

# Herramienta de Tele-Enseñanza de Robótica

## Simulador PUMA 560

José Antonio Salinas Vergara  
Gl. Heliopolis N°1, 41012 Sevilla  
[jsalinas4@amena.com](mailto:jsalinas4@amena.com)

### Resumen

*HETERO (Herramienta de Tele-Enseñanza de Robótica) es un proyecto cuyo objetivo es hacer que un usuario remoto pueda conectarse a un servidor para hacer uso de herramientas, aplicaciones y servicios Web que configuren un entorno de simulación sobre problemas relacionados con los manipuladores robóticos y robots móviles.*

*El proyecto se ha estado desarrollando desde el curso 2001-02, y en el último curso 2003-04, se han actualizado las tecnologías usadas tanto para las implementaciones del portal Web (portal que permite el acceso a HETERO) como la del simulador del brazo robótico PUMA 560. Complementándose así esta herramienta de tele-enseñanza, para una mejor visión y aprendizaje por parte de sus usuarios, sobre los problemas robóticos y de simulación que trata este proyecto.*

## 1 Introducción

### 1.1 Objetivos

El proyecto HETERO fue diseñado e implementado para ofrecer una serie de herramientas, aplicaciones y servicios Web que configuran entornos de simulación sobre problemas relacionados con la robótica, concretamente problemas sobre manipuladores robóticos y el estudio de robots móviles. El usuario remoto hará uso de HETERO para:

- ✓ Obtener los modelos cinemáticos directos de configuraciones de manipuladores robóticos especificados por el usuario.
- ✓ Obtener la matriz Jacobiana de un manipulador robótico definido por el usuario.
- ✓ Obtener la configuración cinemática inversa de un manipulador robótico ante una trayectoria especificada por el propio cliente remoto.
- ✓ Obtener modelos dinámicos de manipuladores robóticos en función de los parámetros del manipulador especificado por el usuario.

- ✓ Generación de trayectorias rectilíneas en el espacio cartesiano o parabólicas, polinomios cúbicos o de mayor grado en el espacio articular para manipuladores robóticos.
- ✓ Estudiar el control de manipuladores robóticos mediante el uso de una estimación de su modelo dinámico, en especial la estrategia conocida como par computado.
- ✓ Acceder a mecanismos para la simulación de distintas configuraciones de locomoción típicas de robots móviles, especificando el algoritmo de seguimiento, así como el camino a seguir.
- ✓ Acceder a herramientas para la visualización 3D de las simulaciones de manipuladores robóticos y robots móviles.
- ✓ Acceder a un entorno de simulación para el brazo robótico PUMA 560, que hace uso de un intérprete del lenguaje característico del manipulador denominado VAL-II, para la interacción entre el usuario y el simulador.

Los clientes de HETERO acceden a todas estas herramientas por medio de un portal Web, en donde, se puede elegir el uso de una de estas herramientas. Dicho portal Web estará alojado en un servidor de la facultad y será accedido de forma pública desde los browsers de los PCs clientes (vía Internet o Intranet).

### 1.2 Perfil de usuario

Las herramientas de HETERO se implementaron haciendo uso de los lenguajes de programación mas flexibles y comunes que se pueden encontrar ya instalados en la mayoría de los PCs clientes, y los servicios que ofrece HETERO son aportados de la forma más transparente posible al usuario (ya que existen ciertas capacidades adicionales que tendrán que ser instaladas o actualizadas en el PC del cliente para que se pueda utilizar todo HETERO). El PC cliente de HETERO tendrá las siguientes características mínimas:

- Pantalla con resolución grafica de 1024x768 píxeles y 256 colores.

- Browser compatible con la especificación HTML 4.0 y JavaScript. [2] (se recomienda usar Internet Explorer 6)
- Plug-in Cortona de ParallelGraphics instalado para dar soporte de VRML al Browser.
- Java 2 1.4.2\_04 o superior en su versión SDK o RE.
- JAVA3D 1.3.1 o superior en su versión SDK o RE y para DirectX.
- Una conexión a Internet (si se accede desde fuera de la facultad).

### 1.3 HEMERO

La capacidad de procesamiento necesario para realizar la computación de cada simulación solicitada por el cliente remoto, es proporcionada al usuario a través de la ejecución local en el servidor, de unas funciones de la herramienta MATLAB-Simulink HEMERO (**H**erramienta **M**ATLAB-Simulink para el **E**studio de manipuladores y **R**obots móviles), desarrollada por los profesores (que son también mis tutores de proyecto): J. Iván Maza y Aníbal Ollero [1]. Esta herramienta es un toolbox que combina MATLAB y Simulink, con el fin de permitir efectuar cálculos numéricos y simbólicos relacionados con la cinemática, la dinámica y las estrategias de control de manipuladores robóticos.

