

# PROYECTO DE LABORATORIO REMOTO DE AUTOMÁTICA: OBJETIVOS Y ARQUITECTURA PROPUESTA

Xavier Giralt Ludevid, David Jofre Hernández, Ramon Costa Castelló, Luis Basañez Villaluenga  
Institut d'Organització i Control de Sistemes Industrials (IOC)  
Escola Tècnica Superior d'Enginyeria Industrial de Barcelona (ETSEIB)  
Av. Diagonal 647,11  
08028-Barcelona  
{costa,basanez}@ioc.upc.es

## Resumen

*En este trabajo se presenta el proyecto de laboratorio remoto de automática que se ha puesto en marcha recientemente en el "Institut de Organització i Control de Sistemes Industrials". Se hace especial incapié en los objetivos del proyecto y la arquitectura que se va a utilizar.*

**Palabras Clave:** Internet, automática, Linux, RTLinux, JAVA.

## 1. INTRODUCCIÓN

Es bien conocido que dentro de la enseñanza campo de la ingeniería, y en particular de la automática, la realización de actividades prácticas que ilustren las exposiciones teóricas es de vital importancia y difícilmente sustituible por otro tipo de actividades tales como la simulación o las clases magistrales [7].

Desagraciadamente, existen diferentes factores que dificultan la realización de dichas actividades prácticas. El gran número de estudiantes y la poca capacidad de los laboratorios docentes, ya sea en medios como en espacio, hace imprescindible restringir el acceso a los laboratorios a ciertas horas preestablecidas. Unido a esto cabe destacar que el uso intensivo de los equipos requiere un esfuerzo económico importante y permanente que a veces resulta complicado de realizar. Por dicho motivo se pretende poner en marcha un sistema capaz de permitir la realización de prácticas remotas que ofrezcan una formación similar a las realizadas insitu.

Las prácticas remotas permitirán que los estudiantes tengan acceso a los laboratorios en horarios sin restricciones y un número ilimitado de veces. Dado que el acceso remoto permitirá tener una densidad de ocupación más elevada con un número de dispositivos menor, será posible usar equipos más costosos que permitan realizar prácticas más pedagógicas y atractivas para los estudiantes. Como valor añadido, la manipulación remota evitará el mal

uso de los equipos, cosa que alargará notablemente la vida útil de los mismos.

La posibilidad de realizar prácticas remotas es de gran importancia también en los cursos de formación continuada ya que introduce una mayor flexibilidad imprescindible en este tipo de estudios.

La realización de prácticas de forma remota requiere que el estudiante pueda percibir sensaciones análogas a las sensaciones que tendría en caso de hallarse en el lugar en que se está realizando el proceso físico especialmente las de visión y sonido[7]. En numerosas ocasiones estas sensaciones remotas son generadas a partir de entornos simulados [3][8], aunque idealmente estas deberían estar basadas en información sensorial obtenida del propio entorno de ejecución.

En el momento de iniciar un proyecto de esta envergadura existen diferentes cuestiones de vital importancia que deben tenerse en cuenta. Una de ellas es la definición de una arquitectura que permita satisfacer los requisitos y, a continuación, la selección de un conjunto de herramientas que permitan llevar a la práctica dicha arquitectura. En este proyecto se han asumido ciertas restricciones que marcan este proceso de diseño y selección. En primer lugar se pretende desarrollar una aplicación que por parte del usuario no implique la instalación de ningún elemento adicional a un navegador estándar, y, en segundo lugar, durante el proceso de implementación se pretende usar únicamente elementos de libre distribución.

## 2. OBJETIVOS DEL PROYECTO

Las actividades prácticas contempladas en el proyecto se enmarcan dentro de la intensificación en Automática de la ETSEIB y el programa de doctorado en Automatización Avanzada y Robótica del IOC. En ambos estudios se imparten diferentes materias relacionadas con el diseño e implementación de sistemas de control. El proyecto se centrará en dar soporte principalmente a este tipo

de asignaturas. Para ello se pretende desarrollar una herramienta que permita la experimentación de algoritmos de control digitales sobre plantas reales de forma remota.

