

# LABNET: UNA HERRAMIENTA PARA EL CONTROL DE PROCESOS A TRAVÉS DE INTERNET

Nourdine Aliane, Diego Gachet Páez  
DARC - Universidad Europea de Madrid,  
Villaviciosa de Odón, S/N (28670) Madrid, [\[aliane, gachet\]@uem.es](mailto:[aliane, gachet]@uem.es)

Aída Martínez, Alberto Fraile, Jaime Ortiz  
Escuela politecnica – Área de informática - Universidad Europea de Madrid

## Resumen

*En este trabajo se presenta una herramienta didáctica para realizar experimentos de control de forma remota sobre unas maquetas de laboratorio. Las características más importantes del sistema son: el acceso simultáneo a varias maquetas, la elección de un controlador predefinido o del tipo de la realimentación (local o remota), la generación automática de informes y la reutilización de plantillas. En este artículo, se explicará la arquitectura del sistema, se comentarán los resultados obtenidos y se darán los detalles relacionados con la implementación.*

**Palabras Clave:** Control de Procesos, Control Remoto, Laboratorios Remotos, Educación, Internet

## 1 INTRODUCCIÓN

Dentro de la enseñanza del control automático, la realización de prácticas experimentales sobre sistemas reales es de gran importancia ya, no solamente ayuda a los estudiantes a asimilar mejor los conceptos teóricos, sino también desvela otros aspectos difíciles de abordar de forma teórica. Desgraciadamente, varios factores dificultan la realización de dichas prácticas [5] como el gran número de estudiantes, la poca capacidad de los laboratorios, ya sea en medios o en espacio, lo que hace imprescindible la restricción del acceso a dichos laboratorios a ciertas horas prefijadas.

El acceso remoto a los laboratorios no solamente nos permite incrementar el ratio de ocupación y la posibilidad de utilizar a tiempo compartido equipos más costosos, sino permite a los estudiantes realizar sus prácticas sin restricciones de horarios y un número ilimitado de veces. Desde el punto de vista pedagógico, las prácticas realizadas de forma remota ofrecen una formación similar a las realizadas in-situ. Finalmente, hay que resaltar que para la formación de tipo semi-presencial o continuada, la

flexibilidad que ofrece un sistema de acceso remoto a un laboratorio es de gran importancia.

La idea de compartir los equipos de un laboratorio usando internet con fines educativos no es reciente. Ya, en los noventa, en el trabajo [1], se propuso un esquema de acceso remoto a los sistemas de control de un laboratorio y como compartirlos con otras universidades. Desde entonces, varios trabajos como [2] se centra en la enseñanza del control automático a través de la red o en [7] que plantea ya problemática de acceso a los equipos de un laboratorio para el modelo de la educación a distancia. En muchos otros trabajos, se ha llevado a la práctica con éxito y podemos nombrar [9], [8] y [4].

A nivel nacional, las cosas no se han quedado atrás y recientemente, en el trabajo desarrollado en la UNED [10], tenemos un caso que muestra la teleoperación de un péndulo invertido a través de internet y en [6] la combinación de un control local y remoto del sistema anterior. A nivel nacional, hay muchos otros trabajos relacionados con el control a través de internet y podemos nombrar algunos ejemplos como [3], donde se desarrolla una aplicación que permite la simulación herramientas Web, o en el trabajo [5], se muestra un control de sistemas a eventos discretos utilizando PLC's.

Pero, en la mayoría de los trabajos comentados antes, se limitan al desarrollo de un prototipo a la cual no se puede acceder. En realidad, pocas universidades disponen de un sistema fiable, con funcionalidades suficientes como para llevarlo a la práctica y satisfacer, así, todas las motivaciones que hemos comentado al principio.

En nuestro trabajo, pretendemos desarrollar nuestro propio sistema que nos permita acceder a las maquetas de control de nuestro laboratorio y con funcionalidades adicionales como veremos más adelante.

En lo que queda del artículo, se va presentar la descripción del sistema dando su arquitectura y su filosofía de funcionamiento. A continuación, se van comentar las funciones más importantes del sistema desarrollado dando las funciones más importantes. Finalmente, finalmente, después de sacar las

Conclusiones más importantes, indicaremos las posibles mejoras de nuestro sistema.