Las funciones MATLAB de HEMERO proporcionan métodos para la generación de trayectorias, que pueden ser empleadas para la generación de referencias tanto para manipuladores robóticos (trayectorias en el espacio), como para robots móviles (trayectorias en el plano). Para el estudio de la robótica móvil se pueden especificar diversas configuraciones de locomoción (bicicleta, triciclo, sistema diferencial, configuración omnidireccional) y diversas estrategias de control (persecución pura o métodos basados en la teoría del control). El análisis cinemático y dinámico que proporciona HEMERO, usa la notación descrita en el libro "Introduction to Robotics" [2]. Las funciones que hacen referencia a la cinemática directa de brazos robóticos posibilitan la generación de expresiones simbólicas que definen el modelo directo a partir de una matriz de parámetros Denavit-Hartenberg [3]. También, se proporcionan funciones que posibilitan el cálculo de la matriz Jacobiana, y permite el uso de modelos simbólicos para resolver los modelos cinemáticos inversos.

También ha sido necesario hacer uso de lenguajes de alto nivel como JAVA3D y VRML, para poder representar gráficamente en la maquina cliente las

simulaciones en 3D de los manipuladores robóticos y robots móviles, según las configuraciones cinemáticas establecidas por el cliente.

Las herramientas, ejemplos y simulaciones que ofrece HETERO en su portal Web (ver Figura 1), están basados en ejemplos presentes en el libro de "Robótica: Manipuladores y Robots móviles" escrito por el profesor Aníbal Ollero [4].



Figura 1: Portal Web de HETERO

## 2 HETERO

### 2.1 Interacción entre el cliente y HETERO

El objetivo de HETERO es que el usuario únicamente tenga que estar familiarizado con entornos Web, y los propios conceptos tratados en la Robótica, ya que HETERO solo proporciona un soporte (a modo de ejemplos prácticos y didácticos para aclarar conceptos ya conocidos) a un texto sobre dichos temas. El esquema de aplicación seguido para implementar HETERO fue el de cliente-servidor para conseguir:

- ✓ *Universalidad*, de acceso siendo la configuración software/hardware del cliente irrelevante, siempre que cumpla con el perfil de usuario expuesto anteriormente (HETERO responde al paradigma de sistema abierto).
- ✓ *Independencia*, de la instalación de software propietario y de la adquisición de licencias de terceras empresas.
- ✓ *Protección*, del código de implementación de HETERO a usuarios no deseados.

El cliente rellena formularios Web que permiten definir los parámetros necesarios para que el servidor

pueda generar las simulaciones. Tras la computación remota en el servidor, este devuelve al cliente algún tipo de resultado, típicamente una matriz simbólica en formato gráfico, un gráfico en 2D o una escena 3D, dependiendo de si el conjunto de valores es susceptible de ser representado de una manera u otra.

## 2.2 Configuración del servidor de HETERO

### 2.2.1 Configuración Hardware

La configuración hardware del servidor esta formado por:

- Ordenador con procesador AMD K7 Athlon de 1.4GHz y con 512 MB de RAM.
- Tarjeta de video aceleradora NVidia TNT 2 de 32MB de SDRAM.
- Disco duro de 40GB con tasa de giro 7200 r.p.m.
- Tarjeta Ethernet 10/100 MBps para conexión LAN.

Esta configuración es mas que suficiente partiendo de la hipótesis de que la tasa acceso al servidor será baja (250 visitas al día) y de la exclusividad de uso del servidor como maquina servidora de HETERO.

### 2.2.2 Configuración Software

La configuración software usada por el servidor esta compuesta por:

- Sistema operativo Windows 2000 SP2.
- Servidor HTTP Apache 2.0.49.
- Gestor de BD MySQL 3.23.
- Modulo middleware de PHP 4.3.4.
- MATLAB Web Server bajo MATLAB R11.
- El sistema de procesamiento de texto LaTeX, MikTEX para entornos Win32.
- Paquete de utilidades para el tratamiento de archivos gráficos Image-Magick.
- Java 2 en su versión sdk 1.4.2\_04.
- JAVA3D en su versión DirectX sdk 1.3.1.

Toda esta configuración software es muy necesaria para poder ofrecer al cliente de HETRO los entornos de simulación desarrollados de la mejor forma. Como se puede ver en la lista, se hace uso de software muy

usado actualmente por las empresas para el desarrollo de aplicaciones y entornos Web eficientes.

## 2.3 El menú de opciones del portal Web

La parte izquierda del portal Web (ver Figura 2), contiene:

- Un link llamado “I.S.A.”, que nos lleva a la pagina Web del departamento “Ingeniería de Sistemas y Automática” de la Escuela Superior de Ingenieros (E.S.I.) de la Universidad de Sevilla.
- Un menú de opciones sobre todos los servicios que ofrece HETERO.
- Un link llamado “Inicio” (y toda la imagen del robot bailarín), que nos muestra la página Web de presentación del portal Web.
- Y por ultimo, 3 links que nos llevan a la página Web de la Universidad de Sevilla, a la página Web de la Escuela Superior de Ingenieros (E.S.I.), y a la página Web de la Escuela Técnica Superior de Ingenieros Informáticos (E.T.S.I.I.).



Figura 2: La parte izquierda del portal Web.

El menú de opciones se implemento haciendo uso de PHP y contiene las siguientes opciones:

- *Ejemplos HETERO*, seleccionando esta opción podremos ver los submenús de acceso a las diferentes herramientas y simulaciones que ofrece HETERO, las cuales se comentan en el siguiente apartado.
- *Ayuda*, permite acceder a un submenú en donde se puede obtener información sobre toda la configuración que el cliente de HETERO

necesita para poder hacer uso de todas las herramientas ofrecidas.

