

LABORATORIO VIRTUAL MULTIPLATAFORMA PARA EDUCACIÓN EN INGENIERÍA

*Rosa M^a Fernández-Cantí; José Antonio Lázaro-Villa;
Aitor Villar-Zafra; Sergio Zarza-Sánchez*
Departamento de Teoría de la Señal y Comunicaciones
Universidad Politécnica de Cataluña
Campus Nord UPC, Edificio D4, 08034 Barcelona
rfernandez@tsc.upc.edu, jose.lazaro@tsc.upc.edu, aitor.villar@gmail.com,
szarza@tsc.upc.edu

RESUMEN

Se presenta un laboratorio virtual autónomo y multiplataforma para docencia. La plataforma implementada incluye un servidor con conexión SSH (Secure SHell) y un repositorio separado que contiene los experimentos virtuales. La programación de los experimentos se ha realizado en Java usando la herramienta EJS (*Easy Java Simulation*) para entornos gráficos y el programa Matlab como motor de cálculo externo. El laboratorio virtual contiene experimentos de control a nivel universitario. Se presentan dos ejemplos de aplicación, en concreto, un levitador magnético y un sistema de péndulo invertido con carrito. El laboratorio virtual ha sido probado con éxito en la formación de estudiantes de Ingeniería Electrónica. Se presentan y comentan los resultados de esta experiencia de *e-learning*.

ABSTRACT

An autonomous and multiplatform virtual laboratory for educational purposes is presented. The implemented platform includes a server with a SSH (Secure SHell) connection and a separated repository containing the virtual experiments. The programming of the experiments is implemented in Java language developed by means the EJS (*Easy Java Simulation*) tool and uses the Matlab language as an external computation engine. The virtual laboratory provides control experiments at University level. Two application examples are described, namely, a magnetic levitator and an inverted pendulum-cart system. The virtual laboratory has successfully been used for education and training of Electronics Engineering students. A discussion of the results of this e-learning experience is also presented.

PALABRAS CLAVE

Código abierto, laboratorio virtual, herramienta educativa multiplataforma, software autónomo, experimentos de control

KEYWORDS

Open source, Virtual laboratory, Multiplatform educational tool, Autonomous software, Control experiments

1. INTRODUCCIÓN

Los laboratorios no presenciales constituyen una herramienta útil en enseñanza/aprendizaje de las diversas disciplinas de la Ingeniería. Las ventajas y los inconvenientes de esta herramienta educativa se han estudiado especialmente en el marco de los sistemas de control y la educación en automática en general. Véase, por ejemplo, [1-4] y las referencias allí citadas.

El interés particular de los laboratorios no presenciales en la docencia de la ingeniería de control se justifica por el elevado coste económico de la compra y mantenimiento de los equipos de prácticas. Además, para dar un servicio adecuado a la educación pública, es necesario adquirir varias unidades del *hardware*, lo que aumenta aún más la inversión económica.

Otro inconveniente, al realizar experimentos de control con equipo físicos, es el tiempo necesario para llevar a cabo las prácticas, ya que muchos estudiantes pueden pasar demasiado tiempo haciendo cálculos y/o probando los equipos para familiarizarse con ellos.

Hay diversas alternativas para solucionar estos dos problemas. Una primera solución es operar laboratorios remotos [5], por ejemplo mediante el uso del *software* propietario Labview. Los laboratorios remotos implican una inversión relativamente baja en *hardware*, ya que sólo se requieren una o dos unidades del equipo experimental. Esta solución necesita la programación del *front-end* y el desarrollo de una aplicación web para acceder al laboratorio a distancia. Sin embargo, en este enfoque, el equipo físico tiene que estar encendido durante largos periodos de tiempo, quizás incluyendo noches y días festivos. Por ello, para asegurar la disponibilidad y correcto funcionamiento de los equipos es necesario realizar tareas intensivas de mantenimiento.

Una segunda solución es el desarrollo de laboratorios virtuales donde se utiliza un entorno gráfico para emular los experimentos que se realizarán con el *hardware* físico. Esta es una buena opción, incluso en los casos en que el *hardware* ya está adquirido, ya que, de esta manera, los estudiantes pueden familiarizarse con el equipo antes de comenzar el experimento presencial, lo que ahorra mucho tiempo en el laboratorio [6]. Por otro lado, este enfoque es especialmente interesante en los casos en que el equipo experimental no está disponible, ya que entonces el profesor tiene la libertad de desarrollar prácticas que impliquen desde problemas básicos de *benchmarking* hasta experimentos complejos que requerirían equipos demasiado caros como para poder disponer de ellos [7].

Este trabajo se centra en el desarrollo de laboratorios virtuales. En primer lugar, una cuestión importante es que el laboratorio virtual debe estar disponible en Internet para todos los estudiantes en cualquier momento. Para lograr este objetivo, el laboratorio debe ser *multiplataforma* ya que los estudiantes pueden utilizar diferentes entornos de trabajo, tales como Linux, Macintosh, u otros. El laboratorio aquí presentado consigue

cumplir esta especificación gracias al uso del lenguaje de programación Java, con ello se logra flexibilidad y la capacidad de integración en cualquier plataforma web.

