

# Utilización de un joystick económico en una interfaz de teleoperación basada en Java

Francisco A. Candelas, Santiago Puente, Fernando Torres, Jorge Pomares

Dept. de Ingeniería de Sistemas y Teoría de la Señal

Escuela Politécnica Superior

Univ. de Alicante

03690, San Vicente del Raspeig, Alicante

[Francisco.Candelas@ua.es](mailto:Francisco.Candelas@ua.es)

## Resumen

En los últimos años se han desarrollado bastantes laboratorios virtuales para la simulación y teleoperación de robots orientados a la docencia, que permiten a numerosos grupos de alumnos realizar prácticas de robótica, sin que la institución docente deba afrontar los grandes costes económicos que implicaría disponer de equipos reales. Pero la gran mayoría de estos entornos disponen de una interfaz de usuario que sólo permite al alumno interactuar con el robot mediante los clásicos dispositivos del teclado y ratón. En este artículo se expone como se puede utilizar un joystick económico en la interfaz de usuario de un laboratorio virtual de robótica desarrollada en Java, presentando los problemas que pueden surgir y una arquitectura software que permite resolverlos.

## 1. Introducción

Tradicionalmente, para la realización de las sesiones prácticas de una asignatura de ingeniería era necesario que el alumno asista a unos horarios fijos en un laboratorio que posee unos dispositivos físicos limitados. Sin embargo, en la actualidad, las nuevas tecnologías, como Internet o la realidad virtual, permite superar esos problemas.

Así, con los denominados laboratorios virtuales, una institución docente puede ofrecer herramientas para que grupos numerosos de alumnos realicen prácticas con recursos escasos [11]. Además el alumno puede realizar los ejercicios prácticos sin necesidad de asistir al aula de prácticas en unos horarios estrictos, a través de Internet, desde otra ubicación como puede ser su casa [3, 4, 8].

En el ámbito de la docencia de robótica, existen bastantes laboratorios virtuales que permiten la simulación de robots, muchos de los cuales permiten además teleoperar un robot real a través de Internet [2]. Entre ellos se encuentra el sistema Robolab desarrollado por los autores de este artículo [6]. Pero las interfaces de usuario de la mayoría de estos sistemas usan la información gráfica o de vídeo como único tipo de realimentación de información al usuario, y además, solo permiten manejar el robot mediante los típicos periféricos de teclado y ratón.

En los ámbitos de investigación y profesional, aparte de la información de realimentación de vídeo, los dispositivos maestros con realimentación de fuerza son otro tipo de dispositivo de mando empleado comúnmente para manejar la simulación y teleoperación de robots [1,9]. Este tipo de joysticks, además de proporcionar un método muy intuitivo para especificar los movimientos, también ofrece la capacidad de transferir la sensación de tacto o fuerza captada por el manipulador al brazo del operador humano.

Sin embargo estos joystick que están específicamente diseñados para aplicaciones de robótica son dispositivos complejos y bastante caros. Por eso, resulta casi imposible para una institución docente disponer de varios de estos dispositivos en un aula de prácticas para grupos con un considerable número de alumnos.

Además, si se desea que el sistema de simulación-teleoperación sea accesible por el alumno desde cualquier ubicación, como puede ser su casa, a través de Internet, no se puede pretender que el alumno disponga de un joystick profesional en su interfaz de usuario.

## 1.1. Robolab II

Robolab es un sistema de simulación y teleoperación de robots, pensado para que un grupo de alumnos pueda practicar comandos de movimiento en un robot industrial, aprendiendo conceptos básicos de robótica, y aspectos relativos a la cinemática y a la generación de trayectorias.

Los alumnos pueden usar todas las funciones que ofrece el sistema mediante un applet creado con el lenguaje Java de Sun [12]. Este applet se puede descargar de una página Web (<http://disclab.ua.es/robolab/>), y ofrece un entorno de simulación 3D que representa el robot y su espacio de trabajo (ver Figura 1), y que es equivalente al entorno real que hay nuestro laboratorio [4].

