

Sistema para el control remoto de un robot mediante un navegador de Internet (I)

Este artículo describe el sistema de control remoto de un robot construido en el laboratorio del Instituto de Investigación Tecnológica de la Universidad Pontificia Comillas. El robot lleva un ordenador personal a bordo con capacidad para recibir las señales de un conjunto de sensores y para enviar órdenes a los distintos actuadores. Se decidió diseñar un sistema de comunicaciones con objeto de conocer en tiempo real los valores de los sensores y para poder solicitar actuaciones.

Álvaro Sánchez Miralles

Ingeniero del I.C.A.I. especialidad electrónica. Pertenece al área de Sistemas Inteligentes del Instituto de Investigación Tecnológica de la UPCO. Áreas de trabajo: Robótica móvil, Comunicaciones y Control wireless, Sistemas de Diagnóstico y Modelado de sistemas no lineales.



Rafael Palacios Hielscher

Ingeniero Industrial del I.C.A.I. Es profesor del departamento de Electrónica y Automática e Investigador del área de sistemas inteligentes del IIT, en la Escuela Técnica Superior de Ingeniería (I.C.A.I.). Doctor Ingeniero Industrial por la Universidad Pontificia Comillas en 1998. Áreas de trabajo: sistemas de monitorización y diagnóstico, análisis de señal y tratamiento digital de imágenes.



En lugar de desarrollar un programa cerrado para la comunicación directa con el robot, se quiso dar mayor flexibilidad y disminuir el tiempo de programación utilizando un navegador de internet como interfaz de usuario. En este artículo se describe tanto la interfaz como el flujo de información entre el robot y cualquier usuario de internet. También se proporciona una dirección de internet donde cualquier persona puede acceder a la aplicación, en modo de simulación.

Introducción

Internet es una tecnología evolutiva que permite conectar electrónicamente lugares alejados. Hasta hace poco tiempo, las redes informáticas servían para el intercambio de datos, fundamentalmente correo electrónico y transferencia de archivos. Sin embargo las nuevas tecnologías y estándares de comunicación han permitido desarrollar sistemas de información dinámica y en tiempo real de imagen, voz y datos. El mejor ejemplo de imagen en tiempo real son las cámaras web (*webcam*), que envían imágenes por la red y se actualizan de forma continua en el navegador. Con un micrófono y un altavoz, conectados al ordenador, se puede charlar en directo con alguien al otro lado del planeta al precio de una llamada local. También es posible consultar en tiempo real los valores de las acciones de la bolsa.

Para sacar total provecho de redes como Internet, sería muy

deseable tener una interacción física real, es decir, que además de ser una herramienta de comunicación, una red puede ser otra manera de actuar a distancia.

En este artículo se va a describir un sistema de actuación remota de un robot autónomo, desarrollado en el Instituto de Investigación Tecnológica (I.I.T.) de la Universidad Pontificia Comillas. Para poder controlar el robot a distancia se ha diseñado una interfaz gráfica que permite el envío de órdenes de actuación y la lectura clara de las medidas de los sensores. Para evitar que se saturen los canales de comunicación ha sido necesario diseñar un sistema que se apoya en la utilización de un servidor que hace de pasarela de todo el flujo de datos.

La interfaz es una página web de control, accesible para cualquier usuario de internet (acceso protegido por contraseña). Con ello se consigue la máxima flexibilidad ya que el robot puede ser monitorizado o manipulado desde cualquier tipo de ordenador conectado a la red. La página web está programada en Java, el lenguaje de desarrollo de aplicaciones de redes soportado en la mayoría de los navegadores y que garantiza la compatibilidad.

Descripción del robot

El robot fue diseñado con objeto de desarrollar algoritmos experimentales que permitan un funcionamiento autó-

nomo. Se trata por tanto de un AGV (*Auto Guided Vehicle*) que toma sus propias decisiones de movimiento y actuación en función del conocimiento de su entorno y basado en ciertos algoritmos.

El robot está formado por un conjunto de sensores y unos circuitos de actuación, el fundamental es el circuito de control de los motores. Recientemente se ha instalado a bordo la placa madre de un ordenador tipo PC, con un disco duro y un conjunto de baterías de alimentación. Todos los sensores y circuitos de actuación se conectan a una placa que se comunica con el PC por medio del puerto paralelo [1]. El PC lee la información de todos los sensores, decide el movimiento que debe realizar y envía las señales a los actuadores.

Utilizar un PC estándar ofrece la máxima flexibilidad para el desarrollo de algoritmos de control sofisticados.

