HETERO v2.0: Herramienta para la tele-enseñanza de la robótica

J. Iván Maza Alcañiz
imaza@cartuja.us.es

Antonio de Castro Orbea, Aníbal Ollero Baturone
acastro@cartuja.us.es, aollero@cartuja.us.es

Escuela Superior de Ingenieros
Departamento de Ingeniería de Sistemas y Automática.
Camino de los Descubrimientos, s/n, 41.092 Sevilla (España)

Resumen

En este artículo se presenta un sitio web desarrollado con objeto de servir de apoyo a la docencia de la robótica. Dicho sitio web es la evolución del trabajo presentado en [3] y permite la ejecución remota de una serie de simulaciones, desarrolladas con la idea de ayudar al usuario a familiarizarse con ciertos conceptos básicos de la robótica. La presentación de resultados se ha cuidado con objeto de que el sitio web sea lo más atractivo posible para los usuarios. Para ello se han empleado tecnologías para la representación de escenarios virtuales como VRML y Java3D.

Palabras Clave: Java, laboratorio virtual, MATLAB, robótica, tele-enseñanza, VRML.

1 INTRODUCCIÓN

El desarrollo de las tecnologías *web* y de representación gráfica de escenas tridimensionales sobre máquinas heterogéneas posibilita el desarrollo de herramientas *software* conocidas como "laboratorios virtuales". En ellos, los usuarios pueden acceder a capacidades de simulación y tele-actuación sobre elementos reales o simulados desde cualquier localización geográfica. Este tipo de soluciones son de utilidad principalmente en entornos académicos, ya que el proceso de enseñanza a los estudiantes de una nueva tecnología es una tarea costosa, que requiere inversiones tanto en la adquisición de la maquinaria adecuada como en su posterior mantenimiento. Por ello, es posible que la escasez de medios disponibles impida el acceso de todos los estudiantes a una determinada tecnología. Desarrollar "*experimentos* virtuales" puede ser una solución a este problema. Estos experimentos pueden basarse en simulaciones *software* o bien en montajes reales accesibles remotamente.

La simulación *software* posee la ventaja de su configurabilidad, ya que puede ser fácilmente adaptada para imitar el comportamiento de distintos sistemas mediante el uso de modelos basados en equipos reales o fruto únicamente de la imaginación del usuario.

Por otra parte, la experimentación remota posee la ventaja de su gran accesibilidad, permitiendo que un mismo recurso sea compartido desde distintas localizaciones. Además, permite trabajar sobre el equipamiento real en lugar de una mera representación matemática de él.

Combinar estos dos enfoques posee incuestionables ventajas. Si se dispone del material en el laboratorio, se podrían usar experimentos con modelos simulados que permitieran que el usuario se familiarizase con la tecnología, logrando formar usuarios más maduros que usen posteriormente los equipos reales de forma remota con mayor seguridad y eficiencia.

2 HETERO

HETERO (HErramienta para la TEle-enseñanza de la RObótica) es un conjunto de aplicaciones *web* que configuran un entorno de simulación para problemas relacionados con la robótica en general, incluyendo tanto manipuladores robóticos como robots móviles. Al tratarse de aplicaciones que se ejecutan en entornos *web*, cualquier usuario puede acceder al servidor *web* donde están ubicadas mediante el uso de un navegador de Internet. Actualmente el sistema de simulación es accesible en la URL http://www.esi.us.es/hetero.

Las aplicaciones están concebidas para desarrollar un diálogo cliente-servidor mediante el intercambio de documentos *web*. El usuario aprovecha la conexión de red para ordenar, desde el navegador cliente, la ejecución de simulaciones en la máquina servidora remota donde tiene lugar el procesado de las peticiones y la computación de las simulaciones requeridas. El sistema proporciona un conjunto de

resultados que son visualizados en el propio navegador del cliente.

