

LABORATORIOS VIRTUALES Y REMOTOS PARA LA PRÁCTICA A DISTANCIA DE LA AUTOMÁTICA

S. Dormido, J. Sánchez, F. Morilla

Dpto. de Informática y Automática, UNED, Avda. Senda del Rey s/n, 28040 Madrid.

Teléfono: 34-91-3987146 Fax: 34-91-3986697

E-mail: sdormido@dia.uned.es, jsanchez@dia.uned.es, fmorilla@dia.uned.es

Resumen

La aparición del WWW ha producido una auténtica revolución en el campo de la enseñanza de manera que en la actualidad muchas materias están siendo reevaluadas en términos de sus propios métodos, técnicas y filosofías. Obviamente las disciplinas que utilizan como un medio para su enseñanza la simulación por computador, y entre las cuales se encuentra en una posición predominante el Control Automático, no han sido una excepción a este fenómeno. Uno de los conceptos más atractivo que ha surgido ha sido el de la "simulación basada en el WWW" que permite plantear nuevos paradigmas de enseñanza distribuida y remota. En este trabajo se exponen los enfoques que dentro de este contexto ha venido desarrollando en los últimos años el Departamento de Informática y Automática de la UNED. En concreto se muestra una nueva concepción de laboratorios virtuales y remotos para la enseñanza de la Automática.

Palabras Clave: Simulación basada en WWW, laboratorios virtuales, laboratorios remotos, docencia en Control Automático

1 INTRODUCCIÓN

En la sociedad actual en que nos encontramos inmersos, la educación a distancia se presenta como la solución idónea para un conjunto de colectivos que exigen el disponer de sistemas de enseñanza mucho más flexibles, accesibles y adaptativos (sin limitaciones espaciales ni temporales). Si bien hasta el momento presente, el modelo educativo de educación a distancia se ha basado, fundamentalmente, en las tutorías telefónicas o en el correo ordinario, las nuevas tecnologías de la información se presentan como las formas alternativas de mejorar la interacción profesor-alumno. Dentro de todo el conjunto de nuevas tecnologías, destacan dos herramientas como las más adecuadas para ofrecer un nuevo enfoque en el modelo educativo de la educación a distancia: los *sistemas hipermedia* como forma

de estructurar la información, y las *redes de comunicación de área extendida* como soporte de la información, es decir, la red Internet.

Pese a que estas dos herramientas se consideran suficientes para la creación de sistemas de apoyo al aprendizaje de materias dotadas de una componente práctica no muy fuerte, la enseñanza del Control Automático, o de alguna otra disciplina con gran contenido experimental, requiere de algo más, de un elemento que permita al estudiante poner en práctica todos los conocimientos que vaya adquiriendo a lo largo del estudio de la materia. Este papel en las enseñanzas tradicionales lo desempeña el *laboratorio de prácticas* (Kheir, *et al.*, 96; Antsaklis, *et al.*, 99), el cual, inexorablemente, requiere de la presencia física del estudiante para poder manipular los sistemas de control y las plantas existentes en un entorno controlado bajo la supervisión del profesor. Por consiguiente, trasladando este entorno práctico a la enseñanza a distancia, el elemento necesario para abordar la realización de prácticas sobre Control Automático es la existencia de un sistema de apoyo a la enseñanza de la Automática consistente en un *laboratorio virtual y de telepresencia accesible a través de una red basada en protocolos TCP/IP que permita al alumno practicar de una forma lo más similar posible a como si estuviese en las dependencias del laboratorio, dándole la posibilidad de manejar las simulaciones o interactuar con las plantas reales.*

En consonancia con lo anterior, el Departamento de Informática y Automática de la Universidad Nacional de Educación a Distancia está trabajando en el desarrollo de nuevos paradigmas de laboratorios para la realización a través de Internet de experiencias prácticas de Control Automático sobre plantas y sistemas reales o simulados.

La exposición del trabajo se ha organizado de la forma siguiente: En la Sección 2 se presentan brevemente algunas consideraciones sobre el papel que debe jugar la interactividad en la enseñanza del Control Automático. La Sección 3 se dedica a plantear lo que significa el nuevo concepto de

simulación basada en el WWW (SBW) y se proporcionan criterios de clasificación que permiten discernir las diferentes arquitecturas que se han propuesto. En la Sección 4 se describe nuestra propuesta de *Laboratorio Virtual*. La SBW del Laboratorio Virtual (simulación remota) se presenta en la Sección 5. En la segunda parte del trabajo se expone como ampliar la idea del Laboratorio Virtual para convertirlo en un auténtico Laboratorio Remoto para la realización de prácticas reales de Control Automático a través de Internet. En la Sección 6 se describe el entorno de telepresencia del Laboratorio Remoto. Los problemas de visualización remota se abordan en la Sección 7. Finalmente en la Sección 8 se exponen algunas propuestas para el seguimiento y evaluación de los alumnos dentro del entorno que se ha desarrollado.

2 LA INTERACTIVIDAD EN LA ENSEÑANZA DEL CONTROL

Para diseñar sistemas técnicos o simplemente para comprender las leyes físicas que describen su conducta, los científicos e ingenieros a menudo utilizan computadores para calcular y representar gráficamente diferentes magnitudes. En el campo del Control Automático, estas cantidades incluyen entre otras las siguientes: respuestas en los dominios del tiempo y de la frecuencia, localización de los polos y ceros en el plano complejo, diagramas de Bode, Nyquist y Nichols, lugar de las raíces, plano fásico, etc. Con frecuencia, estas magnitudes están relacionadas entre si ya que constituyen diferentes visiones de una misma realidad. La comprensión de estas relaciones es una de las claves para lograr un buen aprendizaje de los conceptos básicos y permite al alumno estar en disposición de realizar diseños de sistemas de control automático con cierto sentido.

Tradicionalmente, el diseño de los sistemas se realiza siguiendo un proceso iterativo. Las especificaciones del problema no suelen utilizarse directamente para calcular el valor de los parámetros del sistema ya que no hay una fórmula explícita que los relacione directamente. Esta es la razón de dividir cada iteración en dos fases. La primera, conocida como *síntesis* consiste en calcular los parámetros desconocidos del sistema basándose para ello en un conjunto de *variables de diseño* (que están relacionadas con las especificaciones). Durante la segunda fase, llamada *análisis*, se evalúa el comportamiento del sistema y se compara con las especificaciones. Si no concuerdan, se modifican las variables de diseño y se vuelve a efectuar una nueva iteración. Es posible sin embargo fusionar ambas fases en una sola cuando el efecto de modificar los parámetros produce inmediatamente la actualización de un gráfico. De esta forma el proceso de diseño se hace realmente dinámico y el alumno percibe el

gradiente del cambio del criterio de comportamiento con respecto a los elementos que manipula. Esta capacidad interactiva permite identificar mucho más fácilmente los compromisos que se pueden lograr.

Los recursos de cálculo y los entornos de programación que se disponen en la actualidad han permitido que surjan una *nueva generación de programas para el aprendizaje interactivo del control automático*. Estas herramientas se basan en objetos que admiten una manipulación gráfica directa. Durante estas manipulaciones, los objetos se actualizan inmediatamente, de manera que las relaciones entre los objetos se mantienen en todo momento. *Ictools* y *CCSdemo*, desarrollados en el Departamento de Control Automático del Lund Institute of Technology [Johansson *et al.*, 98], [Wittenmark *et al.*, 98], y *Sysquake* en el Institut d'Automatique de la Escuela Politécnica Federal de Lausanne [Piguet *et al.*, 99] son magníficos ejemplos de esta nueva filosofía educativa del Control Automático. Para los que inician su formación en este campo muchos de los conceptos no son muy intuitivos cuando se abordan por primera vez ya que sus propiedades se expresan en dos dominios diferentes: los dominios temporal y frecuencial. La conducta transitoria (p. ej. tiempo de asentamiento y sobreelongación) y el riesgo de saturación se analizan típicamente en el dominio del tiempo; mientras que conceptos como estabilidad, rechazo al ruido y robustez se expresan más fácilmente en el dominio de la frecuencia. Los mecanismos básicos que los relacionan y otros fenómenos como por ejemplo los efectos del muestreo y la inclusión de no linealidades por citar unos pocos se pueden ilustrar de una manera muy efectiva utilizando estas herramientas.