## 2 DISEÑO DEL SISTEMA

En este apartado, se va a definir la arquitectura de nuestro sistema, dando detalles sobre los elementos que vamos a utilizar así como los métodos de control que queremos realizar.

### 2.1 ARQUITECTURA DEL SISTEMA

Nuestro sistema consiste en una aplicación cliente/servidor orientada a la práctica del control de procesos con maquetas de laboratorio de forma remota utilizando la red internet. Por ahora, nuestro sistema puede cablear y activar hasta 4 maquetas. Estas maquetas se conectan al servidor, que se ejecuta sobre un PC Pentium-III, mediante 2 tarjetas de adquisición de datos de Advantech PCLAB-816 sobre ranuras de tipo ISA.

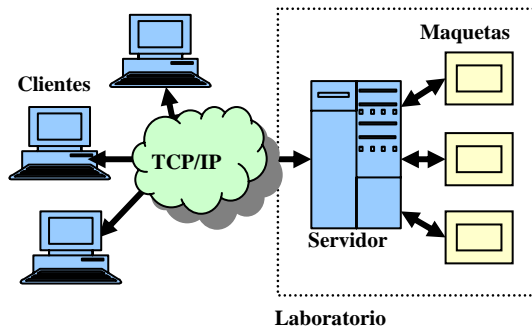


Figura 1: Arquitectura del sistema

El cableado físico de las maquetas con el servidor así como la configuración de las tarjetas de adquisición de datos corresponde al administrador y la operación consiste en realizar una correspondencia exacta entre una maqueta real y su equivalente virtual en el servidor para la correcta elección de varios parámetros como la polaridad de la señal de control, su escala, etc.

### 2.2 MAQUETAS UTILIZADAS

Las maquetas de laboratorio proporcionan sistemas de control de procesos reales mediante los cuales el alumno pueda acercarse a la realidad de sistemas industriales. Las maquetas (ver figura 2) utilizadas por nuestro sistema son: la maqueta para el control de nivel, para el control de temperatura y una tercera que es un sistema de estabilización de barcos.

Todas estas las maquetas admiten entradas y salidas normalizadas en voltajes en su rango de linealidad: de 0 a 10v en el caso de las maquetas unipolares y de -10 a 10v en el caso de las bipolares.

En la pagina web [11], podemos ver más detalles sobre las características y funcionamiento de las maquetas de nuestro laboratorio.

