas para el acceso desde el entorno de

Matlab. La estructura de ficheros se muestra en la Figura 1.

Las rutinas que se han programado para el acceso a estas tarjetas desde el entorno de Matlab permiten las conversiones Analógico/Digital, las conversiones Digital/Analógico, lectura de las señales de encoders, entradas y salidas digitales así como la programación de temporizadores. Todas estas funciones han sido programadas con el lenguaje C++. Las funciones ejecutables de Matlab se obtienen mediante ficheros DLL generados con el comando mex y el compilador Visual C++ v6.0.

```
>> mex read_812.cpp
```

2.2 Configuración de WinCon

WinCon es una aplicación de tiempo real (bajo Windows 2000 o XP) que permite la ejecución en tiempo real de código generado por diagramas Simulink en un PC local o en uno remoto (Quanser Consulting, 2002).

WinCon está compuesto por 2 partes distintas que se comunican con el protocolo TCP/IP: *WinCon Client* (*WCC*) y *WinCon Server* (*WCS*). *WCC* es un componente software de tiempo real que ejecuta código generado desde diagramas Simulink. Recibe el controlador desde *WCS* y lo ejecuta en tiempo real. Además, mantiene la comunicación con *WCS* para la transferencia de las variables de la ejecución.

WCS se encarga de la compilación del diagrama de Simulink para crear un controlador ejecutable. Además, mantiene una comunicación con Simulink para realizar los cambios en línea de los parámetros del controlador, así como el dibujo y almacenamiento de los datos de la respuesta del sistema.

WinCon soporta 2 posibles configuraciones: la configuración local y la remota. En la configuración local, tanto *WCC* como *WCS* se ejecutan simultáneamente en el mismo computador. En la configuración remota, se ejecutan en máquinas diferentes, utilizando el protocolo TCP/IP para su comunicación, de forma que cada *WCS* puede comunicarse con varios *WCC*, y recíprocamente, cada *WCC* puede comunicarse también con varios *WCS*. La figura 2 muestra una posible configuración remota, con un único *WCS* y múltiples *WCC* ejecutándose en nodos distintos en una red TCP/IP.

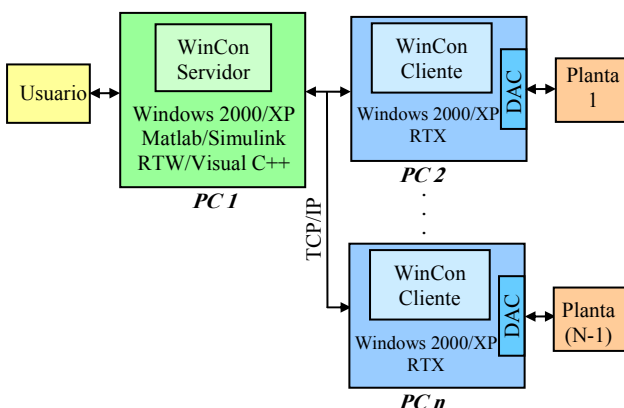


Fig. 2. Configuración remota de *WinCon*

En esta configuración, las comunicaciones entre máquinas distintas se realizan sobre una red (red de área local, Intranet, Internet, etc.), de manera que por ejemplo *WCS1* puede descargar el código sobre *WCC1*, mientras que podría recoger los datos de la ejecución de *WCC2*. Además, otros *WCS* se podrían añadir directamente a la configuración remota.

2.3 Configuración Hardware

La configuración hardware del sistema (Figura 3) se compone de un conjunto de computadores remotos (clientes) y de un servidor. Éste dispone de varias tarjetas de adquisición de datos (*SADs*) industriales de *Advantech*TM y *Quanser Consulting* para el acceso a las señales del sistema a controlar (acciones de control y salidas del sistema). Por último, para la configuración de los laboratorios remotos, se le han conectado diferentes procesos industriales, así como una cámara web que permite visualizar la evolución del sistema real.

La configuración necesaria para los laboratorios virtuales es muy parecida, si bien en este caso, como se trabaja únicamente con simulaciones realizadas en el servidor, no es necesario disponer de ningún dispositivo físico (procesos reales, tarjetas de adquisición de datos, cámaras web etc.).

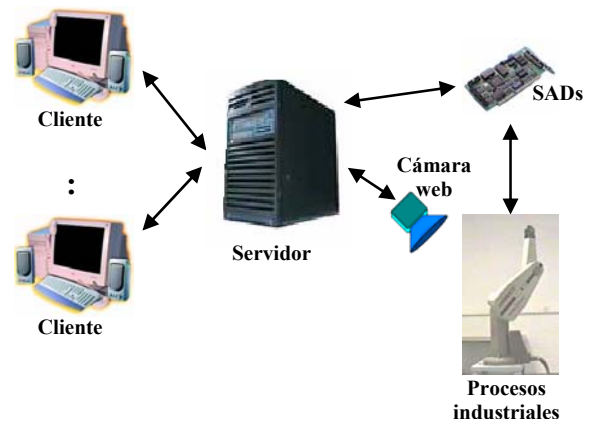


Fig. 3. Configuración hardware de laboratorios remotos

3. DESARROLLO DE APLICACIONES VIRTUALES

Como se ha comentado en la sección 2.1, se necesitan tres ficheros para la ejecución de aplicaciones de Matlab a través de Internet. Se necesita un fichero HTML inicial para el envío de los datos y parámetros a Matlab. Este fichero debe especificar todas las variables necesarias, así como de un botón *submit* para el envío de la información. Los nombres de las variables de estas páginas web deben coincidir con las variables del fichero de Matlab.