3 SIMULACIÓN BASADA EN WWW

La simulación basada en el WWW (SBW) constituye, dentro del extenso campo de la simulación, un nuevo paradigma que está surgiendo con fuerza desde hace 5 años debido a la explosión de las nuevas tecnologías de la información y de las comunicaciones y, concretamente, del World Wide Web. Se puede afirmar que la presentación oficial de la simulación basada en el Web se realizó en el marco de una de las sesiones de trabajo de la *Winter Simulation Conference* del año 1996 celebrada en San Francisco. Más concretamente, el trabajo de P.A. Fishwick [Fishwick, 96] presentado en esta sesión está considerado como uno de los primeros en describir este nuevo tópico como una realidad que ya estaba presente en las ideas y trabajos de multitud de investigadores vinculados al área de la simulación por computador. Dado el interés que se despertó en la comunidad científica e industrial a raíz de aquel evento, ya en la actualidad, es habitual en los congresos de carácter internacional organizados por

diferentes organizaciones científicas internacionales tales como IFAC, IEEE y SCS) encontrar conferencias dedicadas exclusivamente a las múltiples facetas de la simulación basada en el WWW.

P.A. Fishwick define escuetamente la SBW como “una convergencia de metodologías y aplicaciones de simulación por computador dentro del WWW”. Dos son los aspectos en donde la convergencia e interrelación de ambos campos debe ser total:

1. *Enseñanza.* De forma pareja al resto de las disciplinas científicas, el WWW se contempla en el campo de la simulación como un medio clave para distribuir y universalizar la información contenida en las herramientas de simulación. A su vez, destaca la necesidad de potenciar el empleo de las diferentes tecnologías asociadas con Internet (realidad virtual, videoconferencia y multimedia) para potenciar el aspecto didáctico de los materiales educativos y facilitar de esta forma el proceso de aprendizaje a los estudiantes.
2. *Programas de simulación.* Es aquí en donde el WWW y el campo de la simulación forman una simbiosis perfecta. Gracias a Internet, la interfaz gráfica con un software de simulación puede ser manipulada desde cualquier lugar del mundo con tan sólo disponer de un navegador. Incluso la posibilidad de realizar simulaciones distribuidas y posiblemente paralelas adquiere todo su significado al disponer de una vasta red mundial de computadores interconectados por medio de protocolos no propietarios. Pensemos, tal y como sugiere Fishwick, en la reutilización de pequeños modelos para construir grandes sistemas. Mediante el WWW y las actuales técnicas de programación orientada a objetos es perfectamente viable la distribución de componentes a lo largo de la red de forma que en cada computador se podría simular el comportamiento de una parte concreta.

Otro factor común a las simulaciones basadas en Web, derivado de la propia naturaleza del WWW, es que se sustentan en la arquitectura cliente-servidor [Halsall, 98] como medio para transmitir la información al cliente desde una ubicación remota. Esta información puede estar compuesta por el propio programa de simulación que se transmite al cliente para ejecutarse dentro de su navegador WWW, o por los resultados de la simulación efectuados bien en el cliente o en el servidor para su posterior análisis.

De forma más concreta se puede definir la SBW como *el empleo de los recursos y tecnologías ofertados por el World Wide Web para la interacción*

con herramientas de simulación localizadas tanto en el lado del cliente como en el del servidor. Por lo tanto, la característica común de todas las aplicaciones aglutinadas bajo este tópico es que recurren a los navegadores WWW como soporte para las interfaces gráficas que conectan al usuario con la simulación. Es importante observar que la transmisión explícita de un software de simulación desde un servidor al ordenador del cliente y su ejecución como una aplicación independiente del navegador no se enmarca dentro de la categoría de SBW: el navegador WWW siempre tiene que tomar parte activa en el proceso de simulación, bien como mero interfaz gráfico o, adicionalmente, como contenedor del motor numérico de simulación.

Nuestro interés se centra en la fusión de los dos aspectos anteriormente considerados (enseñanza y programas de simulación) para la creación de un laboratorio virtual sobre Internet basado en simulaciones con fines educativos en el campo del Control Automático.

3.1 CRITERIOS DE CLASIFICACIÓN

Pese a que la definición dada de este emergente tópico puede parecer bastante intuitiva es necesario profundizar en ella realizando una clasificación de las diferentes formas de diseñar SBW atendiendo a cuatro criterios:

- *Ubicación del motor matemático de cálculo*
- *Naturaleza del núcleo de simulación*
- *Capacidades de diseño*
- *Grado de interactividad con la simulación*

Acorde con el primer criterio, la *ubicación del motor de cálculo*, se puede discernir entre *local* o *remota*. Dentro del ámbito en que nos movemos, las simulaciones locales se caracterizan porque el motor de cálculo se transmite al computador en el que está trabajando el cliente, de forma que la interfaz gráfica y el núcleo numérico forman un todo que convive en el mismo entorno, es decir, dentro del navegador. Para el desarrollo de tal tipo de simulaciones, Java constituye actualmente la única alternativa capaz de proporcionar total independencia de la plataforma hardware sobre la que se esté trabajando [Carlson *et al.*, 96] [Chatterjee *et al.*, 97] [Alfonseca *et al.*, 99]. También hay investigadores que, aun trabajando en modo local, mantienen una explícita separación entre el motor y la interfaz [Schmid, 97] [Schmid, 99]. Por el contrario en las *simulaciones remotas* el núcleo numérico se ejecuta en un ordenador remoto, no siendo necesariamente en aquél desde el que se ha descargado la página HTML que alberga la interfaz gráfica, aunque lo habitual es que sí lo sea. En cualquier caso, la interfaz gráfica y la simulación se ejecutan en ordenadores diferentes, pudiéndose

comunicar ambas partes por medio de CGI's [Erkes *et al.*, 96] [May, 96] [DIA, 97] [Georgiev y Hogenboom, 99], sockets [Narayanan *et al.*, 99], Java RMI, CORBA, llamadas a procedimientos remotos mediante otros estándares de comunicación, o aplicaciones *front-end* para el software de simulación (en este caso, ambas aplicaciones *front-end* y servidor residen en el mismo computador ya que lo contrario no tendría sentido). Este último supuesto es el que se ha considerado en el laboratorio virtual que hemos desarrollado: un servidor concurrente recibe órdenes desde las interfaces de los diferentes clientes por medio de sockets, estas órdenes son transmitidas al proceso en el que se ejecuta la instancia de la simulación creada para cada cliente concreto (cada cliente posee su propio espacio de trabajo en el servidor) y los datos generados pro cada instancia son recibidos por el servidor para que a su vez, vía sockets, se los entregue al cliente correspondiente.

El segundo criterio de clasificación es la *naturaleza del núcleo de simulación* con independencia de que su ubicación sea local o remota. Este criterio considera si la simulación en sí misma ha sido construida por medio de una herramienta específica orientada al modelado y la simulación (Matlab, Simulink, ACSL, Dymola, Ecosim, etc.) [May, 96] [Schmid, 97] [Schmid, 99], o se haya recurrido a lenguajes de alto nivel de propósito general (C, C++, Fortran, Java) mediante el empleo de librerías específicas orientadas a la simulación [Erkes *et al.*, 96] [Georgiev y Hogenboom, 99] [Narayanan *et al.*, 99]. De acuerdo con este criterio de clasificación, la primera aproximación se puede considerar como la idónea ya que la versatilidad, potencia y prestaciones que brindan las diferentes herramientas gráficas de modelado y simulación no son equiparables a las que se pueden obtener por medio de la programación directa en lenguajes de alto nivel de propósito general.

Las *capacidades de diseño* consideran que el cliente pueda cambiar no sólo los parámetros numéricos del modelo a simular, sino que tenga la posibilidad de modificar su arquitectura. De esta forma, el cliente no se limita únicamente a la introducción de parámetros para configurar el comportamiento del modelo sino que toma parte activa en la construcción del propio modelo. Algunos ejemplos de este tipo de SBW se pueden encontrar en los trabajos de [Boroni *et al.*, 97] y [Chatterjee *et al.*, 97]. En nuestro desarrollo, el usuario puede tomar parte activa en el diseño de los modelos mediante la manipulación de ficheros de parámetros. Mediante éstos, el usuario no sólo se limita a modificar el aspecto y comportamiento de la interfaz gráfica sino que puede modificar las características físicas del objeto simulado y, por lo tanto, cambiar toda la dinámica del sistema.

