

RECOLAB: LABORATORIO DE PRÁCTICAS DE CONTROL DE PROCESOS VÍA INTERNET

Rafael Puerto Manchón, Luis Miguel Jiménez García, Óscar Reinoso García, César Fernández Peris

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

Resumen

En este documento se presenta la arquitectura general de una aplicación para la ejecución remota, vía Internet y en tiempo real, de un regulador sobre un proceso físico determinado. Dicha aplicación se ha realizado utilizando la plataforma Matlab/Simulink¹ para el desarrollo de la misma. La motivación de este trabajo se basa en la escasa disponibilidad de sistemas físicos reales o laboratorios donde realizar los experimentos. Se presenta además, un ejemplo ilustrativo (control de un motor de CC) que muestra la validez y la aplicación de la arquitectura presentada.

Palabras Clave: Control de procesos, tiempo real, control remoto, Internet

1 INTRODUCCIÓN

El presente trabajo pretende proporcionar una arquitectura general para la ejecución remota en tiempo real de procesos físicos por medio de Matlab/Simulink. La principal motivación es la carencia de, por una parte, maquetas o sistemas físicos reales, y por otra, la escasa disponibilidad de horarios en laboratorios donde los alumnos de las asignaturas de control y automática puedan desarrollar los contenidos teóricos dentro de un marco práctico. Así pues, la presente aplicación está orientada al control remoto en tiempo real de un sistema físico a través de Internet y con Matlab como plataforma de desarrollo. Dicha aplicación va a permitir al alumno, primero, simular por medio de Internet el funcionamiento de un regulador dado para un proceso físico determinado, y segundo, ejecutar en tiempo real, sobre el proceso físico en cuestión, dicho regulador. Así pues, se permite al alumno desarrollar dichas actividades sin tener presencia física en el laboratorio donde se encuentra el sistema. Además, la aplicación devuelve al usuario toda la información

referente a la ejecución realizada, además de gráficas y otros datos de interés.

Desde 1994, multitud de educadores, ingenieros, etc., han demostrado la viabilidad de realizar prácticas experimentales utilizando Internet [8]. En [7] y [8] se describen algunas de las características y funciones que debe tener una aplicación de control de procesos vía Internet. Últimamente, algunos autores han realizado importantes avances en el diseño de laboratorios remotos de control. En [10] se describe un laboratorio de control en tiempo real vía Internet, y en [6] se muestra un sistema para prácticas de visión por computador. En estos casos, la mayoría de autores han optado por utilizar herramientas software de distintos vendedores y no siempre compatibles.

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, Simulink y los toolboxes necesarios 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. Por último y no menos interesante es la gran cantidad de investigadores que utilizan esta plataforma como herramienta de desarrollo para simulaciones y generación de aplicaciones.

2 ESQUEMA GENERAL DE LA ARQUITECTURA PRESENTADA

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

¹ Matlab and Simulink are trademarks of The Mathworks Company

- Zona local:
 - Computador
 - Cliente http
 - Conexión a Internet
- Zona remota:
 - Computador
 - Sistema de adquisición de datos
 - Sistema físico a controlar
 - Sistema operativo que permita establecer directivas de seguridad de acceso
 - Matlab R12
 - Simulink
 - Matlab Web Server
 - Real – Time Windows Target Toolbox
 - Real – Time Workshop Toolbox
 - Conexión a Internet

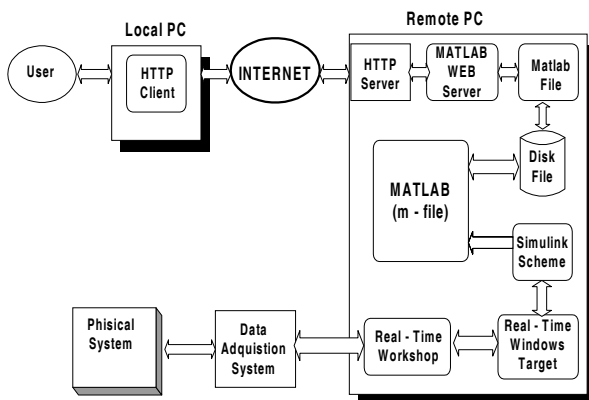


Figura 1: Arquitectura general de RECOLAB

3 FUNCIONAMIENTO INTERNO DE RECOLAB

En esta aplicación cabe destacar dos aspectos bien diferenciados:

- La comunicación cliente – servidor a través del protocolo Html
- La ejecución en tiempo real de un esquema de control sobre un determinado sistema físico.

