

RECOLAB: PRÁCTICAS DE CONTROL SOBRE PROCESOS REALES VÍA INTERNET UTILIZANDO MATLAB

Rafael Puerto Manchón
Luis Miguel Jiménez García
César Fernández Peris
Ramón P. Ñeco García
Nicolás García Aracil

Departamento de Ingeniería
Área de Ingeniería de Sistemas y Automática
Universidad Miguel Hernández
Av. del Ferrocarril s/n
03202 Elche (Alicante)
Spain
EMail: r.puerto@umh.es

Resumen

En este artículo se presentan algunos resultados y avances conseguidos en el uso de un laboratorio remoto para la realización de prácticas de control de procesos. El sistema desarrollado por el área de Ingeniería de Sistemas y Automática de la Universidad Miguel Hernández permite que alumnos de diversas asignaturas/titulaciones puedan realizar prácticas con sistemas físicos reales desde un acceso remoto. Además la integración de sistemas de simulación de amplia difusión como Matlab/Simulink¹ dentro del proceso permite que los alumnos se familiaricen con el sistema de una forma casi inmediata. La motivación de este trabajo se basa en la, a menudo, escasa disponibilidad de sistemas físicos reales o laboratorios donde realizar los experimentos.

Palabras Clave: Prácticas de control a través de internet, control en tiempo real, control remoto.

1 INTRODUCCIÓN

En la actualidad, y sobre todo en las titulaciones técnicas como las ingenierías, es necesario realizar un número de prácticas elevado de forma que el estudiante adquiera unos conocimientos prácticos suficientemente extensos acorde con los conocimientos teóricos que se le suministran y así poder comprobar éstos sobre sistemas físicos reales. Sin embargo, la mayor parte de las universidades no poseen los recursos suficientes para poder crear unos laboratorios con equipos y sistemas físicos reales sobre los que desarrollar las prácticas de

forma eficiente. Ante esta situación se plantean dos alternativas: la primera consiste en realizar prácticas bajo sistemas de simulación que emulen el comportamiento de los sistemas físicos reales, y la segunda, establecer una conexión de los sistemas físicos a través de Internet de forma que los alumnos pueden manejar sistemas físicos reales desde cualquier ordenador conectado a la red disponiendo de una sensación más realista del comportamiento de diversos sistemas físicos.

Mediante este último planteamiento se puede diseñar y construir un conjunto de laboratorios remotos creando una red de laboratorios de forma que el estudiante pueda realizar una gran cantidad de prácticas sobre sistemas físicos reales y distintos a los que pueda tener una única universidad.

En este encuadre, en el presente artículo se presenta un sistema desarrollado en la Universidad Miguel Hernández a través del cual se posibilita la realización de prácticas de control sobre un sistema físico constituido por un servomotor de corriente continua [4][5].

Desde 1994, multitud de educadores, ingenieros, etc., han demostrado la viabilidad de realizar prácticas experimentales utilizando Internet [2], [3], [6]. En [1] y [2] se describen algunas de las características y funciones que debe tener una aplicación de control de procesos vía Internet. En el último periodo se han multiplicado el número de universidades que intentan realizar desarrollos paralelos para la realización de prácticas en laboratorios remotos [6],[7],[8]. Sin embargo, a diferencia de estas propuestas, en el sistema propuesto se utilizan herramientas software (Matlab/Simulink) o sistemas que los estudiantes conocen ampliamente.

¹ Matlab y Simulink son marcas registradas de The Mathworks Company

Los principales motivos por los que se ha optado por la plataforma Matlab/Simulink (con algunos toolboxes adicionales) para el desarrollo de esta aplicación son varios: primero, Matlab y Simulink constituyen una plataforma fiable, conocida y con amplio soporte técnico. Segundo, el tiempo de prototipado y desarrollo es bastante inferior al que presentan otras herramientas y plataformas (programación directa en un lenguaje de programación, etc.). Tercero, dicha plataforma suministra tanto herramientas para la ejecución remota de programas, como para la ejecución en tiempo real sobre un sistema físico, a través de un sistema de adquisición de datos, de un algoritmo de control determinado.