Para simplificar el proceso de implementación y su posterior uso se han planteado dos tipos de controladores.: de estructura fija y de estructura variable.

## 2.1. CONTROLADORES DE ESTRUCTURA FIJA

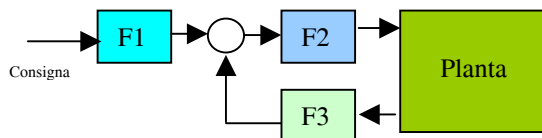


Figura 1. Estructura de un controlador de estructura fija.

De forma genérica, los sistemas de control lineales digitales [1] se pueden descomponer en esquemas de bloques como el presentado en la Figura 1, donde cada elemento  $F_i$  corresponde a un filtro lineal. La transmisión de este tipo de controladores a través de la red resulta extremadamente sencilla pues únicamente es necesario indicar el periodo de muestreo, el orden de cada uno de los filtros y sus coeficientes. En una primera fase se asumirá que se trabaja con este tipo de controladores.

## 2.2 CONTROLADORES DE ESTRUCTURA VARIABLE

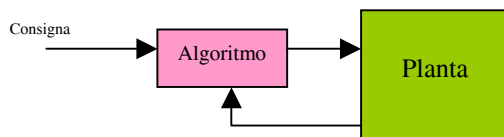


Figura 2. Estructura de un controlador de estructura variable.

Una de las cuestiones frecuentemente olvidadas dentro de los cursos de Automática es la implementación real de los controladores digitales. Aunque la aparición de sistemas microprocesadores de altas prestaciones ha simplificado mucho el proceso de implementación, se ha creído conveniente ofrecer la posibilidad de expresar los controladores como código.

Esta estructura genérica [Figura 2] permitirá también la introducción de controles con características no lineales y por tanto cualquier tipo de ley de control estudiada en los estudios anteriormente citados.

Por dicho motivo se pretende ofrecer la posibilidad de expresar los controladores como algoritmos de forma similar a la que se haría en la práctica. Este proceso es tecnológicamente más complejo que el necesario para el otro tipo de controladores, pues implica la recepción de un código fuente y su posterior compilación y ejecución. Por dicho motivo la implementación de este tipo de controladores se ha dejado para una segunda fase del proyecto.

## 3. PLANTAS

### 3.1. CRITERIOS DE SELECCIÓN

Dado que el objetivo del laboratorio remoto es el de complementar las prácticas presenciales, se ha procurado que esté formado por plantas más atractivas y de complejidad superior a las tradicionalmente disponibles en un laboratorio convencional.

### 3.2. LEVITADOR MAGNÉTICO

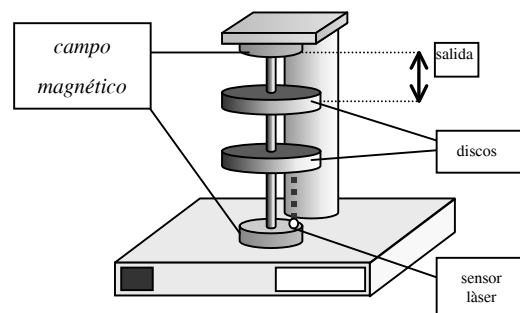


Figura 3: Estructura del levitador magnético usado.

Inicialmente se pensó en un levitador magnético como uno de los elementos que podían formar el laboratorio remoto. Este tipo de planta presenta la particularidad de que es inestable en lazo abierto, hecho poco habitual en las plantas de los laboratorios docentes. Desgraciadamente la mayoría de dispositivos de este tipo que existen en el mercado tienen una bola metálica suspendida en el aire por la acción del campo magnético generada por el electroimán. Dicha bola debe colocarse manualmente en posición al inicio del experimento. Esto dificulta el uso de estos tipos de dispositivos en laboratorios remotos. Para subsanar este problema se ha seleccionado el levitador de ECP systems (Modelo 730, <http://www.ecpsystems.com/controls/maglevit.htm>). En este dispositivo se levitan discos metálicos cuyo movimiento está restringido por un eje central, y por lo tanto no hay necesidad de intervención manual [Figura 3]. Este modelo además presenta un número de configuraciones muy superior al resto de levitadores.