Implantar la aplicación HETERO como un conjunto de servicios *web* que se ofrecen al usuario de forma pública posee atractivas ventajas. La más destacable es la universalidad de acceso. Cualquier sistema que sea capaz de soportar un diálogo HTTP [9] y sea capaz de visualizar documentos codificados en HTML y VRML es un cliente susceptible de interaccionar de manera efectiva con HETERO, sin necesidad, en principio, de ningún *software* adicional. Caso de que el sistema cliente no disponga de soporte para la tecnología VRML se proveen mecanismos para actualizar de manera transparente el sistema (estos mecanismos automatizados sólo son válidos para plataformas cliente Win32).

Otra ventaja para aplicaciones como HETERO que se encuentran en un proceso de continuo desarrollo, es que el usuario siempre accede a la última versión desarrollada sin necesidad de preocuparse por la instalación de actualizaciones o "parches". Una característica adicional que no siempre es deseable, es el hecho de que las funciones MATLAB en las que están basadas las simulaciones son "opacas" para el usuario, de tal modo que se le oculta su código y solamente se le permite acceder a su funcionalidad.

3 PUNTO DE PARTIDA DEL ENTORNO DE SIMULACIÓN

Como motor interno para la computación de las simulaciones se utiliza el conjunto de funciones MATLAB que integran el *toolbox* HEMERO [2]. Esta herramienta facilita un conjunto de funciones MATLAB y Simulink que permite abordar la simulación de una gran variedad de sistemas robóticos, que van desde robots móviles con diferentes configuraciones de locomoción hasta manipuladores robóticos de base fija o móvil. De esta forma se puede considerar a HETERO como una interfaz *web* para simulaciones MATLAB de sistemas robóticos basadas en la invocación de un cierto número de funciones de HEMERO de acuerdo con los deseos del usuario / cliente.

Por otra parte, el sistema de simulación HETERO está concebido para su utilización conjunta con un sistema de enseñanza clásico, a través de un manual o de la docencia reglada, pudiendo practicar el usuario de forma inmediata los conceptos teóricos involucrados. En particular, se ha seguido la estructura de [4] para la organización de los distintos entornos de simulación y se han seleccionado algunos ejemplos de esa misma obra para la elaboración de versiones *web* de los mismos con el objetivo de que el usuario pueda familiarizarse con el sistema de simulación o evaluar las propias capacidades del entorno.

4 ARQUITECTURA DE HETERO

La arquitectura básica del sitio *web* se muestra en la Figura 1. Cada una de las capas que componen esa arquitectura se describen a continuación.

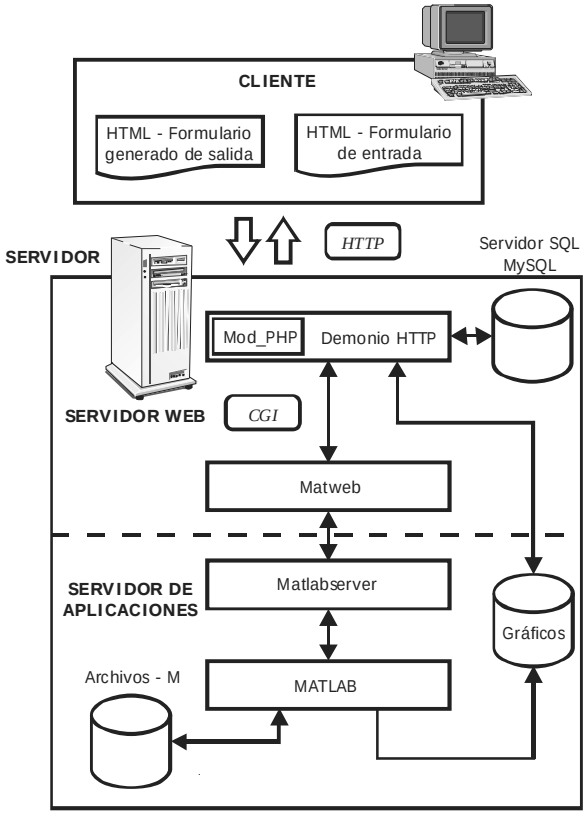

Figura 1: Arquitectura del HETERO.