- Modulo PHP 4.1 para Apache
- Conexión a Internet con ancho de banda elevado si se desea video en tiempo real.

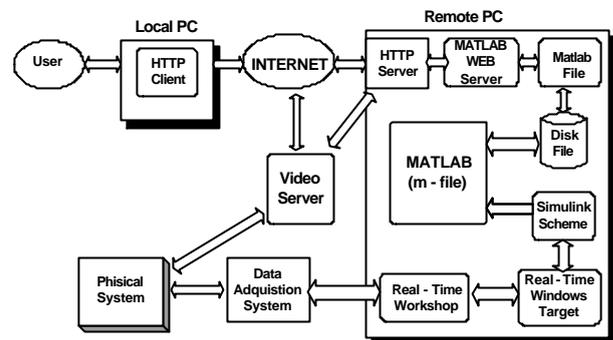


Figura 1. Arquitectura general de RECOLAB

2 ESQUEMA GENERAL DE LA ARQUITECTURA PRESENTADA

El esquema general de la arquitectura se muestra en la figuras 1 y 3. En ellas se puede apreciar los elementos tanto hardware como software necesarios en la zona local y remota:

- Zona local:
 - Computador con conexión a Internet capaz de ejecutar un cliente HTTP que soporte el protocolo HTML 4.0.
- Zona remota
 - Servidor: el equipo de pruebas consiste en un PC Pentium III 800EB con 256 MB. de RAM, disponiendo de suficiente potencia para ejecutar el servidor Web y las aplicaciones de control en tiempo real.
 - Sistema de adquisición de datos: tarjeta NI 6024E con E/S analógicas y digitales.
 - Sistema físico a controlar. Motor DC Feedback 33-002 (fig. 2): dispone de un motor de C/C, amplificador de potencia, tacómetro, encoder absoluto, potenciómetro y freno magnético.
 - Sistema de captación de imágenes y servidor de video (Sony EVI-D31 / servidor Axis 2400) (figura 3)
 - Sistema operativo que permita establecer directivas de seguridad de acceso. Actualmente se utiliza Windows 2000 Professional.
 - Matlab R12.1
 - Simulink 4.1.1
 - Matlab Web Server 2.1.2
 - Real – Time Windows Target Toolbox 2.1
 - Real – Time Workshop Toolbox 4.1
 - Control system Toolbox 5.1
 - Servidor HTTP Apache v. 1.1.1

3 FUNCIONAMIENTO INTERNO DE RECOLAB

En esta aplicación cabe destacar tres aspectos bien diferenciados:

- La comunicación cliente – servidor a través del protocolo HTML
- La ejecución de la aplicación en el servidor, codificada en PHP, que controla el acceso a los recursos y la comunicación entre Apache y las dos aplicaciones Matlab y Matlab Web Server.
- La ejecución en tiempo real de un esquema de control sobre un determinado sistema físico.

Para realizar las tareas referentes a los dos primeros puntos se utiliza el servidor HTTP Apache que gestiona la comunicación con el cliente. La aplicación que se ejecuta en el servidor utiliza el módulo PHP de Apache. Se trata de un lenguaje de programación estructurado que incorpora funciones para integrar de forma sencilla la aplicación con el servidor Web, así como el manejo de bases de datos.

Esta aplicación se encarga de presentar el interfaz de usuario, realizar el control de acceso, enlazar los datos del cliente con Matlab, y sincronizar la ejecución de los diferentes módulos. Para establecer esta comunicación se utiliza una aplicación intermedia, el toolbox Matlab Web Server. Este toolbox permite de forma flexible adaptar los datos enviados por el cliente, así como recopilar datos generados por el núcleo de Matlab y enviarlos a una determinada página Web que es mostrada posteriormente al cliente. Además permite generar también cualquier gráfico que se pueda realizar

desde Matlab y adjuntarlo igualmente a una página Web.