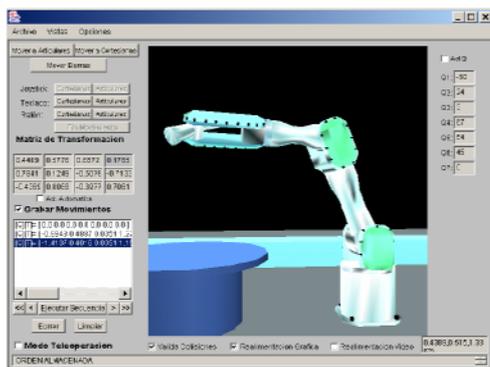


Figura 1. Aspecto de Robolab II

Principalmente, la interfaz que ofrece el applet Java permite al alumno especificar movimientos del robot mediante coordenadas cartesianas del área de trabajo o valores de las articulaciones de diferentes modos, así como los tiempos en los que se deben realizar los mismos en la simulación del entorno virtual. Cuando el alumno obtiene un resultado correcto de la simulación de un movimiento, puede añadir el movimiento a la lista de comandos que aparece en la interfaz. De esta manera, el alumno puede especificar los movimientos para que el robot siga una trayectoria propuesta. La lista de comandos también se puede simular de forma completa. Una vez lograda una lista de comandos validados, un alumno puede solicitar una ejecución remota en el robot, para lo que requiere autenticarse como un usuario registrado.

Robolab II incluye diversas mejoras respecto a la primera versión, entre las que cabe destacar las siguientes [5]. La principal es que la representación virtual del robot se realiza mediante Java-3D, de forma que toda la interfaz de usuario está contenida en un mismo applet, facilitando su utilización. En segundo lugar, el sistema utiliza una biblioteca de clases Java para el modelado de robots basada en Java-3D que permite modelar, simular y tele-operar cualquier brazo robot [7]. La Figura 1 muestra el aspecto de la interfaz con un robot PA-10 de Mitshubitshi.

## 2. Uso de un joystick con Robolab II

Con el objetivo principal de incrementar la capacidad de interacción entre el usuario y la interfaz de Robolab se decidió incorporar la opción de utilizar un joystick para especificar los movimientos del robot simulado de forma más directa que usando otros dispositivos como el teclado o el ratón. Además, de esta manera, la interfaz también resulta más atractiva para el alumno.

Pero teniendo en cuenta la complejidad y el alto coste de un joystick profesional, y dado que Robolab está orientado especialmente a la docencia a través de Internet, se optó por utilizar un joystick para juegos de bajo coste, que incluya además capacidades sencillas de realimentación de fuerza al usuario. Además, son muchos los aficionados a los juegos de ordenador los que disponen de dispositivos de este tipo en sus casas.

Sin embargo, hay que tener en cuenta que los dispositivos para juegos están frecuentemente diseñados para sistemas operativos concretos, y su programación depende de APIs (Applications Programming Interfaces) específicas. Este hecho contrasta con la filosofía de las aplicaciones portables de Java, como es el caso de nuestro sistema Robolab. Además, no hay una solución directa para incluir capacidades de gestión de joysticks de forma genérica en una aplicación Java, y hay que optar por soluciones como la que se describe a continuación.

### 2.1. Arquitectura del software

En nuestro caso, se ha decidido comenzar utilizando modelos de joystick diseñados para la plataforma MS Windows, principalmente por que

son de fácil adquisición y porque se instalan fácilmente en el S.O. En este caso, la forma de programación que permite aprovechar mejor las capacidades del joystick es utilizar el API de Microsoft DirectX. Este API ofrece una biblioteca de funciones a las aplicaciones que hacen transparente el modelo concreto de dispositivo de juegos utilizado, facilitando bastante la programación de las aplicaciones, y la compatibilidad de las mismas con diferentes tipos de dispositivos [10].

Sin embargo, las aplicaciones en Java, independientes del sistema operativo, no pueden acceder directamente a la funcionalidad que ofrece DirectX. Por ello es necesario utilizar un módulo que sirva de puente entre DirectX y la aplicación Java con la interfaz de usuario.

De este modo, para nuestro sistema Robolab, hemos desarrollado una DDL (Dynamic Link Library) para MS. Windows, programada en C, que hace la función de puente. Además, para conseguir que la aplicación Java de interfaz de usuario acceda a las funciones del joystick a alto nivel, independientemente del sistema operativo, hemos desarrollado otro módulo: un paquete de Java para el control del joystick. Actualizando este paquete en el futuro, se podrá incluir otros módulos de puente para plataformas diferentes de MS. Windows - DirectX, sin necesidad de alterar la aplicación.

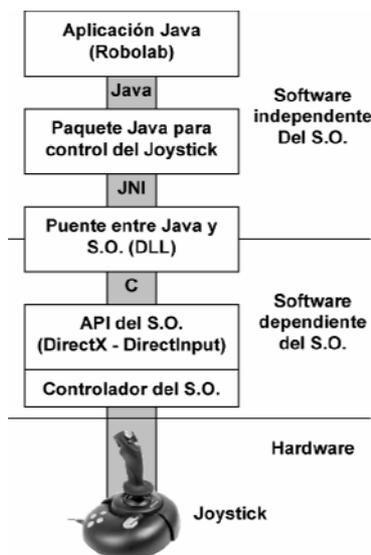


Figura 2. Arquitectura del software del joystick

Finalmente, la comunicación entre los dos módulos se realiza mediante la especificación JNI (Java Native Interface) [12]. La Figura 2 describe esta arquitectura.