4.1 CAPA DE PRESENTACIÓN

La capa de presentación constituye el punto de acceso del usuario/cliente a la aplicación. Presenta al usuario en el navegador de su máquina cliente los documentos *web* intercambiados con el servidor remoto. En ellos se describen las simulaciones o se muestran los resultados de las mismas. La aplicación está orientada a un perfil de usuario típico. Se supone un acceso del cliente a HETERO a través de un navegador de Internet compatible con HTML 4.0 [7] y la especificación DOM [8] del W3C. Además se supone la existencia de soporte para mundos virtuales codificados en VRML97 [6] (bien a través de *plug-ins* del navegador o mediante aplicaciones para la visualización de entornos VRML) y la existencia en la máquina cliente de una máquina virtual de Java 2 [1]. La desviación del cliente de este perfil de usuario típico provocaría una perdida de funcionalidad (más o menos acusada) de la aplicación. La utilización de JavaScript y DOM permite incluir en los documentos presentados al usuario ciertos mecanismos que facilitan su acceso a la información intercambiada. Entre las facilidades

incluidas se encuentran el automaximizado de las ventanas con gran densidad de información, el empleo de menús deslizantes para el acceso a las diversas áreas del sistema de simulación, la actualización del soporte VRML del navegador de la máquina cliente y el bloqueo de los formularios *web* para la solicitud de simulaciones con objeto de evitar bucles de simulación sin respuesta.

4.2 CAPA DE SOPORTE WEB

La misión de esta capa es la de proporcionar un nexo de unión entre la capa de presentación y la capa de cálculo. Realiza tareas de gestión con los datos de entrada (que describen la simulación solicitada por el usuario) y con los archivos que contienen los resultados obtenidos. Asimismo, aporta la flexibilidad necesaria para la definición de los sistemas robóticos y permite superar la rigidez inherente a sistemas basados en el intercambio de documentos HTML estáticos. Además, está encargada de un conjunto de tareas auxiliares como son el control y autentificación de usuarios en el sistema, gestión de estadísticas de acceso, publicación de noticias y la posibilidad de cargar en el servidor mediante el propio interfaz *web* archivos que describan la simulación de interés, evitando tener que rellenar los formularios *web* en simulaciones que involucren un elevado número de parámetros.

El sistema servidor está actualmente instalado en una máquina tipo PC dotada con una CPU K7 Athlon a 1.4 GHz, 512 MBytes de RAM y conectado a Internet mediante una IP "real" a través de la LAN del Dpto. de Ingeniería de Sistemas y Automática. La configuración *software* del servidor está soportada por el sistema operativo Windows 2000 Professional e incluye la presencia del servidor *web* Apache 1.3.23, el módulo para Apache del lenguaje de *script* PHP 4.0.6, el gestor de Base de Datos MySQL 3.23 y la distribución MikTeX de LaTeX para entornos Win32.

La misión del gestor de base de datos es la de proporcionar mecanismos para manejar la información disponible acerca de los usuarios autorizados, de los accesos realizados al sistema y noticias de interés para los usuarios del entorno de simulación.

PHP [5] es un lenguaje de *script* de servidor que permite introducir en la aplicación de simulación HETERO mecanismos para la generación dinámica de contenido *web* en tiempo de ejecución. En particular es de interés la generación de tablas de dimensiones variables, de forma que el cliente dispone así de la suficiente flexibilidad para describir el sistema robótico sin limitarse a un tipo de específico de robots móviles o manipuladores. Entre otros mecanismos aportados por PHP, se incluyen facilidades para la invocación de métodos remotos en el servidor, métodos para la autentificación de usuarios y generación de estadísticas (usando las capacidades del gestor de base de datos) así como la generación de modelos tridimensionales de los robots y su posterior animación de acuerdo con los resultados obtenidos en la simulación.

4.3 CAPA DE CÁLCULO

Como se ha comentado en el apartado 3, las simulaciones solicitadas por el usuario son computadas en el servidor remoto en un entorno MATLAB.