Otro requisito es la *autonomía*, en el sentido de que el laboratorio virtual no debe depender de un único campus virtual. Así, el laboratorio virtual puede ser accedido desde otras universidades, cada una con su propio campus virtual. Esto se ha conseguido mediante el uso de servidores separados, uno para el repositorio de experimentos y otro para el campus virtual.

Finalmente, otra especificación es la *modularidad*, en el sentido de que la aplicación debe permitir la comunicación entre las diferentes partes del laboratorio virtual, pero a su vez, estas partes deben ser completamente independientes entre sí.

Esta comunicación se organiza de la siguiente manera. En el apartado 2 se definen las fases que intervienen en el laboratorio virtual multiplataforma. En el apartado 3 se enumeran las especificaciones del sistema. En el apartado 4 se describen los diferentes bloques que implementan el laboratorio virtual. El apartado 5 presenta dos de los experimentos de control disponibles. El apartado 6 resume los resultados obtenidos con los estudiantes. Finalmente, en el apartado 7 se presentan las conclusiones de esta experiencia y se apuntan algunas líneas futuras de trabajo.

2. FASES DEL LABORATORIO VIRTUAL

La Figura 1 muestra las diferentes fases del laboratorio virtual multiplataforma. Estas son las siguientes:

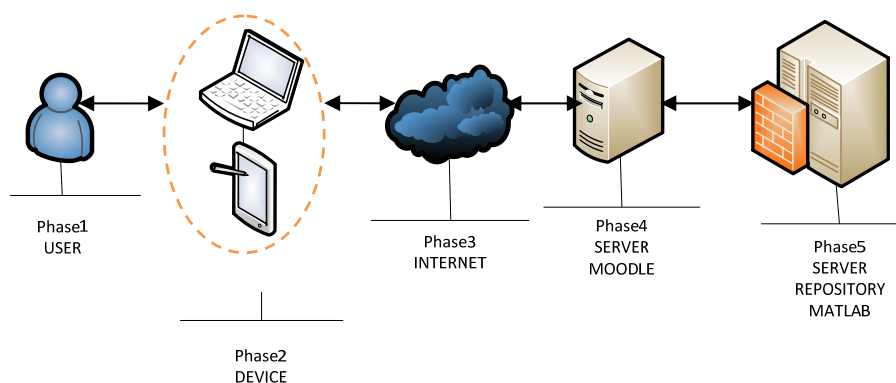


Figura 1. Fases del laboratorio virtual multiplataforma

Fase 1: El usuario

Las ventajas para el usuario incluyen la disponibilidad de los laboratorios las 24 horas del día los 365 días del año para acceder a las prácticas de laboratorio. Además, no hay ninguna limitación en la repetitividad de los experimentos (a menos que el profesor decida lo contrario).

Fase 2: Dispositivo

Dado que el laboratorio está programado en Java, los estudiantes pueden acceder a él desde cualquier terminal en cualquier lugar siempre que el navegador de Internet tenga un cliente Java, que es el caso más común.

Fase 3: Internet

Hoy en día, es posible garantizar el servicio ya que es posible acceder a la nube desde casi cualquier sitio.

Fase 4: Servidor Moodle

El laboratorio virtual presentado en este documento está disponible en el Moodle iLabVir de la UPC (ver Figura 2), pero está diseñado para permitir el acceso al servidor de repositorio desde cualquier otro campus virtual de otra Universidad, siempre y cuando se instale un módulo específico desarrollado por el grupo "GilabVir" del RIMA. El servidor de Moodle es el que permite a los estudiantes acceder a los experimentos.