Para realizar las tareas concernientes al último punto (ejecución en tiempo real de un determinado esquema de control sobre un sistema físico) se utilizan las siguientes aplicaciones y toolboxes: Matlab, Simulink, Real – Time Workshop y Real – Time Windows Target. Estos dos últimos permiten realizar una ejecución en tiempo real de un esquema Simulink sobre un determinado sistema de adquisición de datos (y por tanto sobre el sistema físico conectado a él). Por una parte, Real – Time Workshop proporciona la conexión en tiempo real con el sistema de adquisición de datos, mientras que Real – Time Windows Target permite la ejecución de un esquema Simulink sobre Real – Time Workshop. El hecho de ejecutar directamente un esquema Simulink supone una ventaja añadida ya que el tiempo y la complejidad de trabajar directamente en Simulink se reducen drásticamente, permitiendo una fácil creación y modificación de esquemas.

Aunque el software descrito permite resolver ambos problemas por separado (comunicación y ejecución en tiempo real), no sirve, sin embargo, para tratar el problema de forma conjunta. Esto es, no existe un medio de comunicación eficiente y directo entre Matlab Web Server y la sesión de Matlab que contiene los datos generados por la ejecución en tiempo real. Cabe destacar que aunque Matlab Web Server permite ejecutar programas de Matlab sobre un núcleo del mismo, éste núcleo (incluido en Matlab Web Server) no puede ejecutar esquemas Simulink mediante los toolboxes Real – Time Workshop y Real – Time Windows Target. Estos son los motivos que hacen necesario establecer un mecanismo de comunicación entre una sesión de Matlab (que ejecuta en tiempo real el regulador elegido sobre el sistema físico) y Matlab Web Server.

Para solventar el problema anterior se ha establecido una sincronización y transferencia de datos utilizando una aplicación CGI ejecutada desde el servidor Apache y codificada en PHP. Esta aplicación realiza la sincronización mediante semáforos implementados mediante ficheros.

Es necesario resaltar también que sólo se puede realizar una ejecución por usuario en un mismo instante de tiempo, por lo que es necesario garantizar el acceso al sistema a un único usuario. Esto implica que mientras un usuario esté controlando el sistema, ningún otro podrá hacerlo hasta que éste finalice. No ocurre lo mismo cuando se trabaja en simulación, pues este sistema permite también simular un esquema Simulink con los

parámetros especificados por el usuario (como se tratará más adelante), ya que en este caso sí pueden acceder y simular más de un usuario al mismo tiempo.

La ejecución en tiempo real requiere la validación del usuario en el sistema RECOLAB. La aplicación PHP se encarga de realizar esta validación entonces los módulos del sistema (figura 2).

4 UTILIZACIÓN DE RECOLAB

Para utilizar RECOLAB lo único necesario es un ordenador con un cliente Http y una conexión a Internet, aparte, claro está de tener una cuenta en el sistema. El usuario se conecta a RECOLAB accediendo a la dirección <http://recolab.umb.es>. En la página principal se puede ver la estructura general de RECOLAB. Nos centraremos exclusivamente en la utilización de la ejecución en tiempo real por ser esta la alternativa más importante.

Sea cual sea la ejecución a realizar se deben especificar los siguientes parámetros:

- Sistema físico a controlar (Motor Feedback)
- Modelo de control (Control en Velocidad o Posición)
- Tipo de regulador (por ejemplo P, PI, PD, PID, realimentación del estado, etc.)
- Parámetros asociados al regulador elegido (polos, ceros, ganancia, matriz de realimentación, etc.).
- Eventualmente otro tipo de parámetros como referencia, periodo de muestreo, tiempo de simulación, etc.

Una vez concluida la ejecución se envía al usuario una página con los datos más relevantes de la misma así como gráficas de las señales más importantes (referencia, salida, señal de control, etc.). Se permite además descargar los datos obtenidos en la ejecución con el fin de poder realizar un análisis más profundo. Adicionalmente se dispone de un servidor de video que permite una realimentación visual al usuario de la respuesta del sistema al esquema de control introducido.



