

CONTROL REMOTO DE UN SERVOMECANISMO EN TIEMPO REAL

R. URRETABIZKAYA¹, I. OLEAGORDIA¹, A. ORDAZ², A. NORIEGA²

¹Universidad del País Vasco, EUITI de Bilbao

²Universidad Autónoma de Querétaro, Facultad de Ingeniería.

Este trabajo presenta el control remoto de un servomecanismo lineal, como una aplicación de laboratorio virtual para el desarrollo de las prácticas de las asignaturas de control, sistemas lineales y servomecanismos, con esta herramienta didáctica, los alumnos pueden controlar un sistema real desde cualquier punto de conexión a Internet. En el trabajo se presenta el desarrollo del hardware de control y el sistema, su control en Matlab y la comunicación de este con el usuario remoto a través de applets de java.

1. Introducción

El desarrollo experimentado por los medios de comunicación ha motivado que docentes e investigadores de diversas universidades se planteen la forma de realizar prácticas y experimentos de laboratorio con equipo y sistemas que se encuentran en otros centros o lugares de enseñanza, compartiendo recursos y experiencias de una forma mucho más eficiente y por lo mismo ampliando las fronteras y posibilidades en la educación y el intercambio de conocimiento.

Casi todos los desarrollos de laboratorios virtuales enfocados a la docencia en el área de la electrónica y el control se basan en sistemas cliente servidor[1,2], aunque hay algunas excepciones interesantes como Abler[3] que plantea un sistema distribuido para la “telecolaboración”, que pretende facilitar y mejorar las interacciones alumno-profesor y alumno-alumno, que aunque no es un laboratorio virtual puede resultar de interés como una utilidad extra o añadida de un laboratorio remoto. Pero al margen de este tipo de excepciones podemos asegurar que las soluciones planteadas para un laboratorio remoto tendrán que basarse en una topología cliente servidor.

Dentro de esta línea, encontramos una muy amplia variedad de herramientas utilizadas, técnicas y aplicaciones pero que para efectos de análisis los podemos clasificar en dos grupos de interés. Uno es el que se basa en el uso de comandos de hipertexto para gobernar la aplicación, ya sean estas comandos nativos de HTTP o sus derivados CGI, XML, etc. El otro grupo de interés se caracteriza por el uso de applets de Java para establecer el control de la aplicación. Cabe aclarar que la mayoría de los sistemas basados en HTTP, algunas veces también utilizan en mayor o menor medida applets para algunas tareas de intercambio de datos, pero en todo momento el usuario gobierna la aplicación mediante comandos de hipertexto y recibe por respuesta nuevas páginas WEB, mientras que en las otras el usuario accede a la aplicación a través de una liga de hipertexto, pero inmediatamente una aplicación se descarga y la interacción del usuario es a través del correspondiente applet de Java[4].

La mayoría de los que trabajan en sistemas basados en HTTP, utilizan el LabView como herramienta de control e interfaz con el hardware en el servidor, por un lado porque éste ya cuenta con un servidor web y utilidades para el manejo de comandos CGI en el Internet Tool Kit y G Web Server [5], son sistemas de muy fácil actualización y crecimiento, pero generalmente con dificultades para el control en tiempo real de las aplicaciones[6,7], en consecuencia, esta limitación algunos la compensan con applets[8,9].

Por otro lado los que trabajan basados principalmente en applets suelen escoger como plataforma para el servidor el Matlab, por sus amplias prestaciones para el desarrollo de algoritmos de control y una interfaz inmediata con programas de Java como se aprecia en los trabajos [1,10].

Aunque se conocen diversos métodos y técnicas para el desarrollo de laboratorios virtuales remotos, no existen reglas o guías que nos permitan elegir el método mas adecuado a una aplicación específica, por lo que con el proyecto se pretende adquirir la experiencia y establecer algunos parámetros para la selección de los métodos y técnicas de desarrollo de laboratorios remotos, que nos permitan trabajar en colaboración con otros centros de enseñanza, compartiendo materiales didácticos de instrumentación virtual de forma rápida y eficiente.