Además, también se debe disponer de un fichero “.M” de Matlab (o un esquema Simulink) encargado de realizar la simulación o procesamiento de los datos necesarios. Por último, se necesita un fichero (HTML Resultados) en el que se le devolverá al cliente remoto los resultados de la simulación que se acaba de realizar.

Se ha escogido la herramienta *MSW* de Matlab porque, a partir de la reutilización de funciones Matlab y diagramas Simulink ya existentes, es muy simple y

directo la generación de laboratorios virtuales para el acceso remoto.

3.1 Laboratorios Virtuales con ficheros “.M”

Como se ha comentado, el fichero “.M” para aplicaciones virtuales es una función típica de Matlab que acepta los datos suministrados por el documento de entrada HTML y devuelve los resultados a la página HTML de salida. Existen diferentes opciones para mostrar los resultados:

- *Variables escalares*

El modo más simple de exhibir resultados de salida es mostrar el valor de un escalar o cadena de caracteres. Para hacerlo sólo hay que escribir (entre símbolos \$) el nombre de la variable en el fichero HTML de salida.

Por ejemplo, si se necesita mostrar en una página web dos variables (*var_max* y *var_med*) proporcionadas por Matlab, el código debería ser:

```
:
<p>
Valor máximo: $var_max$
Valor medio: $var_med$
</p>
:
```

- *Matrices*

Si la función de salida de Matlab es más compleja, los resultados se podrían mostrar en una matriz. Para ello, mediante el comando HTML *table*, se puede generar dinámicamente una tabla con la información de la variable de Matlab.

Por ejemplo, si tenemos una matriz de datos *var_matriz* que se desea mostrar, el código HTML que sería necesario es:

```
:
<table border="1" cellspacing="1"
autogenerate="var_matrix">
<tr>
<td align="right"> </td>
</tr>
</table>
:
```

- *Gráficos*

Para mostrar a la vez mucha información es mejor utilizar los gráficos. *MWS* genera gráficos jpeg desde Matlab. Para ello se tiene que definir primero un identificador (*mlid*), utilizándose posteriormente el comando *wsprintjpeg(fig, fichjpeg)* para la creación, mediante el comando *print* con el argumento *-djpeg*, de la figura.

```
:
mlid=getfield(h,'mlid')
s.GraphFileName = sprintf('%graph.jpeg',mlid)
<img border=0 src="$GraphFileName$>
:
```

3.2 Laboratorio Virtuales con Modelos Simulink

Además de los ficheros “.M”, las aplicaciones virtuales se pueden desarrollar también a partir de modelos Simulink. En este caso se necesitan un fichero “.MDL” y un fichero “.M”. El fichero MDL es un modelo normal de Simulink que tiene los bloques necesarios para la realización de la simulación (referencia, funciones de transferencia continuas y discretas, osciloscopios, etc.). El modelo puede ser un esquema general de simulación donde el usuario especifica los parámetros de los distintos bloques de simulación como variables de Matlab (como numerador y denominador de las funciones de transferencia, valores de las constantes, periodos de muestreo, tiempo de simulación, etc.). Debido a que se proporciona toda esta información en el documento HTML de entrada, es muy fácil cambiar el sistema a simular.

Para desarrollar el laboratorio virtual sólo falta un fichero “.M”. Este fichero debe actualizar los distintos parámetros del modelo Simulink y realizar la simulación. Para hacerlo, se utiliza el comando:

```
>> set_param(model,'parameter1',value1,...);
```

Una vez realizada la actualización de los parámetros, el comando *sim* realizará la simulación del modelo:

```
>> [time, state, output]=sim(model);
```

Cuando termina la simulación, el documento de salida HTML se encarga de actualizar y mostrar los resultados de la simulación.

4. DESARROLLO DE PROCESOS VIRTUALES

A partir de los laboratorios virtuales se puede obtener la simulación de cualquier sistema. Sin embargo, puesto que la respuesta que se obtiene suele ser una figura, a menudo los alumnos pueden perder el significado físico de, por ejemplo, si la respuesta del sistema tiene un comportamiento subamortiguado o sobreamortiguado, del valor de la sobreoscilación máxima, el tiempo de establecimiento, etc.

Una solución a este problema se puede obtener a partir de los *procesos virtuales*. Un proceso virtual es un conjunto de funciones de Matlab que simula la respuesta de un proceso real (que se toma como base) por medio de ficheros *avi* (Audio Video Interleave). Si las ecuaciones matemáticas del proceso virtual coinciden con las ecuaciones físicas del proceso real, las respuestas de ambos será prácticamente la misma. Como el resultado es un fichero de animación, el usuario puede ejecutarlo tantas veces como quiera. Además, como éste medio es mucho más intuitivo, se pueden evitar los problemas que se comentaban anteriormente.

Los videos *avi* (unos de los ficheros de vídeo más comunes para computadores personales) se pueden obtener directamente desde Matlab. La forma más simple es, primero, generar el fichero con el comando:

```
>> AVIOBJ=avifile(filename);
```

Después se pueden añadir los fotogramas de la película con el comando *addframe*:

```
>> AVIOBJ = addframe(AVIOBJ,Frame)
```

donde *Frame* es el fotograma obtenido de la figura deseada (*fig*):

```
>> Frame= getframe(fig);
```

Habitualmente los comandos *getframe* y *addframe* se utilizan en un bucle *FOR* para obtener las diferentes figuras que conformarán la animación de la película.