Tanto la DLL como el paquete de control de joystick se han desarrollado de forma que pueden ser utilizadas para otras aplicaciones Java, y no son específicos para la interfaz de usuario de Robolab.

## 2.2. Integración del joystick en la aplicación

El paquete para control del joystick se incluye dentro de la aplicación Java, que en nuestro caso es la interfaz de usuario de Robolab II, como un thread, o hilo de ejecución independiente. Este thread se comunica con la aplicación mediante eventos o a través de variables de estado como muestra la Figura 3, y realiza principalmente tres tipos de funciones:

- Acceder al API de DirectX para iniciar el joystick y obtener información básica como el número de ejes, el número de botones, el modelo o las capacidades de generación de fuerzas.
- Leer periódicamente el estado de cada eje y cada botón del joystick, preprocesar esa información y guardarla en estructuras de datos locales. El periodo de muestreo se puede configurar. Los módulos de la aplicación pueden acceder a esa información cuando lo requieran.
- Informar a los módulos de la aplicación de cuando ocurren cambios de estado en el joystick. Para ello se utiliza el mecanismo de eventos de Java.

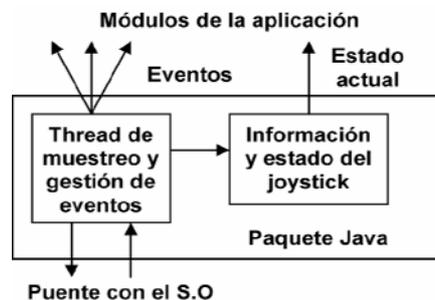


Figura 3. Paquete para control del joystick

### 2.3. Uso del joystick en Robolab II

Con la arquitectura descrita en los apartados anteriores, la aplicación Java de interfaz de usuario de Robolab II permite utilizar un joystick para mover el robot simulado directamente, así como para cambiar el punto de vista de la simulación. En concreto, hemos utilizado modelo "SideWinder Force Feedback" de Microsoft, mostrado en la Figura 4, que dispone de dos ejes principales (X, Y), un acelerador que se puede utilizar para el movimiento en el eje Z, y 8 botones.



Figura 4. Joystick Sidewinder de Microsoft

Cuando se utiliza el joystick para mover el robot, el usuario puede seleccionar con unos botones de la interfaz (Figura 5) si desea mover las articulaciones independientemente o mover el robot dentro del espacio cartesiano. En el primer caso, se utilizan los botones del joystick para seleccionar la articulación, y uno de los ejes principales para cambiar su posición. En el segundo caso se utilizan todos los ejes principales del joystick para especificar la posición de la herramienta.

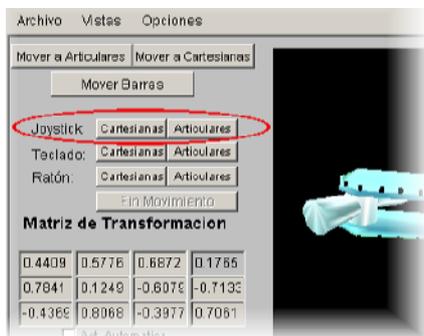


Figura 5. Detalle de los botones para usar el joystick

La aplicación considera distintos modelos de joystick, de forma que adapta los comandos de movimiento a las capacidades del dispositivo utilizado. Por ejemplo, si el joystick incluye algún mando con características de POV (Points Of View), estas estarán disponibles para cambiar el punto de vista de la simulación. De no ser así, se utilizan los ejes principales del mando, en conjunción con algún botón, para cambiar el punto de vista. Del mismo modo, se tiene en cuenta si el joystick tiene dos (X, Y) o tres (X, Y, Z) ejes principales a la hora de realizar movimientos del robot en coordenadas cartesianas, recurriéndose en el primer caso a la combinación del movimiento con la pulsación de algún botón.

Por otra parte, la simulación actúa sobre los motores del joystick para generar fuerzas, y hacer que el usuario sienta más realismo al desplazar el robot. Mientras se realiza un movimiento correcto, se producen fuerzas que acompañan al movimiento, para representar la inercia del robot. Pero cuando se llega al tope de una articulación, o cuando el robot colisiona con algún objeto del entorno, se genera una fuerza contraria al movimiento para dar la sensación de oposición al usuario.