Para realizar las tareas referentes al primer aspecto se utiliza un servidor http y el toolbox Matlab Web Server. Este toolbox permite de forma flexible recopilar datos generados por un núcleo de Matlab y enviarlos a una determinada página Web que se mostrará 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 segundo 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, 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 (entre Matlab Web Server y Matlab) por medio de ficheros. En términos generales, el procedimiento a seguir es el siguiente:

1. En la zona local, el usuario rellena un formulario con los datos de la ejecución a realizar (sistema físico a controlar, tipo de regulador, parámetros del regulador, etc.).
2. Los datos son enviados y recogidos por el servidor remoto que ejecuta un programa sobre Matlab Web Server que se encarga de generar un fichero en disco conteniendo los datos necesarios para la ejecución.
3. Una vez escrito el fichero de datos se lanza una sesión de Matlab (si no hay ninguna activa) que ejecuta automáticamente un programa de Matlab.

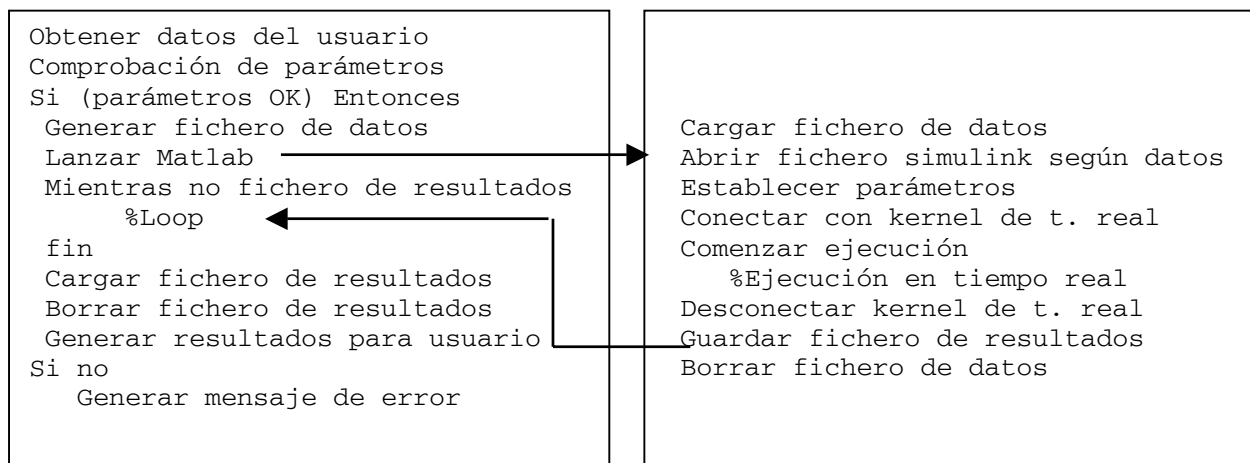


Figura 2: Código de comunicación y sincronización entre Matlab Web Server y Matlab

4. Dicho programa lee el contenido del fichero de datos, y en base a ciertos parámetros contenidos en él, ejecuta el esquema Simulink correspondiente en tiempo real y con los parámetros del controlador especificados.
5. Una vez finalizada la ejecución, Matlab genera un fichero de resultados que es leído por un programa ejecutado por Matlab Web Server. Este mismo programa se encarga de pasar los resultados generados a la página Web de respuesta.

En la figura 2 se describe en pseudocódigo el procedimiento de sincronización y comunicación.

No hay que olvidar 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 este 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.

4 UTILIZACIÓN DE RECOLAB

Aunque el punto fuerte de esta aplicación es la ejecución en tiempo real, se ha pretendido también que RECOLAB sea un punto de referencia y ayuda a los alumnos que cursan asignaturas relacionadas con el control de procesos y dispositivos físicos. Para ello, RECOLAB ofrece información y exposiciones teóricas acerca de los temas impartidos en dichas asignaturas, así como multitud de ejemplos y

tutoriales que ayudarán al alumno a alcanzar los objetivos propuestos.

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 sistema². El usuario se conecta a RECOLAB accediendo a la dirección <http://recolab.umh.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 (por ejemplo motor de CC o cilindro neumático).
- 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 rellenado el formulario con los datos pertinentes, el envío de éste supone la ejecución del algoritmo de control elegido sobre el sistema físico. Notemos que en ningún momento se verifica que el comportamiento del sistema con ese regulador no provoque un comportamiento inestable del mismo (comportamiento éste bastante interesante para el alumno). En la comprobación de parámetros del regulador simplemente se comprueba que éstos estén dentro de unos rangos razonables. Asimismo, para proteger la integridad del sistema, se añaden los bloques de saturación necesarios dentro del esquema Simulink, a la vez que se lleva el sistema a un punto de reposo, cuando se detecta en la misma ejecución

² Sólo necesario para la ejecución en tiempo real

un comportamiento inestable o al finalizar la misma. Todo esto implica, que con esta comprobación tan poco exhaustiva, los procesos físicos que se pueden conectar son aquellos que no puedan provocar daños físicos o materiales al presentar un comportamiento inestable (por ejemplo, motor de corriente continua).

