

LABORATORIOS VIRTUALES Y REMOTOS BASADOS EN EJS PARA LA ENSEÑANZA DE ROBÓTICA INDUSTRIAL

Carlos Alberto Jara Bravo, Francisco Andrés Candelas Herías y Fernando Torres Medina
Grupo de Automática, Robótica y Visión Artificial
Departamento de Física, Ingeniería de Sistemas y Teoría de la Señal, Universidad de Alicante
Apartado de Correos 99, 03080 Alicante
cajb@dfists.ua.es, Francisco.Candelas@ua.es, Fernando.Torres@ua.es

Resumen

El propósito general de este artículo se centra en presentar una serie de herramientas para la enseñanza a distancia de robótica industrial, que permite a los usuarios interactuar con simulaciones de robots reales de una manera intuitiva y económica. Mediante las aplicaciones es posible un aprendizaje rápido de conceptos de robótica, tales como la estructura mecánica, parámetros geométricos, cinemática directa e inversa y planificación de trayectorias. El usuario podrá simular y teleoperar la trayectoria del robot y así, comprobar el comportamiento del mismo. También se realiza una breve descripción de Easy Java Simulations (Ejs), software basado en Java que se ha utilizado para la creación de las aplicaciones.

Palabras Clave: educación, java, interacción remota, robótica, simulación, virtual.

1 INTRODUCCIÓN

Generalmente, la educación y la docencia son de carácter presencial. Sin embargo, debido a diversos factores económicos, espaciales y temporales, este método de enseñanza no ofrece una calidad suficiente en ciertas materias de carácter técnico [1].

Con la llegada de las tecnologías telemáticas y de Internet, aparece una nueva era de conectividad entre máquinas que proporciona innovadoras formas de adquirir, analizar y presentar la información. Este hecho ha abierto un mar de nuevas posibilidades para la formación a distancia, incluidas bajo el término anglosajón *e-Learning* [8].

Este rápido crecimiento de la tecnología ha motivado a los ingenieros a diseñar nuevos sistemas de enseñanza y aprendizaje interactivos que se conocen como *laboratorios virtuales y remotos*. Su aparición ha supuesto una mayor satisfacción tanto en las necesidades académicas como en las de formación, además de romper la barrera interpuesta por los

factores económicos, espaciales y temporales, para proporcionar una calidad adecuada. Los laboratorios virtuales y remotos no pretenden ni suplantar ni competir con los laboratorios tradicionales. De hecho, constituyen una posible extensión de los mismos, abriendo nuevas perspectivas que, dentro de un laboratorio físico, no se podrían explorar completamente a un costo asequible [9].

El artículo se ha organizado de la forma siguiente. A continuación, se describen las ventajas de los laboratorios virtuales y remotos en la enseñanza técnica. Posteriormente, se realiza un breve resumen de las características principales de Ejs. La sección 2 está dedicada a explicar las principales características de los laboratorios virtuales desarrollados. Más tarde, se describirá el laboratorio remoto. Y finalmente, se comentan algunas conclusiones y posibles trabajos futuros.

1.1 EL LABORATORIO VIRTUAL REMOTO: UNA HERRAMIENTA PARA LA ENSEÑANZA TÉCNICA

Actualmente, la formación experimental de los estudiantes de ingeniería se está convirtiendo en la llave que abre las puertas a la prosperidad dentro del campo industrial. Por esta razón, muchos de los grupos de investigación de carácter técnico, trabajan en el desarrollo de laboratorios virtuales y remotos para la docencia en ingeniería [1].

Dentro del área de Educación en Control, muchas de las plataformas de aprendizaje a distancia desarrolladas, se basan en el estudio de la estabilidad de los sistemas lineales y no lineales [6]. Otros estudian el comportamiento de los sistemas reales mediante el acceso al dispositivo físico localizado en el laboratorio remoto [10].

Por otro lado, en el campo de la Robótica, también se han desarrollado varios laboratorios virtuales y remotos. Su funcionalidad se basa en el control y teleoperación a distancia tanto de robots móviles [11] como de industriales [2]. Estas aplicaciones permiten al estudiante adquirir rápidamente conceptos y

habilidades para controlar un robot, comprender su funcionamiento y estar preparado para manejarlo adecuadamente.