La cuarta y última característica que distingue las diferentes aproximaciones que existen para la realización de simulaciones basadas en el Web es el *grado de interactividad con la simulación*. Podemos distinguir dos casos: *pseudo-batch* y *on-line*. En la *simulación pseudo-batch* no hay inmediatez desde que se inicia el proceso de simulación hasta que se obtiene la respuesta en forma de datos numéricos o gráficos. En algunos casos la respuesta del sistema puede corresponder a pasos intermedios del proceso de simulación. Por lo tanto, es posible introducir ciertos parámetros y repetir alguno de los pasos intermedios. Algunos ejemplos se pueden encontrar en [Fishwick, 95] [May, 96] [DIA, 97] y [Georgiev y Hogenboom, 99].

La *simulación on-line* representa el polo opuesto pero, a su vez, es el más sugerente. En este caso, el proceso de simulación avanza de forma continua y dinámica, obteniendo el usuario en cada periodo de muestreo de tiempo simulado los resultados bajo la forma de un flujo continuo de valores numéricos o de gráficos evolucionando de forma sostenida. Otra gran diferencia con respecto al supuesto previo es que, según se modifica un parámetro en la interfaz, la respuesta del sistema es inmediata (evidentemente, la rapidez de la reacción depende de la magnitud del cambio y de la dinámica del modelo). Dadas las características de este tipo de simulaciones, la herramienta por excelencia es el lenguaje Java, aunque también es posible encontrar algunos desarrollos bajo la forma de controles ActiveX [National Instruments]. Otros ejemplos se pueden encontrar en [Carlson *et al.*, 96] [Esquembre *et al.*, 96] [Alfonseca *et al.*, 99] y [Narayanan *et al.*, 99].

Resumiendo, con independencia de las posibilidades de diseño y del grado de interactividad, las SBW más frecuentes son tres:

- La interfaz gráfica y motor de simulación constituyen una aplicación monolítica ejecutándose dentro del navegador WWW y residiendo en el ordenador del cliente (Figura 1a).
- La interfaz gráfica y el motor son aplicaciones independientes residiendo ambas en el mismo computador, es decir, en el del cliente (Figura 1b). La interfaz se localiza dentro del navegador WWW, mientras que el motor habitualmente es un entorno de simulación del tipo Matlab/Simulink o ACSL.
- La interfaz gráfica y el motor son aplicaciones independientes y están físicamente separadas (Figura 1c). Como en el caso previo, la interfaz se localiza en el navegador WWW del cliente pero la simulación reside en el servidor remoto, pudiendo estar constituida por una aplicación

desarrollada para tal efecto o por herramientas de modelado y simulación. En ambos casos, la comunicación con la interfaz se puede realizar de alguna de las formas ya citadas anteriormente: front-ends bajo la forma de CGI o de servidores *ad-hoc* para su comunicación con la interfaz, sockets, etc.

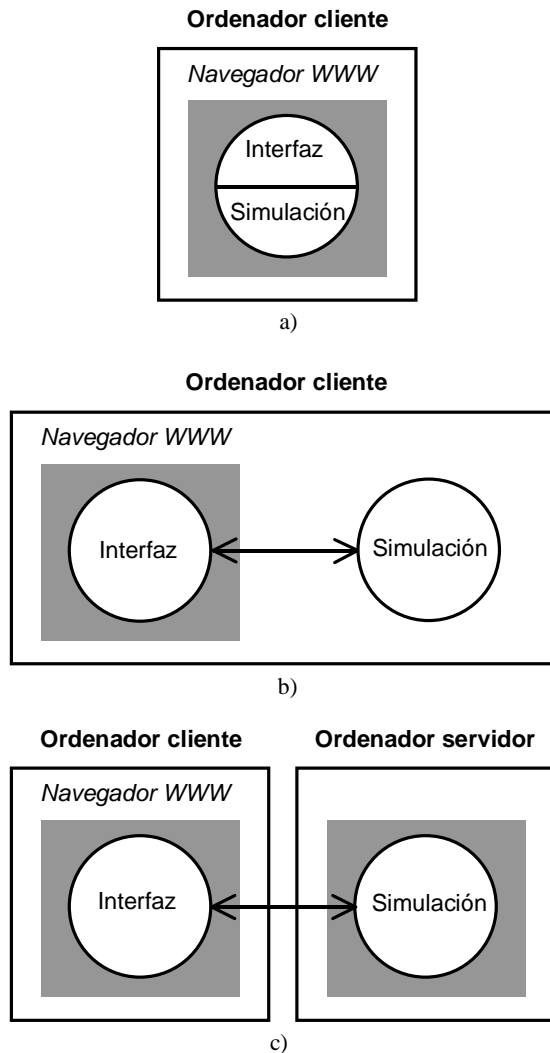


Figura 1 Configuraciones más habituales de SBW

Las herramientas para el desarrollo de las interfaces gráficas suelen ser controles ActiveX (limitados al empleo de navegadores de Microsoft y a plataformas Windows), applets Java (cualquier tipo de navegador con soporte Java), VRML (cualquier navegador con su correspondiente intérprete de VRML) para el desarrollo de interfaces gráficas tridimensionales, y lenguajes de scripts como JavaScript o JScript cuando se realiza la comunicación con la simulación remota por medio de formularios y CGIs (habitualmente simulaciones de carácter pseudo-batch).

Evidentemente, las definiciones y los criterios anteriores intentan abarcar de forma estructurada la

mayor parte de las diferentes formas de realizar este tipo de simulaciones, pero hay algunos casos que no se ajustan exactamente a las definiciones dadas. Por ejemplo, ¿podemos considerar dentro de este paradigma las aplicaciones independientes que se conectan directamente a través de Internet con un ordenador remoto que contiene el motor de cálculo, tal como sucede en [Gillet *et al.*, 97]? Desde nuestro punto de vista, sí. La razón es que, aunque la interfaz no forme parte de un navegador Web, esto no indica que la arquitectura del sistema no se base en los mismos fundamentos e ideas que las SBW. Por otra parte, debido a que el lenguaje Java se ha convertido en un estándar de facto en este tipo de desarrollos, su empleo para la realización de interfaces de usuario permite con suma facilidad que un mismo fichero se puede comportar tanto como una aplicación independiente como un applet incrustado en la página HTML de un navegador.

Concretando lo hasta aquí expuesto y de acuerdo con los criterios que se han definido para la clasificación de las SBW, el desarrollo de nuestro laboratorio virtual que se presenta en la siguiente sección se caracteriza por ser completamente on-line, con capacidades de diseño, con ubicación remota del motor de simulación y estando éste constituido por un entorno específico de modelado y simulación (Matlab/Simulink).

4 LABORATORIO VIRTUAL

En el laboratorio virtual que se ha desarrollado, en su primera versión, para la realización de simulaciones dinámicas e interactivas han prevalecido los aspectos cualitativos sobre los cuantitativos. Tal como se muestra en la Figura 2 el entorno del laboratorio virtual consta de dos partes perfectamente diferenciadas.

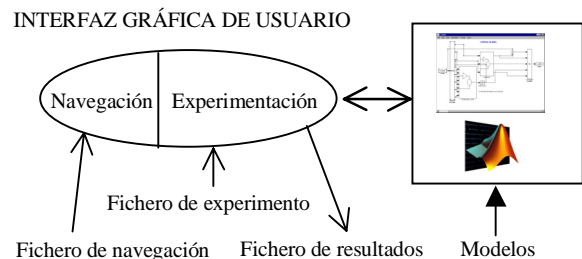


Figura 2 Estructura del laboratorio virtual

- Una *potente interfaz gráfica de usuario (GUI)* para interactuar con los elementos propios de un laboratorio de Control. Básicamente la interfaz está constituida por los esquemas de determinados procesos industriales (plantas, controladores, tuberías, válvulas, etc.) más un conjunto de diagramas de señal para analizar la evolución de

los parámetros y variables a lo largo del tiempo de simulación.

- Un *entorno de cálculo matemático* en el que se ejecutan las simulaciones de las plantas que son de interés por su carácter didáctico: intercambiadores de calor, péndulos invertidos, tanques de líquido, columnas de destilación, etc. Como el entorno ha sido concebido tanto para la enseñanza como para el entrenamiento de operarios de plantas industriales, éste permite realizar una completa configuración de los modelos matemáticos (perturbaciones programadas, parámetros físicos, actuación de los controladores, etc.) y de las interfaces gráficas (rango de las variables, grado de interactividad de las variables) por medio de ficheros de experimentos y, de esta forma, establecer los objetivos pedagógicos que los alumnos tengan que alcanzar interactuando con la GUI.

