

HEATFLOW: UN LABORATORIO BASADO EN WEB USANDO EASY JAVA SIMULATIONS Y LABVIEW PARA EL ENTRENAMIENTO EN TÉCNICAS DE AUTOMATIZACIÓN

H. Vargas*, R. Dormido*, N. Duro*, J. Sánchez*, S. Dormido-Canto*, G. Farias*, S. Dormido*, M. Canto*
F. Esquembre[†]

*Dpto. de Informática y Automática, UNED
C/. Juan del Rosal nº 16, c.p. 28040, Madrid, España

[†]Dpto. de Matemáticas, Universidad de Murcia
Campus de Espinardo, c.p. 30071, Murcia, España

E-mails: {raquel, nduro, jsanchez, sebas, sdormido, macanto}@dia.uned.es,
{hvargas, gfarias}@bec.uned.es, fem@um.es

Resumen — Las nuevas tecnologías basadas en Internet posibilitan el nacimiento y desarrollo de nuevos paradigmas en el modelo educativo tradicional. De esta forma, uno de los usos de esta infraestructura informática es la educación a distancia de los estudiantes por medio de laboratorios virtuales y remotos. El presente artículo muestra el estado de desarrollo de un laboratorio completo para la realización de experiencias en automática mediante sesiones de experimentación virtual y remota a distancia: “El sistema *HeatFlow*”. El laboratorio permite a los usuarios trabajar tanto en modo virtual como remoto con la misma interfaz gráfica. El lado del servidor utiliza *LabVIEW* de National Instruments para la adquisición de datos, generación del lazo de control y comunicación a través de Internet. La interfaz del cliente ha sido creada utilizando el sistema de desarrollo conocido como *Easy Java Simulations*, una herramienta *open source* para generar potentes aplicaciones y *applets* sin la necesidad de tener conocimientos avanzados en programación.

Palabras Claves — Educación a distancia, experimentación basada en el web, laboratorio virtual, laboratorio remoto, control automático, educación en ingeniería.

1 Introducción

Los laboratorios basados en web son entornos de experimentación en el que los usuarios pueden operar sobre una serie de componentes gráficos, cada uno representando un elemento importante del experimento, y llevar a cabo la tele-operación o manipulación remota de un laboratorio virtual (donde la interfaz de experimentación trabaja contra una simulación del proceso) o remoto (donde el proceso es real y accesible a través de Internet) con el simple manejo del ratón o el teclado. De esta manera, los laboratorios basados en web extienden el área de aplicación de las tradicionales sesiones de los laboratorios presenciales con entornos de experimentación virtual y remota a distancia.

Muchas son las áreas del conocimiento e investigación en ingeniería que se han nutrido de los avances de las tecnologías basadas en Internet y la importancia que esta ha tenido en la enseñanza-aprendizaje basada en web. El *control automático* es una de las áreas técnicas de la ingeniería en la que el impacto de estas tecnologías ha sido especialmente significativo en todo lo relativo al desarrollo de herramientas para el aprendizaje y la experimentación (Dormido 2004, Guillet *et al.*, 2005). En un laboratorio de control basado en web, los estudiantes pueden acceder al laboratorio desde una localización remota que les permite interactuar con el experimento real. Es posible cambiar los parámetros

de control, ejecutar los experimentos, descargar datos y visualizar resultados a través de Internet. Con esta perspectiva, el Departamento de Informática y Automática de la UNED en España está trabajando en el desarrollo de nuevos paradigmas de laboratorios para la realización a través de Internet de experiencias prácticas de ingeniería de control sobre plantas y sistemas reales o simulados.

En este contexto, el sistema *HeatFlow* presenta características muy interesantes para la investigación y la educación de los fundamentos de control. Es posible desarrollar un modelo sencillo de primer orden y su control asociado como primer paso a las técnicas de identificación, para luego en subsiguientes experimentos incrementar el grado de dificultad analizando técnicas de identificación y control de sistemas con retardo de tiempo.

