

Estrategias de Enseñanza a Distancia sobre Control No-Lineal Aplicada al Péndulo Invertido

Sergio García-Nieto*, Miguel Martínez, Ángel Llosá y Javier Sanchis

Universidad Politécnica de Valencia, Departamento de Ingeniería de Sistemas y Automática,
Camino de Vera N°14, 46022 Valencia-España
(e-mail: [sergarro, mmiranzo, anllogui, jsanchis]@isa.upv.es)

*autor a quien debe ser dirigida la correspondencia

Resumen

Este artículo describe el desarrollo de una herramienta software/hardware destinada al estudio, en el ámbito de la enseñanza universitaria, de sistemas de control distribuidos en tiempo real. El sistema se empleará en el control de procesos no lineales. Esta herramienta presenta tres partes claramente diferenciadas, las cuales pueden ser analizadas y modificadas por los estudiantes: i) se introduce el modelo físico del péndulo invertido; ii) el controlador es implementado en un sistema distribuido de tiempo real; y iii) todo el sistema es manejado mediante una aplicación desarrollada en JAVA. Por lo tanto, los estudiantes tienen todos los elementos necesarios para trabajar con sistemas complejos no lineales, como el problema del péndulo invertido que se discute en este trabajo. Las pruebas realizadas muestran que la plataforma proporciona un método rápido y seguro de experimentación para los estudiantes.

Palabras clave: cursos a distancia, estrategias de enseñanza, control de procesos, péndulo invertido

Strategies for Distance Learning Courses on Nonlinear Control Applied to the Inverted Pendulum

Abstract

This paper presents a software/hardware tool to study, at university level, distributed control systems in real time. This platform is applied to control nonlinear processes. The system has three different parts which students can evaluate and modify; i) a virtual model represents the physic model of the inverted pendulum; the controller is implemented by a real time distributed control system; and iii) the whole system is managed with a JAVA application. Therefore, students have all the necessary elements to practice with complex nonlinear systems, such as the inverted pendulum discussed in this work. The tests done show that the platform represents a fast and reliable method for experimentation by the students.

Keywords: distance learning, teaching strategies, process control, inverted pendulum

INTRODUCCIÓN

Los conceptos teóricos asociados al campo de control resultan, normalmente, complejos y difíciles de asimilar por parte de los estudiantes. En cambio, la aplicación práctica de la teoría es una forma muy eficaz de asimilar conocimientos (Duan et al., 2005). Por tanto, resulta imprescindible el desarrollo de plataformas software/hardware destinadas a la implementación práctica de las técnicas de control planteadas de forma teórica (Sánchez et al., 2002).

El presente artículo introduce una nueva estrategia educacional, para el estudio de sistemas de tiempo real y control no lineal vía Internet (Benmohamed et al., 2005). Esta nueva estrategia se basa en la inclusión de elementos reales dentro de los modelos de simulación (componentes de comunicación, actuadores y sensores reales, etc.). Esta propuesta pone de manifiesto las diferencias existentes entre los procesos reales y las clásicas simulaciones, así como la problemática asociada al empleo de componentes reales. Los estudiantes pueden comprobar el comportamiento real de un sistema de control complejo. Además, pueden profundizar y afianzar sus conocimientos en la teoría de control no lineal, tal y como se introduce en Apkarian (1999). Por otra parte, el desarrollo de prototipos donde comprobar las diferentes metodologías de control resulta complejo y costoso a nivel económico.

Por tanto, con el objeto de hacer frente a las cuestiones ya mencionadas, este trabajo propone sustituir los procesos reales por *modelos virtuales*, compuestos por un simulador que se ejecuta en un sistema de tiempo real (Sánchez et al., 2004). El *modelo virtual* cuenta con un conjunto de tarjetas de adquisición que permiten la interacción con el entorno. Por tanto, el sistema de control interactúa con el *modelo virtual* como si éste fuese el proceso real. Adicionalmente, se plantea el uso de un bus de campo como sistema de comunicación, con el objetivo de ilustrar la influencia de este tipo de componentes dentro del sistema de control. Consecuentemente, uno de los elementos del bus de campo es un controlador basado en una arquitectura PC. Este controlador emplea los sensores y actuadores distribuidos para interactuar con el modelo virtual. Por último, se ha desarrollado una aplicación que actúa como interfaz entre el usuario y la plataforma, lo cual permite gestionar todo el sistema vía Internet (Schilling et al., 2001). En resumen, la plataforma desarrollada consta de los cuatro elementos principales:

- xPC *modelo virtual*.
- I/O distribuidas basadas en CAN bus.
- Controlador PC basado en RTLinux.
- SCADA desarrollado en JAVA.

El xPC es una potente herramienta incluida en el conocido software de computación científica Matlab. Esta herramienta permite la ejecución de diagramas *Simulink* en tiempo real. Gracias al xPC, el proceso real se sustituye por un *modelo virtual* descrito por un modelo Simulink. En particular, se ha implementado un modelo no lineal basado en las ecuaciones fundamentales del péndulo invertido (Aracil y Gordillo, 2004; 2005).

Por otro lado, las I/O distribuidas se encuentran dispuestas en un bus de campo basado en CAN bus con protocolo de comunicaciones CANOpen. Donde uno de los nodos es un PC estándar que hace las funciones de controlador. El cual, interactúa con el *modelo virtual* mediante las I/O distribuidas en el CAN bus. Gracias al empleo de RTLinux como SO en el controlador, se asegura la ejecución de distintas tareas en tiempo real.

Finalmente, el SCADA desarrollado es un interfaz de usuario desarrollado mediante el lenguaje de programación JAVA. Esta aplicación puede gestionar todas las opciones de configuración del controlador PC, así como visualizar la actividad del sistema (variables manipuladas, acciones de control, etc.). El SCADA ha sido diseñado para establecer una conexión remota con la plataforma a través de Internet (Schilling et al., 2001). El diagrama que se muestra en la figura 1 describe la interconexión entre los diferentes componentes de la plataforma (García-Nieto et al., 2005).