- *Noticias*, permite acceder al boletín de noticias que presenta HETERO.
- *Zona privada*, seleccionando esta opción se puede acceder a un menú interno de HETERO, que ofrece mas servicios, pero al cual sólo pueden acceder usuarios registrados en HETERO y el administrador.
- *Sugerencias*, permite escribir y enviar un e-mail al administrador de HETERO.

## 2.4 Entornos de simulación desarrollados

HETERO esta conformado por 6 diferentes áreas de entornos de simulación que hacen referencia a 6 capítulos específicos del libro [4]. Para acceder a cada uno de estos entornos de simulación, se debe elegir en el menú de opciones que ofrece el portal Web (ver apartado 2.3) la opción de “Ejemplos HETERO”. Y en el submenú se podrán encontrar ejemplos sobre los capítulos:

- *Capítulo 3: Representación de la posición y la orientación.*

Se introduce al usuario en el concepto de la transformación de un sistema de referencia. El usuario puede describir la transformación de 2 sistemas de referencia respecto de otro sistema y ver el resultado de dichas transformaciones como una animación en 3D realizado en VRML.

- *Capítulo 4: Modelos cinemáticos de robots.*

El usuario podrá obtener el modelo directo de un robot manipulador a partir de los parámetros Denavit-Hartenberg del manipulador de interés, calculándose las matrices de transformación intermedias  ${}^{i-1}_i T$ . Así mismo, con los mismos datos, podrá realizar el cálculo de la matriz Jacobiana de un manipulador, y generar una representación 3D de dicho manipulador haciendo uso de VRML.

- *Capítulo 5: Modelos dinámicos de robots.*

El modelo dinámico de un robot manipulador junto con las matrices M (matriz de masas del manipulador), G (vector de términos gravitatorios), y V (vector que considera los pares de fuerzas centrífugas), pueden ser obtenidas después de que el usuario defina todos los parámetros dinámicos sobre dicho manipulador, incluyendo los parámetros del modelo cinemático. Además, se puede simular el

comportamiento dinámico del manipulador ante determinados pares definidos por el usuario, obteniéndose la trayectoria articular del manipulador, y representado dichos resultados mediante gráficos bidimensionales (donde se ven las evoluciones de cada variable articular) o mediante una animación en VRML del comportamiento de dicho manipulador.

- *Capítulo 8: Control de las articulaciones de un robot manipulador.*

Se le da al usuario la posibilidad de simular la utilización de la estrategia de control conocida como par computado para un robot manipulador del que se debe indicar un conjunto de parámetros dinámicos, así como, la trayectoria que se desea que siga el manipulador. Los resultados se pueden ver como una escena animada en VRML o mediante gráficos bidimensionales que indican la evolución de cada una de las variables articulares.

- *Capítulo 9: Control de robots móviles.*

Aquí el usuario puede configurar una trayectoria a seguir por un robot móvil, y verificar si realmente dicho robot llega a cumplir con dicha trayectoria especificada o no, haciendo uso de la estrategia de control (para el seguimiento de caminos) “Seguimiento mediante persecución pura”. Los resultados sobre los valores de las variables de control del manipulador son representados mediante graficas bidimensionales en donde se puede ver las evoluciones de dichas variables, así como, una animación 3D en VRML sobre la trayectoria final realizada por el robot.

- *Capítulo 10: Generación de trayectorias.*

El usuario puede simular la orden MOVE disponible en muchos lenguajes de alto nivel orientados a manipuladores robóticos, y así mover el manipulador desde un punto inicial a otro final siguiendo un segmento rectilíneo que los une mediante interpolación lineal. El resultado de dicho movimiento se puede apreciar mediante una animación 3D en VRML, junto con los gráficos bidimensionales de la evolución en el tiempo de las variables articulares del manipulador. Además, también se permite al usuario generar una trayectoria para el manipulador a partir de una secuencia de puntos de control obligados. Y por último, también se puede especificar una trayectoria a seguir por un robot móvil, y obtener las señales de control necesarias (según la configuración de la locomoción empleada) para que el robot móvil

recorra la trayectoria definida (este ejemplo se puede considerar como la implementación de una estrategia de control de bucle abierto y esta muy relacionado con el escenario de simulación de seguimiento de trayectorias de robots móviles por persecución pura).

### 3 Simulador del PUMA 560

#### 3.1 Introducción

HETERO, además de ofrecer todas las herramientas y simulaciones que se han comentado antes, se le agrego también, una plataforma de simulación específica para el manipulador robótico PUMA 560. Este simulador se comenzó a desarrollar desde el comienzo de todo este proyecto, pero no ha sido hasta este año 2004 en que se ha terminado su implementación, y modificación.

El manipulador robótico PUMA 560 es uno de los más usados en las universidades e instituciones para la educación e investigación, por ser buenos para probar conceptos teóricos sobre sistemas, control y algoritmos. Por tanto, este simulador es una muestra de la puesta en práctica de esos conocimientos teóricos, incluyendo los conocimientos necesarios para su diseño e implementación sobre la tecnología JAVA.