Posteriormente se pensó en la posibilidad de diseñar un sistema de comunicación para su-

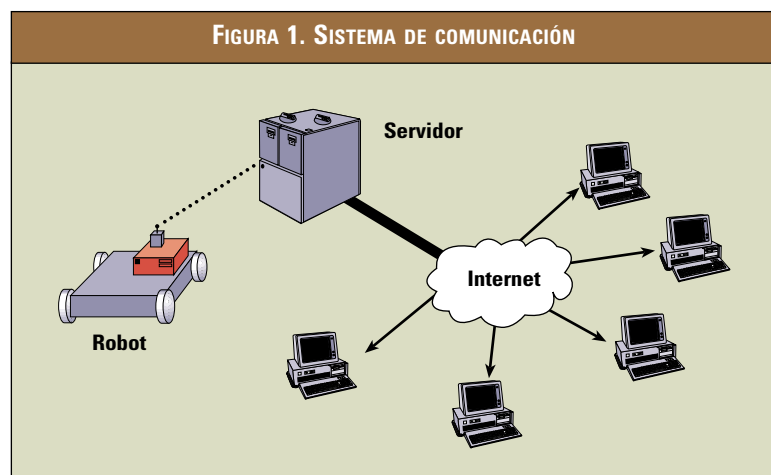
pervisar el funcionamiento del robot, de los algoritmos de control e incluso para imponer ciertas actuaciones o movimientos con prioridad sobre los algoritmos de autoguiado. El mecanismo de comunicación debe estar basado en un sistema de radio para evitar la utilización de cables que interfieran con el movimiento del vehículo.

Sistema de comunicación

Para poder controlar el robot desde cualquier lugar del mundo se ha optado por introducirlo en Internet como se describe a continuación.

Comunicación con el robot

El PC de control funciona bajo sistema operativo Linux y comunica con la red mediante un enlace sin cables. El PC de

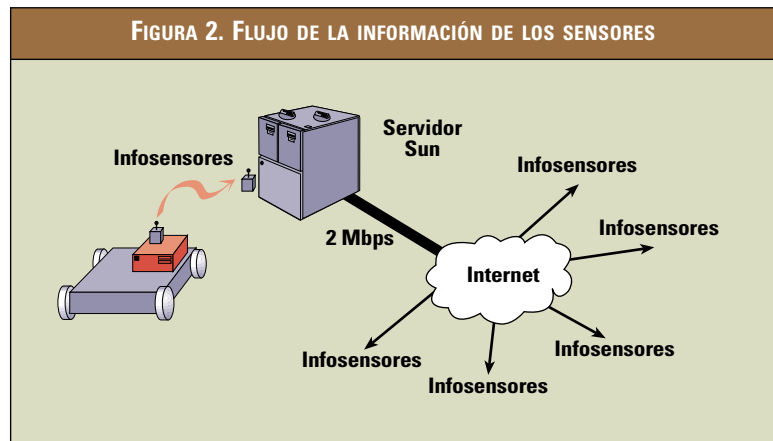


control está en comunicación directa por radio con un ordenador denominado servidor central. Generalmente las comunicaciones por radio tienen un ancho de banda pequeño y son lentas, por tanto es necesario minimizar el tráfico de información del robot.

Para solucionar el problema del ancho de banda se ha diseñado el esquema de comunicación que se ilustra en la Figura 1, que se basa en la utilización de una estación de trabajo Sun que hace de servidor.

El servidor está conectado a Internet mediante una línea de 2 Mbps y controla todo el flujo de información entre el robot y el exterior. Cualquier ordenador puede conectarse con el servidor para consultar los datos de los sensores, pero el servidor es el único ordenador que se comunica directamente con el robot. Esto disminuye al mínimo el tráfico del enlace por radio, que es el más débil. Por ejemplo, la información de los sensores del robot se envía al servidor una sola vez cada cierto tiempo. El servidor almacena esta información y se la ofrece a todos los ordenadores de Internet que la soliciten (el flujo de información se muestra en la Figura 2).

La comunicación entre el robot y el servidor puede realizarse por distintos medios siempre que no utilicen cable: comunicación directa por radio, por infrarrojos, comunicación indirecta. Finalmente se decidió utilizar, por sencillez, un modem GMS de manera que el robot se conecta por radio con un prove-



edor de internet. Sin embargo esta conexión sólo se utiliza para comunicarse con el servidor pero el robot no actúa como servidor *web* y no admite información directa de otros ordenadores de la red.