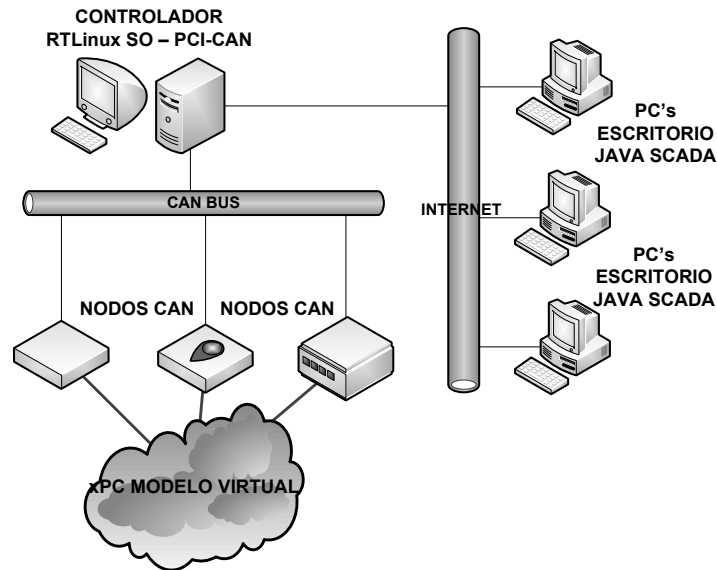


Fig. 1: Diagrama funcional del sistema

EL PÉNDULO INVERTIDO: DESARROLLO TEÓRICO

El péndulo invertido en sus dos variantes: péndulo invertido plano (Aracil y Gordillo, 2004; 2005) y péndulo invertido giratorio (conocido éste último como péndulo de *Furuta*) (Furuta et al., 1992), ejemplo de proceso subactuado, ha sido objeto de estudio e implementación en muchas universidades del mundo como banco de pruebas de diferentes metodologías de control. Las razones del interés en este problema podrían asociarse a la inherente dificultad de controlar un proceso tan no lineal como éste, lo que constituye un reto importante para la comunidad científica. Pero al mismo tiempo, por la facilidad de formulación de las ecuaciones físicas que ligan el comportamiento del sistema, lo que permite disponer de un modelo de una precisión aceptable.

Normalmente, el modelado es el primer paso para el diseño de un controlador, por lo que se va a proceder a establecer los modelos matemáticos para cada uno de los péndulos. Para dicha modelización se suelen emplear las ecuaciones de Euler-Lagrange que proporcionan dos ecuaciones diferenciales ordinarias de segundo orden no lineales, que se transforman posteriormente en ecuaciones de estado. El cálculo se inicia mediante la formulación del *Lagrangiano* que se define como la diferencia entre la energía cinética y la potencial del sistema, y la aplicación de las ecuaciones de Euler-Lagrange en notación vectorial a este *Lagrangiano*:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}} \right) - \frac{\partial L}{\partial q} = f \quad (1)$$

donde $L(q(t), \dot{q}(t)) = E_{cin} - E_{pot}$, q es el vector de variables generalizadas y f es el vector de fuerzas generalizadas aplicadas externamente. Puesto que la energía cinética responde a una estructura dada por: $E_{cin} = \frac{1}{2} \dot{q}^T M(q) \dot{q}$ con $M(q)$ matriz cuadrada, es fácil demostrar que:

$$\ddot{q} = M(q)^{-1} \left[- \left(\frac{d}{dt} M(q) \right) \dot{q} - \frac{\partial (E_{pot} - E_{cin})}{\partial q} + f \right] \quad (2)$$

Péndulo invertido plano

En la figura 2 se presenta un esquema del péndulo plano, en este caso resultan las variables generalizadas descritas por la ecuación (3).

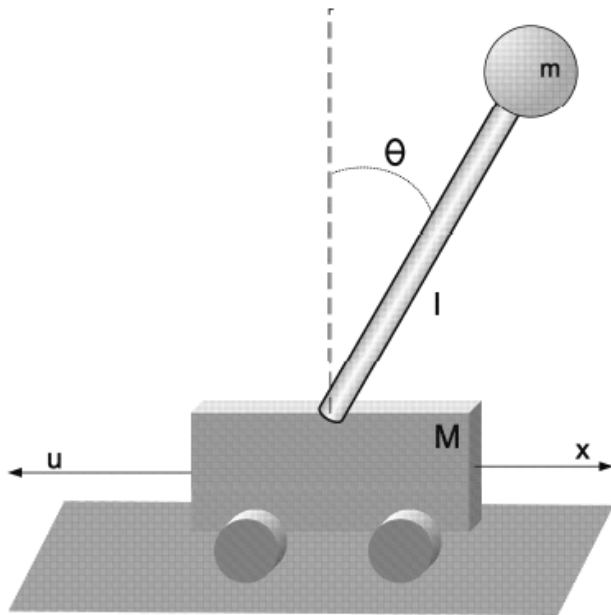


Fig. 2: Péndulo plano.

$$q = \begin{bmatrix} \theta \\ x \end{bmatrix}; \quad \dot{q} = \begin{bmatrix} \dot{\theta} \\ \dot{x} \end{bmatrix} \quad (3)$$

$$\ddot{\theta} = \frac{1}{\Delta} [\gamma w \sin \theta - \beta^2 \dot{\theta}^2 \sin \theta \cos \theta - \beta \cos \theta u] \quad (4)$$

$$\ddot{x} = \frac{1}{\Delta} [-\beta w \sin \theta \cos \theta + J \beta \sin \theta \dot{\theta}^2 + J u] \quad (5)$$

donde:

$$\Delta = J\gamma - \beta^2 \cos^2 \theta$$

$$w = mgl$$

$$\gamma = M + m$$

$$\beta = ml$$

$$J = ml^2$$

Las expresiones (4) y (5) representan el conjunto de ecuaciones no lineales de segundo grado que describen el comportamiento del sistema. Aplicando la simplificación $M \gg m$ en la ecuación del carrito (variable x) y adaptando a un sistema en variables de estado definido como $\theta = x_1$; $\dot{\theta} = x_2$; $x = x_3$, u fuerza aplicada al carrito, resulta:

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = \frac{1}{J} [mg l \sin x_1 - ml \cos x_1 \frac{u}{M+m}] \quad (6)$$

$$\dot{x}_3 = \frac{u}{M+m}$$

Con el fin de simplificar en los cálculos de los apartados posteriores se asumirá que todos los coeficientes de las ecuaciones anteriores son unitarios.