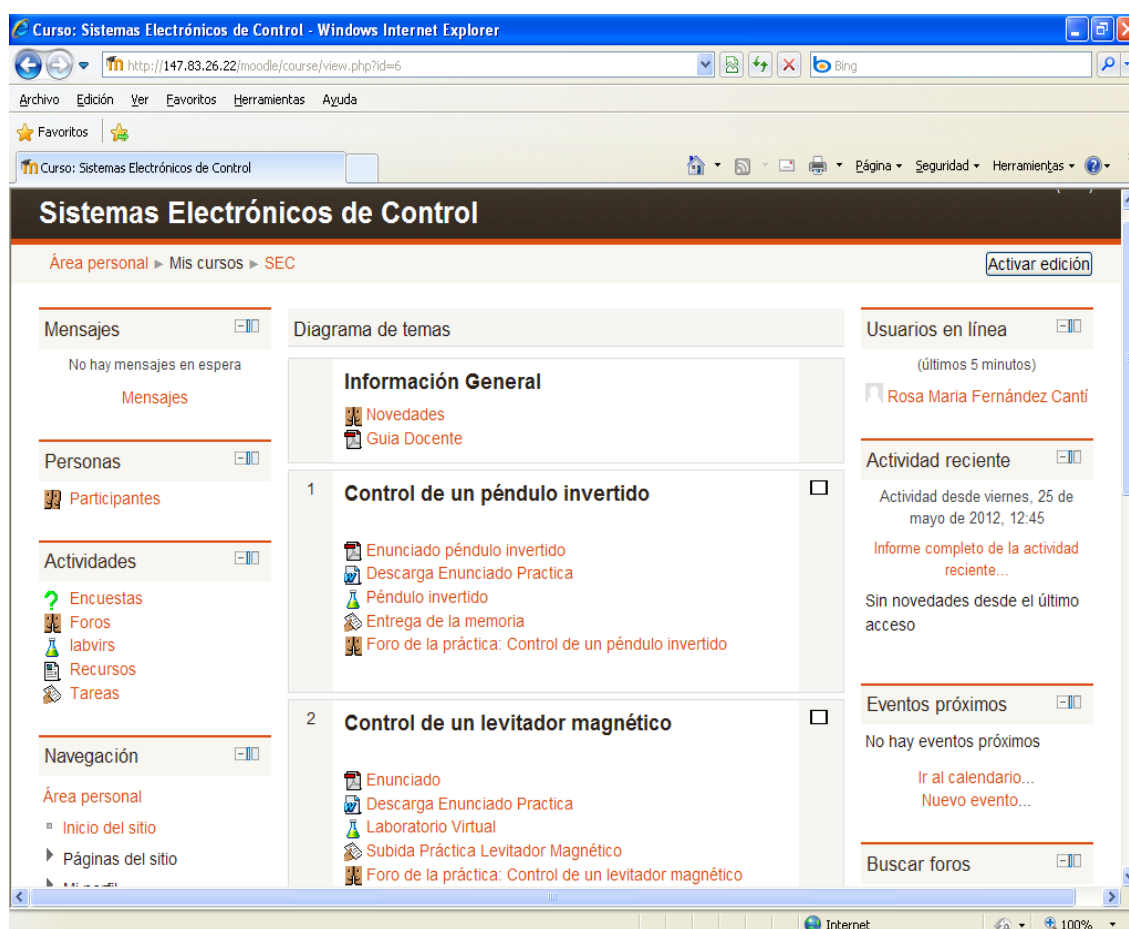


Figura 2. Moodle iLabVir

Fase 5: Servidor del repositorio y motor de cálculo

El servidor de repositorio está separado del servidor de Moodle con el fin de dar servicio a más de un campus virtual. Esto hace posible tener un motor de cálculo en la misma máquina donde está el servidor del repositorio. Nuestro motor de cálculo es Matlab, pero se puede utilizar cualquier otro programa, incluyendo los de *software* libre, por ejemplo, Octave. Por último, se ha instalado un cortafuegos entre el servidor de repositorio y el servidor de Moodle a fin de tener un control de los diferentes campus virtuales que deseen acceder a los experimentos.

3. Especificaciones del sistema

Desde el punto de vista de los usuarios de laboratorio virtuales, los principales requisitos incluyen aspectos tales como la disponibilidad, la facilidad de uso, y la repetitividad de los experimentos. Pero desde el punto de vista del diseñador las especificaciones de mayor interés son la autonomía, multiplataforma, y la modularidad.

Autonomía

La implementación de un sistema autónomo significa que el laboratorio virtual debe ser totalmente independiente del campus virtual. De esta manera, los experimentos son accesibles desde cualquier otra universidad o institución educativa. O bien, el sistema puede ser desarrollado como un recurso abierto para la educación (*Open Educational Resource*, OER) a fin de poder comparar los resultados entre diferentes grupos de trabajo. Además, el requisito de la autonomía mejora la seguridad del sistema, que es un problema importante cuando se accede desde diferentes campus.

Multiplataforma

En cuanto a la especificación multiplataforma, el primer paso es desarrollar la aplicación en un lenguaje de programación de tal manera que cualquier navegador de Internet pueda llevar a cabo el experimento sin problemas. De esta manera cualquier usuario con cualquier conexión a Internet puede acceder a su campus virtual particular y ejecutar el experimento. Un lenguaje que satisface plenamente estos requisitos es Java.

Modularidad

Por último, la especificación de modularidad conduce a la implementación de un sistema distribuido formado por diversos bloques separados entre sí. De esta manera nuevas actualizaciones o futuras modificaciones no implicarán cambios en todos los bloques.

4. Arquitectura del laboratorio virtual

En esta sección se describen los diferentes bloques que constituyen nuestro laboratorio virtual y se explica cómo se lleva a cabo la comunicación entre las diferentes partes.

La Figura 3 muestra los diferentes bloques constitutivos. Estos son los siguientes:

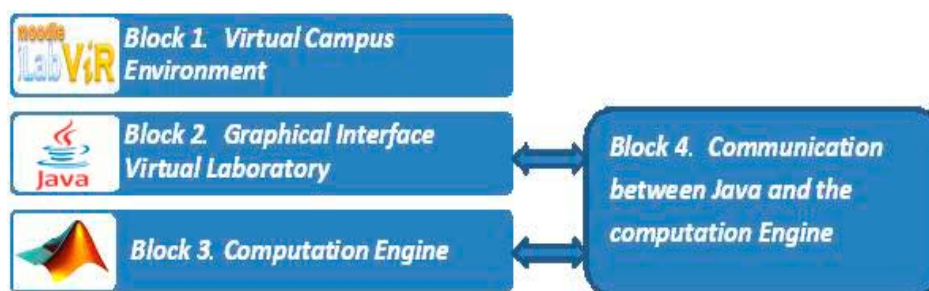


Figura 3. Diagrama de bloques del laboratorio virtual

Bloque 1. Entorno del campus virtual

El laboratorio virtual presentado en este documento se ha instalado en el Moodle v2.0 de la ETSETB (*Escola Tècnica d'Enginyers de Telecomunicació de Barcelona*). Como se mencionó en la sección anterior, para operar los experimentos virtuales es necesario instalar un módulo que ha sido desarrollado por el grupo "GiLabVir" del proyecto RIMA. Este módulo es de código abierto y está disponible en [8].