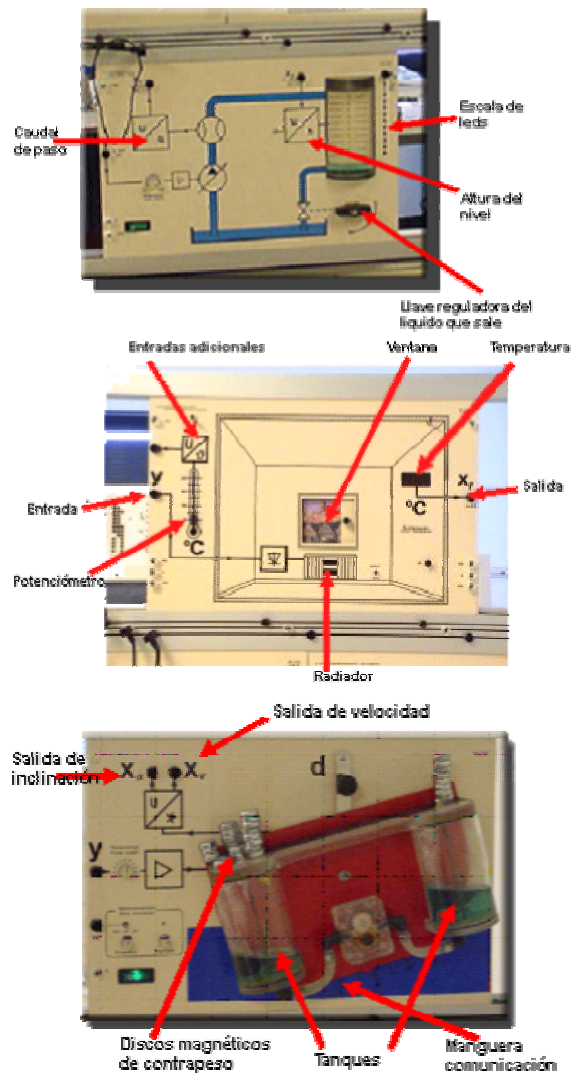


Figura 2: Las maquetas de laboratorio

### 2.3 MODOS DE CONTROL

Nuestra aplicación se apoya en dos configuraciones básicas: control local y control remoto.

En el control local, la realimentación se realiza en el servidor. En este caso, el cliente realiza las acciones de configuración y accionamiento del experimento, su parada, etc. El cliente recibe una vez que haya finalizado el experimento o en tiempo real.

En el control remoto, el controlador está ubicado en lado del cliente. En esta configuración, los retardos de transmisión por la red se introducen en el lazo de control y su efecto sobre el comportamiento global del sistema depende de varios factores: el protocolo utilizado (TCP, UDP), el medio utilizado para la conexión (cable o la red inalámbrica de nuestro

Campus), tráfico en la red en el momento de la experimentación, etc.

En la figura 3, podemos ver una ilustración para aclarar del tipo de realimentación utilizado por nuestro sistema.

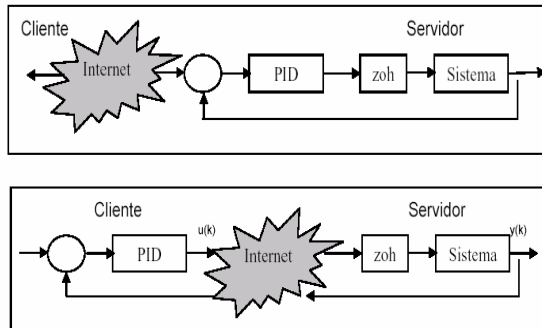


Figura 3: Realimentación local o remota

En cuanto al controlador propiamente dicho, nos hemos limitado a la implementación del controlador PID con 3 variantes: el PID teórico, PI-D con D-filtrado y el PID con Antiwindup.

### 3 LA HERRAMIENTA LABNET

La herramienta LABNET se articula en dos partes bien diferenciadas: el cliente y el servidor. En este apartado explicaremos las funcionalidades más destacadas de ambas partes.

#### 3.1 EL CLIENTE

El cliente es directamente utilizado por los usuarios finales para conectarse al sistema LABNET. Para facilitar la conexión y configurar un experimento de control de forma rápida e intuitiva, se ha diseñado un cliente con un entorno GUI que recoge todas las funcionalidades necesarias del sistema. La figura 4 muestra el aspecto del entorno gráfico del cliente

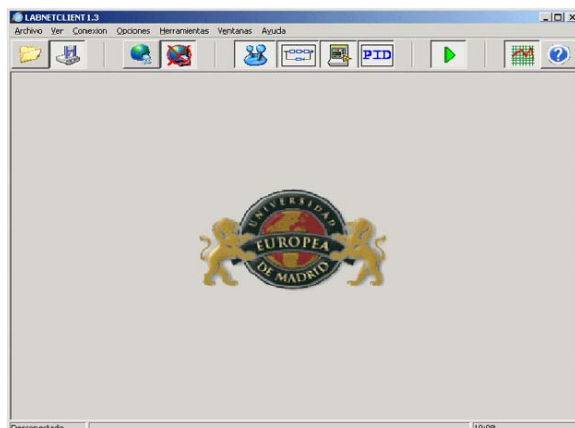


Figura 4: Entorno GUI del cliente LABNET

A continuación vamos a explicar las funcionalidades más destacadas del cliente LABNET

#### 3.1.1 Configuración de un experimento

Un usuario se conecta al sistema mediante un usuario y una password. Una vez activado el cliente, el usuario puede configurar los diversos parámetros mediante la introducción de datos en ventanas contextuales. Los parámetros más importantes son:

- Elección de la maqueta
- El tipo de control (local o remoto)
- Elección de la señal de referencia
- Elección del regulador PID (El teórico, con D-filtrado, con Antiwindup) y sus diversos parámetros

En la figura 5, podemos ver un ejemplo de como se elige un controlador y el ajuste de sus parámetros

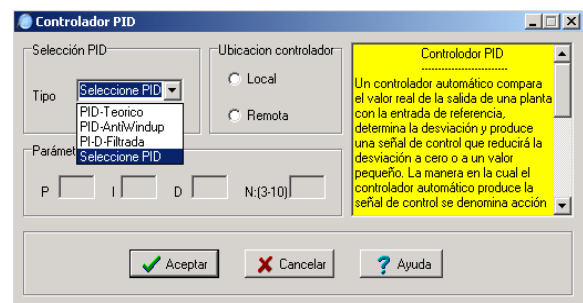


Figura 5: Elección del tipo de controlador

#### 3.1.2 Visualización de datos e informes

Para visualizar los datos de nuestros experimentos, el cliente dispone de una herramienta versátil y con las siguientes funcionalidades:

*Visualización grafica:* Esta opción nos permite visualizar los datos de las variables elegidas como la salida del sistema, la señal de control.

*Exportación de los datos:* Una vez finalizado el experimento, podemos exportar los datos y los gráficos a formatos más conocidos como (Excel, Matlab, html, jpeg, bmp, etc).

*Generador de informes:* Esta opción nos crea un documento plano con los datos del usuario, los parámetros de configuración y una serie de enlaces a todos los archivos generados por el sistema.

En la figura 6, podemos ver un ejemplo de informe generado por el cliente LABNET.

#### 3.1.3 Gestor de plantillas

Para acelerar el proceso de toma de datos, el cliente LABNET dispone de un gestor de plantillas que consiste en guardar todos los datos de configuración de un experimento en un archivo. El cliente dispone de la opción de cargar los datos de configuración

desde un archivo, y lanzar el experimento a ejecución. Las plantillas de los experimentos se pueden modificar utilizando una herramienta de edición como el *notepad* o similares.

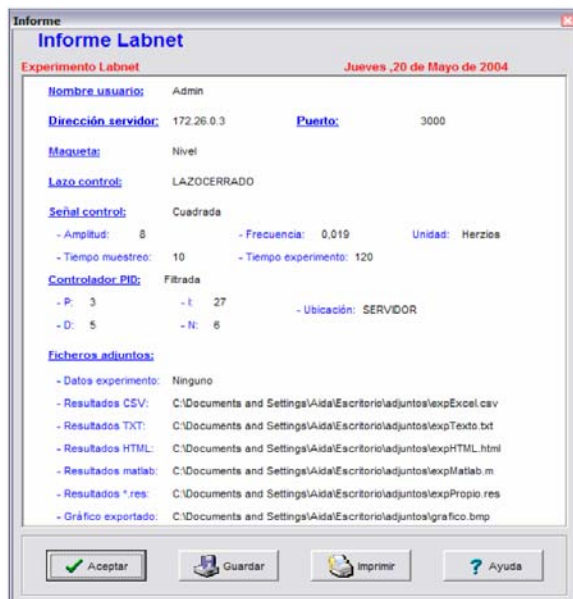


Figura 6: Ejemplo de informe generado por el cliente

## 3.2 EL SERVIDOR

El servidor ante todo, constituye un puente entre los clientes y las maquetas. Para realizar las diversas funciones de administración, se ha desarrollado un entorno GUI. Las funciones más importantes son la administración de los usuarios, la gestión de las conexiones y la gestión de las maquetas. La figura 7 nos muestra el servidor y podemos apreciar todas las opciones de administración en forma de pestañas