El interfaz entre una aplicación *web* y MATLAB es proporcionado por el *toolbox* "MATLAB Web Server" a través de una conexión con el servidor HTTPD ("demonio" HTTP) mediante un protocolo CGI. El usuario introduce en los campos del formulario *web* de entrada (ver Figura 2) los parámetros de la función MATLAB a invocar (cuyo identificador se indica mediante un campo oculto para el usuario). "MATLAB Web Server" llama a la función de interés que tras el procesamiento de los datos de entrada genera un conjunto de resultados, que generalmente son volcados en archivos textuales o gráficos. Estos resultados constituyen los campos de una estructura de salida que en último término es "mapeada" en un archivo HTML de salida que funciona como plantilla, en el que se incrustan los resultados que han de ser devueltos al usuario. Existen funciones para realizar dicho "mapeo" y volcar las figuras generadas en un archivo gráfico en formato JPEG.

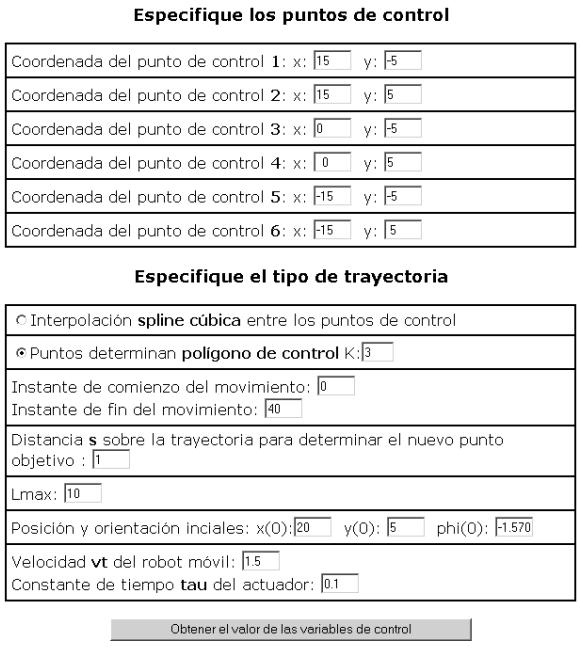

Figura 2: Ejemplo de formulario de entrada.

Salvo la estructura fija de los parámetros de entrada / salida de la función MATLAB invocada, el cuerpo de dicha función sigue la sintaxis habitual de MATLAB, y desde ella es posible llamar a cualquier función de HEMERO para computar la simulación requerida.

4.4 PRESENTACIÓN DE RESULTADOS

Debido al carácter eminentemente pedagógico del sistema de simulación se ha considerado interesante realizar un esfuerzo en la presentación de los resultados al usuario.

4.1.1 Generación de ecuaciones en formato gráfico

Una de las lagunas del lenguaje HTML se encuentra a la hora de mostrar en el navegador cliente expresiones de naturaleza matemática. Aunque se han propuesto estándares como MathML [10], basado en una representación XML de la expresión a mostrar al cliente, su difusión es aún meramente testimonial. Durante la implementación de HETERO, se encontró un método capaz de ofrecer una legibilidad satisfactoria para los resultados. Consiste en volcar la expresión matemática simbólica a un archivo gráfico en formato GIF. Para llegar a esa representación es necesario engarzar una serie de pasos de forma secuencial:

1. Se genera un archivo de texto en formato LaTeX a partir de la expresión matemática que se va a mostrar al cliente mediante el uso de la función MATLAB `matlab2latex`. LaTeX en un paquete de procesamiento y edición de textos muy extendido en la comunidad científica y que presenta elevadas facilidades para la edición de expresiones matemáticas. Esta función reconoce el uso de subíndices y el uso de letras del alfabeto griego en los modelos obtenidos, representándose como tales en la descripción LaTeX.

2. El archivo se compila usando una distribución de LaTeX para entornos Win32, obteniéndose como resultado un archivo PostScript.

3. Del archivo PostScript se extrae la expresión matemática en formato gráfico y se transforma (en tiempo de ejecución) a un determinado formato de archivo gráfico. En concreto se ha elegido el formato GIF por ser la opción que presenta un mejor balance entre compresión del archivo obtenido y legibilidad de la expresión matemática que representa.

En la Figura 3 se puede ver un ejemplo que ilustra el resultado de aplicar este método.