La característica principal de este módulo es que permite tener un repositorio de experimentos virtuales separado del servidor Moodle. De esta manera no es necesario instalar los experimentos en el propio campus virtual lo que permite que diferentes campus virtuales puedan tener acceso a los experimentos. Los experimentos se compilan en Java e incrustan en HTML.

También cabe destacar que el repositorio está en una máquina virtual Xen, en el que también se instala el *software* que se utiliza como motor de cálculo (Matlab, en nuestro caso). El motor de cálculo es el responsable de la mayor parte del consumo computacional. Puesto que se instala en el mismo repositorio que los experimentos, ello significa que los experimentos no necesitan consumir los recursos del servidor de Campus virtual.

En cuanto a las cuestiones de seguridad, la comunicación entre el servidor de Moodle y el servidor del repositorio se establece mediante la configuración de los puertos de comunicación de ambos y las IPs del cortafuegos que hay entre ellos dos. Esto

implica que los estudiantes no tienen acceso directo al repositorio de los experimentos. La única entidad que puede acceder al repositorio es el servidor de Moodle, identificada por medio de su IP. Así, Moodle simplemente desempeña el papel de marco para el experimento pero el experimento no se ejecuta en él. Por medio de las herramientas de edición del campus virtual Moodle, el profesor puede establecer qué experimentos estarán a disposición de los estudiantes y cuáles no.

Bloque 2. Interfaz gráfica del laboratorio virtual

Nuestra aplicación está desarrollada en lenguaje Java. En particular, la interfaz gráfica de usuario (*Graphical User Interface*, GUI) de los experimentos se ha programado utilizando el paquete *Easy Java Simulation* (EJS) [9-10]. Se trata de una herramienta de código abierto especialmente diseñada para el desarrollo de aplicaciones Java que requieren de un entorno gráfico para desarrollar la simulación. Como resultado del trabajo con EJS, se obtiene un archivo *.jar. Este archivo es la compilación del proyecto de EJS y puede ser incrustado en una página HTML, obteniendo así la estructura típica de un Applet de Java.

Bloque 3. Motor de cálculo

El uso de un motor de cálculo externo u otro *software* matemático resulta muy interesante dependiendo de la complejidad del experimento que queremos virtualizar. Puesto que los experimentos virtualizados corresponden a sistemas de control automático, se han usado las funciones de la *Control Systems Toolbox* de Matlab [11]. Puesto que el motor de cálculo y el repositorio están en la misma máquina (véase Figura 1), no existe la necesidad de utilizar la red Ethernet para comunicar la parte de Java con la parte de Matlab. Esto reduce considerablemente el tiempo transcurrido entre la llamada al experimento desde el ordenador del estudiante y la ejecución del experimento. De todas maneras, si fuera necesario, también se podrían tener el motor y el repositorio en máquinas diferentes.

Por último, y siguiendo el requisito de la modularidad, EJS se utiliza sólo para el desarrollo del *front-end* gráfico (incluyendo los efectos de animación), mientras que el archivo-m de Matlab es el responsable de generar las diferentes señales que intervienen en los experimentos. Por lo tanto, el archivo-m contiene las ecuaciones dinámicas que describen el comportamiento del sistema, controlado o no, y la pantalla de Java es la que permite introducir los parámetros del experimento (señal de excitación, la configuración del controlador y sus coeficientes, las condiciones iniciales) y presentar los resultados (efectos de animación, gráficos de la respuesta).

Esta disposición facilita el desarrollo de nuevas prácticas y experimentos, ya que, si queremos cambiar el comportamiento del experimento o generar nuevas prácticas, sólo tenemos que modificar los archivos-m. Destacar, una vez más, que en lugar de Matlab, es posible utilizar cualquier otro motor de cálculo, incluyendo los de *software* libre como, por ejemplo, Octave y su *Control Suite*.

Bloque 4. La comunicación entre Java y el motor de cálculo

Un aspecto importante del laboratorio virtual es cómo comunicar Java con el motor de cálculo de una manera sencilla y no costosa (a pesar de que el motor de cálculo es propietario y ofrece una solución comercial para ello). En nuestro caso hemos resuelto este problema mediante el establecimiento de la comunicación entre Java y el motor de cálculo a través de SSH (Secure SHell). SSH es un protocolo de red para la comunicación de datos entre dos entidades que garantiza que el intercambio de información es completamente seguro. Para llevar a cabo la conexión SSH se utiliza la biblioteca de código abierto de Java JSch [12]. El uso de JSch proporciona los objetos y clases de Java necesarios para implementar una conexión SSH en el lenguaje Java. Esta biblioteca se incluye en el archivo *.jar de la compilación. De esta manera la conexión entre el repositorio y el motor de cálculo es completamente segura.