Por consiguiente, los autores presentan en este artículo una serie de laboratorios virtuales y remotos para la enseñanza y adiestramiento de robótica industrial, que permite a los estudiantes interactuar con entornos virtuales y remotos. Mediante estas aplicaciones es posible un aprendizaje rápido de conceptos de robótica, tales como la estructura mecánica, parámetros Denavit-Hartenberg, cinemática directa e inversa y planificación de trayectorias. Además, son muy sencillos de utilizar, incluso para la gente que no está familiarizada con el mundo de la robótica. Para la realización de estas aplicaciones se ha utilizado Ejs, software basado en Java ideado para el desarrollo de aplicaciones docentes.

1.2 PRINCIPIOS DE EASY JAVA SIMULATIONS

Ejs es una herramienta *open-source* elaborada en Java para la creación de laboratorios virtuales interactivos [7]. Fue parte del proyecto *Open Source Physics*, que se estableció con el objetivo de crear y distribuir simulaciones para su uso pedagógico en la enseñanza de la física [4]. Las simulaciones creadas con Ejs pueden ser usadas como programas *stand-alone* bajo diferentes sistemas operativos o ser distribuidas como *applets* a través de Internet.

Las simulaciones Ejs constan de dos partes bien diferenciadas: el modelo y la vista. El modelo está constituido por las variables de la simulación, su valor inicial y por las ecuaciones matemáticas que rigen la evolución del sistema. Para su implementación, Ejs proporciona dos posibilidades:

un editor de ecuaciones diferenciales ordinarias (ODEs) y una conexión con Matlab/Simulink [5].

La vista es la interfaz de usuario donde se muestra la representación gráfica de los diferentes estados del sistema. Ambas partes se encuentran interconectadas. Cualquier cambio en el estado del modelo es visualizado en la vista, y si el usuario interactúa con ella (controles de la interfaz), es capaz de modificar el valor de una variable del modelo.

La principal razón de escoger Ejs para el desarrollo tanto de los laboratorios virtuales como del remoto, es su simplicidad de uso y que no es necesario ser un programador experto para crear una aplicación. Por lo tanto, no ha sido necesario emplear una gran cantidad de tiempo para crear una interfaz gráfica interactiva y ha permitido concentrarse más en la descripción del sistema robótico.

2 LABORATORIOS VIRTUALES

Los laboratorios virtuales son herramientas que dan la posibilidad de experimentar y explorar el comportamiento del sistema robótico. De esta manera, ayudan al usuario a comprender complejos conceptos a través de la simulación.

En este apartado se van a mostrar dos laboratorios virtuales desarrollados donde se simulan un robot de 6 grados de libertad (gdl) tipo PA-10 y uno de 3 gdl tipo RRR. Su principal diferencia se basa en el método empleado en la construcción del modelo robótico. Mientras que en el primero (Figura 1) se ha utilizado la conexión entre Ejs y Matlab, en el segundo tan sólo se ha empleado el editor de ODEs (Figura 2).