Una vez concluida la ejecución se envía al usuario una página con los datos más relevantes de la misma así como gráficas de la 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.

Por último, los resultados generados por cada usuario se guardan en ficheros únicos con el fin de poder detectar fallos en el sistema.

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.

El equipo con el cual se está trabajando consiste en:

- Ordenador PC PIII, 800 MHz., 256 Mb. RAM.
- Tarjeta de adquisición de datos NI 6024E.
- Motor de corriente continua Feedback 33-002.
- Software:
 - Sistema operativo Windows 2000 Professional (SP 2)
 - Matlab R12.1
 - Simulink 4.1.1
 - Real – Time Workshop
 - Real – Time Windows Target
 - Matlab Web Server
 - Control System Toolbox 5.1
 - Servido http Apache v. 1.1.1

El enunciado del problema planteado al alumno podría ser el siguiente³:

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

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

donde $\dot{\theta}$ 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.

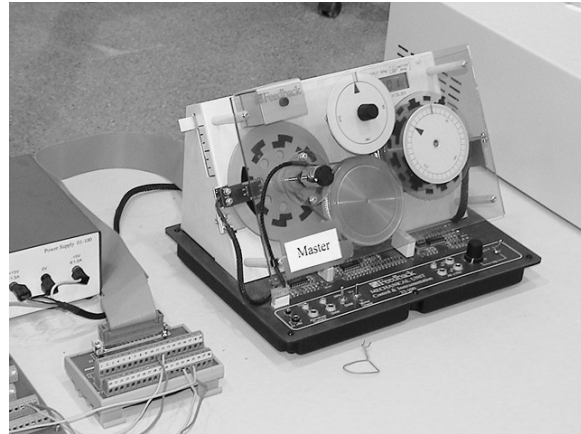


Figura 3: Motor Feedback 33-002

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.

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

³ Este ejemplo es el caso más sencillo. En particular, para el sistema elegido existe un gran abanico que reguladores implementados tanto para control en posición como en velocidad.

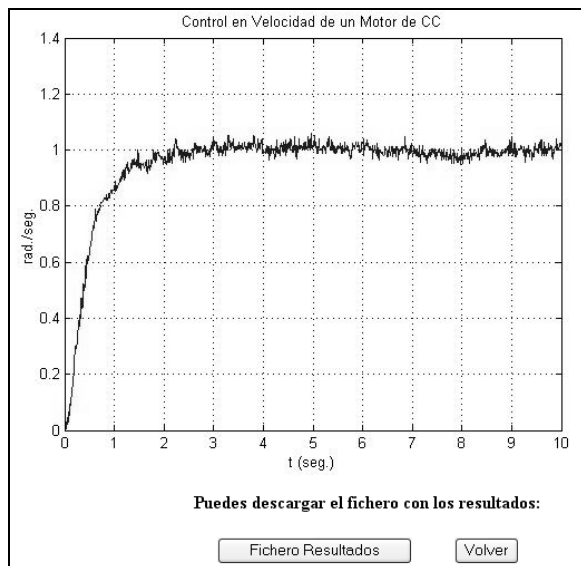


Figura 5: Pantalla de resultados.

Como se puede observar en la figura 5 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.

Como trabajos futuros se plantea la inclusión de video como parte de la respuesta hacia el usuario así como de otras técnicas que incrementen la interacción usuario – sistema.