4.1 EL ENTORNO DE EXPERIMENTACIÓN

Para cubrir las necesidades de los dos usuarios potenciales, profesor y estudiante, la GUI tiene dos componentes interrelacionadas, las ventanas de *navegación* y de *experimentación* ambas configurables mediante parámetros.

El componente de navegación permite que los experimentos se ajusten a la estructura jerárquica de un libro de texto o de un curso. Se consideran tres niveles de navegación: capítulos, lecciones y experimentos. De esta forma tan simple, el profesor será capaz de adecuar el entorno para enseñar un curso con un cierto perfil y el alumno podrá seleccionar un experimento de entre los existentes de una forma ordenada de acuerdo con los conceptos o situaciones que desee estudiar u observar

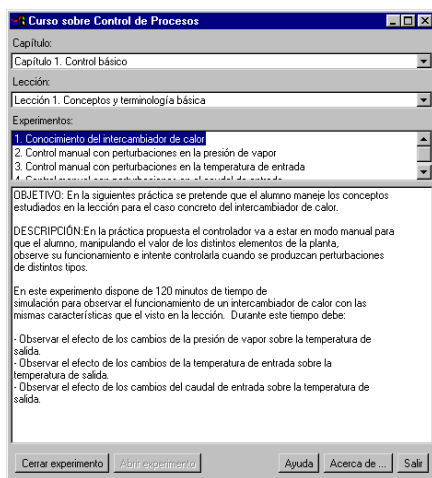


Figura 3 Ventana de navegación.

La Figura 3 muestra la ventana de navegación del entorno desarrollado. Su estructura basada en hiperíndices emplea menús y permite la navegación hasta seleccionar un experimento particular. Con el fin de proporcionar más información al usuario, simultáneamente con la selección de un experimento, se proporciona una descripción textual de los objetivos que se persiguen.

La jerarquía de la navegación así como las descripciones de los experimentos se localizan en un fichero de texto denominado *fichero de estructura de navegación*. Por lo tanto, este fichero debería contener tanto la jerarquía de navegación (capítulos, lecciones y experimentos) y la definición de los experimentos (plantas, fichero de experimentos y fichero de parámetros).

Una vez seleccionado el experimento, el usuario inicia la práctica mediante la interfaz de experimentación gráfica. El aspecto visual de la interfaz así como la conducta del proceso habrán sido previamente establecidos por el profesor de forma que el experimento se focaliza hacia la consecución de un cierto objetivo de acuerdo con la descripción del experimento que se muestra en la ventana de navegación.

4.2 LA INTERFAZ DE EXPERIMENTACIÓN

Con el fin de reforzar el dinamismo y los aspectos cualitativos la Interfaz de Experimentación Gráfica (GEI) se caracteriza por su interactividad con la simulación, la visualización dinámica, la animación de los elementos (cuando el sistema lo permite) y el registro histórico de las variables y de los eventos. La GEI consta de los siguientes elementos: el diagrama del proceso, los paneles de control, los registros gráficos, el panel multiseñal y el fichero de resultado. A continuación se describen cada uno de ellos:

Planta	Variables interactivas	Variables no interactivas
Intercambiador de calor con control básico	Presión de vapor (bar.) Temperatura de entrada (°C) Caudal de entrada (m ³ /h) Control manual (%) Set point de temperatura (°C)	Tiempo de simulación (min.) Caudal vapor entrada (Tm/h) Temperatura de salida (°C) Apertura de válvula (%)
Intercambiador de calor con control en cascada	Presión de vapor (bar.) Temperatura de entrada (°C) Caudal de entrada (m ³ /h) Set-point caudal vapor (Tm/h) Control manual de TC (%) Control manual de FC (%) Set point de temperatura (°C)	Tiempo de simulación (min.) Caudal vapor entrada (Tm/h) Temperatura de salida (°C) Apertura de válvula (%)
Control de nivel de un tanque	Caudal de entrada (m ³ /h) Set-point de nivel (%) Control manual de LC (%)	Tiempo de simulación (min.) Nivel del tanque (%) Caudal de salida (m ³ /h.) Caudal de rebosé (m ³ /h)
Control básico y avanzado de una columna de destilación	Caudal de carga (m ³ /h) Composición (%C3) Set-point temp. cabeza (°C) Set-point temp. fondo (°C) Control manual reflujo (m ³ /h) Control manual vapor (m ³ /h) L/D control manual V/F control manual	Tiempo de simulación (min.) Temperatura de cabeza (°C) Temperatura de fondo (°C) Ímpurezas en cabeza (%C4) Destilado (m ³ /h.) Ímpurezas en el fondo (%C3) Caudal de fondo (m ³ /h.)

Tabla 1. Señales de proceso de algunos modelos.

a) *Diagrama de proceso.* Está compuesto por un esquema gráfico del proceso (algunos de los objetos presentes pueden tener animación) con visualización alfanumérica de las señales y unidades más importantes, más un esquema de la estrategia de control con acceso a los parámetros y a los cambios de modo de los controladores. En las Figuras 4, 5 y 6 se muestran los diagramas de proceso de dos plantas y en la Tabla 1 se recogen las señales más importantes de cada una de ellas.

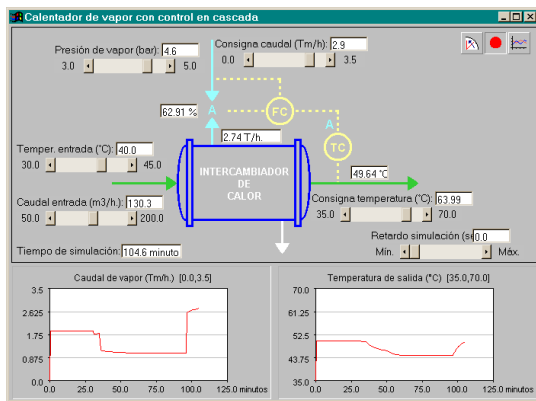


Figura 4 Control en cascada de un intercambiador

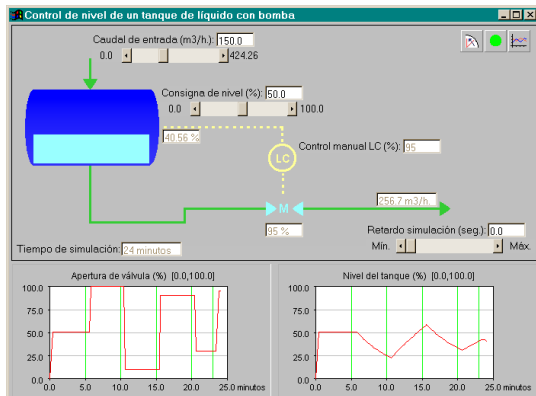


Figura 5 Control de nivel de un tanque

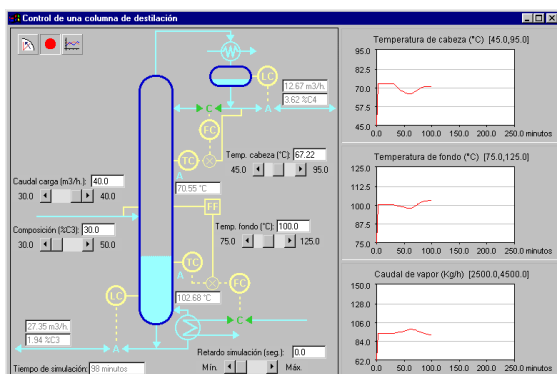


Figura 6 Control de una columna de destilación

Un aspecto importante de estos diagramas es que puede cambiar la representación esquemática de los elementos físicos. Por ejemplo, el tanque de la Figura 5 se puede configurar como vertical u horizontal, con la existencia o no de una bomba de drenaje, y aunque es el mismo, sus características varían completamente.

b) *Paneles de control.* Tienen tres tipos de elementos (botones, deslizaderas y campos) que se pueden agrupar en tres categorías:

- *Panel de control.* Permite la parada, continuación y re arranque de la simulación del proceso. Se localiza en la parte superior de la interfaz.