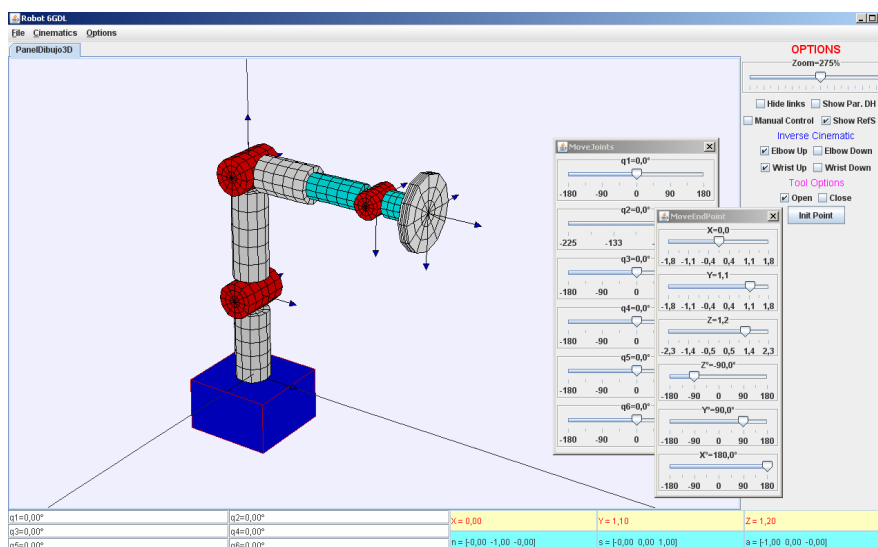


Figura 1: Interfaz gráfica del robot de 6gdl

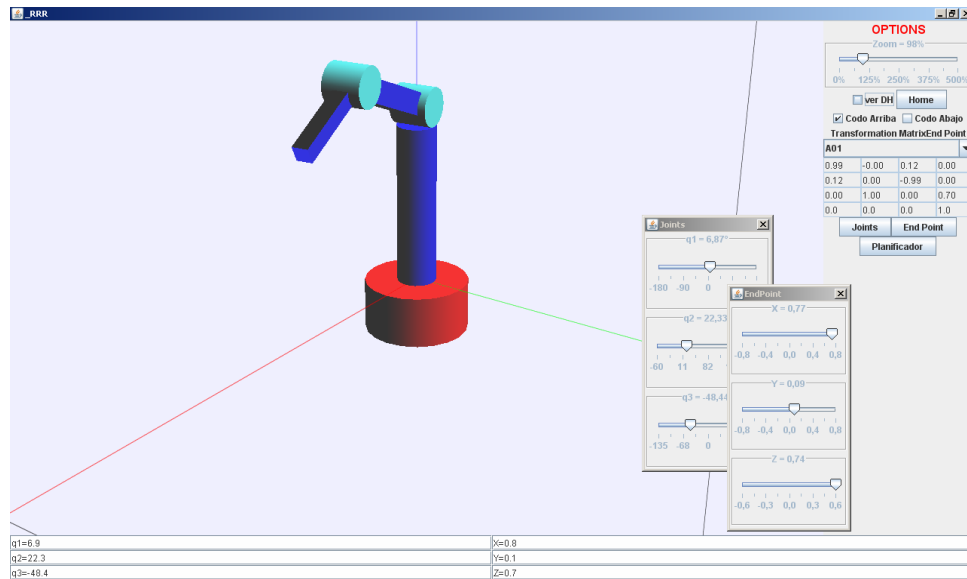


Figura 2: Interfaz gráfica del robot de 3gdl

La vista principal de ambos laboratorios virtuales se ha organizado de manera similar. La parte superior de la interfaz contiene una representación 3D del espacio de trabajo donde se visualiza el robot. Podemos observar también, una serie de diálogos que contienen controles mediante los que se mueve el robot (ver sección 2.1). Finalmente, en la parte derecha de la vista podemos ver que hay una serie de controles con más opciones como: zoom, y mostrar/ocultar los sistemas de referencia Denavit-Hartenberg, etc,... En las siguientes subsecciones se describirán las distintas posibilidades de ambos laboratorios virtuales.

2.1 CINEMÁTICA DIRECTA E INVERSA

El movimiento del robot se basa en la implementación del modelo cinemático directo e inverso del mismo [9]. Dentro de esta opción, es posible mover el robot mediante controles articulares dando valores a cada una de las articulaciones del robot (controles “q”), o mediante controles cartesianos dando valores al punto final (controles X, Y, Z). Las interfaces muestran en todo momento la matriz de transformación del extremo del robot y los valores de posición de las articulaciones.

2.2 PARÁMETROS GEOMÉTRICOS

En los laboratorios virtuales es posible editar y cambiar los valores que definen la geometría de los robots. Existe un diálogo específico con varios controles para realizarlo. Además de los parámetros geométricos, también es posible cambiar el radio de los eslabones. La simulación gráfica se actualiza

dinámicamente mientras el usuario va cambiando los parámetros.

2.3 PLANIFICADOR DE TRAYECTORIAS

Las aplicaciones desarrolladas permiten la simulación de distintos interpoladores en el espacio articular. A continuación se describen los distintos interpoladores implementados, mostrando el resultado gráfico de cada uno de ellos.