Un vez que se ha añadido el último fotograma, el comando *close* cerrará el fichero *avi*:

```
>> AVIOBJ=close(AVIOBJ);
```

La figura 4 muestra un ejemplo de proceso virtual desarrollado: un motor de corriente continua virtual basado en el Servomotor SFT 154 de Feedback[®]. El eje de este motor de continua describe un movimiento circular que, dependiendo de la acción de control, puede alcanzar un ángulo desde 0 hasta 2π radianes.

Tal como se muestra, el motor virtual proporciona, además de la fotografía del proceso real tomado como base, la posición del eje (figuras de la izquierda) y la acción de control (figura inferior derecha).

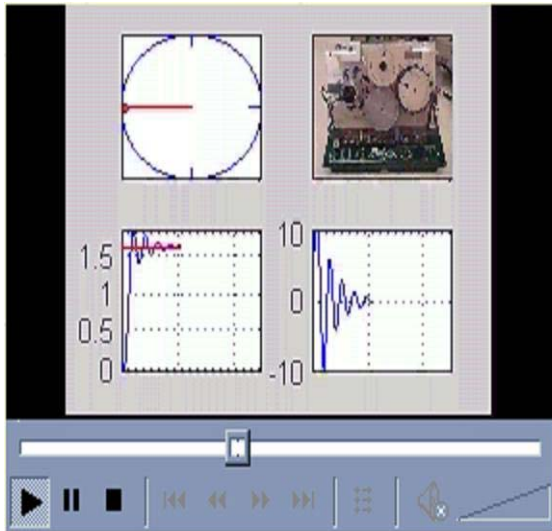


Fig. 4. Fichero Avi de un Proceso Virtual (Motor CC)

Para desarrollar este motor virtual se ha programado el sistema para generar ficheros *avi* con la evolución del sistema. Este fichero de animación muestra la evolución del cursor del eje del motor imitando fidedignamente la respuesta del motor real. La dinámica del sistema se describe por las ecuaciones en diferencias de la función de transferencia del sistema.

Además de la gráfica animada de la posición del cursor se han añadido dos más, mostrando la posición del eje y las acciones de control proporcionadas en función del tiempo. Las tres gráficas animadas evolucionan simultáneamente de forma que los usuarios pueden analizar de una forma directa y fácil la evolución del sistema. Cuestiones como comportamientos sobreamortiguados o subamortiguados, sobreoscilación máxima, tiempo de posicionado etc. pueden entenderse perfectamente, evitando así las lagunas que suelen aparecer entre las gráficas tradicionales de las simulaciones y el comportamiento del sistema real.

Después de que se haya generado el fichero *avi* con la respuesta del proceso virtual, los usuarios remotos pueden obtener la animación en el documento HTML de salida proporcionado por la librería Matlab Web Server. El proceso virtual mostrará la evolución de la posición del eje del motor y las acciones de control aplicadas. Además, puesto que se trata de un fichero *avi*, los usuarios pueden visualizar la respuesta del sistema repetidamente sin más que pulsar el botón correspondiente, lo que les permite analizar la respuesta del sistema (estabilidad, error en régimen permanente, etc.) de una forma simple, rápida e intuitiva. Con esta información se podría diseñar un controlador y simular la respuesta del sistema con las herramientas de Matlab o con el proceso virtual. Una vez que las simulaciones del sistema son correctas, los usuarios pueden obtener el permiso para acceder al control remoto del proceso físico real.

5. EJEMPLOS DE LABORATORIOS VIRTUALES

Para analizar las características de la herramienta desarrollada y la facilidad de acceso a la plataforma vía Internet se han desarrollado varios laboratorios virtuales. Para ello se han generado (basándose en lo presentado en el apartado 3) y utilizado varios ficheros “.M” y “.MDL”.

Actualmente se tienen disponibles laboratorios virtuales relacionados con el diseño de filtros digitales y el diseño de controladores, contenidos muy comunes en cursos básicos de control (Díez, *et al.*, 2002,). De esta forma, cualquier estudiante puede acceder mediante Internet a los contenidos desarrollados en estos laboratorios remotos. Para ello, el único requerimiento que necesitan los estudiantes es tener un computador con conexión a Internet y un buscador web, evitando así por ejemplo que se deba tener una instalación local de Matlab.

La estructura de los laboratorios consiste en hipervínculos desde la página principal (*index.html*) a las páginas principales de diseño de controladores y de filtros, lo que permite que sea muy simple actualizar o ampliar los contenidos y laboratorios virtuales.

A continuación se muestran un conjunto de laboratorios, para que se pueda comprobar la herramienta desarrollada.

5.1 Laboratorio de Diseño de Filtros IIR.

Este laboratorio incluye varias actividades, como:

- Generación de señales
- Diseño de filtros
- Análisis del filtrado de señales

La opción de Generación de señales permite que Matlab genere señales complejas sumando componentes senoidales. Para ello el usuario debe proporcionar los parámetros de las señales que quiere obtener (amplitud, frecuencia y fase, habitualmente).

Después que se introducen los parámetros como cajas de entrada de texto se debe pulsar el botón de Submit para hacer llegar los datos al servidor de aplicaciones de Matlab. Cuando le lleguen, un fichero “.M” procesará la

información (por medio de los comandos de Matlab *plot* y *fft*) y devolverá los resultados al computador del cliente.