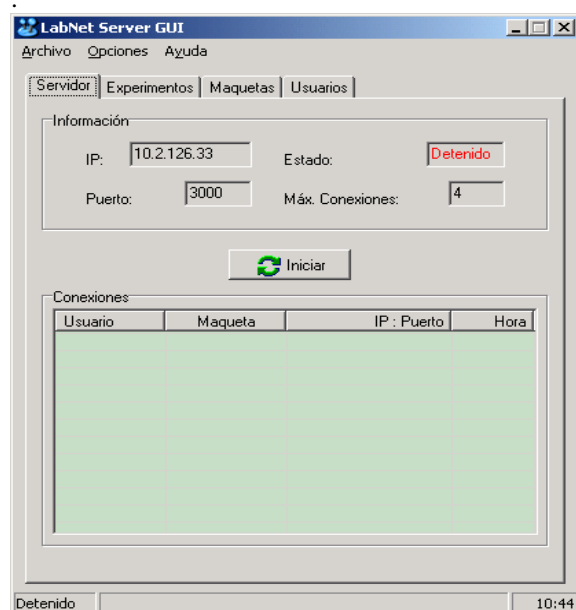


Figura 7: El servidor LABNET

### 3.2.1 Administración de usuarios

La aplicación permite dar de alta o de baja a usuarios finales. La autenticación de acceso al sistema se realiza mediante la utilización de un nombre de usuario y una clave. Una vez iniciado, el servidor está preparado para establecer conexiones con clientes y realizar las configuraciones oportunas para iniciar un experimento.

### 3.2.2 Administración de maquetas

Nuestro laboratorio cuenta con 10 maquetas, pero el servidor LABNET está preparado para conectar un máximo de 4. El cableado físico de las maquetas con el servidor así como la configuración de las tarjetas de adquisición de datos corresponde al administrador.

La administración de las maquetas consiste en establecer una correspondencia exacta entre las maquetas reales y sus equivalentes virtuales definidas en el servidor. Esta operación es importante para la correcta elección de varios parámetros como la polaridad de la señal de control, su escala, etc.

El sistema lleva la cuenta de las maquetas activadas, de las maquetas ocupadas y de las maquetas que aún están libres. Un usuario dispone de esta información justo al acceder al sistema.

La gráfica 8 muestra las diferentes maquetas que podemos activar y las tarjetas de adquisición de datos utilizadas.

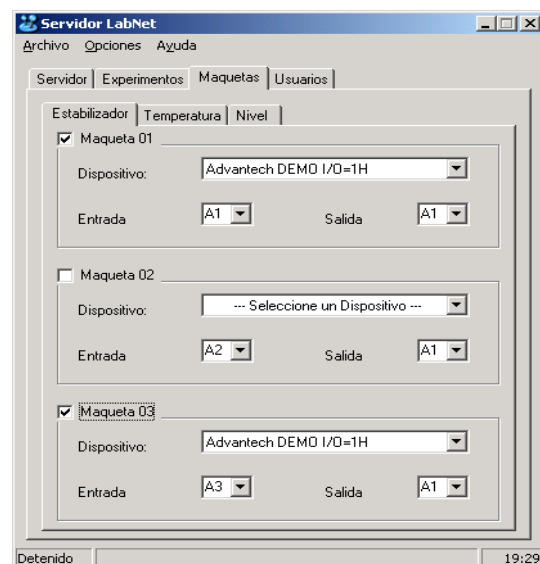


Figura 8: Administrador de las maquetas

### 3.2.3 Administración de los experimentos

El servidor es de tipo concurrente y puede atender a hasta un máximo de 4 clientes. Una vez validado un usuario del sistema y haya iniciado un experimento,

la opción experimentos del servidor nos muestra toda la información relativa al usuario y a sus experimentos en curso tal y como nos lo muestra la siguiente grafica