NOTA: El nombre y el password son sensibles a mayúsculas

Figura 2. Validación en RECOLAB

5 EJEMPLO: CONTROL EN TIEMPO REAL DE UN MOTOR DE CC.

En este ejemplo se muestra los pasos a seguir para realizar el control en tiempo real de un motor de corriente continua (figura 3).

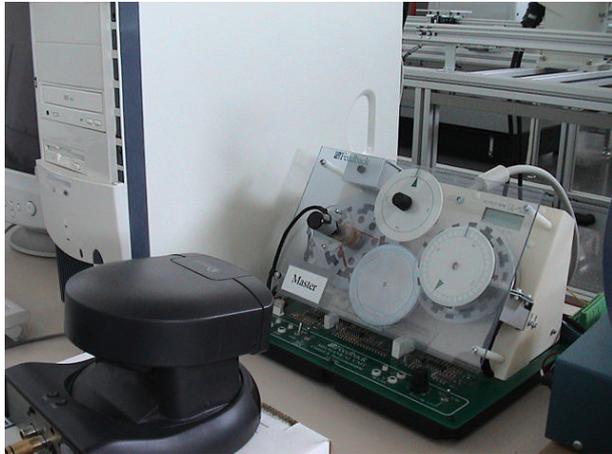


Figura 3 Motor Feedback 33-002

El enunciado del problema planteado al alumno podría ser el siguiente:

Se desea controlar en velocidad un motor de corriente continua cuya función de transferencia viene dada por

$$G(s) = \frac{q}{V} = \frac{2.304}{s + 2.22}$$

donde q es la velocidad angular en rad./seg. y V es la tensión de entrada aplicada al mismo. Diseñar el mejor regulador tipo PID discreto ($T = 0.01$ seg.) tal que la salida del mismo presente un error de posición nulo.

Los pasos que debe seguir el alumno son:

- Discretización del sistema al periodo de muestreo indicado
- Verificación de las especificaciones y elección del regulador, en este caso un regulador PI
- Ajuste de parámetros del regulador mediante el método del lugar de las raíces.

En este caso el regulador pedido tiene la forma

$$G_R(z) = \frac{K_p(z-a)}{(z-1)}$$

donde los valores de K_p y a , para que se cumplan las especificaciones requeridas, toman los valores 0.045 y 0.69 respectivamente.

El siguiente paso sería introducir los parámetros deseados, tal y como se muestra en la figura 4.

Figura 4. Pantalla de introducción de datos.

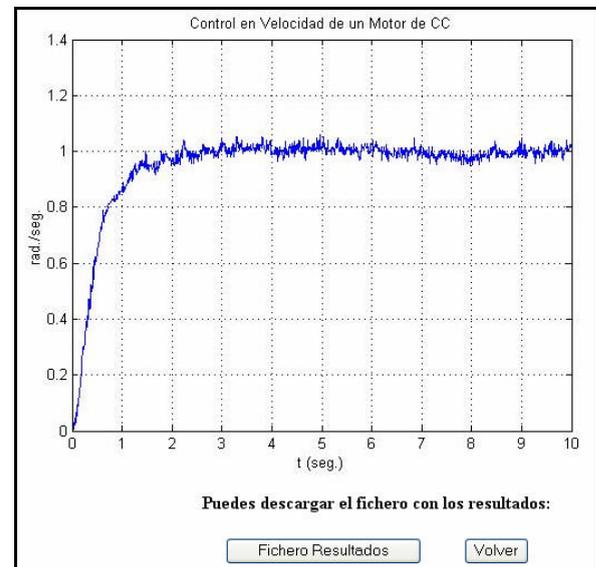


Figura 5. Pantalla de resultados.

Una vez lanzada la ejecución se remiten los resultados de la misma al usuario tal y como se muestra en las figuras 5 y 7.