Por último, si el Applet de Java utiliza un recurso externo (por ejemplo, una conexión a una base de datos o a un disco duro), es necesario firmar digitalmente el archivo *.jar.

Flujo de datos

La Figura 4 muestra los diferentes tipos de flujo de datos entre las diversas entidades que constituyen nuestro laboratorio virtual.

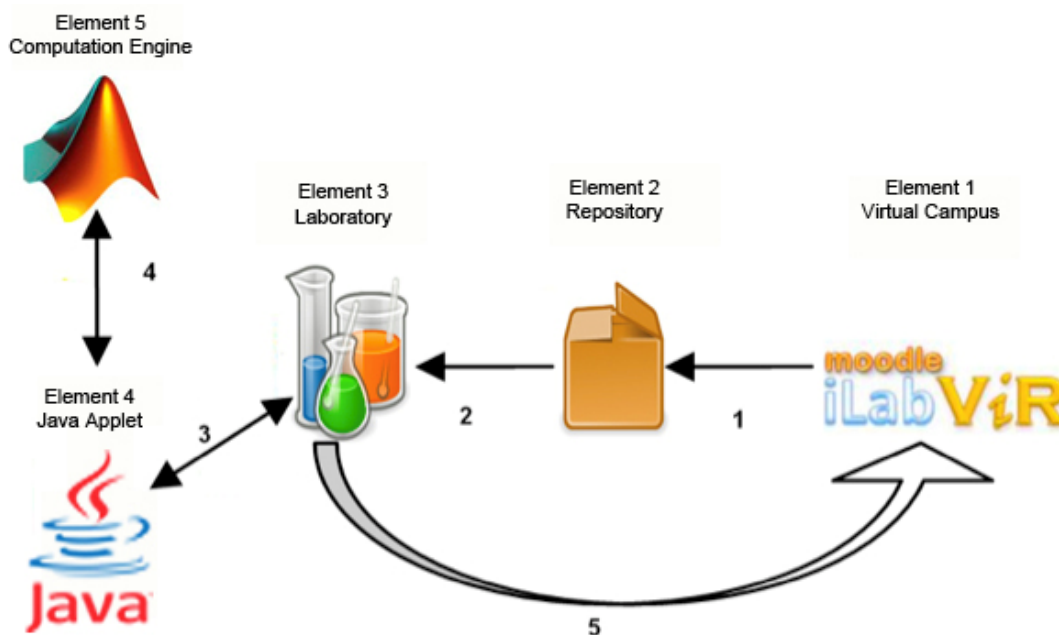


Figura 4. Flujo de datos en el laboratorio virtual

La flecha marcada como "1" representa el flujo de datos entre el campus virtual y el repositorio. Es unidireccional, es decir, los datos se transmiten únicamente del primer al segundo elemento. La información transmitida está relacionada con la conexión del usuario a la lista de experimentos almacenados en el repositorio.

La segunda flecha, "2", representa el flujo de datos entre el repositorio y el laboratorio virtual. De nuevo, es unidireccional, es decir, los datos se transmiten únicamente del segundo al tercer elemento. Estos datos corresponden a la elección del experimento que se desea ejecutar.

La flecha marcada como "3" representa el flujo de datos entre el laboratorio y el Applet de Java. Los datos se transmiten entre el bloque tercero y cuarto bidireccionalmente. Cuando la información viaja desde el laboratorio al Applet de Java, el flujo de datos contiene los parámetros necesarios para iniciar la simulación. En sentido contrario, cuando la información viaja desde el Applet de Java hacia el laboratorio, los datos contienen la información relativa a la simulación gráfica de los resultados del experimento.

La cuarta flecha, "4", representa el flujo de datos entre el Applet de Java y el motor de cálculo y es bidireccional, los datos viajan entre el tercer elemento y el cuarto (Applet de Java). Se trata de datos numéricos que contienen tanto los parámetros específicos de la simulación como los valores, muestra a muestra, de las señales implicadas en los experimentos.

Por último, el la flecha, "5", representa la transmisión de los datos obtenidos en la simulación en el Applet de Java y el Campus Virtual, que es lo que se muestra a los estudiantes.

5. Experimentos de control

Para ilustrar los experimentos disponibles en este laboratorio virtual, considerar los *layouts* mostrados en la Figura 6 y la Figura 7.

Levitador magnético

El primer ejemplo es un levitador magnético virtual. En este caso, los experimentos virtuales emulan a un equipo docente real, el levitador magnético (MagLev) de Feedback Instruments Ltd., modelo 33-210 [13], ver Figura 5. Los parámetros físicos en el modelo virtual se corresponden con los ajustes reales del MagLev. Además, para asegurar que el comportamiento del levitador virtual coincide con el del real, diversos efectos no lineales se han incluido en la programación del archivo-m.