#### 3.2 El diseño del simulador

Al igual que el resto de herramientas que ofrece HETERO, ha esta también se le exigían los mismos requerimientos básicos sobre su uso e interacción con el cliente (ver apartado 2.1). Su acceso se realiza a través del portal WEB de HETERO, y la interacción con el simulador ocurre en una ventana nueva y personalizada del browser en el cliente. En esta ventana se ejecuta un applet de JAVA, cuyo código compilado ha sido previamente descargado desde el servidor de HETERO.

El simulador completo consta de 4 módulos o partes:

1. Visualizador 3D.
2. Panel de mandos de control.
3. Consola de ejecución de comandos.
4. Mini panel de utilidades.

##### 3.2.1 Visualizador 3D

El visualizador 3D es una pequeña ventana en donde se puede ver la animación en 3D del estado actual del brazo robótico PUMA 560, cuyo comportamiento estará ligado a los comandos introducidos en la consola (ver apartado 3.2.3) y/o a lo que el usuario realice sobre el panel de mandos (ver apartado 3.2.2). En principio, el diseño geométrico del brazo

robótico, se realizo haciendo uso de una herramienta grafica para entornos 3D, con la cual se generó el diseño geométrico en formato VRML, y gracias a este diseño se pudo desarrollar un nuevo diseño geométrico haciendo uso de JAVA3D.

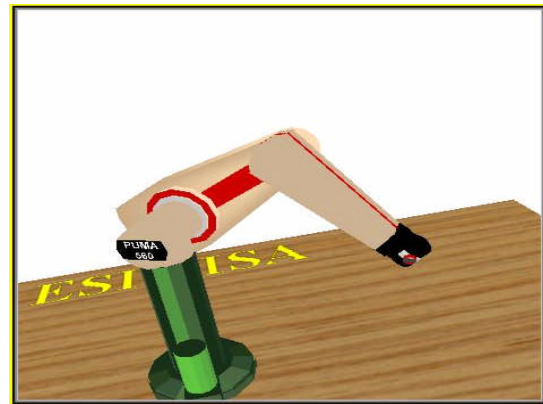


Figura 3: Visualizador 3D del manipulador robótico.

##### 3.2.2 Panel de mandos de control

Este panel de mandos, permite la interacción con el brazo robótico mediante 6 barras de desplazamiento (para girar cada una de las 6 partes que componen el brazo robótico), así como de botones para la conexión/desconexión de la comunicación entre el cliente y el servidor, y la activación de las entradas 1 y 2 del simulador. Además, permite ver el estado de las salidas 1 y 2 del simulador.

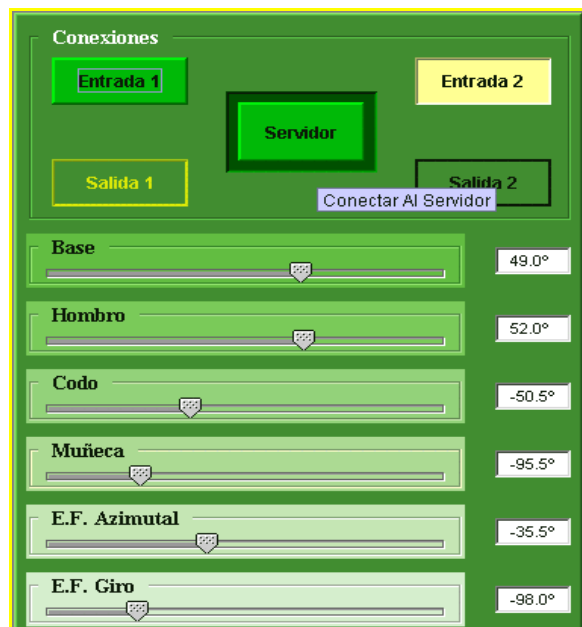


Figura 4: Panel de mandos de control.

Las entradas 1 y 2, son botones cuyo estado cambia entre ON y OFF cuando el usuario pulsa sobre ellos. Las salidas 1 y 2, son botones de visualización de

estado, es decir, cambian de color según el estado ON - OFF en el que estén. Las entradas/salidas 1 y 2 del simulador sirven para poder ser utilizadas en los programas escritos en el lenguaje VAL-II y que se quieren ejecutar en este simulador, dando así más posibilidades de programación.

### 3.2.3 Consola de ejecución de comandos

Esta consola tiene una doble funcionalidad. Es una consola de ejecución de comandos del lenguaje VAL-II, y es una ventana de estado en donde se podrá comprobar el estado de la conexión con el servidor, el estado de la ejecución de comandos, posibles errores, etc...



Figura 5: Consola de ejecución de comandos.

Los comandos del lenguaje VAL-II que se pueden ejecutar en esta consola se muestran en la siguiente Tabla 1.

Tabla 1: Comandos VAL-II disponibles.