Flujo de información en el servidor

En lugar de escribir un programa cerrado para el servidor, se decidió desarrollar una interfaz *web* que permitiese controlar el robot desde cualquier ordenador. En este desarrollo se ha diseñado una página *web* y unos mecanismos de envío y recepción de mensajes del robot.

Cualquier documento *web* debe cumplir los siguientes requisitos:

- Máxima compatibilidad
- Eficiencia del lado del servidor
- Funcionalidad y facilidad de manejo por cualquier persona.

Para ello se ha optado por utilizar código HTML y *applets* Ja-

va [2][3][4]. El código HTML describe las páginas *web* con el texto, los gráficos y los enlaces a otras páginas. Los *applets* son pequeños programas en código Java que se encuentran en el servidor, pero se ejecutan en el navegador del usuario. Java es un lenguaje de programación de alto nivel orientado a objetos independiente de la máquina en la que se ejecute, por lo que se convierte en un candidato ideal para aplicaciones *web* dado que los navegadores pueden ejecutar el código sin problemas de compatibilidad.

Para manejar la información de las actuaciones se ha optado por la tecnología CGI (*Common Gateway Interface*) [5][6] dentro del servidor. CGI permite a una página *web* lanzar un programa independiente que se ejecuta en el servidor para llevar a cabo cualquier tipo de tarea. Esta técnica está muy extendida y probada en Internet, además su implantación en sistemas Unix es muy sencilla y no requiere ninguna formación especial. Por estas razones se ha elegido la tecnología CGI frente a otras posibilidades como ASP (*Active Server Page*) [7] o *Servlet* [8].

La página para controlar realmente el robot está protegida por contraseña, de manera que sólo las personas involucradas en los proyectos de vehículos autoguiados del IIT tiene acceso al control del robot. Sin embargo un ejemplo de la misma página está accesible en modo de simulación en la siguiente página:

<http://www.iit.upco.es/~palacios/robot>

Envío de órdenes al robot

La petición de actuaciones se realiza a través de la página *web*. Para mandar una orden al robot, el usuario tiene que pulsar un botón u otros componente de la página. Haciendo eso, el *applet* Java llama a otra página que ejecuta un programa CGI (*Common Gateway Interface*). Por ejemplo, al seleccionar el sonido Sound5 se genera una llamada a la página de control de acciones:



.../control.shtml?Sound5

Como parte de la dirección se indica el tipo de acción a realizar, que será interpretando como un argumento según las especificaciones CGI [5][6]. El contenido de la página de control es:

```
<!--#exec cgi="robot.cgi" -->
```

Esta línea llama al programa `robot.cgi`, que es un *script* de

FIGURA 3. TRES ACCIONES DEL PROGRAMA CGI

```
#!/bin/sh
echo "Content-type: text/plain"
echo ""
echo $QUERY_STRING > action.dat
echo "\n"
rcp actions.dat robot:/home/robot

case $QUERY_STRING in
    . . .
    "Sound2"*) echo "\"Sound nº2\" sent to robot" ;;
    "LedsAnim3"*) echo "\"Leds Animation nº3\" sent to robot" ;;
    "goleft"*) echo "\"The robot goes to the left\" sent to robot" ;;
    . . .
    *)
esac
```

unix, y le pasa como argumento el comando Sound5.

El programa CGI ejecuta tres acciones secuenciales, ver Figura 3.

- La primera acción que va a llevar en cabo es escribir la orden dentro del archivo `actions.dat`. La variable `$QUERY_STRING` permite al CGI leer qué hay después del signo de interrogación de la dirección *web*. (Sound5 en el ejemplo anterior).
- Para mandar esta orden al robot, el programa CGI utiliza un *remote copy* (`rcp`) [9] que sirve para copiar el archivo `actions.dat` sobre el disco duro del robot. De esta manera, el robot recibe la orden que el usuario ha mandado.
- La tercera tarea del programa es escribir un mensaje sobre la pantalla del usuario, para informar que la orden ha sido enviada al robot.

A continuación se muestra la estructura del contenido del archivo

```
~palacios/public_html/robot
/robot.cgi
```

Al otro lado del sistema de comunicación, en el PC del robot, un programa escrito en lenguaje C está siempre a la escucha de una nueva orden. Cuando aparece un archivo `actions.dat` se ejecuta la orden y se borra el archivo. Así se realiza una comunicación CGI entre el usuario y el servidor, y una comunicación `rcp` entre el servidor y el robot como se muestra en la Figura 4.