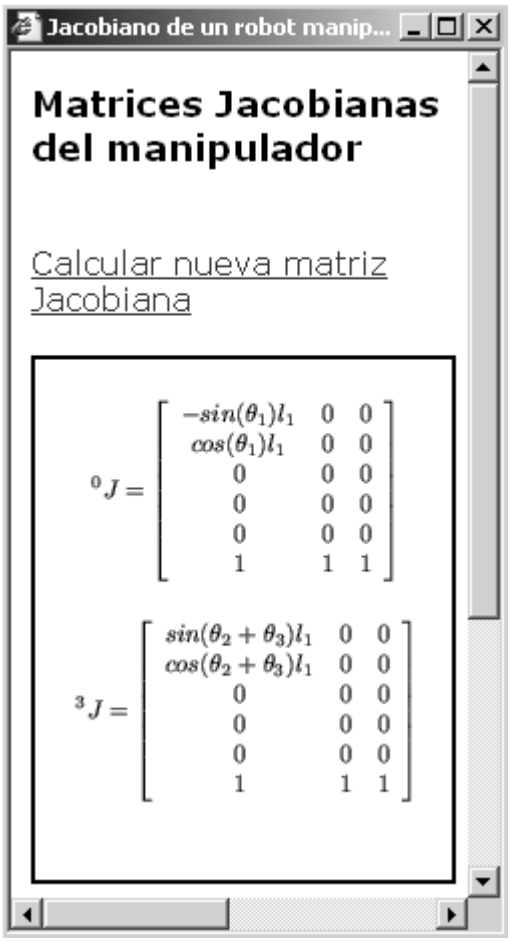

Figura 3: Representación en LaTex de los resultados.

4.1.2 Generación de escenas VRML

La creación de la escena VRML que muestra la animación del sistema robótico simulado por el usuario consta de 2 fases:

1. Un *script* PHP (`dh2VRML`, `creadiferencial` o `creatriciclo`) genera, según los parámetros de la simulación descrita por el usuario, un modelo tridimensional en formato VRML del robot protagonista de la simulación.

2. La simulación del sistema robótico produce un conjunto de secuencias numéricas: la evolución temporal de los valores de las variables articulares (en el caso de que se simule un manipulador robótico), o de las variables de control de la configuración de locomoción utilizada (en el caso de que se simule un robot móvil). Esas secuencias numéricas son empleadas por la función MATLAB `fig2VRML` (basada en la función `VRMLplot`) para animar el modelo tridimensional del robot generado en la primera fase. El usuario dispone de una barra de control para activar la ejecución de la animación, así como un conjunto de vistas predefinidas que le facilitan la visualización de la escena (ver Figura 4).

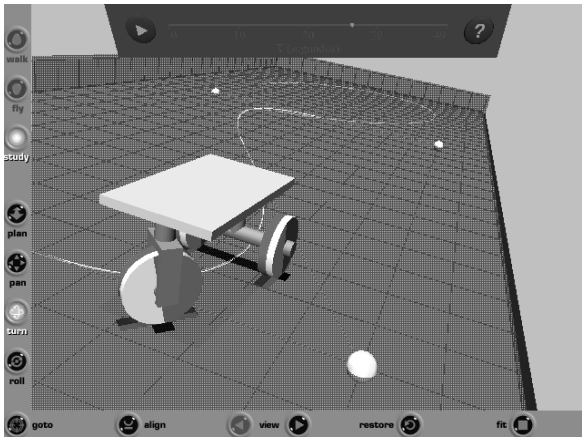

Figura 4: Ejemplo de la representación en VRML.

5 ENTORNOS DE SIMULACIÓN IMPLEMENTADOS

5.1 MANIPULADORES ROBÓTICOS

El actual estado desarrollo de la aplicación incluye una serie de entornos de simulación que permiten al usuario obtener (en [4] se explican en detalle todos estos conceptos):

- Modelo directo y matriz jacobiana del manipulador.
- Representación tridimensional del manipulador a partir de los parámetros Denavit-Hartenberg.
- Modelo dinámico del manipulador.
- Configuración cinemática inversa correspondiente a una trayectoria generada por el propio usuario.
- Trayectorias articulares resultantes de aplicar un conjunto de pares.
- Resultados de la aplicación de la estrategia de control denominada “par computado”.
- Generación de trayectorias mediante el uso de funciones lineales con enlaces parabólicos y de polinomios de orden 3 y 5.