- *Panel de variables interactivas.* Se extienden sobre todo el diagrama y consiste en deslizaderas y campos alfanuméricos para modificar el valor de las perturbaciones, los puntos de consignas y las señales de control manual. Con el fin de permitir que el tutor guíe las acciones del alumno, el profesor puede configurar la forma en que el usuario interactúa con este panel. Hay tres configuraciones posibles para cada variable interactiva: totalmente escondida, visible pero no modificable, visible y modificable (ver la segunda columna en la Tabla 1).

- *Paneles de los controladores.* Estos paneles o ventanas están generalmente escondidas y se muestran cuando el usuario pulsa sobre el símbolo de un controlador (en nuestro caso, el controlador se representa mediante un círculo con dos letras en su interior que indican la variable de control: LC controlador de nivel, TC controlador de temperatura, y FC controlador de caudal). Tal como se puede ver en la Figura 7, es posible modificar los parámetros de los controladores de una forma interactiva así como el modo de operación (manual, automático o cascada).

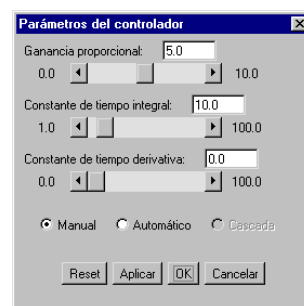


Figura 7 Panel del controlador

Cuando sucede con las variables interactivas, y de acuerdo con los objetivos del experimento, el profesor puede modificar la forma en la que el usuario interactúa con los controladores: rangos, valores iniciales de los parámetros, modo inicial de control, acción de control directa o inversa, etc.

- c) *Registros gráficos.* Proporcionan la visualización gráfica de las variables principales del sistema. Al mismo tiempo cuando la simulación progresa, estos registros reflejan de forma dinámica y continua cualquier cambio en las variables que toman parte en el proceso. Los cambios en las señales del sistema se pueden deber a las acciones del usuario (movimiento de una deslizadera en el diagrama del proceso) o a perturbaciones preprogramadas por el profesor para este experimento. En el fondo de la Figura 4, se puede apreciar la evolución temporal de dos señales del intercambiador de calor.
- d) *Panel multiseñal.* Con la intención de dar al usuario una visión global, se ha dotado a la interfaz gráfica de un panel especial que permite mostrar en una única ventana todas las señales y perturbaciones (ver Figura 8).

Un aspecto importante de todos estos registros gráficos es que cualquier evento preprogramado por el profesor se presenta en la ventana como una línea vertical. De esta forma el usuario puede saber como y cuando ha sido introducida en el sistema una perturbación y como este ha evolucionado a partir de ese instante.

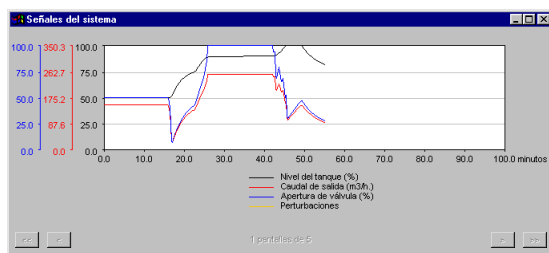


Figura 8 Panel de registro multiseñal

- e) *Registro histórico de eventos y acciones.* Posteriormente el alumno o el profesor podrán utilizar una herramienta de análisis para estudiar de una forma cuantitativa todo lo acontecido durante el proceso de simulación.

4.3 PROPUESTAS DE EXPERIMENTOS

Una vez han sido descritos los atributos principales de la interfaz de experimentación (dinamismo, interactividad y configuración), es conveniente enumerar alguno de los ejercicios que el profesor

puede proponer a los estudiantes con el objetivo de estudiar cualitativamente las características dinámicas de cualquiera de las plantas. Estos ejercicios se pueden agrupar en tres categorías genéricas: a) conocimiento del proceso, b) control manual, y c) control automático. Detallemos cada uno de estos puntos.

- a) *Conocimiento del proceso.* El objetivo de esta clase de ejercicios es familiarizar al usuario con las reacciones del proceso cuando se provocan cambios en las señales de control y en las perturbaciones. Algunos ejercicios posibles de este tipo son:

- *Descripción cualitativa.* Al estudiante se le pide que complete una matriz cualitativa de estados estacionarios del modelo con el que está realizando la práctica. Para realizar este tipo de ejercicios el profesor habrá configurado el experimento con control manual y permitiendo que todas las variables interactivas puedan manipularse libremente sin ningún tipo de restricciones y con los paneles de los controladores desactivados.
- *¿Qué señal de entrada ha cambiado?* El profesor programa un cambio brusco en una señal de entrada (evento preprogramado), y esta variable permanecerá oculta durante todo el experimento. Al alumno se le indica que tiene x unidades de tiempo de simulación para observar la salida del proceso y descubrir que entrada ha cambiado y en que sentido (aumento o disminución). Para configurar este ejercicio el profesor habrá desactivado los paneles de las variables interactivas y los paneles de los controladores.

- b) *Control manual.* El estudiante se ejercita con la planta utilizando el controlador en modo manual. Previamente el profesor habrá desactivado en el controlador la posibilidad de cambiar de modo. Algunos ejercicios posibles son:

- *Contrólolo por usted mismo.* El profesor programa una perturbación en una variable de entrada y se le indica al alumno que tiene x unidades de tiempo de simulación para contrarrestar dicha perturbación por medio de la apertura o cierre de las válvulas que tiene a su disposición. Como en el caso previo, la variable perturbada estará oculta y el alumno solo será capaz de ver el efecto que causa sobre el sistema. Debe observarse que de esta forma se pueden obtener una batería considerable de ejercicios, preprogramando la modificación de otras señales del proceso, variando el número de cambios y sus instantes y el número de señales que se perturban simultáneamente.

- *¿Qué acciones de control debería utilizar?* Se le pide al alumno que analice la conducta del proceso para determinar que tipo de acción (directa o inversa) debería corresponder al controlador.
- c) *Control automático.* En este caso el profesor preprograma los experimentos de forma que el alumno no puede colocar el controlador en modo manual para sintonizar los parámetros del controlador PID si la conducta del proceso no es la apropiada. El alumno se ve forzado a corregirla. Algunos ejercicios de este tipo son:
 - *Objetivo de la acción de regulación.* El experimento se comienza con los controladores en modo automático y se le pide al alumno que realice cambios individuales en todas las variables interactivas (excepto en los puntos de consigna) y que observe si el controlador es capaz de contrarrestar estas acciones con sus parámetros actuales. También se le indica que rellene una matriz cualitativa del estado estacionario del sistema de control.
 - *Objetivo de la acción servo.* El experimento se comienza con los controladores en modo automático y se le pide al alumno que realice cambios en los puntos de consigna y que observe si el controlador es capaz de lograr que las variables del proceso alcancen el nuevo punto de consigna.
 - *Acciones básicas con el controlador PID.* Esta clase de ejercicio tiene como objetivo centrar la atención del alumno en las acciones de control básico: proporcional, integral, y derivativa. Inicialmente, el profesor sintonizará al controlador en modo automático y con control P, PI, o PID. El alumno hará un cambio en las variables interactivas y observará si el controlador es capaz de contrarrestar estas perturbaciones con sus parámetros actuales. Posteriormente se le pide que elimine las perturbaciones para retornar al estado inicial, que modifique y que repita la experiencia previa de introducir perturbaciones.
 - *Sintonía de los parámetros del controlador.* En este grupo existen muchos ejercicios. De una forma genérica dos alternativas posibles son: 1) el controlador se preprograma con algunos parámetros con valores no muy apropiados y se le pide al alumno que introduzca cambios en las variables interactivas y que observe la conducta del sistema; 2) se le pide que elimine las perturbaciones para volver al estado estacionario inicial y que a continuación sintonice los parámetros del controlador para mejorar la conducta global del sistema; debe repetir la

experiencia hasta que obtenga algún buen juego de parámetros del controlador.

- *Análisis de las limitaciones del actuador.* La finalidad en esta clase de ejercicios es que el usuario comprenda que ciertos objetivos de control no pueden ser alcanzados en el mundo real debido a las limitaciones físicas de los componentes del sistema.
- *Estudio del rango de operación.* Este grupo incluye algunos experimentos similares a los de la sección anterior. En este caso se pretende que el alumno descubre el rango de operación de cualquier planta y el hecho de que su sistema de control está siempre limitado.