2.3.1 Interpolador lineal

Este interpolador consiste en mantener la velocidad de movimiento entre los puntos inicial y final de la trayectoria (q_i , q_f). La ecuación (1) muestra la fórmula que rige la evolución de la trayectoria, donde T es el tiempo total y dt es el valor del diferencial de tiempo.

$$q(t) = \frac{(q_f - q_i)}{T} \cdot dt + q_i = V_q \cdot dt + q_i \quad (1)$$

Esta trayectoria asegura la continuidad en posición, sin embargo origina saltos bruscos en la velocidad provocando aceleraciones de salto infinito. Aunque en la práctica este tipo de interpolador no es posible, se ha implementado en la simulación, resultando gráficas como la que se muestra en la Figura 3.

2.3.2 Interpolador tipo *splin*

Este tipo de interpolador asegura que la trayectoria que une los puntos articulares presente continuidad en velocidad.

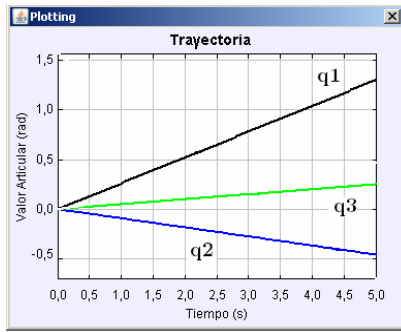


Figura 3: Interpolador lineal

De este modo, el grado del polinomio que rige la evolución temporal de la trayectoria debe ser de grado 3 o mayor. En este caso, se ha implementado un *splin* de quinto grado asegurando también la continuidad en aceleración (Ecuación 2).

$$q(t) = a \cdot t^5 + b \cdot t^4 + c \cdot t^3 + d \cdot t^2 + e \cdot t + f \quad (2)$$

El resultado obtenido en la simulación se muestra en la Figura 4, donde se puede observar la suavidad y continuidad del movimiento.

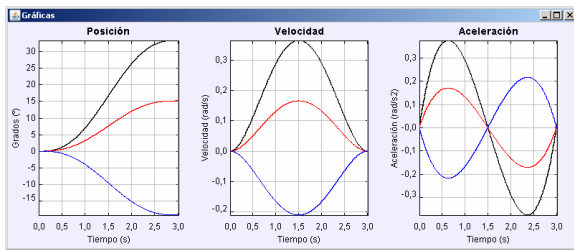


Figura 4: Interpolador *splin* de grado 5

2.3.3 Interpolador a tramos

El interpolador a tramos consiste en descomponer la trayectoria articular en tres fragmentos consecutivos: aceleración, velocidad constante y deceleración.

Para los tramos de aceleración y deceleración pueden emplearse *splins* para obtener curvas suaves. En este caso, se ha implementado con un valor constante, resultado en la simulación gráficas como la que se muestra en la Figura 5.

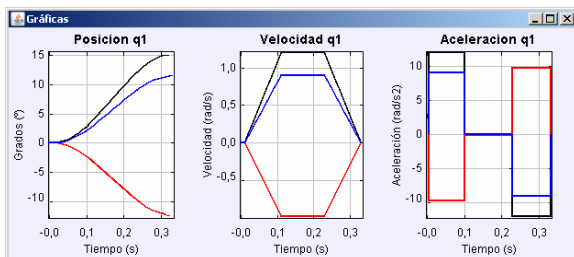


Figura 5: Interpolador a tramos

3 LABORATORIO REMOTO

A parte de los laboratorios virtuales desarrollados, se ha creado un laboratorio remoto que permite la simulación y teleoperación de un robot tipo Scorbot ER-IX (de Intelitek).

El usuario, mediante un *applet* desarrollado con Ejs, es capaz de simular y teleoperar un robot a través de Internet, con sólo teniendo instalado en su PC la máquina virtual de Java (*Java Virtual Machine*) y un explorador web. A continuación vamos a describir las características principales del laboratorio remoto.

3.1 SIMULACIÓN LOCAL

El aspecto gráfico de la interfaz se muestra en la Figura 6. Como se puede observar, posee una gran similitud con las interfaces desarrolladas para los laboratorios virtuales.