Péndulo invertido giratorio

En la figura 3 se presenta un esquema del péndulo giratorio, donde el parámetro I representa el momento de inercia de la base giratoria. Por otro lado, la ecuación (7) muestra las variables generalizadas del sistema. Aplicando una deducción similar al caso anterior y suponiendo $I \gg ml^2$ y $I \gg mr^2$ para la ecuación de la base giratoria (variable ψ) y adaptando a un sistema en variables de estado definido como $\theta = x_1$; $\dot{\theta} = x_2$; $\psi = x_3$, M_t par aplicado a la base giratoria se obtiene la ecuación (8). Por otro lado, como en la subsección anterior, con el fin de simplificar en los cálculos de los apartados posteriores se asumirá que todos los coeficientes de las ecuaciones anteriores son unitarios. Es importante destacar que, en esta situación, resulta un sistema idéntico a (6) correspondiente al del péndulo plano. Donde se substituye la posición del carro x por el ángulo de la base giratoria ψ , y la fuerza u aplicada al carro por el momento M_t aplicado a la base giratoria del péndulo giratorio. Por tanto, formalmente y con la debida interpretación, es posible representar ambos problemas con el mismo conjunto de ecuaciones.

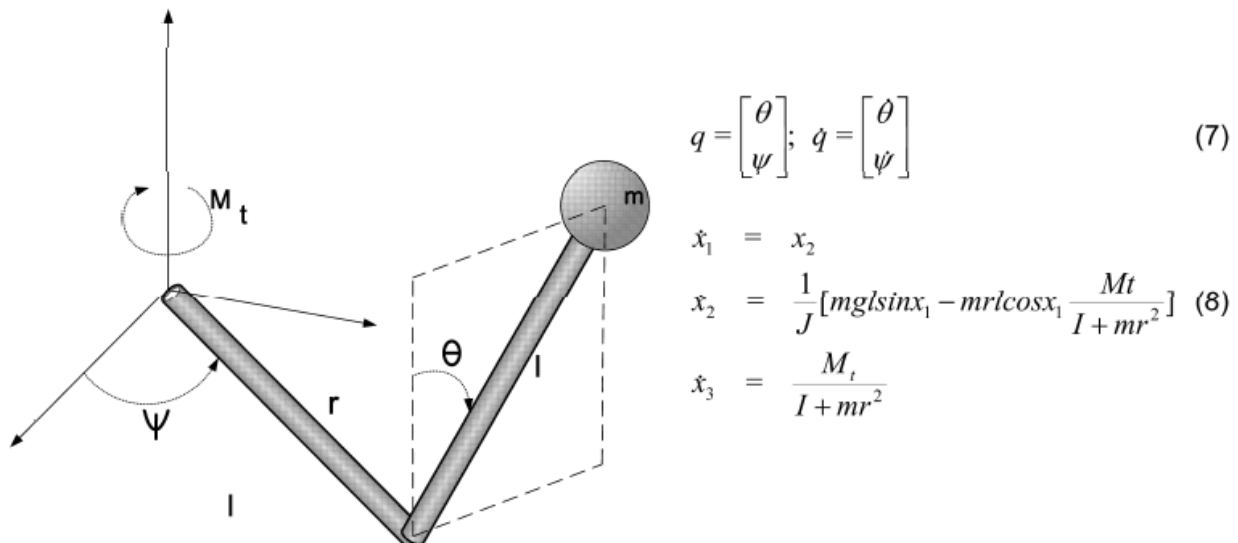


Fig. 3: Péndulo giratorio.

xPC COMO MODELO VIRTUAL

El software xPC de Mathworks (The MathWorks, Inc., 2002) es una potente herramienta en el prototipado rápido de modelos virtuales en tiempo real. El sistema xPC es un procesador de señales digitales (DSP) empleando el entorno Matlab. Esto permite desarrollar código DSP a partir de un diagrama de bloques de Simulink, ejecutándose en un PC equipado con tarjetas de adquisición en tiempo real. xPC emplea una estructura Host-Target, donde el host es un PC que ejecuta Matlab/Simulink, y el target es un PC ejecutando un SO con kernel de tiempo real. El PC host compila el código DSP a partir del diagrama Simulink y lo carga en el PC target via TCP/IP. En conclusión, se deben realizar cuatro pasos a la hora de implementar un sistema mediante el toolbox xPC (figura 4):

1. Construcción del modelo Simulink mediante estructuras básicas.
2. Compilación del modelo mediante el compilador de Matlab.
3. Transferencia del código compilado al PC target.
4. Inicialización del sistema PC target.

El diagrama Simulink de la figura 5 muestra el modelo desarrollado péndulo invertido.

BUS DE CAMPO

Los buses de campo son un elemento cada vez más frecuente en proyectos de automatización y control distribuido. La industria considera que los buses de campo son un perfecto sustitutivo al cableado punto a punto convencional, ya que reducen costes, espacio y tiempo, redundando en un ahorro significativo. Frente a los primeros sistemas propietarios, se están consolidando los buses de campo basados en estándares abiertos. La principal ventaja de dichas soluciones es que amplían el abanico de suministradores de equipos compatibles, facilitando también el mantenimiento de las instalaciones y su crecimiento. Actualmente la mayoría de fabricantes incorporan en sus equipos interfaces con uno o más estándares abiertos. En particular, se ha empleado un bus de campo basado en CAN y con protocolo de comunicaciones CANOpen.

A continuación se describen los módulos I/O empleados en el bus:

- 2 BK5120 BECKHOFF Industrie Elektronik GMBH CAN modules.
- 1 CAN-CBM-AI410 Esd electronic system design GMBH CAN module.
- 1 CANbloc-Mini Esd electronic system design GMBH CAN module.