Algunos de los ejercicios enumerados suponen que el control se lleva a cabo con un controlador PID en modo manual o automático. Por ejemplo esto acontece en el control básico del nivel de un tanque (Figura 5). No se han enumerado los posibles ejercicios sobre control avanzado: en estas prácticas se necesitan dos controladores PID (por ejemplo, el intercambiador de calor con control en cascada) y otros elementos (control de relación y compensación feedforward) tal como sucede en la columna de destilación.

4.4 HERRAMIENTAS SOFTWARE USADAS

Las herramientas de software se pueden agrupar en dos categorías, una para cada uno de los dos elementos que integran el entorno: la interfaz de usuario y el motor de simulación. Para la primera se ha escogido Java por las siguientes razones: (Hamilton, 96):

- La *posibilidad de desarrollar applets y aplicaciones*, así como el proceso simple de conversión de una en otra. Esta característica permite que el software completo desarrollado para su operación de forma independiente se transforme fácilmente en un applet.
- Su *orientación a Internet y sus herramientas APIs para cálculo distribuido*. RMI (Remote Method Invocation) y JavaIDL (Java Interface Definition Language) constituyen una garantía del grado de implicación en el desarrollo de aplicaciones distribuidas.
- Su *orientación a objeto* ofrece todas las características beneficiosas de este paradigma.
- Su *portabilidad* permite que los applets y aplicaciones de Java se puedan ejecutar en cualquier máquina que disponga de Java Virtual Machine.

- La posibilidad de *extenderlo con otros lenguajes de más bajo nivel*. Gracias a JNI (Java Native Interface), es posible desarrollar librerías en C para acceder a dispositivos periféricos o para la programación de sistemas en tiempo real.

Como herramienta para el motor matemático y de simulación se ha empleado Matlab 5.x y Simulink 2.x. La razón para esta elección es que ambas herramientas se pueden considerar de facto como un estándar en el campo del Control Automático y de la Simulación. Pero hay algunas razones más:

- La *gran diversidad de sistemas operativos sobre los cuales se puede usar Matlab*. Gracias a eso, es posible transportar directamente el modelo Simulink directamente de un sistema a otro.
- *Los alumnos pueden adquirir una versión limitada de ambas herramientas a un precio muy asequible*. Todas las simulaciones del laboratorio virtual han sido realizadas con estas versiones, lo que facilita la distribución del entorno entre los alumnos.
- *Matlab se interconecta con lenguajes externos*. Es necesario recalcar la integración con ActiveX, o el empleo de Matlab como un motor numérico de otros lenguajes, ambos en plataformas Windows y Unix.

Centrándonos en ActiveX, Matlab soporta el protocolo Automation de forma que puede ser controlado y a su vez controlar a otros componentes que también soporten ActiveX. Cuando Matlab se controla por otro componente, denominado el cliente, se dice que es un servidor, y cuando controla a otro componente Matlab se transforma en el cliente y el otro componente actúa como servidor. En nuestro caso, Matlab solo interesa trabajando como servidor para poder controlarlo por medio del código de Java desarrollado para la interfaz de simulación.

Aunque el número de herramientas de desarrollo en Java es considerable, se ha escogido el entorno Visual Microsoft Java++. Esto ha estado motivado por la posibilidad de interconectar de una forma sencilla el código de Java con Matlab, gracias al soporte que el entorno de Microsoft hace de ActiveX.

5 SIMULACIÓN REMOTA

El laboratorio virtual que se acaba de presentar se puede transformar en un verdadero SBW (sistema de simulación remota basada en el WWW) de acuerdo con los principios expuestos para este tipo de sistemas en la Sección 3. Los elementos adicionales que hay que considerar son los siguientes:

- Un *servidor concurrente* como mecanismo para el intercambio de información a través de Internet entre los applets de experimentación y el núcleo de cálculo/simulación. Con el objeto de que el entorno ofrezca la posibilidad de que varios estudiantes trabajen simultáneamente, la plataforma software para la que se está desarrollando el prototipo es un sistema operativo *nux (cualquier variante de Unix). Esto viene motivado por la posibilidad de que, gracias a la interfaz de comunicación externa de Matlab para *nux, es posible disponer de tantas sesiones de trabajo Matlab/Simulink como usuarios se conecten al entorno de simulación remoto en un instante dado mediante la creación de procesos Matlab/Simulink asociados en exclusiva a procesos o *threads* denominados PE o *procesos de experimentación* (Figura 9).

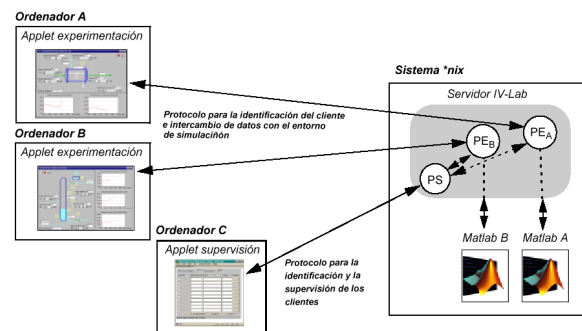


Figura 9 Esquema conceptual del entorno de experimentación remota

- Un *sistema de supervisión y monitorización* con el objeto de que el profesor de forma remota supervise on-line el trabajo que está siendo realizado por los estudiantes mediante los applets de experimentación. Para ello, se ha diseñado un prototipo de *applet de supervisión* que permite monitorizar algunos parámetros de las sesiones de trabajo de los estudiantes: hora y tiempo de conexión, modelos utilizados, comandos emitidos, respuestas del entorno de simulación, etc. Análogamente a la existencia de procesos PE, el servidor cuenta con un proceso o *thread* PS (*proceso de supervisión*), encargado de dialogar con el applet de supervisión y, simultáneamente, de intercambiar datos con los proceso PE por medio de colas de mensajes.
- Un *conjunto de presentaciones conceptuales* embebidas en páginas HTML como forma de proporcionar o completar las bases teóricas necesarias para abordar la realización del trabajo práctico, así como para señalar al estudiante los objetivos que se pretenden alcanzar con la consecución de la práctica.

6 TELEPRESENCIA

Utilizando la misma filosofía que en el laboratorio virtual se ha desarrollado un entorno de telepresencia para el empleo remoto de tarjetas DSP y de ciertas plantas de entrenamiento (péndulo invertido, tanques de agua, intercambiadores de calor, etc). Con el empleo de plantas reales se intenta, en la medida que la tecnología lo permita, transmitir a los estudiantes las sensaciones y responsabilidades que tendrán cuando trabajen en entornos industriales reales. Pero evidentemente, el término *telepresencia* debe ser tomado con las limitaciones propias de la tecnología actual, ya que de los cinco sentidos que posee el ser humano, sólo dos de ellos, la vista y el sonido, pueden ser transmitidos a través de una red a un coste moderado (Aktan, *et al.*, 96). La transmisión de información digital referente a los otros tres sentidos es viable, pero tanto el coste de los transductores necesarios, como la posible adecuación a un marco de experimentación universitario en la enseñanza de la Ingeniería de Control hace que, por el momento, se desestimen.

El esquema conceptual del sistema de telepresencia será similar al del laboratorio virtual: applets Java para las interfaces gráficas de experimentación, applet de supervisión y un servidor *ad-hoc* que haga de unión con el lazo de control en tiempo real. Pero a diferencia de las simulaciones remotas, en que es necesario que el servidor genere procesos PE en exclusiva para cada cliente, el servidor ante una petición de conexión, generará un único proceso que se comunicará en tiempo real con las plantas a través de las tarjetas de adquisición de datos.

La Figura 10 recoge un esquema gráfico resumido del entorno de teleoperación planteado. Debido a la necesidad de dialogar en tiempo real con la planta, nada más activar el servidor, éste inicializa las tarjetas de adquisición de datos e instala el lazo de control en una rutina de interrupción activada por la propia tarjeta de adquisición de datos. Establecido el lazo de control con unos parámetros por defecto, la planta permanecerá en estado estacionario hasta que un applet de experimentación establezca una conexión con el servidor. En ese momento, el servidor creará un proceso PE para atender la conexión y transmitir los parámetros de control desde el applet hasta el lazo de control de la planta, traduciendo las acciones del usuario sobre el applet de experimentación en acciones sobre la planta real. Análogamente, la conexión de un applet de supervisión al servidor ocasionará la creación de un proceso de supervisión PS con capacidades de monitorizar al estudiante, al lazo de control y a las tarjetas de adquisición de datos.