La aplicación permite al usuario practicar conceptos como la cinemática directa, inversa y planificación de trayectorias. Para esta última opción, la aplicación permite al usuario simular trayectorias articulares de tipo *síncrono* y *asíncrono*. Además, estas trayectorias se pueden guardar en una lista y ejecutarlas todas a la vez creando una trayectoria formada por varios puntos intermedios.

3.2 TELEOPERACIÓN

Una vez que el usuario ha obtenido una serie de trayectorias válidas, puede solicitar la ejecución remota de la lista completa de comandos en el robot real. Dichos comandos son enviados *servidor del robot*, PC que se encarga de comprobar son válidos antes de trasladarlos al lenguaje del robot. Esta acción garantiza el buen uso del sistema real.

Para que el usuario pueda seguir el movimiento real del robot, la aplicación dispone de dos opciones de realimentación [3]. La primera es mediante un flujo de video comprimido que se transmite de forma *on-line* al ordenador del usuario y mediante una actualización de la simulación con información *on-line* recibida del controlador del robot de sus posiciones articulares reales (Figura 7).

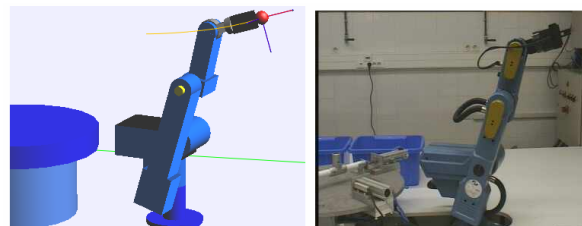


Figura 7: Opciones de realimentación

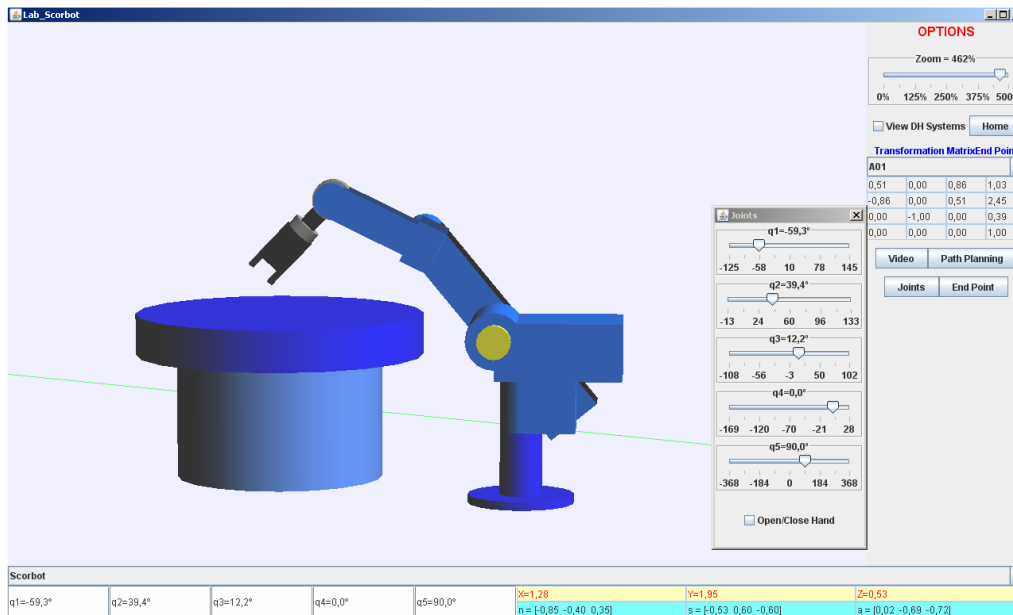


Figura 6: Interfaz del laboratorio remoto

4 TRABAJOS FUTUROS

Actualmente, los autores continúan desarrollando las aplicaciones descritas en este artículo, incluyendo nuevas características tanto en la interfaz gráfica como en el modelo del sistema robótico simulado.

Con respecto a la interfaz, se está trabajando en la posibilidad de añadir nuevos robots de diferentes características en el espacio de trabajo, ya que las aplicaciones actuales sólo contienen un robot. De esta forma, el usuario podrá practicar con diferentes tipos de robots en un entorno más real. También, se está trabajando en mejorar el aspecto gráfico 3D de la versión actual de Ejs. Así, se conseguirá más realismo en las simulaciones.