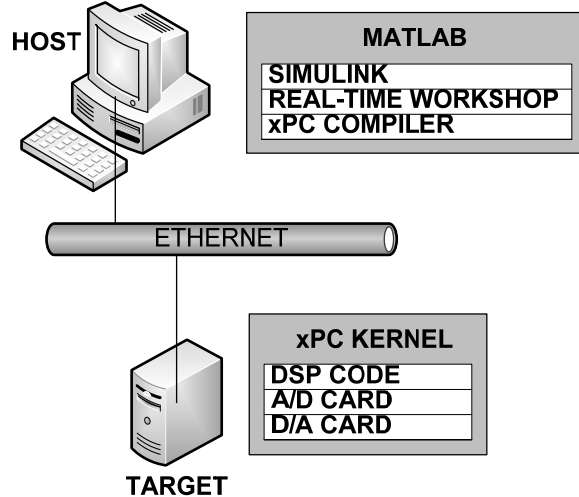


Fig. 4: Implementación xPC

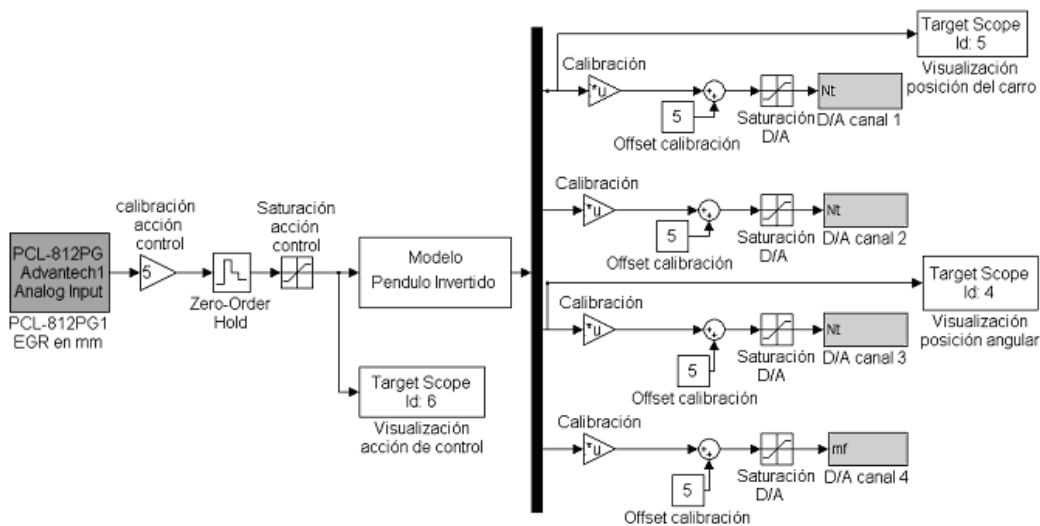


Fig. 5: Modelo Simulink

CAN Bus

CAN (Controller Area Network) es un bus de campo, desarrollado originalmente para aplicaciones en la industria de automoción. El protocolo CAN fue estandarizado en 1993 como ISO 11898-1. CAN define la capa de datos y parte del nivel físico del modelo OSI de siete capas. En cuanto al nivel físico CAN incorpora un estándar ISO: "physical signaling", que incluye bit encoding y decoding (NRZ), bit timing y sincronización. Además, existen diferentes protocolos basados en CAN que comprenden las capas OSI superiores, como CANOpen (Boterenbrood, 2000). Desde la perspectiva del usuario, CAN proporciona dos servicios de comunicación: el envío de mensajes y la petición de mensajes. Además incluye otros servicios de bajo nivel y transparentes al usuario como procesamiento de errores y retransmisión automática.

Algunas de las principales funciones que proporciona CAN son las siguientes:

- Comunicación broadcast. Un emisor de información transmite a todos los nodos a través del bus. Todos los nodos reciben el mensaje y deciden si es relevante o no para ellos. Este mecanismo garantiza la integridad de la información de todos los nodos.
- Priorización de mensajes. El protocolo proporciona mecanismos hardware que garantizan que no se produzcan colisiones entre mensajes cuando varios nodos desean emitir mensajes.

- Mecanismo sofisticado de detección de errores y retransmisión de mensajes fallidos.
- Estructura multi-maestro, que permite construir redes con tolerancia a fallo (redundantes).

Las principales aplicaciones de CAN incluyen: coches de pasajeros, camiones y autobuses, electrónica marítima, aérea y aeroespacial, automatización de factorías, control de maquinaria industrial, ascensores y montacargas, equipamiento medico, equipamiento no industrial, etcétera.

Si se desea más información sobre CAN, puede consultarse la Web de CiA (CAN in Automation), la organización internacional de usuarios y fabricantes que desarrolla y da soporte a los protocolos de alto nivel basados en CAN (American CANopen Networks, 2005).

CANOpen Protocol

CANopen es un protocolo basado en CAN que implementa la capa de aplicación. Actualmente esta ampliamente extendido, y ha sido adoptado como un estándar internacional. Las especificaciones de CANopen cubren la capa de aplicación y los perfiles de comunicación (CiA DS 301) así como una estructura para dispositivos programables (CiA 302), recomendaciones para cableado y conectores (CiA 303-1) y sobre el empleo de unidades del sistema internacional y representación mediante prefijos (CiA 303-2). Aunque a continuación se describen algunas de las características fundamentales del protocolo, es evidente que una descripción exhaustiva del mismo requiere mayor extensión. Además de la documentación de la CiA.

Los principales elementos que el protocolo define son los siguientes:

- Capa de aplicación (application layer)
- Perfil de comunicación (communication profile)
- Diccionario de objetos (OD, Object Dictionary)
- Servicios de gestión de red (NMT)
- Perfiles de dispositivos (device profiles)

La capa de aplicación y el perfil de comunicación de CANopen (EN 50325-4; CiA 301) dan soporte el acceso directo a los parámetros de cada dispositivo y la transmisión de la información propia del proceso (por ejemplo señales de sensores, acciones de control, referencias, mensajes de error, etcétera). Son por tanto la base fundamental del protocolo.

El diccionario de objetos (OD, Object Dictionary) describe completamente la funcionalidad de cada dispositivo y permite su configuración mediante mensajes a través del propio bus. En el se describen tanto los parámetros de funcionamiento del dispositivo, como los mensajes con información del proceso que recibe y transmite. Dicha información se almacena de forma estandarizada y estructurada, en un archivo de configuración genérico, formado por una lista de entradas (cada entrada se define con un índice de 16 bits y un subíndice de 8).

Además, los servicios de gestión de red (NMT, network management services) de CANopen facilitan el diseño de la red, su configuración, puesta en marcha e integración, así como el diagnóstico de fallos. Por último, CANopen define una serie de perfiles de dispositivos (device profiles) que estandarizan la información contenida para diferentes tipos de dispositivos: módulos de entrada/salida, dispositivos de control, PLC's, encoders, inclinómetros, control de puertas, etcétera. De esta manera el usuario final puede utilizar indistintamente dispositivos de diferentes fabricantes, siempre que estos cumplan dichos estándares.

PC CONTROLLER

El controlado ha sido implementado en una arquitectura de tipo PC con el SO Linux (distribución Debian) como base.

El kernel de Linux ha sido modificada con RTLinux-GPL (Yodaiken, 2006), un núcleo de tiempo real con licencia GPL. Por otra parte, las características del PC empleado son:

- 1 Desktop PC, con procesador AMD-K6 400 MHz y 128 Mb RAM.
- 1 PCI-CAN 7841 ADLINK CARD.