Partiendo de la base de que por sus características, las técnicas basadas en comandos de hipertexto resultan mas fáciles y rápidas de implementar pero requieren, para la misma tarea, una velocidad y calidad de servicio en la red IP mucho mayor que las técnicas basadas en Applets, en proyectos anteriores, hemos controlado sistemas con un tiempo de respuesta muy elevado (orden de minutos) por lo que sus requerimientos de periodo de muestreo y por ende velocidad de comunicación son mucho menores, para el control de estos sistemas se utilizaron comandos de hipertexto y paginas WEB obteniendo excelentes resultados, pero esas mismas técnicas resultan inapropiadas ahora que el sistema a controlar es un servomecanismo lineal con un tiempo de respuesta de pocos segundos por lo que requerirá una mayor tasa de muestreo. De esta forma y probando ambos sistemas sobre las dos técnicas de comunicación en laboratorios virtuales remotos, podemos establecer las limitaciones y alcances de cada una de ellas.

2. Objetivos y metodología

Con este proyecto, se pretende aprovechar las nuevas tecnologías para a mediano plazo poner al alcance de los alumnos una mayor cantidad y diversidad de experimentos prácticos y casos de estudio que favorezcan la construcción del conocimiento relacionado con la teoría del control automático y con la práctica de la disciplina

El objetivo principal de este trabajo es desarrollar e implementar un laboratorio virtual que permita compartir recursos y experiencias entre los laboratorios de electrónica de la Universidad del País Vasco y la Universidad Autónoma de Querétaro, de donde se desprenden varios objetivos particulares y específicos que se describen a continuación:

- Poner al alcance de los alumnos de ambas universidades una mayor variedad de equipos, prácticas y facilidades para su desarrollo.
- Involucrar a los alumnos en el manejo de nuevas tecnologías vinculadas a las comunicaciones (TIC) y electrónica.

Dar un primer paso en el establecimiento de una metodología para el desarrollo de este tipo de aplicación, que abra las puertas a futuros desarrollos en esta materia

La metodología seguida en el desarrollo de este proyecto consiste, a grandes rasgos, en la siguiente secuencia de acciones:

- Determinación de las características y necesidades del sistema.
- Descripción de las etapas de hardware, software y comunicación.
- Desarrollo del hardware del sistema y manejo de comunicaciones RS232.
- Desarrollo del programa con el algoritmo de control y que interactúa con el hardware del sistema de forma local.
- Desarrollo de los applets de java, paneles frontales para manejo remoto y enlace de estos con Matlab para la publicación de los instrumentos virtuales hechos en Matlab.
- Pruebas locales de funcionamiento de PC a PC dentro del ámbito de la intranet (propio laboratorio)
- Instalación, configuración y puesta en marcha de un sistema de administración de usuario
- Prueba de los instrumentos virtuales con usuarios ajenos al grupo de investigación

3. Desarrollo

El desarrollo del sistema esta compuesto por tres elementos, el hardware, que realiza la interfase entre el proceso (servomecanismo) y el ordenador servidor, el software de control y servidor de aplicación y el applet de interfase de usuario, un esquema general puede observarse en la figura 1.