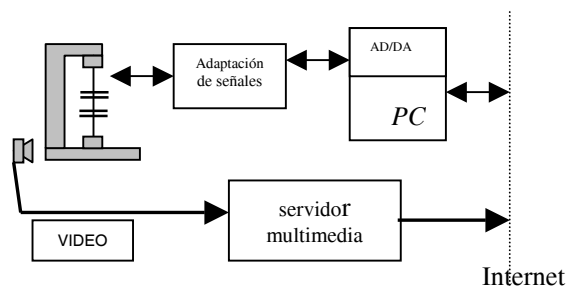


Figura 4: Componentes del sistema de levitación.

El sistema levitador estará conectado a un PC a través de una etapa de adaptación señales. Dicha etapa subministrada por ECP systems, se encarga de convertir la lectura de los sensores al rango de entrada de los conversores A/D, y dotar de potencia suficiente a las señales de control generadas por el D/A. En el PC se dispone de una tarjeta AD/DA y de conectividad a la red, lo que permite enviar la información al usuario remoto [Figura 4]. Con el fin de mejorar la realimentación sensorial del usuario remoto se introducirá un sistema de visión conectado directamente a internet (probablemente uno de los sistema comercializados por Axis, <http://www.axis.com/es/index.htm>).

### 3.3. Dirigible

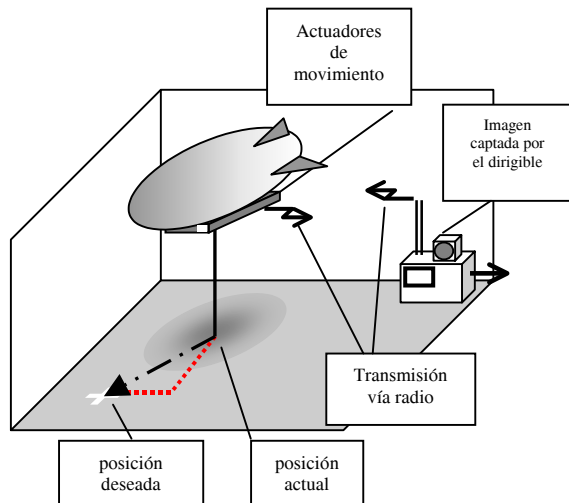


Figura 5: Estructura del dirigible.

La segunda planta que se pretende incorporar en el laboratorio remoto es un sistema formado por un dirigible controlado por radio, y equipado con cámaras de vídeo. El objetivo del control del dirigible es seguir una trayectoria dibujada en el suelo. Para realizar dicha función el dirigible dispone de dos rotores a modo de propulsores, conectados a un PC central mediante un enlace de radio. De forma similar el conjunto de cámaras que permiten observar el objetivo están conectados con el mismo PC mediante

enlaces de radio (para ello se usan los dispositivos proporcionados por Microvideo, <http://www.microlog/%7emicrovid/>) [Figura 5].

Desde el punto de vista docente esta planta presenta la característica de que resulta complejo obtener modelos de su comportamiento basados en leyes físicas. Adicionalmente incorpora como sensores cámaras de video, que requieren un procesamiento adicional.

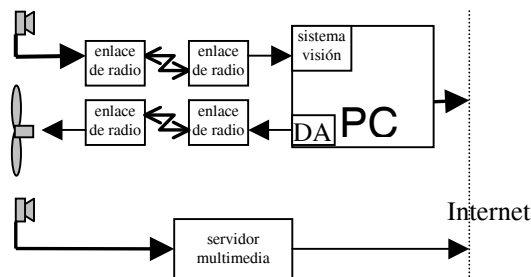


Figura 6: Componentes del sistema de posicionamiento.

Para poder llevar a la práctica el sistema se ha dotado al PC de control de un sistema de video encargado de capturar la imágenes que provienen del enlace de radio, y una tarjeta DA encargada de generar las consignas a los actuadores. Con el fin de transmitir al usuario la realimentación sensorial se instalará un sistema de cámaras conectadas a un servidor multimedia encargado de monitorizar la evolución del dirigible y su objetivo en el interior de la sala de trabajo [Figura 6].