La levitación magnética es un proceso muy interesante en la enseñanza de la ingeniería de control, ya que se trata de un proceso inestable y no lineal, mientras que el modelo analítico que describe su comportamiento dinámico es relativamente fácil

de obtener. Por esta razón, este problema puede ser dirigido tanto a estudiantes universitarios como a investigadores en el campo del control.



Figura 5. Levitador magnético de Feedback Instruments Ltd.

En el caso del levitador magnético se pueden realizar diferentes experimentos y, en todos ellos, el alumno puede ver los movimientos de la bola en tiempo real. El objetivo del sistema de control consiste en mantener la bola a una distancia determinada del electroimán. Dicha distancia puede ser fija o seguir una referencia del tipo señal cuadrada o sinusoidal. Si el control falla la bola o bien cae al suelo o bien se queda pegada al imán.

Los estudiantes tienen que seleccionar una estrategia de control y diseñar el controlador, es decir, calcular sus parámetros. Los controladores pueden ser fácilmente probados y las estudiantes pueden ver si sus diseños son correctos o no, pudiéndolos mejorar si es necesario.

Los principales conceptos ilustrados con el experimento del levitador magnético incluyen la regulación PID (Proporcional-Integral-Derivativa), el diseño mediante el Lugar Geométrico de las Raíces de Evans, y el diseño de controladores óptimos ITAE (*Integral Time Absolute Error*).

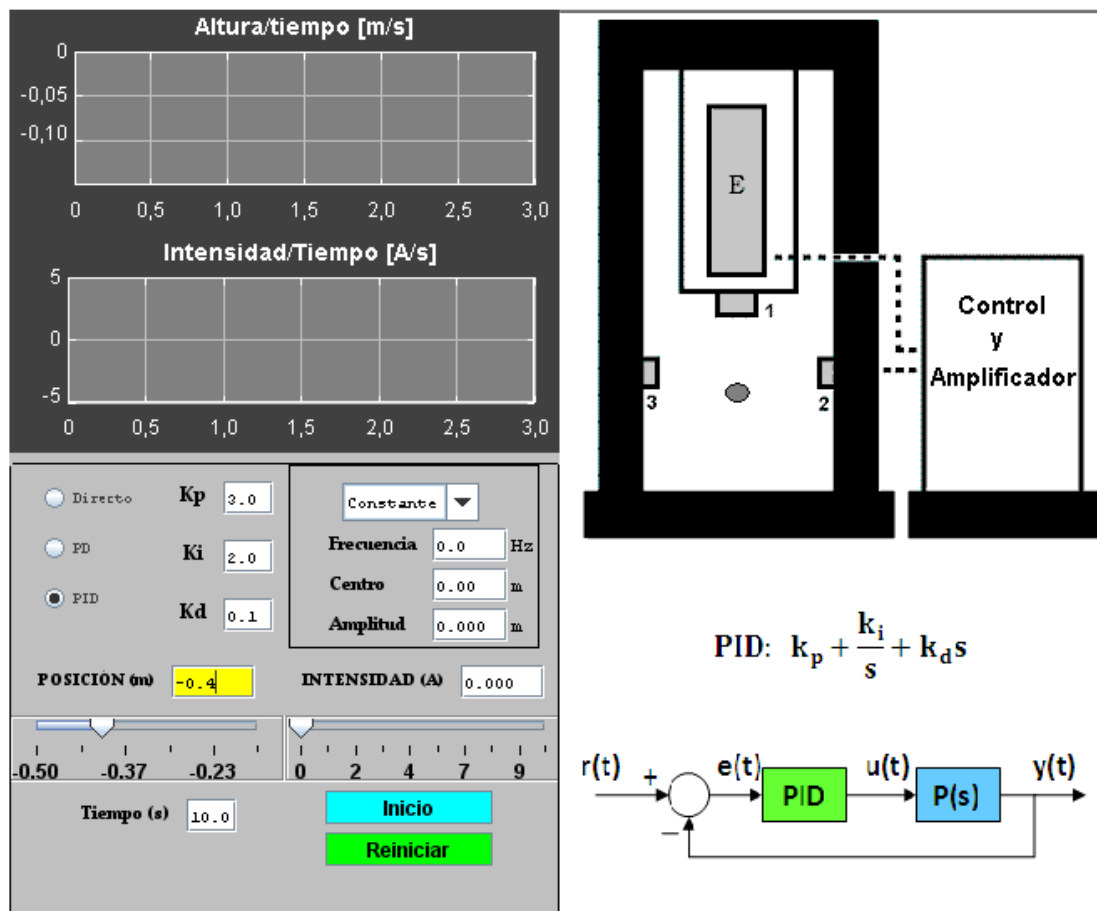


Figura 6. Levitador magnético

Sistema péndulo invertido-carrito

El sistema péndulo invertido-carrito que se muestra en la Figura 7 constituye otro experimento clásico en la docencia del control. En nuestro caso, no disponemos de ningún equipo de prácticas correspondiente a este experimento. Por esta razón, la selección de los parámetros de configuración y control se ajustó a fin de ilustrar mejor los conceptos del curso.