El artículo se organiza de la siguiente manera. El apartado 2 introduce los conceptos básicos de *Easy Java Simulations*. En la apartado 3 se describe el sistema de control de flujo de calor “*Heatflow*”. El apartado 4 presenta la construcción del laboratorio remoto. Cada sección de este apartado muestra: la confección de la vista del laboratorio elaborada en *Ejs* (el cliente), la lógica de control y acceso a los recursos del laboratorio remoto (el servidor) y el camino que los usuarios finales del sistema deben seguir para realizar una reserva del experimento, respectivamente. Para finalizar, el apartado 5 recoge algunas conclusiones y consideraciones generales del trabajo.

2 Fundamentos de Easy Java Simulations

Ejs es una herramienta de software gratuita diseñada para la creación de simulaciones interactivas en Java (Esquembre, 2004). El usuario al que está dirigida *Ejs* son estudiantes, profesores e investigadores de ciencias e ingeniería, que poseen un conocimiento básico de programación de computadores, pero que no disponen de una gran cantidad de tiempo para crear una simulación gráfica con un elevado grado de interactividad.

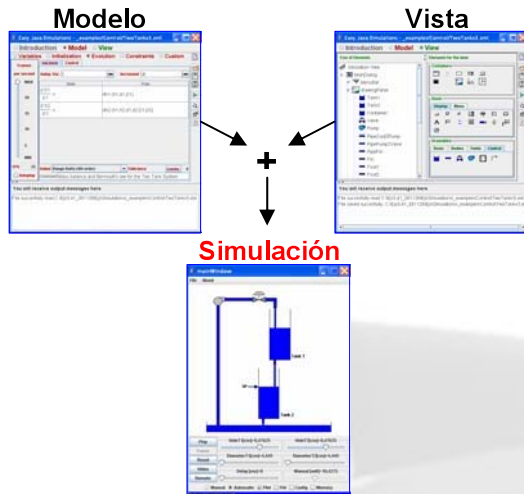


Figura 1. Pasos para crear una simulación en *Ejs*.

Las simulaciones en *Ejs* son estructuradas en dos partes, el *Modelo* y la *Vista* (Fig. 1). En el modelo se describe el comportamiento del sistema mediante variables y código Java o ecuaciones diferenciales ordinarias. Mientras que la vista provee el aspecto gráfico o visual de la simulación. La interfaz de usuario de *Ejs* se presenta en la Figura 2.

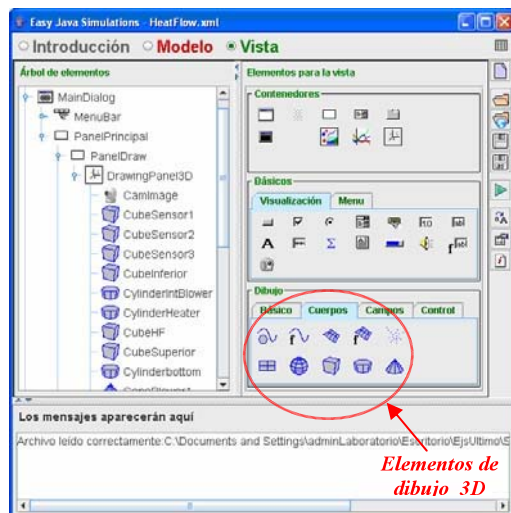


Figura 2. Interfaz gráfica de usuario de *Ejs* para la creación de la vista de una simulación. El árbol de la izquierda corresponde a la vista de la Figura 5.

Ejs se puede utilizar de forma independiente para crear simulaciones interactivas, ya sea como aplicación o como applet, sin embargo, también es posible utilizar *Ejs* para agregar interactividad a modelos Simulink (Dormido *et al.*, 2005).

3 El Sistema Heatflow

El desarrollo del laboratorio virtual y remoto utiliza el sistema Heatflow fabricado por la empresa canadiense Quanser (Quanser, 2006).

La planta consiste en una caja equipada con los siguientes componentes: un calefactor y un ventilador ubicados en un extremo de la estructura y tres sensores de temperatura S1, S2 y S3 localizados en diferentes posiciones a lo largo de un ducto (Fig. 3). La potencia liberada al calefactor es controlada aplicando un voltaje entre 0 y 5 voltios a la entrada V_h . Igualmente, la velocidad del ventilador puede ser controlada aplicando un voltaje comprendido entre 0 y 5 voltios a la entrada V_b . La temperatura medida en cada sensor se obtiene a través de las salidas S_1 , S_2 y S_3 como un voltaje analógico en el rango de 0 a 5 voltios y cuyo valor final en grados Celsius se obtiene multiplicando por un factor de ganancia lineal de 20 [°C/voltio]. La velocidad de giro del ventilador se mide utilizando un tacómetro y se puede usar para diseñar controladores de velocidad.