5.2 ROBOTS MÓVILES

Actualmente es posible la obtención de la secuencia de señales de control necesarias para seguir un camino generado por el usuario. Para ello se dispone de la estrategia de seguimiento de persecución pura y eventualmente de los modelos inversos de los robots móviles para el caso en que se quiera ver cual es la postura que adopta el robot para seguir un determinado camino.

El usuario puede generar los caminos de dos formas:

- Definiendo los puntos de paso, en cuyo caso se emplea interpolación mediante *splines* cúbicas.
- Especificando los vértices del polígono de control de la trayectoria, en cuyo caso la trayectoria queda inscrita dentro de dicho polígono.

Se sugiere consultar [4] para tener una referencia de los conceptos teóricos involucrados en las simulaciones correspondientes.

6 APPLET DE SIMULACIÓN DEL MANIPULADOR PUMA 560

El *applet* Java de simulación permite la interacción del usuario con un modelo virtual del robot manipulador PUMA 560. El *applet* aprovecha las capacidades de la plataforma Java 2 (en especial la tecnología de diseño de interfaces de usuario *Swing*) y de la librería de representación de escenas tridimensionales Java3D. Por tanto, es necesario que el navegador de Internet usado por el cliente incluya una máquina virtual de Java 2 (bien de forma nativa como en el caso de Netscape 6 o bien a través de un programa *plug-in*).

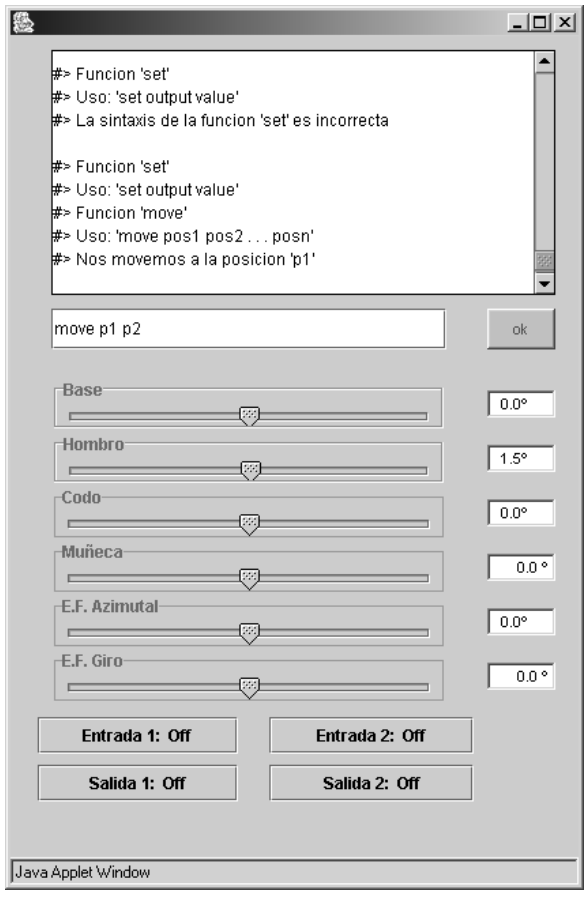

Figura 5: Interfaz para el manejo del Puma 560.

En su estado actual de desarrollo el *applet* presenta al usuario dos ventanas. En la primera se representa la escena virtual tridimensional desde el punto de vista del usuario. Este punto de vista es modificable mediante operaciones de rotación, *zoom* y traslación

mediante el uso conjunto de desplazamientos del ratón sobre la escena y la pulsación de los botones del mismo. La segunda ventana (ver Figura 5) muestra una consola de control que permite al usuario la interacción con el modelo virtual del manipulador. Esta consola incluye controles deslizantes para la manipulación individual de cada articulación del manipulador, una línea de texto para que el usuario teclee instrucciones de control y una ventana de texto donde es posible monitorizar el estado de ejecución de las instrucciones ordenadas por el usuario.