Una vez más, los estudiantes e investigadores pueden seleccionar distintas estrategias de control (controlador serie, control en el espacio de estados), ajustar los parámetros del controlador y probar el diseño. En el caso de los estudiantes, se plantean diferentes niveles de dificultad en función del límite de tiempo para estabilizar la planta. De esta manera la selección de polos rápidos y los compromisos de diseño se convierten en aspectos importantes del problema. Como en el caso del levitador magnético, el alumno puede visualizar el comportamiento dinámico tanto en forma de resultados gráficos como con los movimientos en tiempo real del péndulo y el carro.

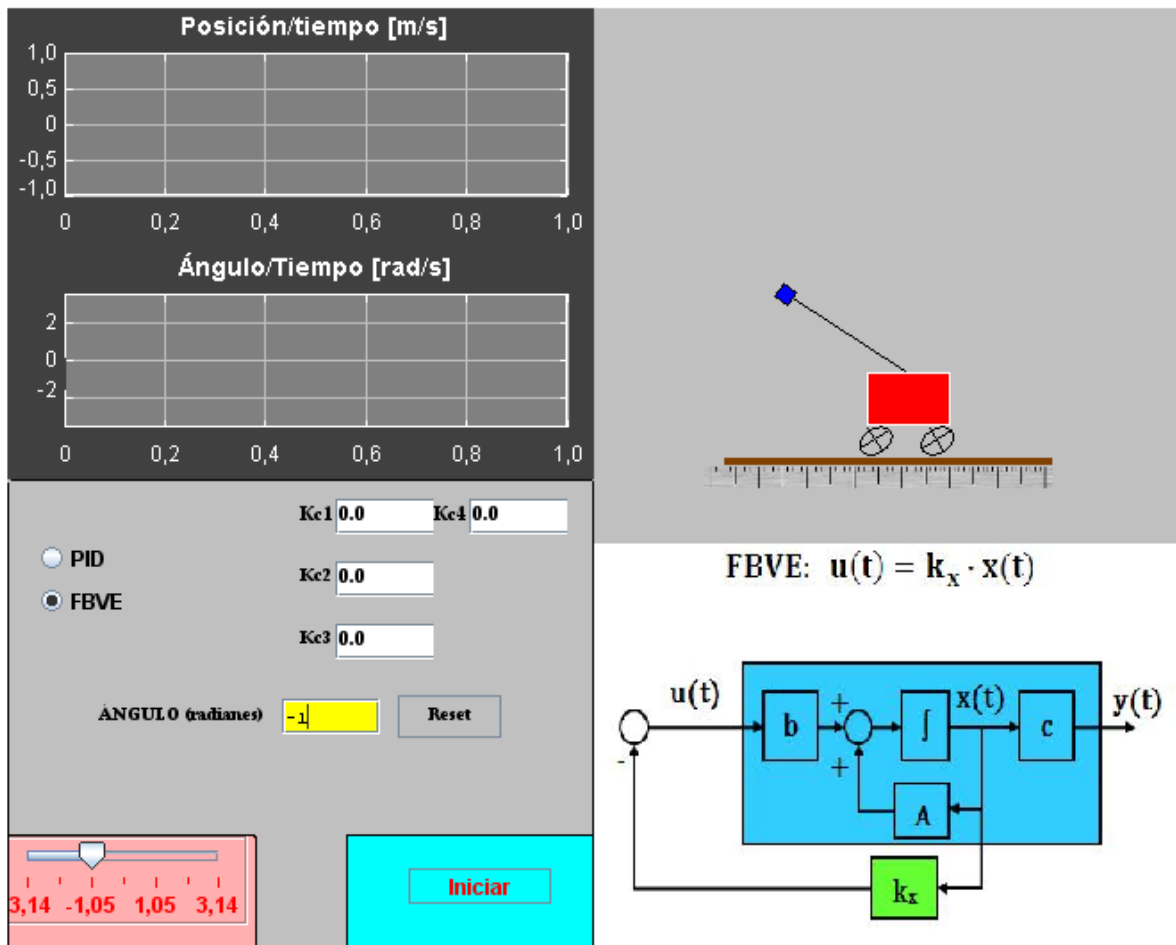


Figura 7. Sistema péndulo invertido-carrito

Todos los experimentos de control permiten a los estudiantes descargar los archivos con los resultados numéricos para su posterior análisis. Además, se pueden capturar los gráficos y grabar vídeos (formato GIF animado) del comportamiento del sistema.

6. Resultados

Los dos experimentos de la sección anterior han sido probados por diferentes grupos de estudiantes de la asignatura *Sistemas Electrónicos de Control* de la titulación de Ingeniería Electrónica de la ETSETB durante los cursos académicos 2010/2011 y 2011/2012. En total se han matriculado 28 estudiantes.

A fin de extraer conclusiones sobre el funcionamiento y utilidad del laboratorio virtual, se diseñó una encuesta de satisfacción a contestar por los estudiantes una vez realizados los experimentos. En general, los estudiantes están muy satisfechos con el laboratorio virtual. Valoran muy positivamente la posibilidad de realizar prácticas a distancia, en horario flexible y el efecto “realista” de las animaciones, que les resulta sumamente gratificante cuando consiguen diseñar un controlador correcto. Aun así,

siguen demandando la posibilidad de realizar prácticas con equipos reales y tener al profesor cerca para hacerle preguntas.