## 4. ARQUITECTURA DEL SISTEMA

### 4.1. Estructura general

El software que se pretende desarrollar seguirá una estructura cliente-servidor. Desde este punto de vista, el usuario remoto dispondrá de una aplicación que le permitirá de forma local diseñar y validar controladores, y simultáneamente podrá solicitar la realización de ciertos experimentos. Estos se realizaran de forma remota. El dispositivo que se halla cerca de la planta se puede considerar como un servidor de experimentos.

La máquina que funcionará realizando la función de servidor de experimentos deberá ejecuta dos tipos de funciones claramente diferenciadas. Por un lado deberá gestionar todo el protocolo de red, gestión de usuarios y gestión de usuarios, y por otro, deberá realizar de interfaz con los dispositivos de bajo nivel, a la vez que ejecutar la ley de control. Estas dos funcionalidades están claramente diferenciadas y por tanto se diseñaran e implementaran por separado.

## 4.2. Sistema de Tiempo Real (TR)

Los componentes de software encargados de realizar la interfaz con los dispositivos de bajo nivel y la función de los controladores deben satisfacer especificaciones de tiempo real [2]. El cumplimiento de estas especificaciones requiere el uso de componentes que estén expresamente diseñados con esta finalidad. Dado que se pretende realizar una programación a alto nivel, se plantea el uso de un Sistema Operativo de Tiempo Real, como por ejemplo QNX [QNX] (<http://www.qnx.com>), o bien un sistema operativo que presente extensiones de tiempo real como Linux (<http://www.realtimelinux.org/>).

En el IOC se tiene experiencia en los tipos de entornos mencionados, pero para esta aplicación se ha optado por utilizar Linux i RTLinux (<http://www.fsmlabs.com/>), a fin de poder hacer uso de la gran cantidad de software de libre distribución disponible y simultáneamente tener prestaciones de tiempo real en la misma máquina.

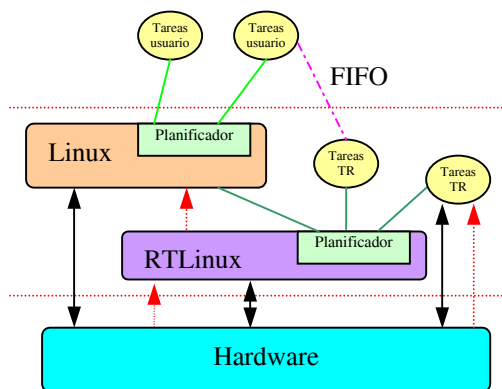


Figura 7: Arquitectura RTLinux.

RTLinux modifica el núcleo de Linux para que el sistema pueda ofrecer prestaciones de tiempo real. El resultado es un sistema compuesto por tareas de tiempo real ejecutadas a nivel de núcleo del sistema operativo. La menos prioritaria de las cuales es el propio Linux [Figura 7]. Adicionalmente Linux tiene sus propias tareas que se ejecutan a nivel de usuario. En este planteamiento cualquier programa desarrollado para Linux puede ser ejecutado en concurrencia con las tareas de tiempo real.

La programación de la tareas de tiempo real sigue los estándares POSIX [4], por lo que resulta sencillo implementar este tipo de funciones. Una funcionalidad de gran interés ofertada por RTLinux es la posibilidad de comunicar tareas de tiempo real con tareas Linux, a través de memoria compartida, aunque las últimas versiones del sistema enmascaran el procedimiento a través de unos mecanismos en

forma de colas FIFO visibles tanto desde Linux como desde RTLinux.

La implementación de la tarea de control se realizará siguiendo el esquema de la Figura 8. En ella se puede ver la existencia de un módulo de entrada/salida encargado de realizar la interfaz con los diferentes elementos hardware del sistema, una tarea de control que será una tarea periódica encargada de implementar el controlador propiamente dicho, un gestor de comandos encargado coordinar la ejecución de la tarea de tiempo real y establecer sus parámetros de ejecución, y, finalmente, una tarea adicional que actuará a modo de *watchdog*.