La razón para esta elección es que RTLinux-GPL y Linux poseen una licencia de tipo GPL. Esto implica, entre otras cosas, la posibilidad de obtención del código fuente para su modificación o ampliación. En el caso de investigación, este punto es muy importante, debido a la necesidad del investigador a saber, en caso de necesidad, como funcionan las herramientas en las que está basando sus investigaciones, o modificarlas para poder ampliar el uso de éstas y adaptarlas a los objetivos propios de la investigación.

RTLinux toma como proceso de máxima prioridad el “Módulo de control” y mantiene a Linux como otro proceso de menor prioridad. Esto significa, que mientras tenga algo que ejecutar el módulo de control, Linux pasará a la espera. Por lo tanto, en caso de que el tiempo necesario de ejecución del control fuese superior al tiempo asignado previamente, Linux no se ejecutaría, por lo que el sistema daría la impresión de estar colgado, ya que el módulo de control se apoderaría completamente de la CPU. Cuando el modulo de control no se está ejecutando, entra en funcionamiento Linux, y se ejecutan el resto de procesos del sistema.

SCADA

El Scada ha sido creado mediante el empleo del entorno de desarrollo integrado (IDE) NetBeans. Este término hace referencia tanto a la plataforma de desarrollo de aplicaciones Java (The Java Virtual Machine Specification, 2005), como al entorno de integrado de programación (NetBeans, 2006). La plataforma NetBeans es un framework que simplifica el trabajo de desarrollo de aplicaciones para entornos desktop. Cuando una aplicación basada en NetBeans se activa, la clase *Main* de la plataforma se ejecuta. En ese momento, se localizan los módulos necesarios y se insertan en el registro de memoria, iniciando las tareas de inicialización de los módulos. Normalmente, el código de un módulo sólo se inserta en memoria si es usado por la aplicación. Por tanto, las aplicaciones pueden instalar los módulos de forma dinámica. La plataforma ofrece todas las herramientas necesarias para la creación de aplicaciones, permitiendo que el desarrollador se centre en la creación del código específico de su aplicación. Algunas de las características que ofrece la plataforma son:

- Manejo de interfaces de usuario (menús, barras de herramientas, etc.)
- Manejo de las configuraciones de usuario
- Manejo del almacenamiento de datos
- Manejo de ventanas gráficas
- Framework wizard (soporte de diálogos paso a paso)

El IDE NetBeans es un entorno de desarrollo Open Source escrito íntegramente en Java utilizando la plataforma NetBeans. Este IDE soporta el desarrollo de todo tipo de aplicaciones Java (J2SE, web, EJB, etc). El Scada desarrollado empleando NetBeans se muestra en la figura 6. Los aspectos más importante de la aplicación son:

- Botón Connect, establece la conexión con el controlador PC)
- Botón Start/Stop, inicio o finaliza la ejecución del sistema
- Visualización gráfica del comportamiento del Péndulo invertido.
- Visualización de la ecuación del controlador.
- Visualización numérica de la variables y almacenamiento de datos.
- Visualización de la evolución temporal de las variables del sistema.

La comunicación via Internet entre el Scada y el controlador PC se realiza mediante protocolo TCP/IP y *ssh* (como se muestra en la figura 1). El Scada se comunica con el controlador via *ssh* con privilegios *root*. El controlador PC tiene un servidor *ssh* que permite la conexión remota a través de Internet. Por tanto el Scada se puede ejecutar en cualquier PC con *la máquina virtual de JAVA* y conexión a Internet. RTLinux y otros módulos necesarios del SO del controlador PC son cargados en memoria con privilegios de *Administrador* para un correcto funcionamiento del sistema. En cualquier

caso, todas las cuestiones referentes a la comunicación via Internet son gestionados automáticamente por el Scada. El software desarrollado para todo el sistema (Scada, algoritmos de control, manejo del bus de campo, etc) se distribuyen bajo licencia GPL y se encuentra disponible en la siguiente dirección web: <http://ctl-predictivo.upv.es/programas.htm>.

GUÍA DE USUARIO

La presente sección describe el manual de usuario para el correcto funcionamiento y utilización de la plataforma descrita en la sección 3. Basicamente, los estudiantes deben seguir tres pasos para utilizar apropiadamente el sistema.

Inicialización

El primer paso es la inicialización de la aplicación. El botón *Connect* (figura 6, punto 1) establece la conexión con el controlador PC y prepara los elementos de visualización.

Selección del Controlador

El segundo paso es la selección de la ley de control que se empleará. El Scada incorpora ocho algoritmos de control diferentes, tal y como muestra la tabla 1. La *herramienta combo* (figura 6, punto 2) permite la selección de la ley de control deseada.

Ejecutando la Aplicación

Finalmente, el botón *Run* inicia el sistema de control y la transferencia de información de las variables del proceso a través de Internet. La figure 6 muestra la aplicación durante una ejecución. La visualización de las variables (figura 6, puntos 3 y 5), está optimizada para conseguir una visualización realista y apropiada. El Scada actualiza la información que se visualiza cada 40 ms. Este tiempo de muestro produce una animación continua y razonablemente adecuada.

Almacenamiento de la Información

La información recibida a través de Internet y visualizada en las gráficas y displays numéricos, puede ser almacenada en un fichero mediante la selección del *checkbox Save Dates* (figura 6, punto 5). Esta opción almacena toda la información en un fichero de tipo .dat cuyo nombre se establece como:

<TipoControlador_Fecha+Hora.dat >

La estructura de datos en el fichero es (por columna) de la siguiente forma:

<i>Formato del Fichero por Columna</i>					
Tiempo	Fuerza	Ángulo	Velocidad Angular	Velocidad del carro	Tiempo de muestreo

Experimentos Propuestos

Los estudiante pueden modificar el sistema para realizar diferentes análisis y comprobaciones prácticas, con el objetivo de obtener un mejor conocimiento del comportamiento de sistemas de control reales. A continuación se proponen algunas posibilidades:

- *Modelos de incertidumbre*: Los modelos de Simulink que se ejecutan en el xPC pueden ser modificados, con el objetivo de introducir cierto grado de incertidumbre. Por ejemplo, se pueden modificar algunos de los valores numéricos de los parámetros del modelo (incertidumbre paramétrica).
- *Ruido en las señales*: La inclusión de ruido blanco en el modelo Simulink, junto con el ruido existente en los elementos de comunicación, puede emplearse para analizar una el comportamiento del sistema en entornos con un elevado ruido en los elementos de medida.

- Errores de comunicación: La introducción de retardos (delays) en el diagrama Simulink, puede ser interpretados como errores de comunicación en una red CAN sobrecargada.