Para el diseño de filtros se ha creado una página inicial HTML que envía al servidor el ancho de banda de la señal a filtrar y los parámetros del filtro. Estos parámetros son, por ejemplo, el tipo de filtro (Butterworth, Chebyshev I, Chebyshev II, Elíptico), las características del filtro (pasa-bajo, pasa-alto, pasa-banda, rechazo de banda), las frecuencias de paso/rechazo, etc.

Una vez que se han introducido los parámetros en los menús y cajas de entrada de texto, pulsando el botón de Submit se enviarán los datos, obteniéndose al cabo de poco tiempo una gráfica (proporcionada por el comando *freqz* de Matlab) con el análisis frecuencial del filtro diseñado.

La tercera opción permite simular el comportamiento de los filtros. La salida que proporciona esta opción (Figura 5) muestra la respuesta temporal y frecuencial de las señales original y filtrada.

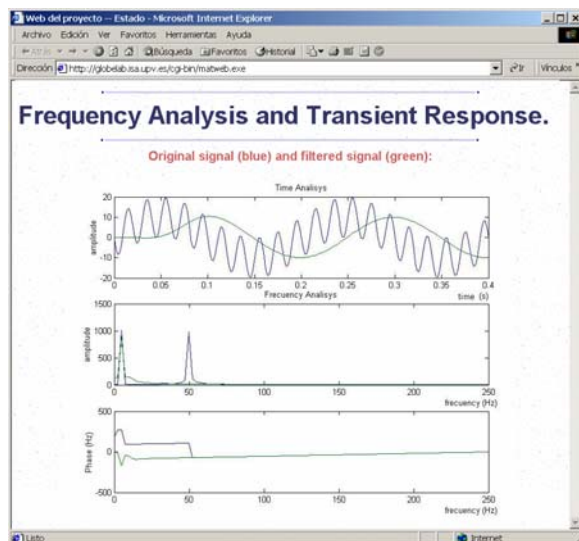


Fig. 5: Resultados del Filtrado de Señales

5.2 Laboratorio de Diseño de Controladores

Desde la página principal se puede acceder a otros laboratorios virtuales, como por ejemplo el laboratorio de diseño de controladores. Este laboratorio se ha desarrollado a partir de modelos Simulink y de ficheros ".M" de Matlab. Los posibles ejercicios disponibles hasta la fecha son los siguientes:

- Análisis en sistema en bucle abierto
- Análisis en sistema en bucle cerrado
- Diseño de controladores

La figura 6 muestra la página inicial para el análisis de sistemas en bucle abierto. Con esta página se puede enviar (mediante las cajas de entrada de texto) el numerador y denominador de la función de transferencia, el periodo de muestreo y el tiempo total de simulación, utilizando el servidor el comando *dstep* de Matlab para realizar la simulación del sistema.

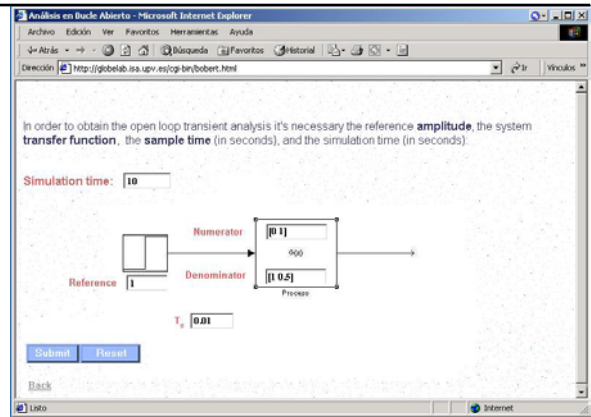


Fig. 6: Página inicial para el Análisis en Bucle Abierto

El análisis de sistemas en bucle cerrado se hace de una forma parecida puesto que con la página inicial se envían, entre otros parámetros, el numerador y denominador de la función de transferencia a analizar de manera que los usuarios remotos acceden al comando de Matlab *rlocus* para las tareas de diseño de controladores.

La última opción de este laboratorio permite realizar simulaciones del comportamiento del sistema controlado en bucle cerrado. En este caso se debe especificar los numeradores y denominadores de las funciones de transferencia del controlador y de la planta a regular. En esta opción se utilizan los comandos de Matlab *set_param* para actualizar los parámetros del esquema Simulink y *sim* para realizar la simulación del sistema. La respuesta del servidor para esta opción es una gráfica donde aparece la respuesta temporal del sistema controlado y la referencia introducida.

6. CONTROL DE PROCESOS VIRTUALES

Los laboratorios remotos permiten a los estudiantes establecer el control de sistemas reales sin la necesidad de disponer el proceso real y de las herramientas software. Esto es sumamente interesante puesto que el precio de ambas cosas suele ser muy caro. El único inconveniente que tienen los laboratorios remotos reside en el número de usuarios que pueden trabajar con él, ya que habitualmente sólo un usuario (o grupo de trabajo) puede acceder simultáneamente al proceso real.

Por supuesto, los usuarios pueden analizar y simular el sistema con los laboratorios virtuales, utilizando para ello las actividades descritas en el apartado 5. Sin embargo con esta solución pueden aparecer otros problemas puesto que en ocasiones los resultados pueden ser un tanto abstractos y por lo tanto difíciles de entender por los alumnos, perdiendo la relación entre las simulaciones y la respuesta real del sistema.