Referencias

- [1] Aracil, A. y Albertos, P. (1993): Problemas de Regulación Automática. Servicio de Publicaciones de la ETSIIM
- [2] Aracil, R. y Jiménez, A. (1993): Sistemas Discretos de Control. Servicio de Publicaciones de la ETSIIM.
- [3] Kuo, B.C. (1996): Sistemas de Control Automático. Prentice Hall
- [4] Ogata, K. (1998): Ingeniería de Control Moderna. Prentice Hall
- [5] Puerto, R., Reinoso, O., Neco, R., García, N., Jiménez, L.M. (2001): Remote Lab for Control Applications using Matlab. IBCE 2001. Madrid.
- [6] Sebastián J.M., García D., Santo, D, Campoy, P. (1999): “Proyecto Títere. Realización de prácticas de laboratorio en puestos de trabajo remotos mediante la transmisión de imágenes por red telefónica conmutada”. XIX Jornadas de Automática, pp. 21-27.
- [7] 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.
- [8] 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
- [9] Schmid, Chr. (1992): “Real – Time Control with CADACS-PC”. Recent Advances in computer – Aided Control Systems Engineering. M. Jamshidi and C.J. Herget (Editors), 337 – 355. North – Holland, Amsterdam, 1992.
- [10] Schmid, Chr. (2000): “Remote Experimentation in Control Engineering”. 45th International Scientific Colloquium. Ilmenau Technical University. October 4 – 6, 2000.
- [11] Schmid, Chr. (1999): “Virtual Laboratory for Engineering Education”. Proc. 19th World Conference on Open Learning and Distance Education, ICDE'99, Wien, Paper u2a1.

[12] 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

[13] The Mathworks Inc. (2000): Matlab R12, Simulink, Real – Time Workshop, Real – Time Windows Target and Matlab Web Server. Online Manuals.

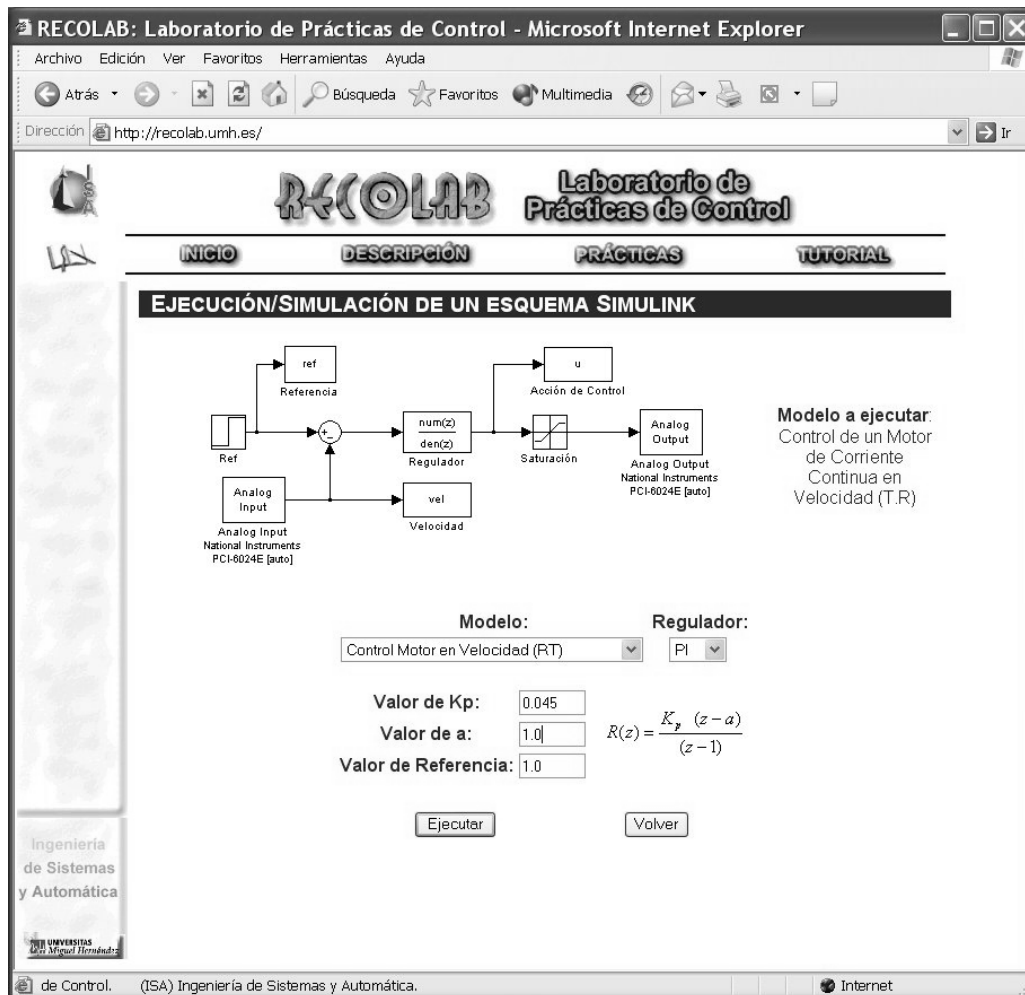


Figura 6: Vista general de la aplicación Web