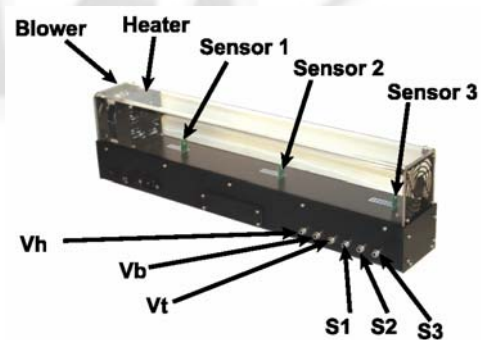


Figura 3. El sistema HeatFlow de Quanser.

En principio, puede definirse un modelo genérico de la forma:

$$\dot{T}_n = F(V_h, V_b, T_a, x_n) \quad (1)$$

Donde:

- T_n es la temperatura en el sensor n .
- V_b es el voltaje aplicado al ventilador.
- V_h es el voltaje aplicado al calefactor.
- T_a es la temperatura ambiente.
- x_n es la distancia del sensor n al calefactor.

Debido a que un modelo matemático completo que describa la termodinámica del sistema es difícil de derivar, se han estimado los parámetros del modelo mediante técnicas de identificación. Para ello, se procede a determinar las funciones de transferencia para cada sensor de temperatura en función del voltaje aplicado al calefactor. A partir de experiencias mediante respuesta a escalón en lazo abierto se registran las respuestas del proceso para cada sensor. Para que el registro sea válido, la salida debe haber alcanzado el estacionario. En cada experiencia, el voltaje aplicado al ventilador se mantiene constante y se utiliza como variable asociada a una *perturbación* para las funciones de transferencia resultantes del sistema. La identificación final de los modelos se ha realizado con el toolbox de identificación de sistemas de Matlab - IDENT.

Bajo estas consideraciones, el modelo que mejor se ajusta al comportamiento del sistema es de la forma:

$$G(s) = \frac{T_n(s)}{V_h(s)} = \frac{K_p (1 + \tau_3 s) e^{-\tau_d s}}{(1 + \tau_1 s)(1 + \tau_2 s)} \quad (2)$$

- $T_n(s)$ es la temperatura en el sensor n .
- $V_h(s)$ es el voltaje aplicado al calefactor.

donde la ganancia, constantes de tiempo y el retardo dependen de cual de los tres sensores se está utilizando para cerrar el lazo de control de temperatura. La Tabla 1 contiene una lista con los parámetros del modelo obtenido para cada sensor.

Tabla 1. Parámetros de los modelos.

	K_p	τ_1	τ_2	τ_3	τ_d
S1	26.95	38.74	3.74	18.35	0
S2	22.37	50.46	4.00	22.42	0.5
S3	22.00	56.45	4.29	26.70	1.15

4 El Laboratorio Virtual y Remoto

La arquitectura del laboratorio virtual y remoto se basa en la conocida estructura cliente/servidor (Duro *et al.*, 2005). El ordenador que implementa el lado del servidor contiene el servidor web que da acceso al applet de estado y al sistema de control en tiempo real (Fig. 4). La interfaz del cliente se describirá en la sección 4.1, y corresponde a la vista construida con *Ejs*. Esta vista interactiva puede funcionar como laboratorio virtual utilizando el modelo matemático de la planta descrito mediante el editor de ecuaciones diferenciales de *Ejs* o como laboratorio remoto con la planta real.