Una opción que se puede utilizar para evitar estos problemas son los procesos virtuales. Debido a que un proceso virtual es muy parecido al proceso real, los estudiantes pueden entender y analizar fácilmente la respuesta del sistema. Además, puesto que se trata de funciones software no existe la limitación en el número de procesos que estén ejecutando simultáneamente, por lo que múltiples usuarios pueden estar trabajando con estos procesos a la vez.

El primer paso para establecer el control de un proceso virtual reside en el desarrollo del sistema virtual. Para ello es preferible basarse en un proceso físico real que se conozcan muy bien las ecuaciones dinámicas y los valores de sus parámetros.

Con los procesos virtuales se pueden desarrollar un gran número de actividades, sin limitaciones del número de procesos que se pueden ejecutar a la vez. Para ello, además del proceso virtual, se tiene que programar una página Web inicial que permita a los usuarios proporcionar los datos de entrada, como por ejemplo la referencia que tiene que seguir el sistema o el numerador y denominador de la función de transferencia del controlador. Además, este documento también proporciona una breve descripción de las actividades que se pueden realizar, así como del significado de los distintos parámetros de entrada y salida etc. En (MathWorks, 1999), (Vallés, *et al.*, 2002), se describe de una forma detallada como se puede llevar a cabo.

Una vez se tienen los parámetros suministrados por el usuario, Matlab puede ejecutar las funciones de control como se hace de forma habitual. Por ejemplo, en el caso del motor de corriente continua virtual mostrado en el apartado 4, los estudiantes deben establecer el control de la posición del motor. Después de haber proporcionado al servidor la referencia y los parámetros del controlador, éste devuelve al usuario un documento HTML de salida con el fichero *avi* (como el mostrado en la Figura 4) que contiene la evolución de la posición del eje del motor, así como las acciones de control generadas por el controlador.

7. CONTROL REMOTO DE PROCESOS INDUSTRIALES

Los laboratorios remotos pueden trabajar de dos formas diferentes: por lotes (*batch*) o en línea (*on-line*). Los laboratorios por lotes son más fáciles de desarrollar puesto que las referencias y los parámetros de los controladores se envían al servidor antes de empezar el experimento, y una vez finalizado se devuelven a los computadores remotos las salidas del sistema. El desarrollo de los laboratorios remotos en línea, por el contrario, generalmente es mucho más complejo puesto que permite una interacción continua entre el operario y el experimento remoto.

En este trabajo se han desarrollado los 2 tipos de laboratorios remotos. Para los laboratorios remotos tipo por lotes se ha utilizado *MWS*, mientras que los laboratorios en línea se han desarrollado mediante *WinCon*. Así, cualquier usuario con un PC conectado a Internet y un buscador Web puede establecer un control por lotes de procesos industriales, sin necesidad de tener una licencia de Matlab (Díez, *et al.*, 2001). Para el control en línea de procesos físicos se necesita además una licencia de WCS.

En los laboratorios remotos se ha añadido además una cámara (webcam) que permite observar la evolución real del sistema mediante el software libre de Windows Media Technologies 7 (Microsoft Corp, 2000). Para ello se suministra un hipervínculo en la página principal del experimento que permite la emisión de la señal de vídeo

de la cámara. Cuando se escoge la visualización mediante esta cámara

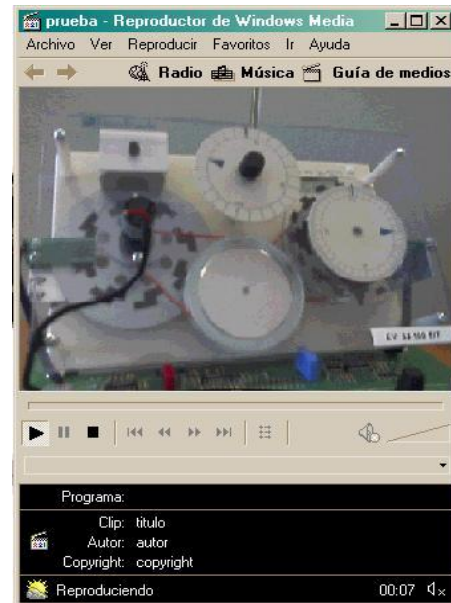


Fig. 7. Visualización del Proceso Real mediante la Cámara del Laboratorio Remoto.

Windows Media Player abre una figura como la que se muestra en la Figura 7 para que los usuarios pueden visualizar en tiempo real el control que se está llevando a cabo.

A pesar que se han utilizado y comprobado otros procesos físicos, en el trabajo se van a presentar 3 laboratorios remotos: un motor de corriente continua, un robot industrial y una mesa vibratoria. Cualquier otro proceso industrial se puede añadir de una forma directa, siempre que sean accesibles las señales del sistema: acciones de control y salidas.

La figura 8 muestra la configuración general utilizada para el control de estos experimentos. En este trabajo se ha utilizado la tarjeta industrial de *Advantech™ PCL-812* para obtener las salidas analógicas (velocidades y posiciones) del sistema físico, la tarjeta *PCI-1711* para proporcionar las acciones de control a los procesos a controlar, y la tarjeta *PCL-833* encargada de la lectura de las señales de los encoders. Además, también se ha instalado una tarjeta multifunción de *Quanser Consulting: MultiQ-2E*.