Como se puede observar en la figura 4 se permite al usuario descargar un fichero con los datos numéricos (en formato Matlab) más relevantes de la ejecución (posición, velocidad, referencia, acción de control, etc.).

En la figura 6 se muestra una vista general del servidor Web RECOLAB.

6 CONCLUSIONES

En este trabajo se ha presentado la arquitectura básica de un laboratorio remoto de control de procesos. Las características principales de esta aplicación se pueden resumir en los siguientes puntos:

- Se ha utilizado una plataforma de desarrollo (Matlab) ampliamente utilizada y por ello con amplio soporte técnico.
- El empleo de dicha plataforma supone un menor tiempo de desarrollo y mantenimiento.
- Se ha solventado los problemas de comunicación y sincronización que aparecen al utilizar distintos elementos de dicha plataforma.
- La arquitectura diseñada es completamente modular, lo cual implica que el coste de añadir nuevos modelos o modificar el comportamiento de los ya existentes es bastante reducido.

7 BIBLIOGRAFÍA

- [1] Shor Molly H. and Robson Robby (2000): "A Student – Centered Feedback Control Model of the Educational Process". 30th ASEE/IEEE Frontiers in Education Conference. Kansas City, MO, October 18 – 21, 2000.
- [2] Shor Molly H. (2000): "Remote – Access Engineering Educational Laboratoires: Who, What, When, Why and How?", Proceedings of the 2000 American Control Conference. Chicago, IL, June 28 – 30th, 2000
- [3] Schmid, Chr.(2000): "Virtual Control Laboratories and Remote Experimentation in Control Engineering". Proc. 11th EAEEIE Annual Conference on Innovations in Education for Electrical and Information Engineering, University of Ulm, Ulm, S.213-218
- [4] Puerto, R., Jiménez, L.M., Reinoso, O., Fernández, C., (2002): RECOLAB Laboratorio de prácticas de control de procesos vía Internet. EIWISA'02. III Jornadas de Enseñanza vía Internet/Web de la Ingeniería de Sistemas y Automática. Alicante. pp. 33-38
- [5] Puerto, R., Jiménez, L.M., Reinoso, O., Fernández, C., (2002): Laboratorio vía Internet para Control de Procesos. VirtualEduca'02. Valencia, Junio 2002
- [6] Ludevit X., Hernández D., Castelló R., Basañez L.: Proyecto de Laboratorio Remoto de Automática: Objetivos y Arquitectura propuesta. EIWISA'02. III Jornadas de Enseñanza vía Internet/Web de la Ingeniería de Sistemas y Automática. Alicante. pp. 93-98
- [7] Salichs M., Khamis A. Rodríguez F. Rivero M.: Laboratorio a distancia vía internet en Robótica Móvil. EIWISA'02. III Jornadas de Enseñanza vía Internet/Web de la Ingeniería de Sistemas y Automática. Alicante. pp. 71-74
- [8] Pastor R., Sánchez J., Dormido S.: Ejecución remota de modelos simulink utilizando related. EIWISA'02. III Jornadas de Enseñanza vía Internet/Web de la Ingeniería de Sistemas y Automática. Alicante. pp. 121-126