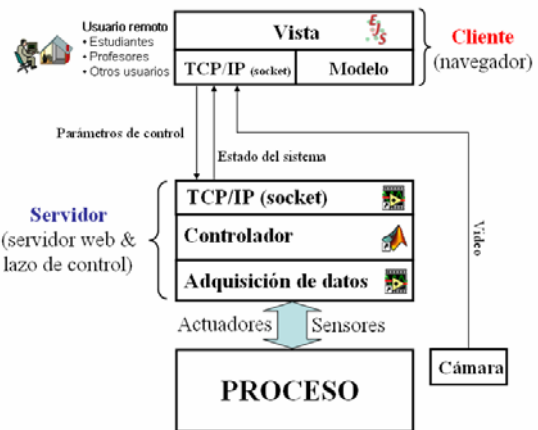


Figura 4. Arquitectura cliente - servidor.

La implementación del servidor desarrollada con *LabVIEW* se describirá en detalle en la sección 4.2. En el servidor se realizan las tareas de control de acceso, transmisión continua de datos entre el cliente y el servidor, comunicación con el sistema físico a través de tarjetas de adquisición de datos y cierre del lazo de control.

4.1 El Cliente

En la Figura 5 se muestra la ventana principal del laboratorio virtual y remoto construida con *Ejs*. La parte superior contiene una representación 3D del sistema heatflow cuyo color varía en función del estado del proceso (se utiliza el modelo identificado en el apartado anterior cuando el sistema se ejecuta como laboratorio virtual; cuando se utiliza la opción remota el sistema accede al proceso real del laboratorio).

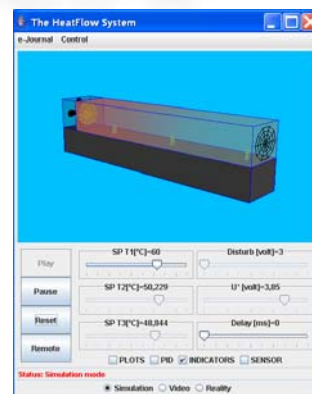


Figura 5. Interfaz gráfica de usuario (vista *Ejs*) creada con la librería de elementos gráficos de *Ejs*.

La parte inferior de la ventana principal contiene un panel de control que permite escoger el tipo de experiencia (virtual o remota), y un conjunto de sliders y botones que permiten definir diversas situaciones en la dinámica del proceso (por ejemplo, realizar un cambio en la consigna o introducir alguna perturbación variando el voltaje en el ventilador).

Los cuatro botones localizados a la izquierda del panel de control (*Play*, *Pause*, *Reset*, y *Remote*) permite al usuario controlar las operaciones principales sobre la evolución del sistema. Si se pulsa el botón *Remote* el sistema se conectará a la planta real ubicada en el laboratorio de la universidad, si el nombre de usuario y la contraseña introducida son correctas (esta característica se describe en la sección 4.2). Si este botón no se acciona la aplicación trabaja en modo simulación. A la derecha de los botones, seis sliders permiten ajustar los valores de las consignas para cada sensor de temperatura (*SP T1*, *SP T2*, y *SP T3*), el voltaje aplicado al ventilador (*Disturb*), el voltaje suministrado al calefactor (*U**), y ajustar un retardo en la simulación (*Delay*). Adicionalmente, cuatro botones de selección permiten la apertura de cuatro ventanas auxiliares (Fig. 6).

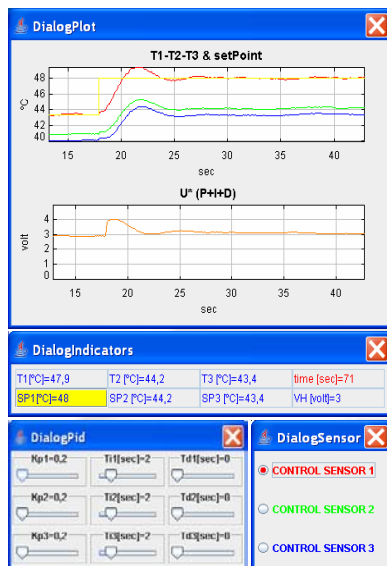


Figura 6. Ventanas auxiliares del cliente *Ejs*. La gráfica superior muestra la respuesta del sistema en modo remoto a un cambio en la consigna del primer sensor de 43.4 °C a 48 °C.