Comando	Descripción
<i>Here</i>	Guarda la posición actual del robot con el nombre indicado. (ej. Here pos1)
<i>Delete</i>	Borra la posición cuyo nombre se indica. (ej. Delete pos1)
<i>List</i>	Muestra una lista con todas las posiciones existentes en memoria. (ej. List)
<i>Delay</i>	Realiza una espera de los milisegundos indicados. (ej. Delay 1000)
<i>Speed</i>	Selecciona la velocidad de movimiento del robot (1-100). (ej. Speed 5)
<i>Set</i>	Selecciona el valor de una de las salidas del robot. (ej. Set out1 1)
<i>Home</i>	Mueve el robot a la posición de descanso. (ej. Home)

<i>Move</i>	Mueve el robot a una posición guardada en memoria. (ej. Move pos1)
<i>Movet</i>	Mueve el robot en línea recta a una posición guardada. (ej. Movet pos1)
<i>Appro</i>	Aproxima el robot a una cierta distancia en un eje de una posición guardada en memoria. (ej. Appro pos1 z 10)
<i>Depart</i>	Aleja el robot a una cierta distancia en un eje de la posición en la que este actualmente. (ej. depart x 5)
<i>Read</i>	Carga un fichero de programa en VAL-2. (ej. Read bucles.val)

Como se puede apreciar en la Tabla 1, el último comando permite cargar programas escritos en VAL-II para poder ejecutarlos. Dichos programas pueden contener comandos como los de la tabla, e incluso se permite el control de flujo de sentencias mediante las estructuras especificadas en la Tabla 2.

Tabla 2: Estructuras de control del flujo

Estructura	Descripción
<i>If - Else - EndIf</i>	Estructura de ejecución condicional. (ej. If in1 == 0 HOME Else Set out2 1 EndIf )
<i>For - EndFor</i>	Estructura de ejecución iterativa. (ej. For 1 to 360 by 8 Depart y 10 EndFor )
<i>While - EndWhile</i>	Estructura de ejecución iterativa condicional. (ej. While in2==1 Appro p1 x 4 EndWhile )

Para hacer referencia al estado de las entradas y las salidas del simulador se utilizan las palabras reservadas 'in#' y 'out#', en donde, # puede ser 1 o 2, según la entrada o la salida que se quiera hacer referencia. Los valores que pueden tomar dichas variables in# / out# son 1 si están en ON y 0 si están en OFF. El contenido de los ficheros es un listado (sin delimitadores de línea como en otros lenguajes de programación) de comandos y estructuras de control, que serán luego interpretadas y ejecutadas gracias al Parser del lenguaje VAL-II que esta integrado en el simulador.

### 3.2.4 Mini panel de utilidades

Este mini panel de utilidades que se encuentra justo debajo de la consola (ver Figura 6), esta compuesto por 3 botones que son los siguientes:

- *Cargar*: Sirve para poder cargar en el servidor ficheros que contengan programas escritos en el lenguaje VAL-II y que se quieran ejecutar en el simulador.
- *Ayuda*: Abre una ventana en donde se muestra el documento de ayuda del simulador.
- *Acerca de*: Permite conocer a los desarrolladores y a los tutores de este proyecto, así como, la fecha de la última verificación o corrección realizada sobre la implementación del simulador.

### 3.3 La implementación del simulador

La implementación del visualizador 3D, el panel de mandos de control, y de la consola, se realizo mediante un applet de JAVA. Este applet necesitó tanto para su desarrollo como para su ejecución de que se tuviese instalado JAVA2 SDK\_1.4.2\_04 y la librería de JAVA3D SDK\_1.3.1 (para DirectX). Esta versión de la maquina virtual de JAVA y de la librería de JAVA3D son necesarias porque traen muchas mejoras y correcciones que sus anteriores versiones no tenían y fallaban en conseguirlo, aunque también es posible usar en el futuro, versiones posteriores a estos para ejecutar el applet sin problemas.

Aunque no presento aquí el listado de todas las mejoras que ofrecen las nuevas versiones, si puedo dar como ejemplo que la nueva versión de la maquina virtual tiene corregido la implementación de su Plug-in (el cual ofrece muchos servicios no funcionales como una Cache para los paquetes JAR descargados desde sitios Web y mucho más), y que por ejemplo, la nueva versión de la librería de JAVA3D tiene corregido el cargador de imágenes para usarlas como texturas para las figuras geométricas generadas. Para mas información sobre estas mejoras puede consultar los siguientes sitios Web [5], [6], y la guía [7].

Algunos aspectos a resaltar sobre la implementación del applet del simulador son:

- Se hace uso del Plug-in que trae JAVA para usar la Cache que ofrece (de esta forma el paquete JAR se descarga 1 sola vez desde el servidor), para usar el versionado de paquetes JAR (permite especificar que versión del paquete JAR se debe descargar del servidor, si es que todavía no ha sido descargado), para poder comunicar

parámetros desde la pagina PHP (que carga al applet) hacia el applet, y también para realizar algo muy interesante que se comentará luego en el apartado 3.4.3.