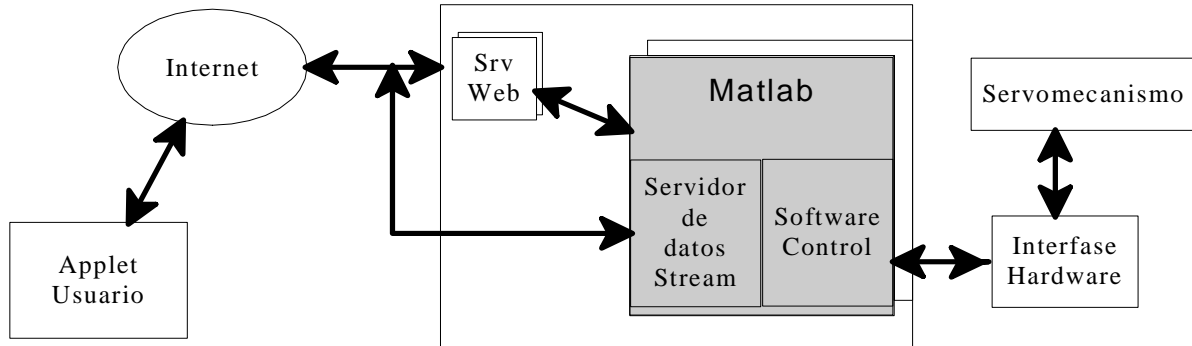


Figura 1. Esquema general a bloques del sistema

El desarrollo del sistema esta compuesto por tres elementos, el hardware, que realiza la interfase entre el proceso (servomecanismo) y el ordenador servidor, el software de control y servidor de aplicación y el applet de interfase de usuario, un esquema general puede observarse en la figura 1.

3.2. Desarrollo de hardware

El Hardware que desarrollamos, se compone de una tarjeta para la adquisición de datos y manejo de señales de control que cuenta con comunicación RS232 a 115200 baudios para la comunicación con el PC, 2 generadores de señales PWM, 8 señales de control, entradas digitales acondicionadas para la lectura de un encoder de cuadratura y 8 canales analógicos con rango de -5v a 5v. La potencia al motor del servomecanismo se suministra con un puente H L298 que se maneja con una señal PWM y una señal de control de sentido, el diagrama esquemático y su implementación se observan en las figuras 2 y 3 respectivamente.

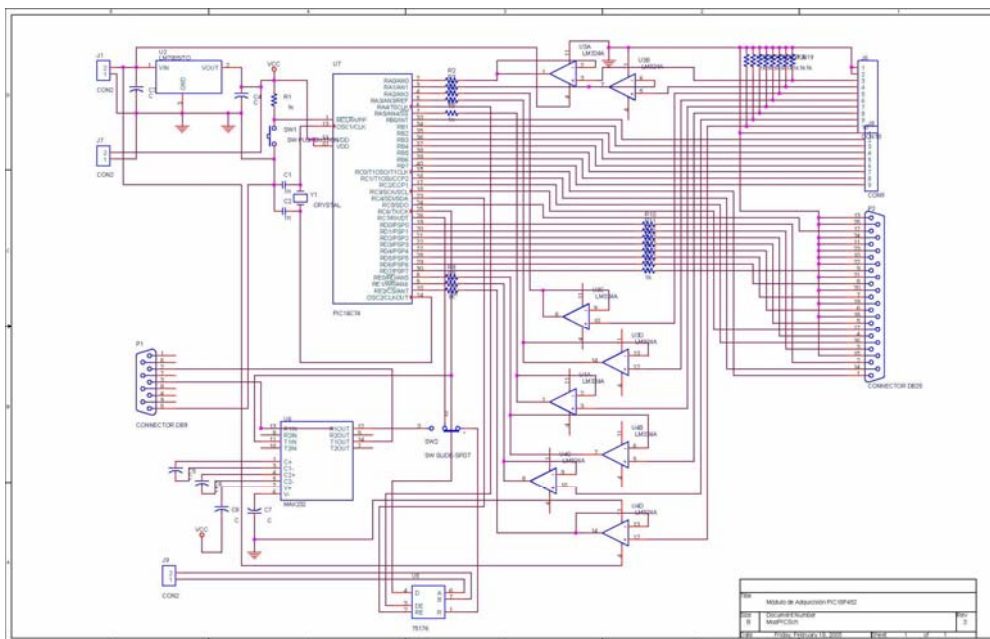


Figura 2 Diagrama esquemático de la tarjeta.