Desde el punto de vista de utilidad docente, a la pregunta de si el laboratorio virtual les ha servido para consolidar/aclarar conceptos, la gran mayoría de estudiantes se han mostrado de acuerdo o muy de acuerdo. Este dato es consistente con los resultados obtenidos en los trabajos de fin de curso, siendo significativamente de mayor calidad los trabajos de los dos cursos en que el laboratorio virtual ha estado activo.

7. Conclusiones

En este trabajo se ha presentado un laboratorio virtual autónomo, multiplataforma y modular. El repositorio de experimentos y el motor de cálculo se encuentran en un servidor diferente del servidor del campus virtual. Ello aumenta la velocidad de respuesta, aumenta la seguridad del conjunto, reduce el uso de los recursos de los servidores y permite el acceso a diferentes campus virtuales o, lo que es lo mismo, permite a diferentes instituciones educativas acceder a nuestros experimentos para su explotación.

La implementación presentada aquí se ha realizado con ayuda de la herramienta EJS y el programa Matlab, comunicados por medio de SSH. Pero si se desea realizar el diseño usando íntegramente *software* libre, es posible sustituir el Matlab por un paquete equivalente *open source* como, por ejemplo, Octave. La modularidad de la plataforma facilita el rápido desarrollo de nuevas prácticas y aplicaciones, incluyendo experimentos de otras asignaturas.

La buena aceptación del laboratorio virtual, tanto por parte de los estudiantes como de los profesores, nos indica que se trata de una herramienta que puede tener un gran futuro en la Universidad, por su atractivo y utilidad. Por ello, seguiremos trabajando en este sentido ampliando las posibilidades de la pantalla de visualización, virtualizando aplicaciones más complejas, editando manuales más completos sobre el entorno y los experimentos y promoviendo el uso del foro para las consultas a los profesores y el trabajo en equipo de los estudiantes.

Agradecimientos

Los autores agradecen al equipo de Sistemas Informáticos del Departamento de Teoría de la Señal y Comunicaciones el intercambio de ideas y su apoyo en el desarrollo de este proyecto. Al Institut de Ciències de l'Educació (ICE) y a l'Escola Tècnica Superior d'Enginyeria de Telecomunicació de Barcelona (ETSETB) de la Universitat Politècnica de Catalunya (UPC) por su apoyo a esta iniciativa. Y, finalmente, a los miembros del Grup d'interés en Laboratoris Virtuals i remots (GiLabVir) del proyecto de Recerca i Innovació en Metodologies d'Aprenentatge (RIMA) por sus fructíferos debates.

Referencias

- [1] S. Dormido, "Control Learning: Present and Future" *Annual Reviews in Control*, 28(1): 115-136, 2004
- [2] J.L. Guzman, M. Berenguel, F. Rodríguez , *Laboratorio virtual y remoto para el control de clima en una maqueta de un invernadero*, Universidad de Almería, <http://greenhouse.ual.es>
- [3] F. A. Candelas, J. Sánchez, "Recursos didácticos basados en Internet para el apoyo a la enseñanza de materias del área de ingeniería de sistemas y automática" *Revista Iberoamericana de Automática e Informática Industrial*, 2(2): 93–101, 2005.
- [4] A. Cardoso, M. Vieira, and P. Gil, "A Remote and Virtual Lab with Experiments for Secondary Education, Engineering and Lifelong Learning Courses", *iJOE*, 8(2):49-54, 2012.
- [5] I. Calvo, E. Zulueta, F. Oterino, J.M. Lopez-Guede , " A Remote Laboratory for a Basic Course on Control Engineering", 2009 [doi:10.3991/ijoe.v5i3.925](https://doi.org/10.3991/ijoe.v5i3.925)
- [6] <http://www.aurova.ua.es/robolab/index.html>
- [7] A. García, N. Duro, R. Dormido and S. Dormido "The Reaction Wheel Pendulum: An Interactive Virtual Laboratory for Control Education" , 2010. [doi:10.3991/ijoe.v6i3.1348](https://doi.org/10.3991/ijoe.v6i3.1348)
- [8] <http://moodlelabs.svn.sourceforge.net/>
- [9] M. Domínguez, "Simulaciones interactivas en EASY JAVA de equipo Feedback MS-150", *XXXI Jornadas de Automática*, 2010.
- [10] F. Esquembre, "Easy Java Simulations: a software tool to create scientific simulations in Java", *Computer Physics Communications*, vol. 156, issue 2, pp. 199-204, 2004. [http://dx.doi.org/10.1016/S0010-4655\(03\)00440-5](http://dx.doi.org/10.1016/S0010-4655(03)00440-5)
- [11] The Mathworks, *Control Systems Toolbox for Matlab. User's Guide*, 2011
- [12] <http://www.jcraft.com/jsch/>
- [13] Feedback Instruments Ltd., *Magnetic Levitation Installation & Comissioning*, 33-9421C, 2006.