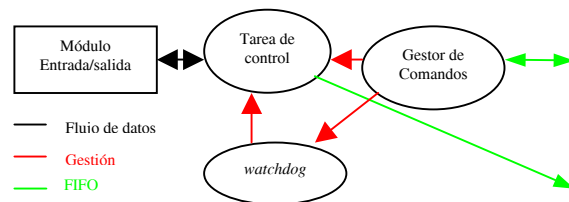


Figura 8: Arquitectura sistema de control.

RTLinux utiliza la estructura de módulos de Linux, lo que permite añadir y gestionar las tareas de tiempo real en caliente, es decir sin necesidad de reinicializar el sistema. Esta característica es de vital importancia cuando se pretende implementar algoritmos de estructura variable, pues ello implica introducir nuevo código en el núcleo del sistema.

## 4.3. Sistema de gestión de red

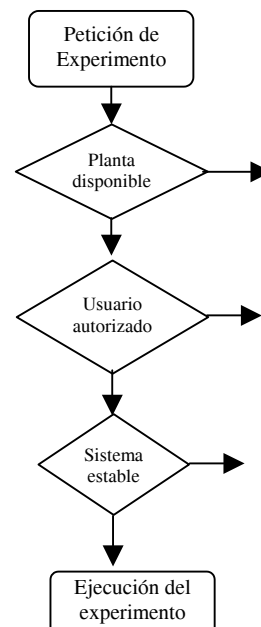


Figura 9: Esquema de acceso al sistema

Un elemento de vital importancia del sistema que se pretende desarrollar corresponde a la gestión de red. Además de ofrecer la conectividad del servidor a la

La conectividad a la red forma parte del sistema operativo que se está usando, es decir Linux y en concreto la distribución *Red Hat 7.2* (<http://www.redhat.com>). Como servidor de Web se está utilizando *Apache 1.3* (<http://www.apache.org/>), este es el servidor de WEB más popular en la actualidad, se trata de una herramienta de libre distribución disponible para casi todo tipo de plataformas, y que incorpora la mayoría de funcionalidades, ya sea a nivel de interacción con otros componentes o a nivel de seguridad.

PostgreSQL ofrece una interfaz en lenguaje SQL cosa que simplifica y estandariza los mecanismos de acceso y gestión de la información. Otra característica importante es que presenta *'bindings'* para los principales lenguajes de programación, por lo que resulta fácilmente integrable.

lo que se ha creído ideal para el tipo de procesamiento que se pretende realizar. Alternativamente, dado que el servidor de experimentos es de diseño propio se ha seleccionado CGI para enlazarlo pues se trata de un mecanismo más versátil.

```

graph TD
    subgraph Hw
        PC_LAB_Hw[PC-LAB (Hw)]
        Net_Card[Net Card]
    end

    subgraph Kernel_Linux_modules [Kernel Linux modules]
        PC_LAB_Device[PC-LAB Device]
        Interrupt_Hw_RTLinux_Kernel["[Interrupción Hw] RTLinux Kernel"]
        Tarea_RTLinux((Tarea RTLinux))
        Net_Linux_O_S["[Net] Linux O.S."]
    end

    subgraph Server
        Tarea_FIFO((Tarea))
        Simulacion[Simulación y validación de los controladores]
        CGI[CGI]
        HTTP_Server[HTTP Server]
        Base_datos_SQL[(Base de datos SQL)]
        Usuarios_Experimentos[Usuarios - Experimentos]
    end

    PC_LAB_Hw --> PC_LAB_Device
    PC_LAB_Hw --> Interrupt_Hw_RTLinux_Kernel
    Net_Card --> Net_Linux_O_S

    PC_LAB_Device --> Interrupt_Hw_RTLinux_Kernel
    Interrupt_Hw_RTLinux_Kernel --> Tarea_RTLinux
    Interrupt_Hw_RTLinux_Kernel --> Net_Linux_O_S

    Tarea_RTLinux --> Tarea_FIFO
    Net_Linux_O_S --> Tarea_FIFO

    Tarea_FIFO -- FIFO --> Simulacion
    Tarea_FIFO --> CGI
    Tarea_FIFO --> HTTP_Server

    CGI --> HTTP_Server
    HTTP_Server --> Base_datos_SQL
    HTTP_Server --> Web
    Base_datos_SQL -- PHP --> HTTP_Server
    Base_datos_SQL --> Usuarios_Experimentos
  
```

En la Figura 10 se presenta un esquema de bloques con todos los elementos software descritos anteriormente, se puede observar que aunque existen elementos de naturaleza muy dispar se ha obtenido una arquitectura compacta y simple. En la actualidad esta arquitectura está siendo puesta a punto y sometida a diferentes tipos de experimentos, que permitirán validar su correcto funcionamiento.