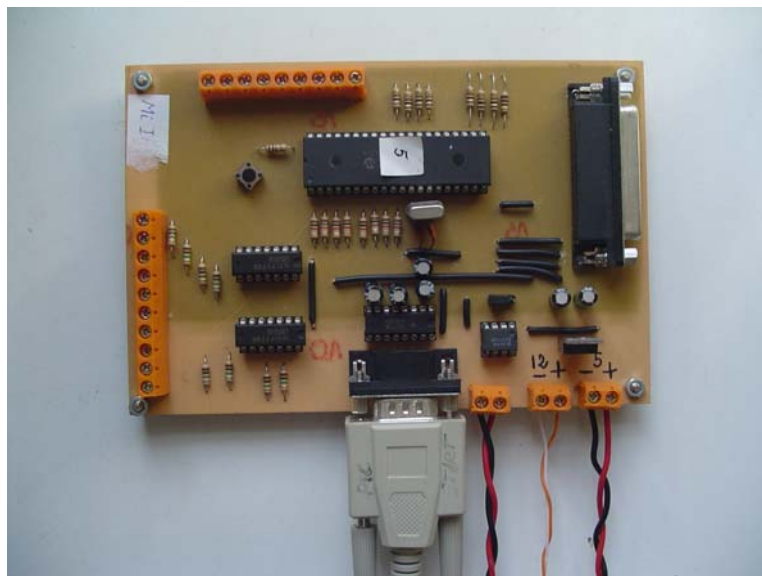


Figura 3 Foto de la tarjeta del módulo de adquisición y control

El núcleo de este módulo es un microcontrolador de nueva generación PIC18F452[11], los datos de configuración del mismo se almacenan en memoria EEPROM. Toda la configuración del módulo se realiza por software, a excepción de la selección del puerto serie RS232 ó RS485 que se realiza mediante un interruptor en hardware. La selección del puerto serie o paralelo es automática, si al momento de encender el módulo no puede detectar la conexión del puerto paralelo, conmuta a modo serie, donde permanecerá hasta un nuevo reset. Para realizar esto, el programa del modulo tiene una rutina dentro de la función de inicialización que detecta las impedancias correspondientes al puerto ECP del ordenador, si estas no están presentes configura su funcionamiento para trabajar por puerto serie, de lo contrario quedará configurado para comunicarse por el puerto paralelo

El programa se desarrolló en lenguaje C con el compilador PICC de CCS, le estructura general del programa esta basada en eventos, es decir que el módulo permanece inactivo mientras no ocurra un evento que induzca una acción, los eventos que motivan una reacción por parte del μ C son: Solicitud de un dato por comunicación, escritura de dato por comunicación y fin del periodo de muestreo del controlador, para implementar esto se utiliza el esquema de interrupciones del microcontrolador para manejar los eventos

El Servomecanismo lineal que se esta usando (Figura 4) consiste en un motor de CD de 24v, con una correa dentada de precisión que lo acopla a un sinfín de doble entrada de 20cm de longitud sobre el que esta acoplado un encoger de 48 pulsos por vuelta y el carro de desplazamiento sobre el que se observa el movimiento lineal. Además cuenta con 2 interruptores ópticos de fin de carrera.

3.2. Desarrollo de software

El software desarrollado para el laboratorio remoto consta de tres módulos independientes, uno es el software de control, que contiene el algoritmo de control y maneja las comunicaciones con el hardware, el segundo modulo es una clase de java para comunicación que abre los túneles de conexión streaming con el usuario remoto, obtiene los datos de las variables principales del software de control y las envía al usuario, así como recibe los comandos e instrucciones del usuario y las ejecuta sobre el programa de control. El tercer módulo de software es un applet de java que se ejecuta en el ordenador del usuario, contiene la interfaz gráfica con el usuario, la gestión de datos y se comunica vía streaming con el segundo módulo mencionado.