Si se selecciona la opción *PLOT* se visualiza un panel compuesto por dos gráficas en las que se recoge la evolución de las variables más importantes del proceso, es decir, las medidas de los sensores de temperatura y la acción de control *U** sobre el calefactor; seleccionando la opción *PID* se obtiene un panel con los parámetros del controlador PID asociado a cada sensor. El botón *INDICATORS* despliega una nueva ventana con un conjunto de campos numéricos que muestran el valor de las distintas variables de interés del proceso. El botón *SENSOR* permite escoger el sensor para cerrar el lazo de control de temperatura.

Un campo de texto no editable ubicado debajo del panel de control sirve para visualizar mensajes sobre el estado de la conexión entre el cliente (vista *Ejs*) y el servidor (planta real).

Tres botones radiales (*Simulation*, *Video*, y *Reality*) localizados en la parte más baja de la aplicación permiten seleccionar tres formas diferentes de ver la planta (Fig. 7). El primero muestra la planta creada con los elementos propios de *Ejs*; el segundo, muestra el proceso real que está en el laboratorio remoto; y el tercero, permite ver la planta real en el laboratorio remoto y superpuesta a ella la planta simulada.



Figura 7. Vista del cliente *Ejs* en modo *Video* y *Reality*.

Un menú desplegable (*e-Journal*) localizado en la parte más alta de la consola muestra un submenú con cinco botones: *SaveGraph*, *SaveData*, *ReadData*, *Start Record*, y *Stop Record* que permiten guardar una imagen de la evolución de las variables en formato .gif, salvar el estado del sistema en un fichero de datos binario .dat, leer un fichero de datos binario .dat, grabar en un fichero .m los parámetros de los controladores y los valores de la evolución temporal de las variables controladas y manipuladas, y parar la grabación de datos (esta opción está activa una vez que la grabación ha comenzado). Finalmente, otro menú desplegable (*Control*) localizado a la derecha del anterior permite conmutar las acciones de control de modo manual a automático, y viceversa.

4.2 El Servidor

El lazo de control en el servidor se ha desarrollado utilizando *LabVIEW* (LabVIEW, 2006) y *Matlab/Simulink*. El intercambio de datos entre la planta y el ordenador se lleva a cabo mediante una tarjeta de adquisición de datos DAQ de National Instruments desde LabVIEW, donde además se cierra el lazo de control y se configura el socket servidor que queda a la espera de conexión desde el cliente. Los controladores PID están programados en Matlab/Simulink, y se accede a ellos mediante un bloque para llamar y evaluar un *scripts* de Matlab desde programas *LabVIEW*. Esta estructura de diseño del servidor presenta una gran flexibilidad para incluir nuevas estrategias de control (por ejemplo, controladores predictivos o difusos) con un pequeño esfuerzo en programación.

Al producirse una conexión desde el cliente, la vista *Ejfs* envía una estructura de datos al servidor, obteniendo como respuesta un vector con el estado actual de la planta (Fig. 4). Los parámetros de control remitidos desde el cliente al servidor son: modo de control (*Manual/Automático*), sensor (S_1, S_2, S_3), voltaje manual al calefactor (U^*), parámetros de los controladores PID ($K_{p1}, T_{i1}, T_{d1}, K_{p2}, T_{i2}, T_{d2}, K_{p3}, T_{i3}, T_{d3}$), consignas de temperatura (sp_1, sp_2, sp_3) y voltaje manual al ventilador (*Disturb*), formando un paquete de datos de 60 bytes. La información que la vista obtiene como respuesta está compuesta de cinco valores: instante de muestreo (t), temperatura en los sensores (S_1, S_2, S_3) y voltaje aplicado al calefactor (U^*). El tamaño de este paquete de datos es de 20 bytes.