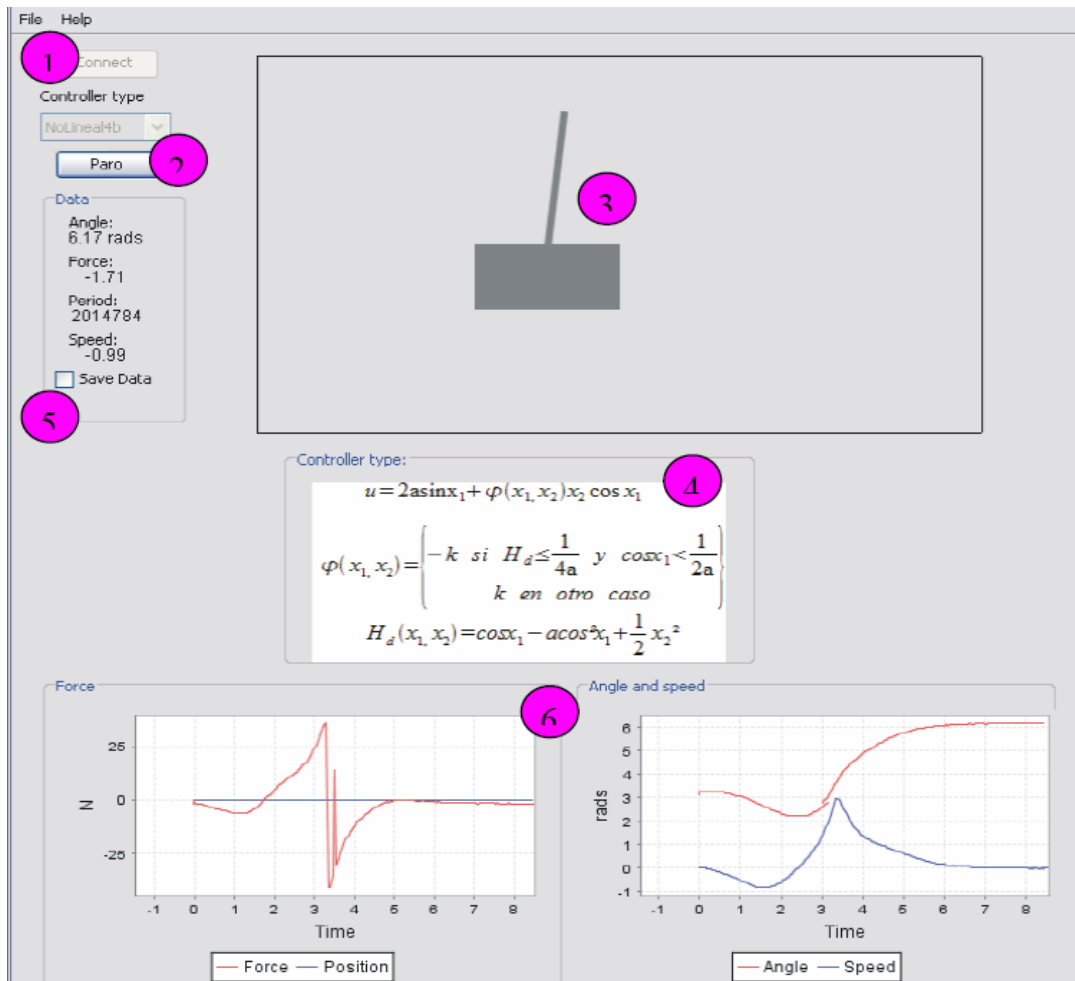


Fig. 6: Pantalla Principal del SCADA

Tabla 1: Controladores Predefinidos

	Ley de Control	
Control 1 (con a=2):	$u = 2a \tan x_1$	(9)
Control 2 (con a=2):	$u = 2a \sin x_1$	(10)
Control 3 (con a=2 y k=5):	$u = 2a \tan x_1 + kx_2$	(11)
Control 4 (con a=2 y k=5):	$u = 2a \sin x_1 + \varphi x_2 \cos x_1$	(12)
	$\varphi = \begin{cases} -k & \text{si } (H_d \leq \frac{1}{4a}) \text{ y } (\cos x_1) < \frac{1}{2a} \\ k & \text{otro caso} \end{cases}$	(13)
	$H_d(x_1, x_2) = \cos x_1 - a \cos^2 x_1 + \frac{1}{2} x_2^2$	(14)
Control 5:	Ley de control (12) con a=1 y k=2	
Control 6:	Ley de control (15) con $\varepsilon_2=0.5$, $\varepsilon=0.25$, $\lambda=-1$, k=0.6, a=1 y $x_{1_0} = \pi$	
Control 7:	Ley de control (16) con $\varepsilon_2=0.5$, $\varepsilon=0.5$, $\lambda=-1$, k=2, a=2 y $x_{1_0} = \pi$	
Control 8	Ley de control (18) con $\mu=0.6$, k=0.8 y $x_{1_0} = \pi$	

RESULTADOS Y DISCUSIÓN

La presente sección muestra los resultados obtenidos fruto del trabajo iniciado en García-Nieto et al. (2005) y basados en Aracil y Gordillo (2004) y Aström y Furuta (2000).

Ley de control basada en la inyección de energía y amortiguamiento

Las ecuación (15) muestra la ley de control empleada y los resultado experimentales obtenidos se muestran en la figura 7.

$$u = \underbrace{2a \sin x_1}_{u_c} + \underbrace{\varphi_\varepsilon(x_1, x_2)}_{u_d} x_2, \quad \varphi_\varepsilon(x_1, x_2) = \begin{cases} k(H_d(x_1, x_2) - \frac{1}{4a} - \varepsilon) & \text{if } \cos x_1 < \frac{1}{2a} \\ k & \text{en otro caso} \end{cases} \quad (15)$$

Péndulo con carro

Se emplea la ley de control (16), la cual presenta dos términos diferentes. La estabilización del péndulo se consigue mediante los dos primeros términos de la ecuación. Mientras que el último término consigue que la velocidad del carro sea cero en régimen estacionario.

$$u = 2a \sin x_1 + \varphi_\varepsilon x_2 \cos x_1 - \varepsilon_2 \rho\left(\frac{\lambda x_3}{\varepsilon_2}\right), \quad \rho(y) = \text{sgn}(y) \min\{|y|, 1\} \quad (16)$$

donde el modelo del péndulo es expresado mediante la ecuación (17) y los resultado experimentales obtenidos se muestran en la figura 8.