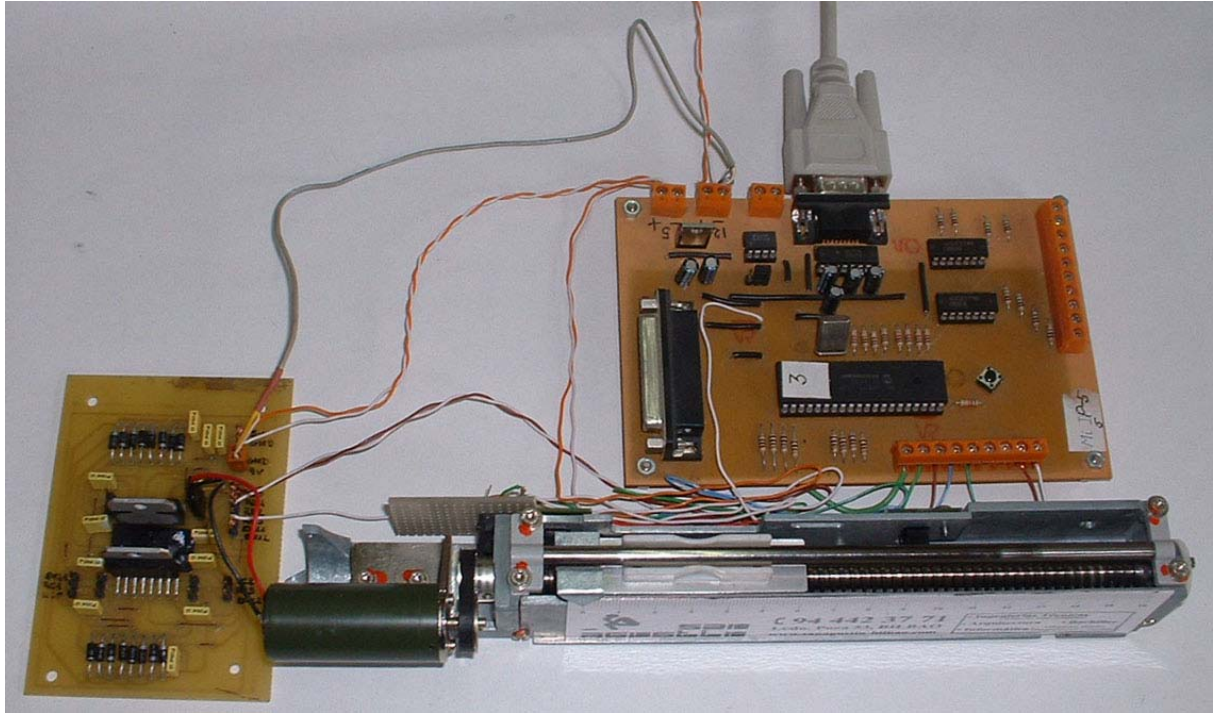


Figura 4. Servomecanismo de pruebas

El software de control se desarrolló en Matlab aprovechando las facilidades que otorga este, tanto desde el punto de vista de algoritmos de control, comunicaciones seriales para el manejo de nuestro hardware y muy especialmente aprovechando la capacidad de Matlab para cargar clases de java y ejecutar sus métodos desde el propio código de nuestro programa como si se tratase de funciones nativas de Matlab.

La comunicación de datos se realiza con los programas en java utilizan la clase SocketStream para enviar y recibir los datos de forma continua, para esto, se abre un canal de comunicación en un puerto IP diferente para cada tipo de dato que se desea enviar, de forma tal que la señal de control se envía por un puerto, la de error por otro y así sucesivamente. Cabe destacar que al utilizar streams la eficiencia de la comunicación es máxima para protocolos IP.

Se hizo una pequeña pagina web, con un único enlace que arranca la aplicación en el servidor con un comando CGI y carga en el cliente (ordenador de usuario) el applet de java que contiene la interfase de usuario, el manejo los streams de datos y realiza las gráficas.

Cabe recordar que al programar el applet en java el usuario solo necesita un navegador Web para ejecutarlo, además de que puede trabajar en cualquier plataforma de hardware y software (Windows, Linux, Unix, Os, etc. Y PC, Mac, PDA, etc.).