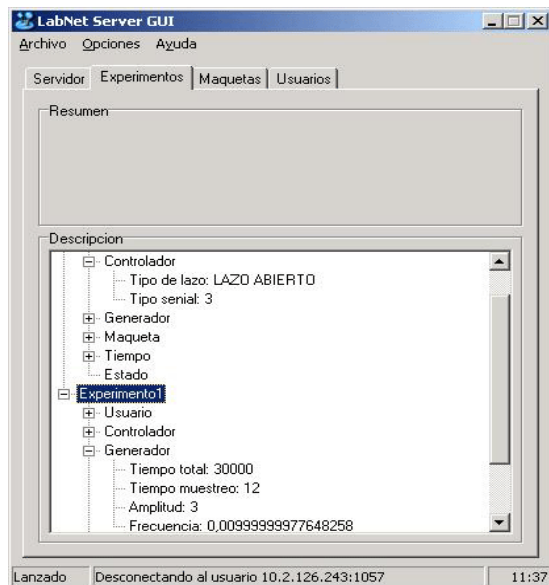


Figura 9: Administración de los experimentos

## 4 EXPERIMENTOS Y PRUEBAS

Durante la fase de desarrollo, hemos realizado varias pruebas parciales. Las más importantes son las pruebas sobre los temporizadores, el generador de señales, las pruebas sobre los controladores y finalmente los retardos debidos a la red

### 4.1 EL TEMPORIZADOR

Las pruebas realizadas sobre el reloj de windows son de tipo estadístico y son diseñadas para evaluar la robustez y la regularidad de los temporizadores. La prueba consiste en programar un temporizador periódico que ejecute una rutina cada un periodo de muestreo determinado. Esta misma prueba se realiza un número elevado de veces y se comparan los tiempos exactos de los experimentos frente a los tiempos reales.

Para los periodos del rango 10...50 milisegundos, y realizando la misma prueba 6000 veces, hemos obtenido el promedio del error del 0,2% y con una desviación estándar del orden de 0,0005. Esta prueba ha sido realizada solo para un temporizador y conviene generalizarla a varios temporizadores.

### 4.2 EL GENERADOR DE SEÑALES

El generador de señales se ha sometido a una prueba que consiste en generar una señal especificando los parámetros de amplitud y frecuencia y sacarla por el

canal de salida de la tarjeta de adquisición. Esta misma señal se captura y se procesa con Matlab/Simulink.

### 4.3 LOS CONTROLADORES

Las pruebas realizadas sobre el control en lazo cerrado en modo local consiste en la comparación de las señales de control de PID implementado y del mismo PID, pero simulado sobre Simulink.

El regulador simulado tiene como entradas las mismas señales utilizadas por el sistema real (la referencia y la salida del sistema). Después de algunos ajustes y escala de señales, no se ha observado ningún error.

La figura 10 muestra el aspecto de la respuesta de la maqueta de temperatura ante un escalón y controlada con un PID con D-filtrada.

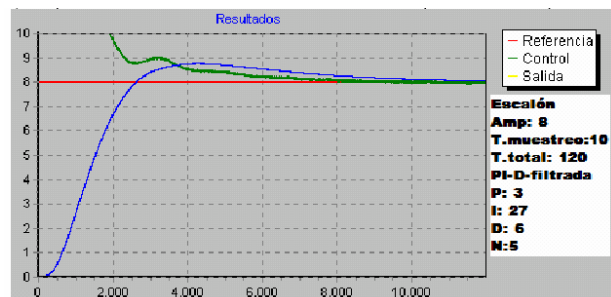


Figura 10: La respuesta de la maqueta de temperatura controlada con un PI-D-filtrada

### 4.4 EL CONTROL REMOTO

En el caso de ejecución del algoritmo de control en la parte cliente, hemos evaluado los retardos que se añaden en la transmisión por la red dependiendo del tipo de conexión (cable o inalámbrica).

Para ello, hemos utilizado el comando "ping" que nos da una aproximación de lo que tardan los paquetes de información en llegar de un equipo a otro, enviando 4 paquetes de 8 bytes cada uno.

Tiendo en cuenta las condiciones experimentales (con poco tráfico en la LAN de nuestro Campus), hemos observado, en promedio, que por el cable tenemos un retardo menor de 1 milisegundo, mientras este se dispara hasta casi 42 milisegundos en el caso del acceso inalámbrico.

El efecto de los retardo sobre la calidad del control remoto es mínimo en el caso del cable, ya que las maquetas que estamos utilizando son sistemas lentos y su ancho de banda bastante limitado. Esto nos permite utilizar periodos de muestreo algo más elevados. En el caso de la red inalámbrica, el retardo es algo elevado, pero aun así, podemos elegir periodos de muestreo más elevados ya que son sistemas lentos.