- Todo el código compilado (los ficheros .class) del applet se encuentra almacenado en un solo paquete JAR, listo para ser descargado y ejecutado (llamando a la clase principal que trae el paquete).
- Todo el código del applet se encuentra estructurado en paquetes, cada uno implementa una parte concreta del simulador, y se han usado algunos conceptos sobre patrones de diseño al estructurar y diseñar las clases dentro de los paquetes.
- Para implementar el visualizador 3D, se usó el diseño geométrico en formato VRML del brazo robótico como base para el nuevo diseño geométrico haciendo uso de la librería JAVA3D. Asimismo, esta librería fue usada para poder crear la animación 3D del brazo robótico, y para dar la interactividad entre el panel de mandos y la animación.
- Gracias al botón de conexión con el servidor, el cliente puede comunicarle al servidor todo lo que se ejecuta en el simulador. Para esto, en el servidor deberá estarse ejecutando la aplicación servidora (que ya ha sido implementada), para que reciba todo lo que el cliente le comunica de forma remota.
- El mini panel de utilidades comentado en el apartado 3.2.4, ha sido implementado haciendo uso de HTML, y no de la tecnología JAVA. Por lo que si el applet del simulador no se carga por cualquier razón, este siempre será visible y se podrá usar.

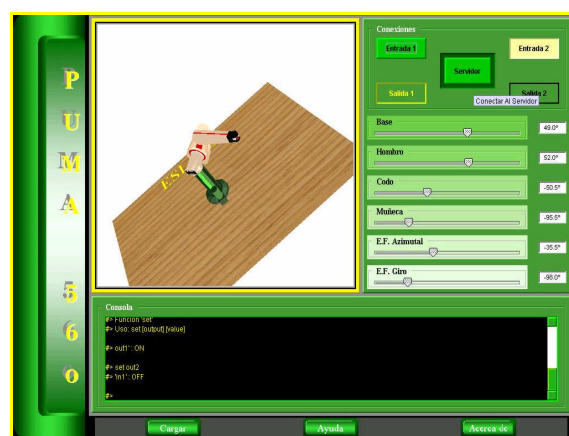


Figura 6: El simulador PUMA560

### 3.4 El cliente del simulador

#### 3.4.1 Acceso privado

El portal Web de HETERO, ofrece también algunos servicios de gestión Web, como son la publicación de noticias, cargador de ficheros para las simulaciones, generación de estadísticas sobre el uso de HETERO, todos estos accesibles desde un menú que sólo se accede al autenticarse como usuario o como administrador, al elegir entrar en la zona privada del portal Web desde el menú de opciones del portal (ver apartado 2.3). Si no se está registrado como usuario se puede dar de alta en la misma página de autenticación de usuario y lógicamente, un administrador tendrá más privilegios que un simple usuario.

Para poder acceder al simulador del manipulador robótico PUMA 560, es necesario, elegir entrar en la zona privada del portal, y autenticarse como usuario o como administrador haciendo uso de su login y password. Una vez autenticado, se podrá acceder a un menú de opciones, en donde se encuentra una opción para poder ejecutar el simulador.

#### 3.4.2 Modos de Simulación

En la página de presentación del simulador, se le recuerda al usuario que debe tener instalado JAVA2 y JAVA3D para poder usar el simulador. De forma opcional, también se le permite al usuario elegir que modo de simulación prefiere ejecutar. Los diferentes modos de simulación sólo se hicieron para demostrar algunas capacidades de animación que permite realizar JAVA3D de forma simple pero que difieren en tiempo de ejecución (es una puesta a prueba de la tecnología usada). De esta forma, se ofrecen 3 diferentes modos de ejecución:

- *Con texturas*, este modo hace uso de 5 imágenes en formato jpg para ser usadas como texturas para las figuras geométricas usadas en la animación 3D (ver Figura 3). Estas imágenes tienen un nombre específico pero pueden ser reemplazadas por otras que tengan el mismo nombre y que se almacenen en la misma carpeta en el servidor.
- *Sin texturas*, como indica el nombre, aquí no se hacen uso de texturas (no se cargan las imágenes jpg), sólo se usan colores sólidos y se pone más en manifiesto las capacidades de iluminación que permite generar JAVA3D en las animaciones (ver Figura 7).
- *Sólo diseño*, este modo permite ver en 3D sólo el diseño geométrico del brazo robótico, el resultado de esta simulación, es lo más parecido

a lo que permiten realizar algunas herramientas gráficas profesionales como el AUTOCAD, o como el 3DStudio (ver Figura 8).

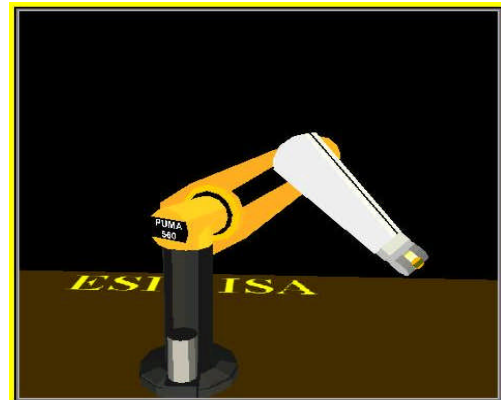


Figura 7: Modo "Sin texturas" del simulador.