En la actualidad el *applet* reconoce un conjunto muy limitado de instrucciones de control que permiten la definición de posiciones predefinidas (instrucción HERE), el movimiento del manipulador de la posición actual a la posición predefinida utilizada como argumento (instrucción MOVE), eliminar la posición predefinida indicada como argumento (instrucción DELETE), listar en la ventada de texto todas las posiciones predefinidas (instrucción LIST) y, por último, ordenar el movimiento del manipulador a una posición de descanso predefinida (HOME).

7 CONCLUSIONES Y DESARROLLOS FUTUROS

Se ha implementado una aplicación que facilita al usuario un conjunto de servicios *web* que permiten la simulación remota de diversos sistemas robóticos, contemplándose tanto el caso de manipuladores robóticos de base fija como el caso de diferentes configuraciones de locomoción de robots móviles. Además se ha implantado un *applet* Java que permite al usuario la interacción con un modelo virtual tridimensional del manipulador robótico Puma 560.
Futuras líneas de desarrollo contemplan la introducción en la aplicación de nuevos entornos de simulación que abarquen cuestiones no contempladas en el estado actual de desarrollo como planificación de movimientos, control mediante estrategias adaptativas, etc. Asimismo se han pensado diversas mejoras en la interfaz gráfica de la aplicación con el usuario.

En la actualidad está en proceso de desarrollo un completo intérprete del lenguaje de programación de robots VAL-II que permitiría la simulación sobre el modelo virtual tridimensional de programas de control que posteriormente podrían ser transferidos a un manipulador real para su ejecución.

Por último, para aprovechar las ventajas comentadas en la sección 1, se planea en un futuro añadir capacidades de teleoperación al *applet* de simulación del manipulador PUMA 560. Para ello sería necesario implantar un proceso servidor en la computadora conectada al controlador del manipulador que atendiera las peticiones del *applet*-cliente. Para gestionar el diálogo cliente-servidor podrían ser utilizada la tecnología CORBA o RMI, ésta última válida únicamente si ambos procesos estuvieran implantados en Java.

Agradecimientos

Se agradece a Israel Cárdenas Romero la colaboración prestada en el desarrollo del sitio web presentado en este artículo.

Referencias

[1] Hamilton, G., (2000), J2SE™ Merlin Release (JSR 59), Sun Microsystems, Inc.

[2] Maza J.I., Ollero, A., (2001), HEMERO: Herramienta MATLAB/Simulink para el estudio de manipuladores y robots móviles, Marcombo-Boixareu editores (publicada en CD-ROM), ISBN: 84–267–1313–0, España.

[3] Maza, J.I., Ollero, A., (2001), HETERO: Una herramienta para la tele–enseñanza en robótica, EIWISA 2001 (II Jornadas de Trabajo Enseñanza vía Internet/Web de la Ingeniería de Sistemas y Automática), Madrid, España.

[4] Ollero, A., (2001) Robótica: Manipuladores y robots móviles, Marcombo-Boixareu editores, España.

[5] Ratschiller, T., Gerken, T., (2001), Creación de aplicaciones web con PHP 4, Pearson Educación, España.

[6] VRML Consortium, ISO/IEC 14772-1, (1997), The Virtual Reality Modeling Language (VRML97), http://www.vrml.org.

[7] W3C World Wide Web Consortium, (1998), HTML 4.0 Specification, , http://www.w3.org /TR/1998/REC-html40-19980424.

[8] W3C World Wide Web Consortium, (1998), Document Object Model (DOM) Level 1 Specification,http://www.w3.org/TR/1998/REC-DOM -Level-1-19981001.

[9] W3C World Wide Web Consortium, (1999), Hypertext Transfer Protocol – HTTP/1.1 (RFC 2616).

[10] W3C World Wide Web Consortium, (2001), Mathematical Markup Language (MathML) version 2.0, http://www.w3.org/TR/2001/REC-MathML2-20010221.