Desde el punto de vista técnico esta arquitectura de control trabaja de la forma siguiente: los potenciómetros y encoders proporcionan la posición de los motores de los sistemas, y los tacómetros proporcionan la velocidad, introduciéndose estas señales en el servidor con los convertidores analógico/digital y las tarjetas de encoders. Con esta información el servidor calcula y aplica las acciones de control con los convertidores digital/analógico. Además, para controlar sistemas más complejos se han programado otros manejadores (drivers), como por ejemplo para obtener entradas y salidas digitales. Una explicación más detallada sobre el control remoto de procesos industriales se puede encontrar en (Díez, *et al.*, 2001).

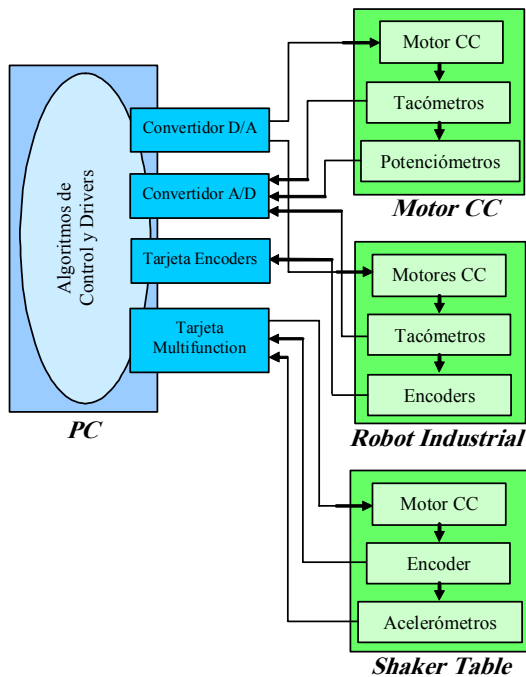


Fig. 8: Configuración Laboratorios Remotos

7.1 Ejemplo: Control Remoto de un Motor

En este primer laboratorio remoto se establece el control de un motor de corriente continua: el Servomotor SFT 154 de *Feedback*. En este caso se puede establecer tanto el control de la posición del eje del motor como el control de la velocidad de giro de su eje.

Para el desarrollo de este laboratorio remoto se necesita un fichero HTML para enviar los datos del control a Matlab. Este fichero debe especificar todas las variables del sistema (referencias, parámetros de los controladores y tiempos de muestro y ejecución etc.), y con el botón de *submit* enviar los datos al programa.

Cuando se pulsa el botón de *submit* los datos se envían al servidor, empezando la ejecución del fichero “.M” asociado. Este fichero es un fichero normal de Matlab, cuya funciones aceptan como parámetros de entrada los datos del documento HTML inicial, ejecutando y devolviendo los datos del experimento al usuario remoto en el documento HTML de salida.

La figura 9 muestra la página de salida del control de velocidad del motor de corriente continua, mostrándose la evolución temporal que ha tenido la velocidad del proceso real.

Para el primer laboratorio remoto (control de posición del motor), el usuario debe especificar la referencia de posición y los parámetros del control, devolviendo los resultados (evolución temporal de la posición) en forma de gráfico.

Cabe destacar que este proceso físico ha sido utilizado como base para el desarrollo de un proceso virtual. Por ello es necesario probar y validar el controlador que se desea implementar en el proceso virtual como paso previo al control del proceso real. Con ello se trata de evitar que el controlador que se suministra pueda provocar que el sistema tenga un comportamiento inestable.

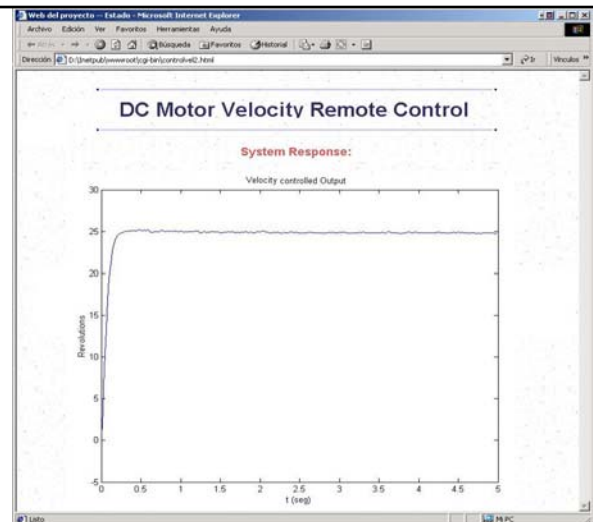


Fig. 9: Página Web de Salida para el Control de Velocidad de un Motor de Corriente Continua

7.2 Ejemplo: Control Remoto de un Robot Industrial

Para el desarrollo de este segundo laboratorio remoto se ha utilizado el robot industrial que se muestra en la Figura 10.



Fig. 10. Robot porticado industrial

El robot tiene tres ejes: X-Y-Z con unos recorridos aproximados de 2,5, 1,5 y 1 metros respectivamente. Además de la parte mecánica, el robot dispone de un armario de control donde se encuentran los variadores y las distintas etapas de potencia que se encargan de alimentar a los motores de corriente continua que generan el movimiento de los tres ejes. A este armario llegan las distintas señales del robot como por ejemplo las señales de la velocidad de los tres ejes que proporcionan los tacómetros, las señales de la posición de los ejes (las señales A, B y paso por cero de los encoders incrementales), el conjunto de señales digitales utilizadas para la detección de los finales de carrera, activación de la herramienta del elemento terminal, etc.