Con respecto al sistema robótico, la principal característica es añadir el modelo dinámico. Las herramientas actuales sólo incluyen la cinemática directa e inversa y el planificador de trayectorias. Dada la complejidad de este trabajo, se recurrirá a la comunicación que posee Ejs con softwares orientados al modelado de sistemas, tales como Modelica o Matlab. Con el modelo dinámico, la aplicación será mucho más completa y el usuario podrá evaluar el par en los actuadores cuando el robot simula una determinada tarea.

Posteriormente a las características comentadas, los autores trabajarán en añadir otras opciones como: reconocimiento 3D de objetos básicos en el entorno de trabajo, la inclusión de objetos virtuales usando realidad aumentada y un intérprete de comandos de voz para que el usuario guíe la tarea.

5 CONCLUSIONES

En este artículo se han presentado una serie de herramientas para la docencia a distancia de robótica industrial. Mediante las aplicaciones, el alumno puede simular y teleoperar robots industriales, aprendiendo conceptos de robótica tales como la cinemática directa e inversa y planificación de trayectorias.

El empleo de los laboratorios virtuales y remotos en el campo de la educación y aprendizaje permite al alumno ser adiestrado de una manera rápida y económica. De esta manera, estará más preparado para afrontar y resolver los problemas de los complejos dispositivos robóticos.

Las aplicaciones se han desarrollado usando Ejs, un *open-source* software diseñado para la creación de simulaciones interactivas. Mediante esta herramienta, el procedimiento para transformar el sistema robótico en un interactivo laboratorio virtual fue más sencillo que con cualquier otro programa. También es importante comentar que no se ha empleado mucho tiempo ni se ha necesitado adquirir habilidades de programación para realizar la interfaz gráfica de las simulaciones.

Agradecimientos

El trabajo presentado en este artículo está financiado por el Ministerio de Educación y Ciencia a través del programa de becas FPI y el proyecto de investigación DPI2005-06222. Los autores quieren agradecer esta financiación.

Referencias

- [1] Candelas, F. A. y Moreno, J. S., (2005). 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*, vol. 2, pp 93.
- [2] Candelas, F. A., Puente, S. T., Torres, F., Ortiz, F. G., Gil, P. y Pomares, J., (2003). A virtual laboratory for teaching robotics. *International Journal of Engineering Education*, vol. 19, pp 363-370.
- [3] Candelas, F. A., Puente, S. T., Torres F., Segarra, V. y Navarrete, J., (2005). Flexible System for Simulating and Tele-Operating Robots Through the Internet. *Journal of Robotic Systems*, vol. 22, pp 157-166.
- [4] Christian, W. y Belloni, M., (2003). Developing open source programs for science and mathematics. *EUROCON 2003*.
- [5] Dormido, S., Farias, G., Sanchez, J. y Esquembre, F., (2005). Adding interactivity to existing Simulink models using Easy Java Simulations. *44th IEEE Conference on Decision and Control*, vol. 2, pp 4163-4168.
- [6] Dormido, S. y E., F., (2003). The quadruple-tank process: An interactive tool for control education. *European Control Conference*. Cambridge.
- [7] Esquembre, F., (2004). Easy Java Simulations: a software tool to create scientific simulations in Java. *Computer Physics Communications*, vol. 156, pp 199-204.
- [8] Foix, C., Zavando S., (2002). Estándares e-Learning. *Informe Técnico*. Centro de Tecnologías de la Información.
- [9] Jara, C. A., (2007). Diseño de herramientas para la interacción remota on-line de robótica industrial. *Memoria de suficiencia investigadora*. Alicante.
- [10] Jiménez, L. M., Puerto, R., Reinoso, O., Fernández, C. y Ñeco, R., (2005). RECOLAB: Laboratorio remoto de control utilizando Matlab y Simulink. *Revista Iberoamericana de Automática e Informática Industrial*, vol. 2, pp 64-72.
- [11] Payá, L., Gil, A., Reinoso O., Jiménez, L.M. y Ballesta, M., (2006). Plataforma distribuida para la realización de prácticas de robótica móvil a través de Internet. *XXVII Jornadas de Automática*, pp. 547-554.