4.-Conclusiones

Experiencias previas [12] nos han demostrado que sistemas basados en comandos de hipertexto ofrecen una buena solución para una gran gama de tareas, su desarrollo, actualización y expansión son rápidos, pero cuando la tarea conlleva implícitamente exigencias de tiempos pequeños o grandes flujos de datos, su rendimiento decrece considerablemente. En el caso del control de un servomecanismo demostramos que el desarrollo basado en applets comunicándose mediante streams de datos posibilita un adecuado control remoto aún de este tipo de aplicaciones.

El sistema permite a los alumnos realizar prácticas de control, sintonización de un PID e identificación, desde cualquier lugar con los mismos resultados que si estuviesen físicamente en el laboratorio. Este sistema

permitirá también, la colaboración entre diversas escuelas y universidades, poniendo a disposición de los alumnos una mayor diversidad, cantidad y calidad de experimentos y prácticas de laboratorio

Es de destacar que aunque la visualización de los datos es a través de Internet y por lo tanto depende de la calidad del servicio con que contemos, el control del proceso lo realiza el servidor de manera local, por lo que en caso de no contar con la conexión adecuada a Internet o tener conexiones lentas, la visualización de los datos se vería retrasada, pero no el control del proceso

AGRADECIMIENTOS

Con el apoyo del Programa Alβan, Programa de becas de alto nivel de la Unión Europea para América Latina, beca nº E04D043264MX

Se agradece también la colaboración de la Universidad Autónoma de Querétaro.

REFERENCIAS

- [1] H. Ewald, G. F. Page. *Client-Server and Gateway Systems for Remote Control in Engineering Education*. Global J. of Engng. Educ., Vol.7, No.2 2003 UICEE
- [2] M. Cassini, D. Pattichizzo, A. Vicino. *The Automatic Control Telelab: A User-Friendly Interface for Distance Learning*. IEEE TRANSACTIONS ON EDUCATION, VOL. 46, NO. 2, MAY 2003
- [3] R. T. Abler, I. G. Wells. *Distributed Engineering Education: Evolution of the Telecollaboration Stations for Individualized Distance Learning*. IEEE TRANSACTIONS ON EDUCATION, VOL. 48, NO. 3, Ago 2005
- [4] G.W. Chang, Z.M. Yeh, H.M. Chang, S.Y. Pan. *Teaching Photonics Laboratory Using Remote-Control Web Technologies*. IEEE TRANSACTIONS ON EDUCATION, VOL. 48, NO. 4, Nov 2005
- [5] "LabView 6.0 Internet Developers" Toolkit for G (G-Server) Manual, National Instruments 2000.
- [6] K.K.Tan, T. H. Lee, F. M. Leu, "Development of a Distant Laboratory Using LabView", The International Journal of Engineering Education, Vol. 16, PP. 273-282, 2000.
- [7] V. G. Gomes, B. COI, G. W. Barton, J. A. Romagnoli, Web-Based courseware in teaching laboratory-based courses, Global Journal of Engineering Education, Vol. 4, PP. 65-72, 2000.
- [8] H. Ewald, G. F. Page, Performing Experiments by Remote Control Using the Internet, Global Journal of Engineering Education, Vol. 4, PP. 287-292, 2000.
- [9] K. Amaratunga, R. Sudarshan, A Virtual Laboratory For Real-Time Monitoring Of Civil Engineering Infrastructure, International Conf on Engineering Education August 18-21, 2002, Manchester, U.K.
- [10] C. C. Ko, Ben M. Chen, J. Chen, Yuan Zhuang, Kay Chen Tan, Development of a Web-Based Laboratory for Control Experiments on a Coupled Tank Apparatus, IEEE TRANSACTIONS ON EDUCATION, VOL. 44, NO. 1, FEBRUARY 2001
- [11] Microchip, "PIC18FXX2 Data Sheet," *Microchip Technology Inc.*, 2002.
- [12] R. Urretabizkaya, I. J. Oleagordia, P. Rodríguez. *Laboratorio Virtual Remoto Para Prácticas de Control en Tiempo Real*. Seminario Anual de Automatización Electrónica e Instrumentación, Santander E