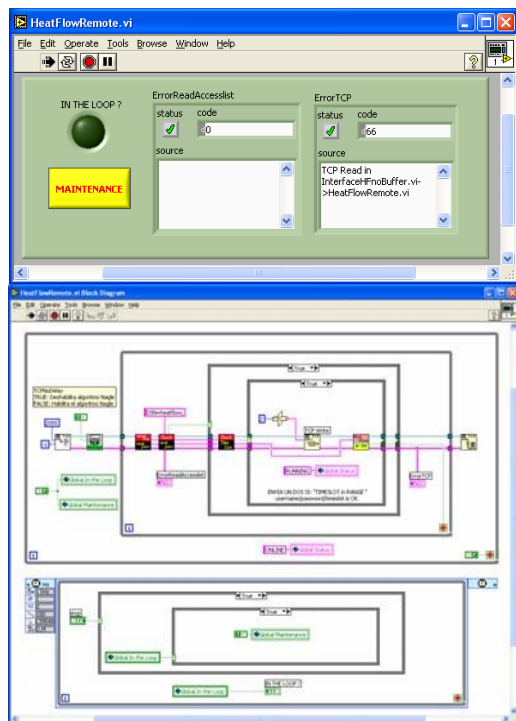


Figura 8. Panel frontal y diagrama de bloques de la aplicación servidora en *LabVIEW*.

La Figura 8 muestra el panel frontal y diagrama de bloques de la aplicación servidora programada en *LabVIEW*. El panel frontal es muy simple e intenta disminuir la cantidad de elementos gráficos con el objetivo de optimizar el rendimiento de la aplicación. Sólo cuenta con dos indicadores que permiten visualizar un posible error de conexión con la base de datos o de comunicación con la aplicación cliente. Esta base de datos (*MySQL*) en el servidor se encarga de la gestión de usuarios y reserva de recursos de acceso a la planta real localizada en el laboratorio de la universidad (esta característica se describe en la sección 4.3). Cada registro en la base de datos corresponde a una reserva del recurso

realizada por un usuario para una fecha y rango de tiempo establecido y que se deberá verificar cada vez que un cliente desee conectarse al recurso remoto. Un *led* denominado *IN THE LOOP* indica si un usuario se encuentra o no operando la planta remota y por último el botón *MAINTENANCE* que permite parar la aplicación.

La Tabla 2 presenta un resumen descriptivo de los bloques utilizados. Cada objeto contiene un nuevo diagrama de bloques que implementa la funcionalidad que se describe a la derecha de cada VI (virtual instrument) de la tabla.

Tabla 2. Bloques (VIs) de la aplicación servidora.

VI	Descripción
	<i>TCP Listen</i> : Crea un socket de escucha y espera por una conexión de red TCP desde el cliente (nativo de LabVIEW).
	<i>TCPNoDelay</i> : Deshabilita el algoritmo Nagle de una conexión TCP/IP establecida entre el cliente y el servidor.
	<i>TCP Write</i> : Escribe datos sobre una conexión o canal de red TCP (nativo de LabVIEW).
	<i>TCP Read</i> : Lee un número específico de bytes desde una conexión de red TCP (nativo de LabVIEW).
	<i>Read User/Pass</i> : Lee el usuario y password (encriptada en MD5) del cliente desde la conexión TCP.
	<i>Check User/Pass</i> : Consulta la base de datos y verifica si el usuario y password existe.
	<i>Check Time Slot</i> : Si existe el usuario y password anterior, verifica si la fecha y hora de reserva son correctas.
	<i>Acquisition Data/Control</i> : Acceso a la planta real y cierre del lazo de control. Realiza el intercambio continuo de datos entre el cliente y el servidor.
	<i>TCP Close</i> : Cierra una conexión TCP establecida entre la aplicación cliente y el servidor (nativo de LabVIEW).

En cada iteración del bucle interno del bloque *Acquisition Data/Control* se realiza la lectura de la estructura de control enviada desde el cliente al servidor, la escritura de la trama de estado desde el servidor al cliente y la comparación del tiempo actual con el tiempo de fin de la sesión de experimentación con el objetivo de cerrar la conexión cuando este tiempo final se haya sobrepasado. Como medida de *seguridad*, ante cualquier evento de término de la conexión (fin de la sesión experimental o corte inesperado de la conexión a Internet) se anula la acción sobre los actuadores del sistema enviando un valor de cero voltios y se cierra la conexión con el cliente mediante el bloque *TCP Close*.

4.3 Método de Acceso a los Recursos

El acceso final a los recursos del laboratorio por parte de los usuarios se lleva a cabo a través del portal de CYBERLAB (Cyberlab, 2006). Cyberlab define una arquitectura de laboratorios orientada a proporcionar un servicio independiente de gestión de usuarios y reserva de uso de los laboratorios adscritos a la red Cyberlab, para facilitar la integración de laboratorios remotos y virtuales de diferentes instituciones (Fig. 9).