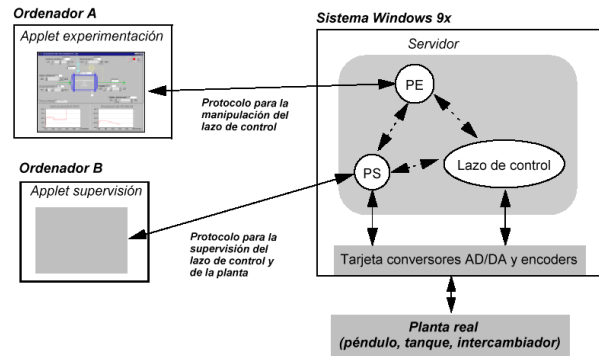


Figura 10 Esquema conceptual del entorno de teleoperación

Obsérvese que con el fin de solventar los problemas de control que implican el retardo en la transmisión de la información a través de la red, se ha recurrido al paradigma de *control supervisado* (Sheridan, 93). Brevemente, el control supervisado significa que en lugar de distribuir el lazo de control entre la planta, el servidor y el applet de experimentación, todo el lazo de control se mantiene cerrado en el propio servidor. De esta forma, no hay retrasos en el lazo que realiza las tareas de control y, por lo tanto, no hay inestabilidades.

De acuerdo con este enfoque, las funciones del control supervisado en nuestro entorno desde el punto de vista didáctico se pueden dividir en cinco puntos, los dos primeros propios del profesor y los tres restantes del estudiante. Estos puntos son: 1) planificar las acciones de la planta en función del objetivo didáctico perseguido con el experimento, 2) confeccionar el plan por medio de los ficheros de experimentos, 3) monitorizar y supervisar la evolución del experimento para lograr los objetivos indicados, 4) si hay anomalías, intervenir a través del applet de experimentación para corregir el problema, y 5) aprender de la experiencia.

7 VISUALIZACIÓN REMOTA

En la actualidad, todo laboratorio remoto de Control que se considere de auténtica telepresencia demanda la existencia de una comunicación visual y auditiva con la planta sobre la que se estén realizando las experiencias. De acuerdo con las ideas esbozadas, el objetivo perseguido con esta sección es la creación de un entorno de trabajo que permita la supervisión visual remota del proceso. Por ello, se puede afirmar que lo que se pretende es construir un *sensor remoto en red* (McDowell, *et al.*, 98), el cual pueda ser gobernado tanto por el profesor como por el estudiante, siempre este último dentro de las especificaciones que marque el primero, es decir, el profesor.

El campo tradicional del empleo del vídeo en entornos de telepresencia o de teleoperación es el de la telerobótica, caracterizándose por constituir un medio para prolongar el sentido humano de la vista más allá de lo que permite la condición humana *per se*. En unos casos el vídeo permite al ser humano operar en entornos peligrosos (minas, centrales nucleares) (Corke, *et al.*, 98) o remotos (espacio exterior, profundidades abisales) (Sheridan, 93); en otros propicia la supresión de barreras espaciales para la aplicación del conocimiento de un experto (teleasistencia médica, teleoperaciones quirúrgicas) (Kitson, *et al.*, 1997); y, cómo no, permite experimentar de forma remota sobre aspectos de la Ingeniería de Control (Aktan, *et al.*, 96; Kondraske, *et al.*, 93; Gillet, *et al.*, 97; Overstreet y Tzes, 99). Este paradigma en el empleo del vídeo es en el que se enmarca nuestra aplicación: *la monitorización visual remota de un conjunto de plantas reales con fines didácticos*.

Algunos de los posibles supuestos de aplicación en entornos pedagógicos de telepresencia o teleoperación aplicados a la enseñanza del Control Automático son los siguientes:

- *Monitorización on-line del proceso.* Se consigue así la visualización directa y en tiempo real de cómo las acciones de control efectuadas sobre la interfaz de experimentación repercuten en la planta real. De esta forma, la realimentación que el alumno obtiene no sólo la extrae del análisis de los valores numéricos sino también de la observación directa del proceso.
- *Comprobación on-line del proceso,* es decir, grabación en vídeo de toda o parte de la experiencia con la planta, de forma que, concluida la sesión de teleoperación se puedan estudiar los pasos dados en caso de que se desee profundizar en algún aspecto destacable.
- *Refuerzo de conceptos.* Empleo de secuencias de vídeo obtenidas de alguna experiencia previa y que permiten reforzar algún concepto teórico.
- *Planteamiento de ejercicios.* Presentación visual de ciertas situaciones de forma que el estudiante, a priori, pueda pensar sobre qué acciones debe tomar en el caso de que se le presente esa situación durante una sesión de teleoperación.

Centrándonos en el diseño del sistema de visualización remota (Figura 4), éste se apoya en la clásica arquitectura cliente/servidor: un servidor se encargará de dialogar con dos tipos de applets de control remoto de la cámara (profesor y estudiante). Como forma de regular el intercambio de información entre ambas entidades software, se ha

diseñado un sencillo protocolo de comunicación denominado PCC (Protocolo de Comandos y Consultas). Este protocolo está compuesto, en su mayor parte, por mensajes que se asocian a paquetes de comandos y consultas soportados por el protocolo de bajo nivel que se emplea para el control de la cámara (protocolo VISCA), más algunos mensajes adicionales para realizar la identificación de usuarios y el control de la comunicación.

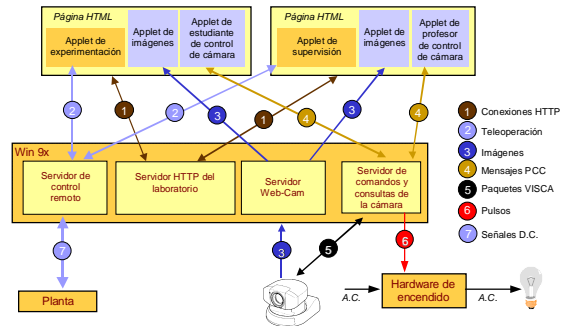


Figura 11 Esquema del sistema de visualización

De acuerdo con lo anterior, una vez que el navegador WWW haya cargado uno de los dos tipos de applets para el control de la cámara, éste establecerá un canal de comunicación con el servidor de comandos y consultas de la cámara (punto 4 de la Figura 11) y ambas entidades comenzarán a dialogar por medio del protocolo PCC. Aunque no se refleja fielmente en la Figura 11, el servidor es concurrente y genera un nuevo thread para atender las comunicaciones con cada applet de control, de forma que el servidor puede continuar a la escucha para atender cualquier nueva petición de conexión.

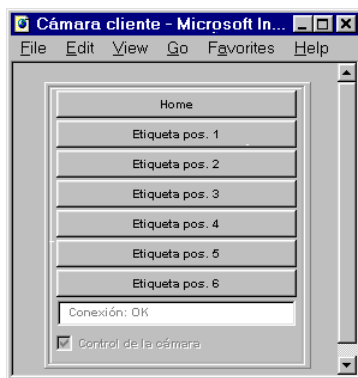
Por su parte, el applet de recuperación de imágenes establece la comunicación con el servidor Web-Cam (punto 5 de la Figura 11). Este último componente software está en conexión permanente con la cámara recuperando imágenes a la velocidad a la que haya sido programado (velocidad que estará en función de la dinámica del proceso: no es lo mismo el control de un péndulo invertido que el llenado de un tanque), aunque en última instancia la velocidad de transmisión está determinada por la capacidad de la red. Obsérvese que los canales de comunicación de los applets de control y de recuperación de imágenes son totalmente independientes (sockets UDP para las imágenes y TCP para los comandos).

Evidentemente, el tipo de applet para el control de la cámara que el servidor HTTP transmite depende del usuario: el del profesor (Figura 12a) dispone de un número muy superior de funciones que las del estudiante (Figura 12b), ya que debe poder preajustar las seis configuraciones que la memoria de la cámara soporta (en este caso una cámara motorizada SONY

modelo EVI-DE30/31). Además, limitando las funciones de la interfaz del estudiante se consiguen dos objetivos: centrar la atención en los aspectos concretos y relevantes de la planta para lograr las metas del experimento, y evitar los consiguientes problemas que ocasionaría un uso inapropiado de la cámara para los posteriores usuarios.



a) Profesor



b) Estudiante

Figura 12 Interfaces para la configuración y el control remoto de la cámara

Por otra parte, la construcción de un entorno remoto de experimentación implica no sólo una independencia física del estudiante con respecto al lugar de experimentación, es decir del laboratorio, sino también temporal. De acuerdo con esto, el instante en que se puede recurrir al entorno remoto para realizar prácticas puede ser cualquier momento del día y, principalmente, para aquellos que simultanean la vida laboral con los estudios universitarios, la tarde-noche es el único momento disponible. De ahí la conveniencia de que el entorno de experimentación esté disponible las 24 horas del día, garantizando no sólo la independencia espacial sino también la temporal.