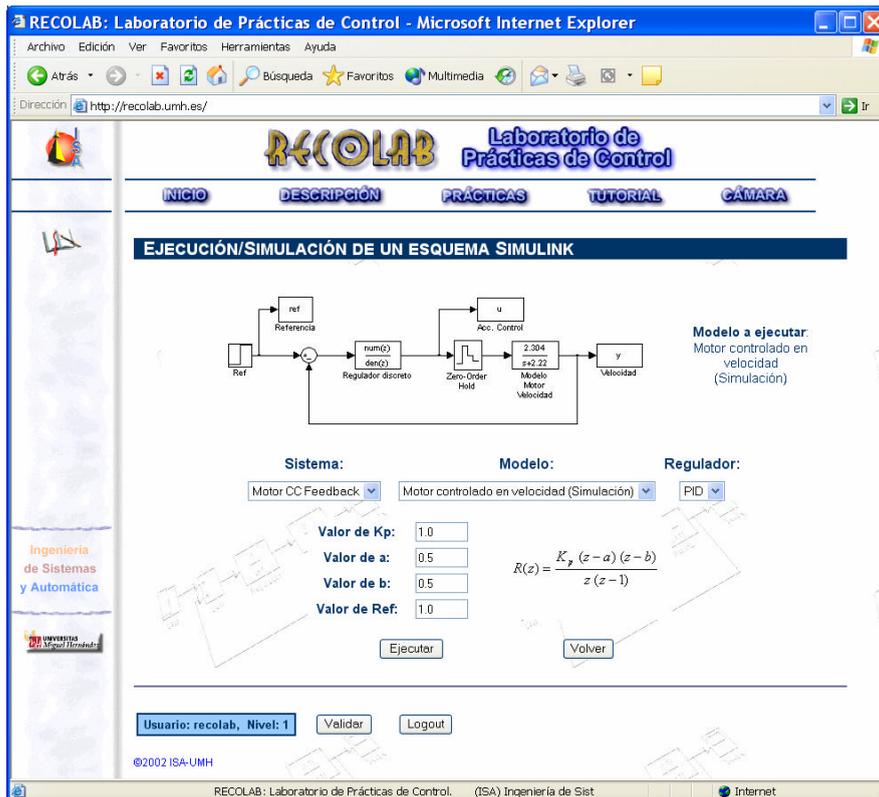


Figura 6. Vista general de la aplicación Web

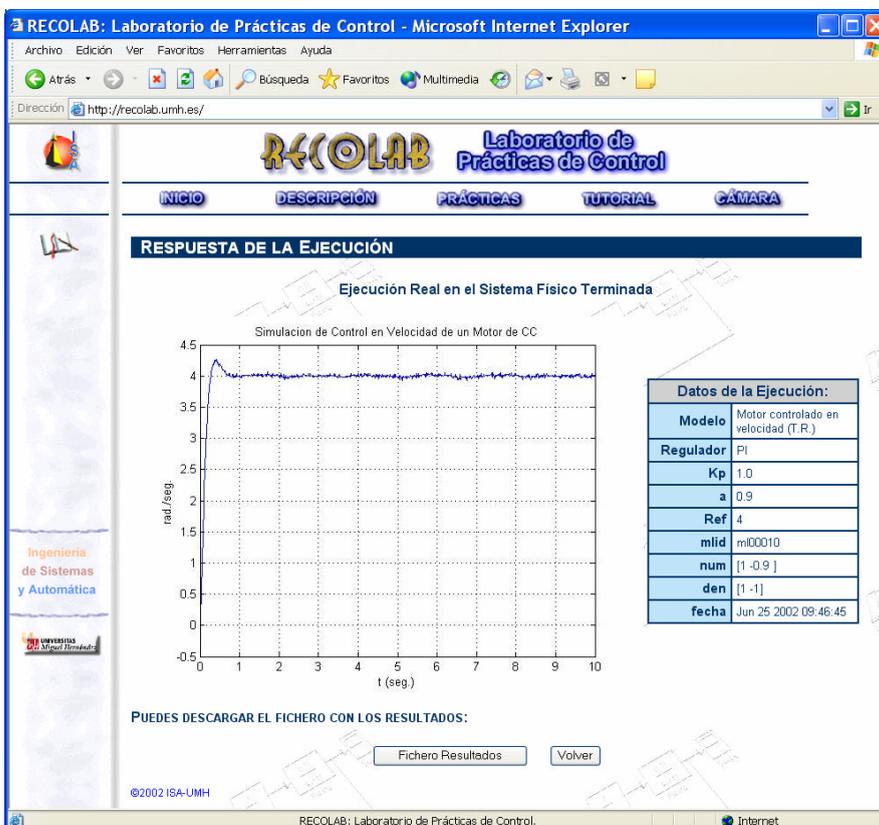


Figura 7. Página Web de resultados de la ejecución en tiempo real