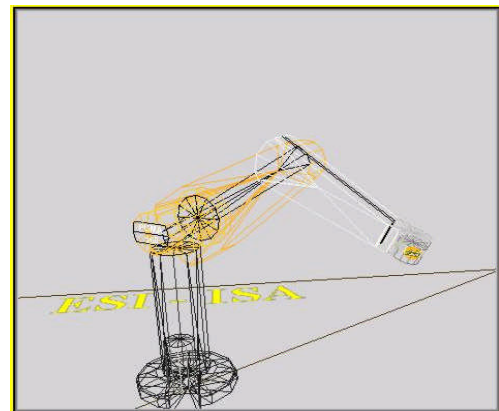


Figura 8: Modo "Sólo diseño" del simulador.

Como se puede intuir los modos de simulación difieren en tiempo de ejecución (de más a menos respectivamente), y otra curiosidad es que, el simulador tarda más tiempo en cargarse la primera vez que se accede a él que en las subsiguientes ejecuciones (esto es, sin cerrar la ventana del portal Web de HETERO, y manteniéndolo en segundo plano).

#### 3.4.3 Ejecución del Simulador

Una vez que el usuario elige el modo de simulación, todos los parámetros necesarios para la ejecución del simulador son pasados desde la página PHP que carga al applet al propio applet (todo esto ocurre de forma transparente para el usuario), y después, se inicia su ejecución, obteniéndose el entorno de simulación ya visto en la Figura 6, con todas las características y capacidades ya comentadas en el apartado 3.2.



Ahora bien, dado que se ha optado por implementar este simulador como un applet de JAVA, el código del applet se tiene que ejecutar en la maquina cliente después de recibir del servidor (de forma transparente al usuario) el paquete JAR que contiene el código compilado del applet. Esto repercute en el objetivo de la independencia de la configuración software del cliente, ya que ahora es necesario que el cliente tenga instalado una versión de la maquina virtual de JAVA, incluyendo una versión de la librería JAVA3D como ya se comento en el apartado 3.3.

Actualmente, la tecnología JAVA ya permite automatizar los procesos de verificación de la versión de la maquina virtual JAVA instalada, al igual que las librerías JAVA instaladas en un PC (gracias a su Plug-in comentado en el apartado 3.3). Además, después de la verificación, permite al cliente decidir si quiere descargarse e instalarse la versión de la maquina virtual JAVA, al igual que las librerías necesarias para poder ejecutar el applet en cuestión.

Por ello, antes de descargarse el paquete JAR que contiene todo el código JAVA compilado del applet desde el servidor, se realiza una verificación sobre la maquina virtual JAVA instalada en el PC del cliente, y se le advierte al cliente cuando es necesario que se descargue y se instale una nueva versión de la maquina virtual, en cuyo caso, el cliente se descargaría del servidor el instalable J2RE\_1.4.2\_04. Sin embargo, no se ha automatizado la verificación sobre si el cliente tiene o no las librerías de JAVA3D instaladas por ser algo redundante (si quieres ver una animación en 3D con JAVA lógicamente necesitas JAVA3D), y por ser mucho mejor que sea el propio cliente el que realice esta verificación y que se descargue las librerías de JAVA3D desde el servidor en caso de necesitarlas (“Y es que tampoco es bueno automatizarlo todo, ya que se pueden generar riesgos, al no conocer del todo al cliente”).

Para realizar estas descargas desde el servidor, se puede elegir en el menú de opciones del portal (ver Figura 2) la opción de “Ayuda”, para acceder a un submenú en donde se da mas información sobre la configuración necesaria para ejecutar el simulador.

## 4 El futuro

HETERO con todo lo que ofrece aún no esta completo y es susceptible a cambiar en el futuro, conforme vayan apareciendo nuevas y mejores tecnologías para el desarrollo de aplicaciones. A continuación, expongo un listado de los posibles cambios que puede tener HETERO en el futuro:

- El simulador PUMA 560 es el primer paso para conseguir la teleoperación del manipulador

robótico PUMA 560 real y de forma remota. La capacidad de comunicación que ya ofrece el simulador a una aplicación servidora ejecutada en el servidor de HETERO, debe permitir realizar esta teleoperación en tiempo real. Por lo que, en el futuro el servidor deberá tener una conexión hardware que le permita comunicar al manipulador robótico todos los comandos (en lenguaje VAL-II) que el cliente remoto le envíe para que sean realizados por el manipulador.

- Posiblemente, también se extienda aun más el Parser que tiene implementado el simulador del PUMA 560, para que se interpreten instrucciones más complejas y nuevas.
- Se podría pensar en extender el número de robots manipuladores y móviles reales que se pueden teleoperar desde HETERO. Empezando primero por crear los simuladores correspondientes a cada uno de estos (como se ha hecho con el manipulador robótico PUMA 560), y finalizando con la conexión hardware del servidor a cada uno de estos robots para la comunicación de las ordenes que el cliente remoto quiere que realicen los robots.
- Además, como medida de seguridad, también podría ser deseable que todo lo que ofrezca HETERO (entornos de simulación, herramientas, etc.) tenga como garantía de autenticidad, una firma digital propia del departamento de Ingeniería de Sistemas y Automática (es decir que todo este firmado digitalmente), por el mismo hecho de que HETERO es accedido de forma pública.
- Lo anterior, sería fácil de hacer, si todos los servicios que ofrece HETERO se hubiesen implementado haciendo uso de JAVA, ya que el Plug-in de JAVA te permite gestionar firmas digitales de forma sencilla, aunque lógicamente, la firma digital se tiene que conseguir mediante otros medios.