Por ello, conjuntamente con el sistema de visualización remota se ha diseñado un sistema de encendido remoto de la iluminación de forma que en el mismo instante en que se establezca una conexión

para iniciar un experimento sobre una planta real se garantice la visibilidad del objeto a monitorizar.

8 SEGUIMIENTO Y EVALUACIÓN

Parece obvio que la completitud de la verdadera meta perseguida en este trabajo, que no es otra que la mejora de los conocimientos del alumno sobre la materia en cuestión, no puede lograrse sin que el estudiante reciba una realimentación sobre la validez del trabajo desarrollado dentro del laboratorio, así como que el profesor conozca el grado de avance experimentado por el alumno. Es por ello que todo el laboratorio virtual no tendría sentido sin la existencia de un módulo de seguimiento y evaluación del alumno que le permitiese a éste autodirigirse en su proceso de aprendizaje, indicándole los puntos sobre los que necesita un mayor refuerzo en su estudio.

Al mismo tiempo, este módulo debe servir al profesor tanto para comprobar que el proceso de aprendizaje que lleva asociado el estudiante es el correcto (seguimiento puntual) como para conocer si se han logrado los objetivos perseguidos tras el estudio completo de la asignatura (seguimiento global). De acuerdo con esto, algunas de las tareas que debe llevar a cabo el módulo de seguimiento y evaluación son: a) generación y corrección de test sobre la materia, b) programación de experimentos para el refuerzo y fijación de determinados conceptos, c) evaluación del resultado de los experimentos y d) seguimiento continuo de las simulaciones y del estado del laboratorio.

9 CONCLUSIONES

El diseño de los sistemas de control asistidos por computador y las herramientas de simulación tienen un fuerte impacto no solo sobre la forma en que diseñamos los sistemas de control sino también en la manera que lo enseñamos a nuestros alumnos. La integración de todos estos nuevos medios en los libros de texto y en los currícula es una tarea necesaria e importante.

Internet representa una gran oportunidad para este nuevo enfoque de la educación en control tanto para la diseminación de todo este nueva clase de material como para el desarrollo de laboratorios remotos, donde el alumno puede obtener experiencia práctica a través de Internet. Aunque el trabajo directo en un "laboratorio real" sigue siendo el método óptimo para formar a nuestros estudiantes, el paradigma de los laboratorios virtuales y remotos puede ser muy beneficioso en algunos aspectos: 1) optimiza la utilización de equipos de laboratorios costosos y de recursos humanos, 2) permite el desarrollo conjunto de una red de laboratorios remotos, 3) rompe las

barreras espacio-temporal en el acceso de los alumnos al laboratorio.

Se puede considerar que el desarrollo de nuevo material educativo para el estudio del Control Automático basados en el WWW está todavía en su infancia. Es posible caracterizar el escenario actual como confuso y caótico debido a lo disperso y variopinto de los sitios Web con material al que le falta una buena documentación y una evaluación de su calidad. Se necesita pues un esfuerzo cooperativo entre las organizaciones internacionales con intereses en el campo del Control Automático y las instituciones universitarias para desarrollar y coordinar la producción de nuevas herramientas basadas en el WWW para la enseñanza de la Automática.

Referencias

- Aktan, B., C.A. Bohus, L.A. Crowl, y M.H. Shor (1996), "Distance Learning Applied to Control Engineering Laboratories", *IEEE Transactions on Education*, vol. 39, nº 3, pp 320-326.
- Antsaklis, P., T. Basar, R. DeCarlo, N. Harris, M. Spong, y S. Yurkovich (1999). "Report on the NSF/CSS WorkShop on New Directions in Control Engineering Education", *IEEE Control Systems Magazine*, vol. 19, nº 5, pp 53-58.
- Corke, P., J. Roberts, y G. Winstanley (1998), "Vision-Based Control for Mining Automation", *IEEE Robotics & Automation*, Vol. 5, nº 4, pp 44-49.
- Garcia, R.C. y B.S. Heck (1999). "Enhancing Classical Controls Education via Interactive GUI Design", *IEEE Control Systems Magazine*, June 1999, pp. 77-82.
- Gillet, D., C. Salzmann, R. Longchamp, y D. Bonvin (1997), "Telepresence: An opportunity to develop practical experimentation in automatic control education", *Proceedings of the European Control Conference*, Bruselas.
- Hamilton, M.A. (1996). "Java and the Shift to Net-centric Computing", *IEEE Computer*, vol. 29, nº 8, pp. 31-39.
- Johansson, M., M. Gäfvert and K.J. Amstrom (1998). "Interactive Tools for Education in Automatic Control", *IEEE Control Systems Magazine*, vol. 18, nº 3, pp. 33-40.
- Kheir, N.A., K.J. Åmström, D. Auslander, K.C. Cheok, G.F. Franklin, M. Masten, M. Rabins (1996). "Control System Engineering Education", *Automatica*, vol. 32, nº 2, pp 147-166.
- Kitson, F.L., T. Malzbender, y V. Bhaskaran (1997), "Opportunities for Visual Computing in Healthcare", *IEEE Multimedia*, vol. 4, nº 2, pp 46-57.
- Kondraske, G.V., R.A. Volz, D.H. Johnson, D. Tesar, J.C. Trinkle, y C.R. Price (1993), "Network-based Infrastructure for Distributed Remote Operations and Robotics Research", *IEEE Transactions on Robotics and Automation*, vol. 9, nº 5, pp 702-704.
- Maly K., H. Abdel-Wahab, C.M. Overstreet, J.C. Wild, A.K. Gupta, A. Youssef, E. Stoica y E.S. Al-Shaer (1997). "Interactive distance learning over Intranets", *IEEE Internet Computing*, vol. 1, nº 1, pp. 60-71.
- McDowell, C.E., B.R. Montague, M.R. Allen, E.A. Baldwin, y M.E. Montoreano (1998), "JAVACAM: Trimming Java Down to Size", *IEEE Internet Computing*, vol. 2, nº 3, pp 53-59.
- Naryanan, S., N. Rao, J. Geist, P. Kiran, H.A. Ruff, M. Draper y M.W. Haas (1999). "UMAST: A Web-Based Architecture for Modeling Future Uninhabited Aerial Vehicles", *Simulation*, vol. 73, nº 1, pp 29-39.
- Overstreet, J. W y A. Tzes (1999), "An Internet Based Real-Time Control Engineering Laboratory", *IEEE Control Systems*, vol. 19, nº 5, pp 19-34.
- Patton, J.B. y P. Jayanetti (1996). "The Making of Multimedia Power Systems Control and Simulation Labware", *IEEE Transactions on Education*, vol. 39, nº 3, pp. 314-319.
- Piguet, Y. y D. Gillet (1999), "Java-based Remote Experimentation for Control Algorithms Prototyping", *Proceedings of the American Control Conference*, San Diego, California, 1999, pp. 1465-1469.
- Piguet, Y., U. Holmberg y R. Longchamp (1999), "Instantaneous performance display for graphical control design methods", *Proceedings of IFAC*, Beijing, China, 1999.
- Schmid. C. (1999). "A Remote Laboratory Using Virtual Reality on the Web", *Simulation*, vol. 71, nº 1, pp. 13-21.
- Shen, H., Z. Xu, B. Dalager, V. Kristiansen, O. Strøn. M.S. Shur, T.A. Fjeldly, J. Lü y T. Ytterdal (1999). "Conducting Laboratory Experiments over the Internet", *IEEE Transactions on Education*, vol. 42, nº. 3, pp. 180-185.
- Sheridan, T.B. (1993), "Space Teleoperation Through Time Delay: Review and Prognosis", *IEEE Transactions on Robotics and Automation*, vol. 9, nº 5, pp 592-606.
- Wittenmark, B., H. Häglund y M. Johansson (1998). "Dynamic Pictures and Interactive Learning", *IEEE Control Systems Magazine*, vol. 18, nº 3, pp. 26-32.