Una cuestión de vital importancia en el desarrollo de el proyecto es la interfaz de usuario, en gran medida el éxito de la plataforma dependerá de la capacidad de diseñar e implementar una interfaz de flexible y amigable que facilite el uso de la herramienta y que además resulte fácil de mantener y ampliar.

La interfaz que se pretende ofrecer al usuario constará de dos partes, una primera de documentación donde se ofrecerá al estudiante una documentación de recordatorio de la teoría expuesta en las sesiones presenciales, y una segunda en la que se ofrecerán las herramientas necesarias para el desarrollo e implementación de controladores.

La primera parte correspondiente a la documentación se está desarrollando mediante la generación de páginas HTML, y el uso de herramientas interactivas como FLASH 5 (<http://www.macromedia.com/>).

La segunda parte, encargada de interactuar con el servidor de experimentos, y ofrecer la realimentación sensorial necesaria se encuentra actualmente en proceso de diseño e integración de los diferentes componentes. Inicialmente, esta parte se pretende desarrollar mediante software *ad hoc* desarrollado en JAVA. En paralelo se está estudiando la utilización de VRML para ofrecer una realimentación mediante realidad aumentada en los casos en los que el ancho de banda disponible no sea suficiente para ofrecer una realimentación basada en imágenes reales.

## 5. CONCLUSIONES

En este trabajo se han presentado los objetivos del proyecto que actualmente se está iniciando en el IOC.

Se ha presentado también la primera tentativa de arquitectura del sistema. La arquitectura propuesta presenta la ventaja de ser compacta, y basada totalmente en software libre. Además de estas características de primordial interés es capaz de satisfacer adecuadamente características tan dispares como la programación en tiempo real, la gestión de datos y la comunicación a través de Internet.

## Agradecimientos

Este trabajo ha sido parcialmente financiado por los proyectos TAP99-0839 y AFC-2001-0388-LD.

## Referencias

- [1] Åström, K. J. and Wittenmark B. (1997). *Computer-controlled Systems*. Prentice-Hall, ISBN: 0-13-314899-8.
- [2] Burns, A. Wellings, A. (2001). *Real Time Systems and Programming Languages* Ada 95, Real-Time Java and Real-Time POSIX. ISBN: 0-201-72988-1. Addison Wesley Longmain.
- [3] Chong, N.Y.; Kotoku, T.; Ohba, K.; Komoriya, K.; Matsuhira, N.; Tanie, K (2000) Remote coordinated controls in multiple telerobot. Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on , Volume: 4 , 2000 Page(s): 3138 -3143 vol.4.
- [4] Bill Gallmeister (1995). *Posix.4*. O'Reilly, 1995. ISBN 1-56592-074-0.
- [5] Jeffay, K.; Hudson, T.; Parris, M. (2001). Beyond audio and video: multimedia networking support for distributed, immersive virtual environments. Euromicro Conference, 2001. Proceedings. 27th , 2001 Page(s): 300 - 307
- [6] QNX Software Systems Ltd (1997), *QNX Operating System, System Architecture*.
- [7] Sánchez, J., Morilla F., Dormido S.. ``Teleoperation of an inverted pendulum through the world wide web''. WORKSHOP ON INTERNET BASED CONTROL EDUCATION.IBCE'01. Madrid, Spain. December 12-14, 2001
- [8] Serrano C., Costa R., y Basañez L. "VRML-JAVA Aplicaciones a la programación fuera de línea de robots industriales". 6to Congreso de la Asociación Española de Robótica y Automatización Tecnologías de la producción. Barcelona, 20-22 Octubre de 1999