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= \sin x_1 - \cos x_1 u \\ \dot{x}_3 &= u \end{aligned} \quad (17)$$

Oscilación autónoma del péndulo invertido

La ley de control (18), propuesta en (Aracil y Gordillo, 2004), produce una oscilación fija y determinada del péndulo invertido. La figura 9 presenta los resultados obtenidos del experimento realizado.

$$u = \frac{x_1 + k\Gamma x_2 + \sin x_1}{\cos x_1}, \quad \Gamma = (x_1^2 + x_2^2 - \mu) \quad (18)$$

Análisis de los Resultados

Las principales diferencias entre los resultados obtenidos mediante el empleo de la plataforma y simulación, son:

- Error constante en las variables de posición y velocidad.
- Comportamiento dinámico es ligeramente distinto.

Las principales razón que explican estos fenómenos son:

- Control Discreto. Las simulaciones han sido realizadas en el dominio continuo, mientras que la plataforma trabaja en discreto, con un period de muestro fijo.

- Presencia de interferencias en los sensores y actuadores. Otra importante cuestión es la existencia de ruido electromagnético, el cual introduce importantes diferencias entre las simulaciones y la experimentación con la plataforma.
- La resolución de la tarjeta de adquisición. La resolución de las variables simuladas es de 2^{64} bits, mientras que la resolución de la plataforma es de 2^{10} bits.

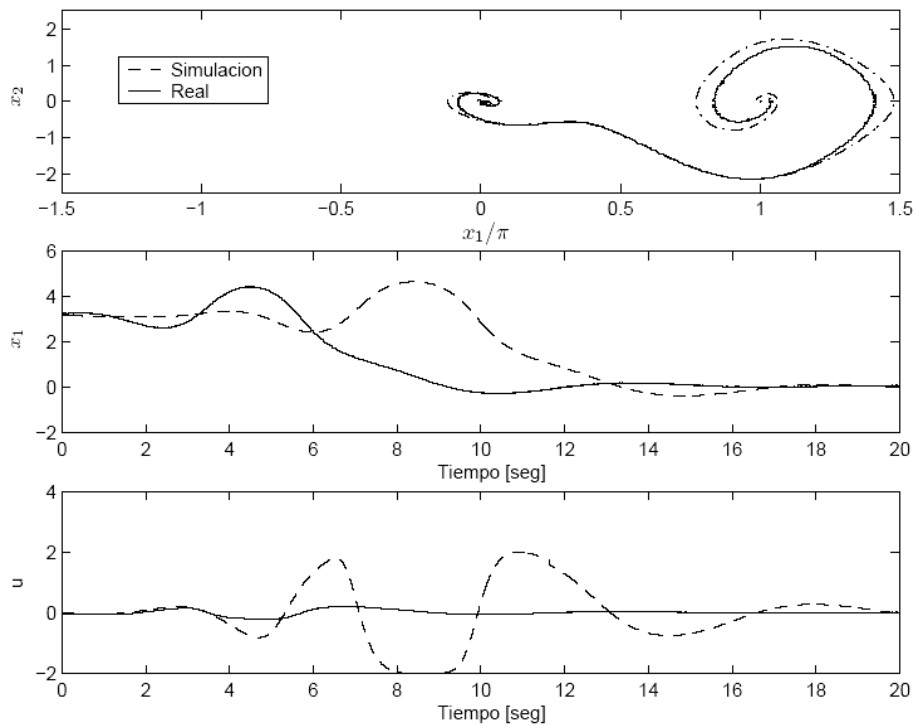


Fig. 7: $\varepsilon_2=0.5$, $\varepsilon=0.25$, $\lambda=-1$, $k=0.6$, $a=1$ y $x_{1_0} = \pi$

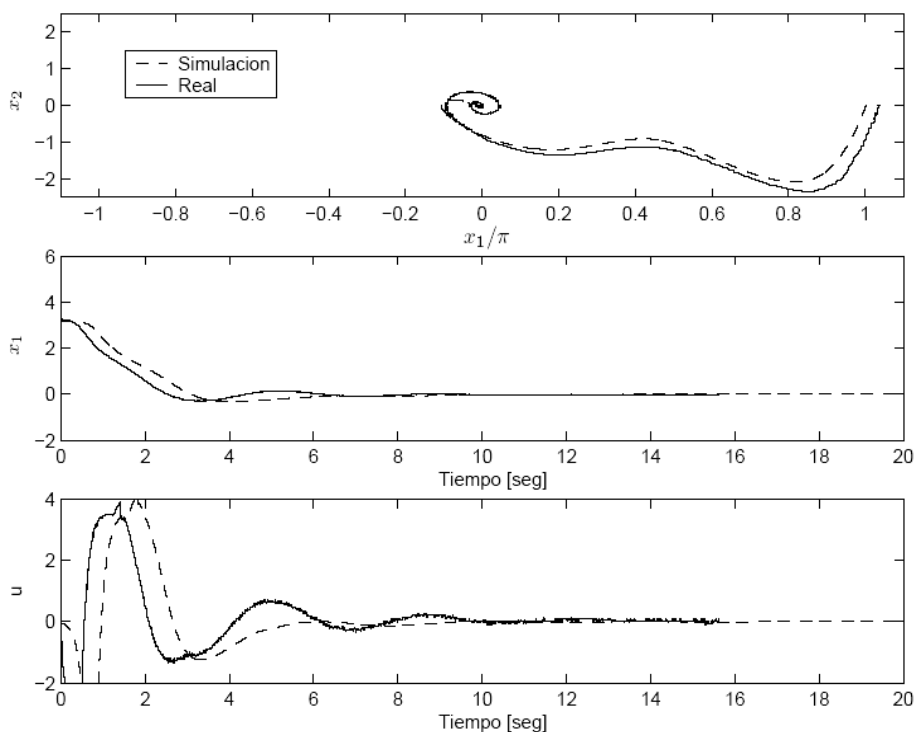


Fig. 8: $\varepsilon_2=0.5$, $\varepsilon=0.5$, $\lambda=-1$, $k=2$, $a=2$ y $x_{1_0} = \pi$

En este caso, la componente que introduce mayor variabilidad respecto a la simulación, es la resolución de la tarjeta de adquisición de datos. El máximo error de cuantificación es 0.01 V (LSB/2), dado que la resolución de la tarjeta es de 0.019 V. Esta diferencia con la simulación resulta determinante en estado estacionario y de menor influencia en el transitorio. A pesar de todo, las leyes de control presentadas son suficientemente robustas frente a los condicionantes del sistema, ya que consiguen estabilizar el péndulo invertido.