Para el control remoto se han tendido que desarrollar varias páginas HTML para proporcionar los parámetros del controlador y para poder obtener los resultados de la salida. La figura siguiente muestra una página HTML inicial para establecer una trayectoria circular. Tal como se puede apreciar, en ella se debe introducir el ángulo y el radio de la trayectoria, así como el tipo del controlador y sus parámetros.



Fig. 11: Página inicial control remoto robot industrial

La figura 12 muestra la referencia y la posición real del extremo del robot. Así mismo se muestra también un detalle de la respuesta inicial.

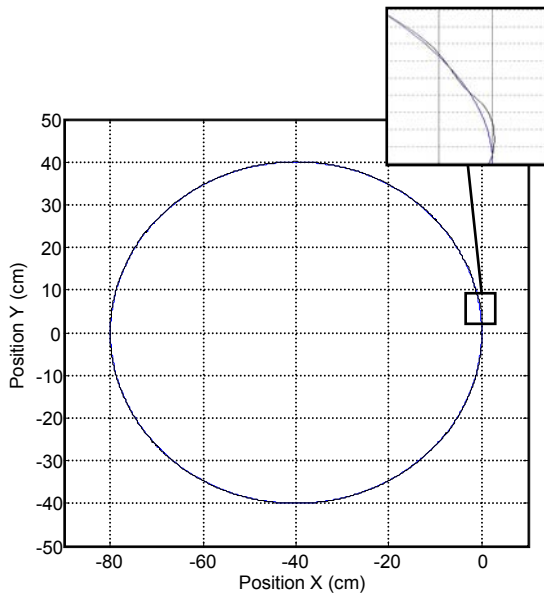


Fig. 12: Salida del control remoto del robot industrial

7.3 Ejemplo: Control Remoto de una mesa vibratoria

Además de los laboratorios remotos por lotes anteriores, también se ha desarrollado un laboratorio remoto en línea. Éste está basado en el prototipo *Shaker Table II*, de Quanser Consulting. Este sistema (que se muestra en la figura 13) está compuesto por una plataforma con 1 grado de libertad controlada por un motor de corriente continua sobre la que se monta una estructura de 2 pisos. Además de disponer de 1 encoder para obtener la posición, la mesa tiene 3 acelerómetros que proporcionan la aceleración de la plataforma y de cada uno de los 2 pisos.

Para establecer el control se han desarrollado distintos esquemas Simulink (controladores tipo PID, de realimentación del estado, no lineales, etc.) a partir de los cuales se obtiene el código ejecutable en tiempo real que se descarga en el WCC. Para facilitar la interacción continua entre el operario remoto y el

proceso a controlar se han generado también diversos *WinCon Control Panel*, que son unos paneles gráficos que permiten al usuario cambiar en tiempo real el valor de los distintos parámetros del control, como son los valores de las referencias (tipo de referencia, frecuencia de la señal, amplitud, etc.) y el tipo y los parámetros del controlador. La figura 14 muestra un panel de control desarrollado. En este caso el usuario dispone de una serie de botones y de barras deslizantes para cambiar de una forma muy simple e intuitiva los parámetros del control.



Fig. 13: Prototipo Shaker II, de Quanser Consulting

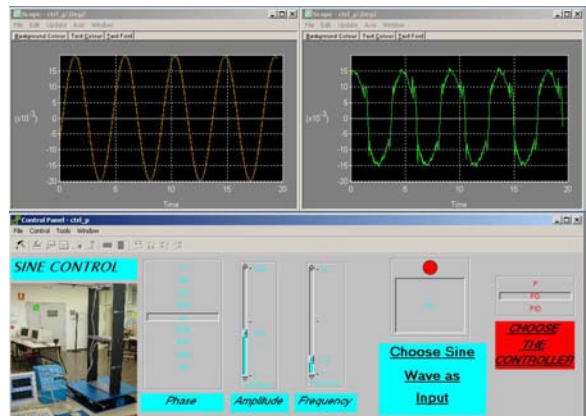


Fig. 14: Panel de control de WinCon desarrollado

8. CONCLUSIONES

Los laboratorios virtuales y remotos proporcionan una herramienta experimental innovadora a los profesores de automática y control de procesos que les permite optimizar los recursos de los laboratorios (compartiendo los procesos físicos y el software), desarrollar material docente innovador (haciendo uso de metodologías pedagógicas avanzadas), o reforzar los conocimientos de los estudiantes en un entorno tiempo-espacio flexible.

El trabajo ha presentado dos soluciones basadas en las aplicaciones de *Matlab Web Server* y *WinCon* para el desarrollo de laboratorios virtuales y remotos.

MWS es una herramienta muy interesante para el desarrollo de laboratorios virtuales, que hace que aplicaciones de Matlab sean accesibles desde cualquier computador remoto, necesitando para ello únicamente un buscador Web. Para hacerlo se tienen que generar

dos ficheros HTML: uno inicial con el que se proporcionan los parámetros de entrada a la función de Matlab, y un fichero final donde se mostrarán los resultados (en forma de gráfico, matriz de resultados etc.) de la ejecución remota.

Sin embargo, *MWS* tiene algunas limitaciones para el desarrollo de laboratorios remotos. Por ello, o se programan una serie de funciones software para el acceso a las tarjetas de adquisición de datos, tal como se ha propuesto en el artículo, o se debe utilizar otra aplicación. En este caso se propone la herramienta de *WinCon*, puesto que los controladores en tiempo real se generan directamente a partir de esquemas de Simulink. Además, los paneles de control que se pueden desarrollar con este software hace que el cambio de los parámetros de los controladores se pueda realizar de una forma muy simple.