Por esto y por mucho más que ofrece JAVA; en el futuro, quizás se decida cambiar de implementación de una parte de HETERO, y aprovechar la tecnología que proporciona JAVA. En este cambio de implementación se podrían hacer uso de las páginas JSP (JAVA Server Pages, páginas Web con etiquetas especiales y código JAVA incrustado, visítese la pagina Web de Sun [8]) y los Servlets (programas de tipo CGI escritos en JAVA, que se ejecutan en el servidor, recibiendo peticiones y generando a partir de ellas una página Web), que permiten ambos hacer uso del lenguaje JAVA para poder crear páginas Web dinámicas ejecutadas desde el servidor, que es parecido a lo que se consigue con PHP y otros CGI's,

solo que se tienen aún más ventajas. Por ejemplo, los Servlets tienen las siguientes ventajas:

- *Eficiencia*, ya que con los CGI tradicionales, cada petición HTTP da comienzo a un nuevo proceso, con lo que si el programa CGI es rápido, el comienzo del proceso se pierde en tiempo de ejecución. Además, si el programa CGI tiene N peticiones simultaneas, entonces el código del programa CGI es copiado N veces en memoria. Ninguno de estos problemas ocurre con los Servlets.
- *Capacidad*, porque permite realizar operaciones que son complejas o imposibles para otros CGI. Por ejemplo, los Servlets pueden comunicarse directamente con el servidor Web, comunicarse con otros Servlets, y mantener la información de peticiones pasadas para ser luego usadas mas adelante.
- *Portables*, estos están escritos en JAVA y siguen la especificación del API de JAVA estandarizado, lo cual permite que puedan ser soportados de forma directa por los servidores Web o mediante el uso de plugins.
- *Coste*, hay muchos servidores que son gratis o que son costosos, y aun así, solo han sido desarrollados para ser usados de forma personal y para mantener sitios Web de poca envergadura. Gracias a la portabilidad que tienen los Servlets, el coste de uso es menor, e inclusive existen ya buenos servidores totalmente gratis como el servidor de Servlets TomCat.

Y el uso de JSPs tiene las siguientes ventajas:

- Permite generar páginas dinámicas haciendo uso del lenguaje de programación JAVA (que es un lenguaje de programación muy popular), combinado con HMTL.
- JavaScript no se puede comparar con JSP, porque JavaScript te permite crear páginas dinámicas en el lado del cliente y no en el lado del servidor (no tiene acceso a los recursos que están en el servidor). Además, JavaScript no permite acceder a los datos del formulario rellenado por el cliente.
- Las paginas JSP son más portables que las páginas ASP, que sólo soportan a sistemas realizados por Microsoft.
- Aunque se podría también usar los Servlets para hacer lo mismo que hace un JSP, la modificación y programación de las páginas JSP es más simple que la de un programa Servlet.

Para mas información sobre JSP y Servlet pueden visitarse las paginas Web [9] y [10]. Por ultimo, es posible que se realicen otros cambios menores para mejorar el entorno de funcionamiento de HETERO, como cambiar el sistema operativo (para tener un servidor neto), el AntiVirus, el FireWall, etc., que quizás se realice pronto, según el departamento I.S.A. de la E.S.I.

## Agradecimientos

*Agradezco a toda mi familia por siempre animarme a que siga adelante y a que nunca me rinda, así como por todo el soporte que me dan para mi educación. Y agradezco a mis tutores de proyecto por darme la oportunidad de formar parte de este gran proyecto, que viene muy acorde con la temática que me ha atraído desde pequeño: los robots.*

## Referencias

- [1] Maza, J.I., Ollero, A. Herramienta MATLAB-Simulink para la simulación y control de robots manipuladores y móviles. Actas de las XXI Jornadas de Automática. Sevilla, 2001.
- [2] Craig, J.J. Introduction to Robotics. Second edition. Addison Wesley, 1989.
- [3] Denavit, J., Hartenberg, R.S. A kinematic notation for lower-pair mechanism based on matrixes. Journal of Applied Mechanics. Junio, 1955. pp. 215-221.
- [4] Ollero, A. Robótica: Manipuladores y robots móviles. Marcombo-Boixerau editores. 2001.
- [5] Página Web de JAVA 2.  
<http://java.sun.com/j2se/index.jsp>
- [6] Página Web de JAVA3D.  
<http://java.sun.com/products/java-media/3D/index.jsp>
- [7] Guía del JAVA Plug-in 1.4.2.  
<http://java.sun.com/j2se/1.4/pdf/plugin-dev-guide-1.4.2.pdf>
- [8] Página Web de Sun para JSP  
<http://java.sun.com/products/jsp/>
- [9] Tutorial sobre Servlet y JSP (en inglés)  
<http://www.apl.jhu.edu/~hall/java/Servlet-Tutorial/>
- [10] Tutorial sobre JSP (en español)  
<http://www.acm.org/crossroads/espanol/xrds8-2/servletsProgramming.html>