Figura 9. Sistema de reservas de CYBERLAB.

Cyberlab define dos tipos de usuarios distintos: proveedores y clientes. Los proveedores se encargan de proporcionar el laboratorio remoto e implementar el mecanismo definido por Cyberlab para indicar la disponibilidad de los experimentos y asignación de tiempos que un usuario final del laboratorio puede realizar. Ese mecanismo lo constituye un servidor central (Cyberlab Provider) que informa al proveedor del laboratorio sobre las conexiones de los usuarios y las reservas de espacios de tiempo generadas a través del portal Cyberlab actualizando la base de datos del proveedor con estas nuevas reservas. Los clientes son aquellos usuarios del portal Cyberlab que realizan reservas de tiempo sobre los laboratorios y acceden a ellos en dichos espacios temporales. El acceso final al laboratorio se hace a través de unas páginas HTML especialmente diseñadas para mostrar el applet de estado que permite al usuario identificarse y acceder al laboratorio.

5 Conclusiones

La investigación en el campo de la experimentación virtual y remota en la educación de las ingenierías puede considerarse ya una tecnología madura. Sin embargo, aún es difícil encontrar en revistas especializadas en la educación del control la aplicación de estos elementos de aprendizaje a niveles de técnico de

automatización. Lo anterior implica ignorar la existencia de este amplio y significativo grupo en el ámbito de la ingeniería del control. En este sentido, y con el objetivo de dar solución a estos problemas, cinco universidades Europeas están llevando a cabo un proyecto piloto denominado "AutoTECH: Automation Technicians Vocational Training Repository" (AutoTech, 2006), cuya meta principal implica desarrollar un conjunto de recursos de aprendizaje basado en web para satisfacer los requerimientos de estas personas. El sistema *heatflow* es uno de estos esfuerzos realizado por uno de los miembros del consorcio AutoTech, el grupo de investigación de la UNED.

La estructura del laboratorio virtual y remoto (*Ejs-LabVIEW*) del artículo presenta grandes ventajas respecto a la filosofía de diseño basada en imágenes (pantalla compartida), como por ejemplo, los paneles remotos de *LabVIEW* o *VNC* (*Virtual Network Computing*). En este caso, la eficiencia de las comunicaciones aumenta ya que el diálogo entre el cliente y el servidor se reduce al intercambio de pequeños vectores de datos estructurados (control y estado) en lugar imágenes que ralentizan la comunicación.

Agradecimientos

Los autores quieren agradecer la financiación de este trabajo a la CICYT en el marco del proyecto DPI2004-01804 y a la Comisión Europea en el marco del proyecto "Leonardo da Vinci Pilot-Project. Nº 2004 N/04/B/PP 165.011".

Referencias

- Dormido, S. (2004) "Control learning: Present and future" Annual Control Reviews, 28, pp 115-136
- Esquembre, F. (2004). Easy Java Simulations: A software tool to create scientific simulations in Java, *Comp. Phys. Comm*, 156 (2), pp. 199-204.
- Pastor, R., Sánchez, J., Dormido, S. (2005). Web-Based Virtual Lab and Remote Experimentation Using Easy Java Simulations, *16th IFAC World Congress*, Prague (Czech Republic).
- Guillet, D., Nguyen, A., Reik, Y. (2005). Collaborative Web-Based Experimentation in Flexible Engineering Education, *IEEE Trans on Education*, vol. 48, nº 4.
- Duro, N., Dormido, R., Vargas, H., Dormido, S., Sánchez, J., Pastor, R. (2005). The Three Tank System: A Remote and Virtual Control Laboratory using Easy Java Simulations, *44th IEEE CDC/ECC*, Sevilla (Spain).
- Dormido, S., Esquembre, F., Farias, G., Sánchez, J. (2005). Adding interactivity to existing Simulink models using Easy Java Simulations, *44th IEEE CDC/ECC*, Sevilla (Spain).
- Quanser (2006). <http://www.quanser.com>
- LabVIEW (2006). <http://www.ni.com/labview>
- Cyberlab (2006). <http://www.cyberlab.org>
- AutoTech (2006). <http://www.pidstop.com>