Lectura de las medidas de los sensores

Cuando un usuario se conecte a la dirección `http://www.iit.upco.es/~palacios/robot` su navegador descargará una página *web* con un *applet*. Los programas Java ne-

cesarios para el funcionamiento del *applet* están guardados en el servidor. Cuando el navegador del usuario accede a la página, se baja este código y empieza a correrlo de forma local.

En el código Java del *applet* existe un método para visualizar el contenido de un archivo que contiene la lectura de los sensores. Este método se ejecuta cada 5 segundos, por lo tanto la información de los sensores se actualiza automáticamente en el navegador del usuario.

El archivo con los valores de los sensores se llama *sensors.txt* y reside en el servidor. El PC del robot se encarga de actualizar este archivo, mediante un *remote copy*, cada vez que lee el estado de los sensores (ver Figura 5). [a](#)

(Continuará en el próximo número)

FIGURA 4. FUNCIONAMIENTO DEL ENVÍO DE UNA ORDEN AL ROBOT

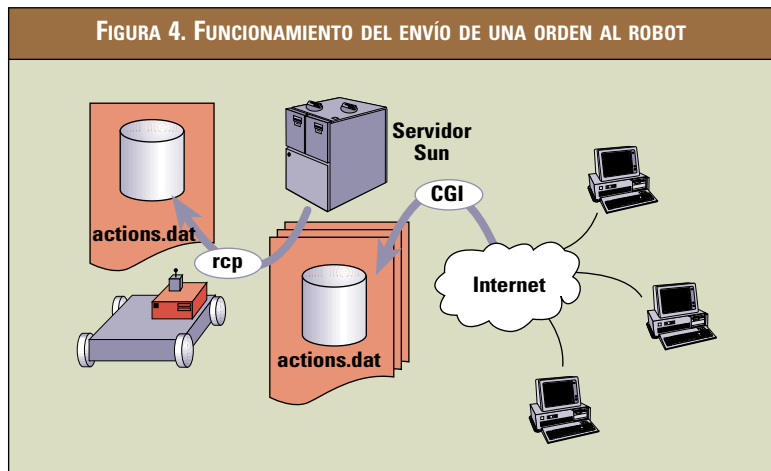
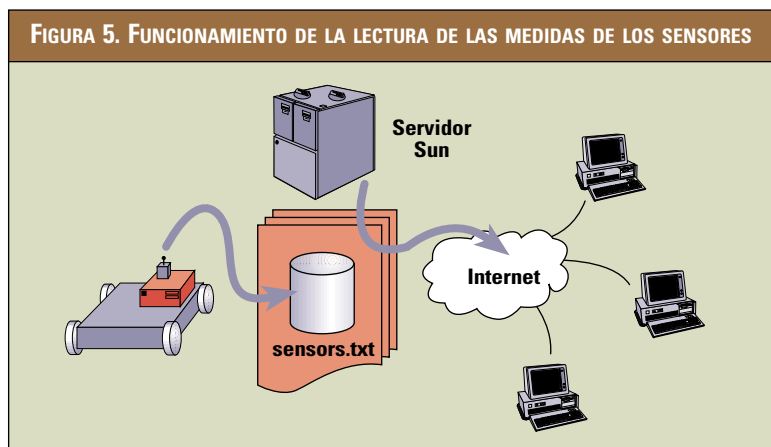


FIGURA 5. FUNCIONAMIENTO DE LA LECTURA DE LAS MEDIDAS DE LOS SENSORES



Bibliografía

- [1] Álvaro Sánchez Miralles, Rafael Palacios Hielscher. Diseño de un vehículo autoguiado controlado mediante un ordenador personal. UPCO-IIT 2000.
- [2] David Flanagan. JAVA in a nutshell. A Desktop Quick Reference. Ed. O'Reilly. 1997.
- [3] David Flanagan. JAVA Examples in a nutshell. A Tutorial Companion to Java in a Nutshell. Ed. O'Reilly. 1997.
- [4] Jean-Francois Macary, Cédric Nicola. Programación Java. Ed. Eyrolles. 1996
- [5] M. Felton CGI Internet Programming with C++ and C. Prentice Hall 1997
- [6] Shishir Gundavaran. CGI programming on the world wide web. O'Reilly 1996.
- [7] Jesús Bobadilla. HTML dinámico, ASP y JavaScript. Marcas comerciales RA-MA, 1999
- [8] Jason Hunter, William Crawford. JAVA Servlet programming. O'Reilly 1998.
- [9] Arnold Robbins. Unix in a Nutshell: System V Edition. O'Reilly, 3rd Edition 1999.