El artículo también ha mostrado como se pueden generar desde Matlab ficheros de animaciones con formato *avi*. Con las animaciones y las ecuaciones físicas del sistema se ha obtenido un nuevo concepto: los *procesos virtuales*. Un proceso virtual permite extender y completar los laboratorios virtuales y remotos de *Ingeniería de Sistemas y Control*.

En el trabajo se han presentado tres ejemplos prácticos: el control de posición y velocidad de un motor de corriente continua, el control de posición de un robot con 2 grados de libertad y el control de una mesa vibratoria. Se puede acceder a éstos y otros ejemplos en la dirección <http://globelab.isa.upv.es>. En todos los casos el desarrollo de la herramienta fue muy simple y rápido. Además, como suele ser habitual en este tipo de laboratorios remotos, se ha incorporado una cámara que permite observar la evolución real del sistema industrial.

Cabe destacar por último que los laboratorios virtuales y remotos presentados en el trabajo se han estado testeando durante los 2 últimos años académicos, utilizándose como una herramienta de apoyo para el profesorado de diversas asignaturas de la Universidad Politécnica de Valencia. Así, por ejemplo, en asignaturas como Introducción al Control por Computador (Escuela Técnica Superior de Ingenieros Industriales) o Laboratorio de Informática Industrial (Facultad de Informática) se ejecutaba de forma remota los controladores sintonizados y/o calculados por los alumnos, analizándose posteriormente la respuesta obtenida por cada uno de los controladores. Esto suele ser muy gratificante para los alumnos puesto que, además de comprobar los resultados sobre un sistema real que pueden observar a través de una cámara web, les proporciona una motivación adicional ya que se introduce una especie de competición entre los distintos grupos de alumnos para ver quién es capaz de proporcionar el mejor controlador.

Una vez superada esta fase de test, se espera que próximamente los laboratorios virtuales y remotos estén completamente a libre disposición de los alumnos matriculados en las asignaturas del área de conocimiento de Ingeniería de Sistemas y Automática de la Universidad Politécnica de Valencia.

Agradecimientos

Los autores desean expresar su agradecimiento al Plan Nacional de I+D, Comisión Interministerial de Ciencia y Tecnología (CICYT) por la financiación parcial de este trabajo bajo los proyectos DPI2002 04432-C03-02 y DPI2002 04432-C04-04

REFERENCIAS

- Bernstein, D.S. and H Ashrafiuon (2004). "Innovations in Undergraduate Control Education", *IEEE Control Systems Magazine*, **24**, ISSN 0272-1708, p 18.
- Carreira F.P.N.F., P.M. Silva, J.M.F. Calado (2004). Remote Virtual Lab – A Control Systems Approach. En: *IFAC-Internet Based Control Education IBCE2004*.
- Díez, J.L., M. Vallés, A. Valera, J. L. Navarro (2001). "Remote Industrial Process Control with Matlab Web Server". En: *IFAC-Internet Based Control Education IBCE2001*, ISBN: 0 08 043984 5, pp. 139-143. Ed. Elsevier.
- Díez, J.L., M. Vallés, A. Valera (2002). "A Global Approach for the Remote Process Simulation and Control", *15th IFAC World Congress of Automatic Control*, ISBN: 008044184X.
- Dormido, S. (2002). "Control Learning: Present and Future", *15th IFAC World Congress of Automatic Control*, ISBN: 008044184X.
- Johansson M., M. Gäfvert, K. J. Aström. (1998). "Interactive Tools for Education in Automatic Control", *IEEE Control Systems Magazine*, Vol. **18**, pp. 33-40.
- Microsoft Corp. (2000). "Microsoft Windows Media™ JumpStart CD".
- Overstreet J. W., A. Tzes. (1999). "An Internet-Based Real-Time Control Engineering Laboratory", *IEEE Control Systems Magazine*, Vol. **19**, pp. 19-34.
- Paproch, K. (1998). *Distance Learning: The Ultimate Guide*. London, Sage Publications.
- Poindexter S.E., B.S. Heck. (1999). "Using the Web in your Courses: What can you do? What should you do?", *IEEE Control Systems Magazine*, Vol. **19**.
- Puerto R., L. M. Jimenez, O. Reinoso, C. Fernandez, R. Neco. (2004). "Remote Control Laboratory Using Matlab And Simulink: Application To A Dc Motor Model." En: *IFAC-Internet Based Control Education IBCE2004*.
- Quanser Consulting (2002). *WinCon 4.1: Hard Real-time Performance at your Fingertips (User's Guide)*.
- The MathWorks Inc. (1999). *Matlab Web Server user's guide*.
- Valera, Á., M. Vallés, J. Tornero. (2001). "LabConRob: Virtual Laboratory of Real-Time Robot Control", *Robotica*. **43**. ISSN 0874-9019, pp. 40-45
- Vallés, M., J. L. Díez, A. Valera. (2001). "Setting up a Virtual Matlab Control Laboratory", *IFAC-Workshop EDCOM2001: Education in Automatic Control*. Hohannes Kepler University Linz, Austria.
- Vallés, M., A. Valera, J. L. Díez. (2002). "Desarrollo y Control de Procesos Virtuales mediante Matlab Web Server", *Enseñanza vía Internet/Web de la Ingeniería de Sistemas y Automática EIWISA'02*, ISBN: 84-8454-159-2.