También se ha añadido a Robolab la capacidad de utilizar el joystick para realizar la teleoperación on-line del robot remoto, sin necesidad de simulación previa para generar listas de comandos. Además, el robot remoto puede disponer de un sensor de fuerzas en su extremo, de forma que la información que genera, tras ser simplificada, se envía a la interfaz de usuario para que actúe, junto con las fuerzas generadas por la simulación, sobre los motores del joystick.

Este modo de funcionamiento se puede utilizar cuando la conexión entre el equipo con la interfaz de usuario y el servidor web se realiza mediante una LAN, con unos tiempos de retardos mínimos. En otro caso, como es sabido, el sistema se volvería inestable debido a los retardos entre las acciones del usuario y las percepciones correspondientes que recibe [1]. Por ello, se trata más bien de una alternativa experimental o docente, con la que se puede enseñar al alumno el efecto de los retardos en una teleoperación online.

### 3. Conclusiones

Desde el punto de vista docente, la utilización de un joystick en la interfaz permite un manejo más directo del robot por parte del alumno, que hace más atractiva la aplicación.

Resulta posible utilizar un joystick para juegos en una interfaz de usuario creada con Java, aunque para ello hay que desarrollar los módulos de software adecuados que garanticen un puente entre la aplicación y el API que ofrece una determinada plataforma con su sistema operativo y su hardware.

Si los módulos se desarrollan adecuadamente, será posible utilizar la aplicación sobre diferentes plataformas, sin necesidad de disponer de varias versiones de la misma. Solo es necesario disponer de un puente para cada plataforma. El siguiente paso de nuestros desarrollos es precisamente este, crear nuevas bibliotecas para otras plataformas, como puede ser Linux.

### Referencias

- [1] Aracil, R. Teleoperación. Sesión Invitada de las III Jornadas de Trabajo Eiwisa'02. Alicante, 2002.
- [2] Candelas, F.A., Sánchez, J. 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 (RIAI). Vol. 2, pp. 93-101, 2005.
- [3] Candelas, F.A., Gil, P., Torres, F., Ortíz, F.G., Puente, S.T., Pomares, J. Virtual Remote Laboratory for Teaching of Computer Vision and Robotics in the University of Alicante. Second IFAC Internet Based Control Education (IBCE 2004), Grenoble (Francia), 2004
- [4] Candelas, F.A., Puente, S.T., Torres, F., Ortíz, F.G., Gil, P., Pomares, J. A Virtual Laboratory for Teaching Robotics. International Journal of Engineering Education. Vol. 19, pp. 363-370, 2003.
- [5] Candelas, F.A., Puente, S.T., Torres, F., Segarra, V., Navarrete J. Flexible system for simulating and tele-operating robots through the internet. Journal of Robotic Systems (Wiley Periodicals, Inc.). Vol. 22, pp. 157-166, 2005.
- [6] Candelas, F.A., Torres, F., Gil, P., Ortiz, F., Puente, S.T., Pomares J. Laboratorio Virtual remoto para Robótica y Evaluación de su Impacto en la Docencia. Revista Iberoamericana de Automática e Informática Industrial (RIAI). Vol. 1, pp. 49-57, 2004.
- [7] Candelas, F.A., Torres, F., Puente, S., Pomares, J., Segarra, V., Navarrete, J. A flexible JAVA Class Library for Simulating and Teleoperating Robots. 11th IFAC Symposium on Information Control Problems in Manufacturing (INCOM04), Salvador, (Brasil), 2004.
- [8] Dormido, S., Sánchez, J., Morilla, F. Laboratorios virtuales y remotos para la práctica a distancia de la Automática, XXI Jornadas de Automática, Conferencia plenaria, Sevilla. 2000.
- [9] Hannaford, B. Feeling is Believing: A History of Telerobotics. The Robot in the Garden: Telerobotics and Telepistemology in the Age of the Internet. Edited by Ken Goldberg. The MIT Press, 2000.
- [10] Microsoft Developer Network (MSDN) Library. Documentación de los entornos de programación Visual Studio de Microsoft. 1998-2000.
- [11] Salzmann, Ch., Saucy, P., Gillet, D., Mondada, F. Sharing of Unique or Expensive Equipment for Research and Education, Informatik / Informatique, Magazine of the Swiss Informatics Societies. Vol. 4, pp. 32-33, 1999.
- [12] The Source for Java: <http://java.sun.com>.