En todo caso, estamos añadiendo un retardo de 1) periodo de muestreo que es inherente a la implementación de un control digital.

Es casi imposible de evaluar con precisión el efecto de los retardos sobre la red utilizando un experimento real ya que no se puede repetir el mismo experimento. Sin embargo, una forma de evaluar este efecto, es comparar la respuesta del control local y remoto de un modelo de simulación.

## 5 CONSIDERACIONES SOBRE LA IMPLEMENTACIÓN

La implementación se ha codificado completamente en lenguaje C++ y se ejecuta sobre ordenadores personales con procesador Pentium III y sistema operativo Windows de 32 bits.

La gestión del tiempo tanto para la generación de señales o la acción de control se realiza utilizando el reloj multimedia de windows, el cual nos permite alcanzar periodos de muestreo de hasta 1 milisegundo. En la pagina [12], podemos encontrar todos los recursos necesarios para su correcta programación.

La concurrencia se ha llevado a cabo mediante hilos. Todos los se crean bloqueados al principio y se desbloquean si fuera necesario.

Finalmente, en cuanto a la implementación digital de los controladores PID, hemos utilizado el esquema de aproximación de Tustin por ser el más estable.

## 6 CONCLUSIONES

En este artículo, en un primer lugar, se han presentado los trabajos más importantes en los últimos años en el diseño de plataformas para el acceso a los laboratorios de forma remota. A continuación, la arquitectura de nuestro sistema ha sido presentada y explicada. Nuestra aplicación presenta una serie de ventajas en varios aspectos como la posibilidad de controlar varias maquetas de forma concurrente, la posibilidad de experimentar una acción de control local o remota, la generación de informes o la reutilización de plantillas. Estos aspectos hacen que nuestra aplicación sea funcional y productiva.

Sin embargo, pensamos que nuestro trabajo se puede mejorar, añadiendo una serie de funcionalidades como el cambio de los parámetro del regulador en tiempo real, la posibilidad de edición de las leyes de

control y su descarga en el servidor, o la posibilidad de incrementar el numero de maquetas de forma independiente del hardware.

## REFERENCIAS

- [1] Aburdene, M. F., Mastascusa, et all "A proposal for a remotely shared control systems laboratory", Proceedings of the ASEE 1991 Frontiers in Education Conference, Session 24A3, 1991, pp. 589-592.
- [2] Aktan, B, C.A. Bohus, "Distance Learning Applied to Control Engineering Laboratories". IEEE Transactions on Education", 1996 Vol 39(3), pp 320-326.
- [3] J. Albino Méndez Pérez et all, Controlweb: una herramienta para el análisis y simulación de sistemas de control en internetm XXIII jornadas de automática, León
- [4] Control Systems Lab, The University of Tennessee at Chattanooga: <http://chem.engr.utc.edu/Webres/Stations/controlslab.html>
- [5] Daniel Gasa Fonollosa, et all "...Laboratorio remoto de automática: plantas de variable discreta" XXIII jornadas de automática, Leon.
- [6] Garcia, R. Murillo et all "Real-time Remote Network Control of an Inverted Pendulum Using ST-RTL", 32nd ASEE/IEEE Frontiers in Education Conference, Boston, MA
- [7] Henry, J. (1996). "Controls Laboratory Teaching via the World Wide Web". In: Proceedings of the ASEE Annual Conference. Washington, USA, Session 3513
- [8] Jamahl W. Overstreet & Anthony Tzes (1999) "Real-Time Control Engineering Laboratory". IEEE Control Systems Magazine, October 1999, pp 19-34.
- [9] Johansson, M., Gäfvert, M. and Åström, K.J. "Interactive Tools for education in Automatic Contro", IEEE Control Systems Magazine, June. 1998
- [10] J. Sánchez and F. Morilla S. Dormido. "Teleoperation of an inverted pendulum through the world wide web" in workshop on internet based control education. ibce'01, pages 12-14, madrid, Spain, 2001.
- [11] [www.esi.uem.es/laboratorios](http://www.esi.uem.es/laboratorios)
- [12] <http://msdn.microsoft.com/library/>