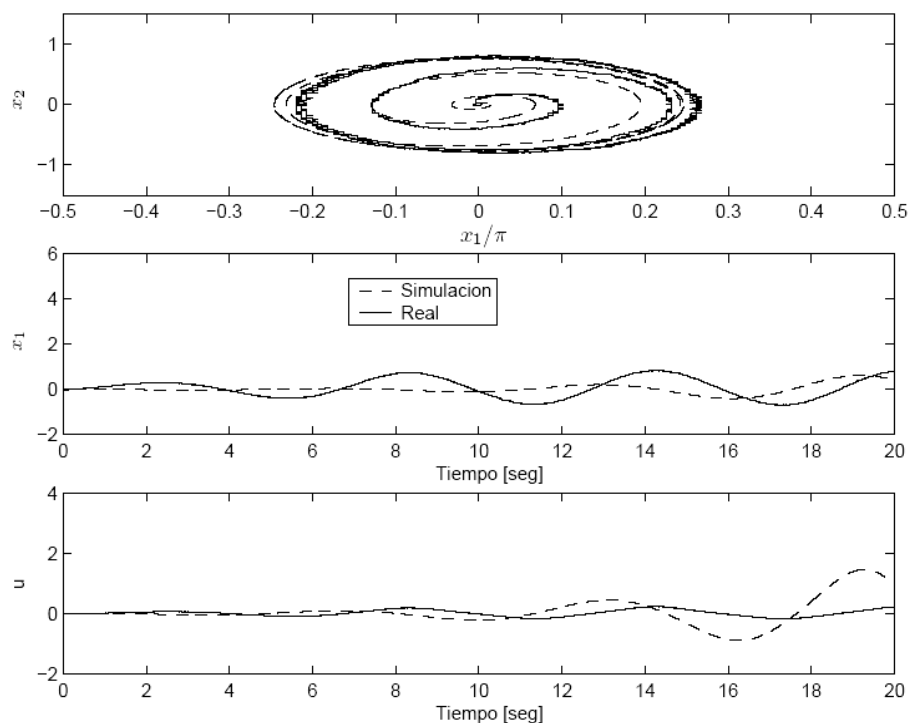


Fig. 9: $\mu=0.6$, $k=0.8$ y $x_{1_0} = \pi$.

CONCLUSIONES

Los resultados obtenidos en los experimentos, muestran las diferencias que aparecen en los algoritmos teóricos cuando estos son implementados en condiciones cercanas a la realidad. Cabe destacar que en este artículo sólo se han incluido, como elementos reales, la parte de comunicaciones y sensorización. Por tanto, si se trabajase con el modelo real del péndulo las discrepancias entre la teoría y la realidad serían aún mayores. Por ejemplo, las fricciones del sistema real no se han modelado y afectan de manera directa en el comportamiento del proceso. Por otro lado, se ha confirmado la validez de la plataforma distribuida en tiempo real. Esto proporciona una herramienta útil a la hora del prototipado rápido y primera fase de evaluación de algoritmos de control. La plataforma proporciona un método rápido y seguro de experimentación, con la ventaja de incorporar comportamientos más reales a nivel de comunicaciones y sensorización. Cuestión fundamental en el control de procesos.

Una de las vías propuestas para trabajos futuros es el reemplazo de la herramienta xPC por una implementación basada en un herramienta Open Source, como Octave (Octave, 2006) o Scilab (Scilab, 2006). Por otro lado, el interfaz de usuario puede ser modificado con el objetivo de incluir mayor funcionalidad y versatilidad.

AGRADECIMIENTOS

Parcialmente financiado por los proyectos de investigación del MEC, Gobierno Español FEDER DPI2004-8383-C03-02, FEDER DPI2005-07835 y Generalitat Valenciana GV06/26

REFERENCIAS

American CANopen Networks; website: <http://www.canopen.us/> (2005).

Apkarian, J.; Internet control, <http://www.circuitcellar.com> (1999).

Aracil, J. y F. Gordillo; The inverted pendulum: a benchmark in nonlinear control. In: Proceedings of the World Automation Congress, Seville (2004).

Aracil, J. y F. Gordillo; El péndulo invertido: Un desafío para el control no lineal, RIAI (2005).

Aström, K.J. y K. Furuta. Swing up a pendulum by energy control, *Automatica*: 36(3), 287-295 (2000)

Benmohamed, H., A. Leleve y P. Prevot; Generic Framework for Remote Laboratory Integration, International Conference on Information Technology Based Higher Education and Training (2005).

Boterenbrood, H.; CANOpen: high-level protocol for can-bus, <http://www.nikhef.nl/pub/departments/ct/po/doc/CANopen30.pdf> (2000).

Duan, B. y otros cuatro autores; An Online Laboratory Framework for Control Engineering Courses. *International Journal of Educational Engineering*: 21(6), 1068 (2005).

Furuta, K., M. Yamakita y S. Kobayashi; Swing-up control of inverted pendulum using pseudo-state feedback, *Journal Systems and Control Engineering*, 206, 263–269 (1992).

García-Nieto, S., A. Llosá, M. Martínez y X. Blasco; Implementación de algoritmos moldeadores de energía en tiempo real, *Jornadas de Automática*, Alicante, 389–396 (2005).

NetBeans developer Tool; <http://www.netbeans.org/> (2006).

Octave, GNU; <http://www.octave.org/> (2006).

Sánchez, J., S. Dormido, R. Pastor y F. Morilla. A Java/Matlab-based environment for remote control system laboratories: illustrated with an inverted pendulum. *IEEE Transactions on Education*: 47(3), 321-329 (2004).

Sánchez, J. y otros cuatro autores; Virtual and remote control labs using Java: a qualitative approach. *IEEE Control Systems Magazine*: 22(2), 8-20 (2002).

Schilling, K., T. Adami y R. Irwin; A virtual laboratory for space systems engineering experiments, 15Th IFAC Symposium on Automatic Control in Aerospace (2001).

Scilab; The open source platform for numerical computation. <http://www.scilab.org/> (2006).

The Java Virtual Machine Specification; website. <http://java.sun.com/docs/books/vmspec/> (2005).

The MathWorks, Inc.; xPC Target Users Guide, Version 2 ed., The Mathworks, INC., Natick, MA (2002).

Yodaiken, V.; The rtlinux manifesto, <http://www.rtlinux.org/> (2006).