

# PLATAFORMA DISTRIBUIDA PARA LA REALIZACIÓN DE PRÁCTICAS DE ROBÓTICA MÓVIL A TRAVÉS DE INTERNET

Luis Payá, Arturo Gil, Oscar Reinoso, Luis M. Jiménez, Mónica Ballesta  
 Departamento de Ingeniería de Sistemas Industriales  
 Universidad Miguel Hernández. Avda. de la Universidad s/n  
 Ed. Torreblanca. 03202 Elche (Alicante) España  
 {lpaya, arturo.gil, o.reinoso, luis.jimenez}@umh.es

## Resumen

*Actualmente, Internet se ha convertido en una potente herramienta aplicable a todos los campos de educación en ingeniería. Este artículo presenta una plataforma distribuida que permite a los alumnos acceder a los robots disponibles en el laboratorio a través de Internet. Mediante esta plataforma, se crea un entorno remoto con el que los alumnos pueden llevar a cabo diferentes experimentos sobre varios equipos, con un horario totalmente flexible. De este modo, los usuarios pueden crear algoritmos básicos de control reactivo y testarlos sobre plataformas robóticas reales. Actualmente, hay tres tipos de robots disponibles. El principal objetivo del trabajo consiste en crear una plataforma común, que permita el acceso a todos ellos utilizando una interfaz de comunicación común, de un modo totalmente transparente para el usuario.*

**Palabras Clave:** Robótica móvil, colaboración entre robots, control distribuido, laboratorio remoto, CORBA.

## 1 INTRODUCCIÓN

La formación experimental de los estudiantes juega un importante papel en la educación en ingeniería. De este modo, se deben proporcionar, a través de los laboratorios, todos los recursos necesarios que permitan al estudiante poner en práctica todo el conocimiento que ha adquirido durante el estudio teórico de las diferentes materias. Tradicionalmente, el elemento que hace posible este aprendizaje es el laboratorio, donde el estudiante puede realizar los experimentos necesarios para conocer los instrumentos y equipos que encontrará durante su vida laboral. Sin embargo, el laboratorio tradicional presenta una serie de carencias que dificultan el proceso de aprendizaje del alumno; requiere la presencia de los estudiantes y profesores en un horario predeterminado y el tiempo disponible para las sesiones prácticas está limitado. Además, el coste de establecimiento y mantenimiento del laboratorio

suele ser alto, con lo cual, el contenido de las clases prácticas estará fuertemente condicionado por el número de equipos disponibles.

El uso de técnicas basadas en Internet supone varias ventajas, solucionando los problemas antes expuestos. Los estudiantes pueden acceder a los laboratorios en un horario totalmente libre, desde su propia casa, y disponiendo de todo el tiempo que requieran para alcanzar los objetivos de cada sesión práctica. Además, los estudiantes pueden acceder individualmente a los equipos, independientemente del número disponible de ellos. También, utilizando un sistema de evaluación online, el estudiante puede conocer los resultados de su evaluación en tiempo real y el profesor puede tener en cuenta no sólo los resultados finales sino el trabajo que el alumno ha llevado a cabo realmente para conseguir dichos resultados. Este sistema permitiría incluso que varios centros compartan los equipos de prácticas y por tanto, los gastos asociados [1]. Además, los estudiantes, están altamente motivados para hacer uso de los recursos disponibles en entornos remotos a través de Internet [11].

Este artículo presenta una plataforma que ha sido desarrollada con el objetivo de que los alumnos puedan hacer uso de los robots móviles disponibles en el laboratorio a través de Internet. De este modo, se hará posible la realización de las sesiones prácticas con estos equipos fuera de los horarios preestablecidos y desde la localización que prefieran. Hasta el momento, se han realizado varios trabajos que permiten el uso de un equipo a través de Internet. Dentro del área de 'Educación en Control', se han desarrollado diferentes laboratorios remotos. [3] presenta un laboratorio remoto para control en automática. Este laboratorio permite al usuario diseñar su propio controlador por medio del entorno MATLAB/Simulink y testarlo a través de una interfaz. En [7] se presenta una estructura que permite laboratorios Web centrados en el aprendizaje de sistemas de control. RECOLAB [4] es otro laboratorio remoto similar en el cual se utiliza un servomotor para testar el controlador diseñado por los alumnos en un entorno remoto. Por otro lado,

también han sido desarrollados varios laboratorios remotos con el objetivo de manejar robots en el entorno remoto. En [2] se presenta un laboratorio virtual para realización de prácticas en robótica que permite tanto la simulación de un brazo robot como la tele-operación del robot real equivalente. En [12] se describe una arquitectura cliente/servidor para el control remoto de un robot manipulador, haciendo uso de programación en Java utilizando TCP y sockets y accediendo al servidor usando cualquier navegador web. [5] presenta la arquitectura 'Developer' para la teleoperación y control del robot B21r. Por último, [10] presenta y discute varias aplicaciones que manejan robots móviles a través de Internet en EPFL, en Lausanne.

El aspecto novedoso de la plataforma propuesta en este trabajo consiste en que se dispone de un equipo heterogéneo de diferentes robots móviles que el alumno debe poder utilizar para testar los algoritmos desarrollados. Por otro lado, también se debe permitir la interacción de un robot con el resto de ellos llevando a cabo una monitorización de todos ellos mientras se está ejecutando una tarea. Cada robot debe poder interactuar con los algoritmos que están corriendo en el resto de robots activos en la plataforma.

El resto del artículo se estructura del siguiente modo. En la siguiente sección, se describe la plataforma desarrollada para la comunicación con los diferentes tipos de robots. La sección 3 muestra como todos estos conceptos han sido aplicados a un tipo determinado de robot, el WiFiBot. En la sección 4 se describe el uso del sistema desde el punto de vista del usuario final. Para finalizar, se exponen los resultados y conclusiones del trabajo.

## 2 ARQUITECTURA DEL SISTEMA

### 2.1 AGENTES MÓVILES

La arquitectura de comunicaciones que se ha planteado tiene como objetivo monitorizar un grupo heterogéneo de robots móviles [8]. En concreto, se dispone de cuatro modelos diferentes de robots móviles, cada uno con características y capacidades sensoriales diferentes. Algunos de estos robots se muestran en la figura 1.

- B21r es un robot de 4 ruedas y cinemática síncrona utilizado fundamentalmente para tareas de investigación. Lleva dos computadores a bordo con el sistema operativo Linux y el software Mobility (librerías basadas en CORBA 2.0 para comunicación con el robot). Dicha comunicación se realiza mediante un enlace ethernet inalámbrico. El robot lleva varios sensores: sonar, infrarrojos, láser y

encoders. Además, lleva en su parte superior una unidad pan-tilt con dos cámaras Sony XC999 paralelas.

- Robots WifiBot: Cuentan con una cámara color DCS 900 y dos sensores infrarrojos de distancia. A bordo, están equipados con un procesador x86 AMD a 20 MHz encargado de controlar sus 4 motores de forma independiente. Además, se ha instalado un PC a bordo, con procesador pentium III funcionando con sistema operativo Linux. Estos robots se comunican con el exterior mediante una comunicación WIFI 802.11b/g.

- Robots EyeBot: Estos robots funcionan con un Microcontrolador Motorola 68332 a 25 MHz y cuentan con 1 MB de memoria RAM. Pueden capturar imágenes en color mediante una cámara con la que pueden extraer una representación simplificada del entorno por el que se están moviendo. También pueden obtener medidas de distancia utilizando dos sensores infrarrojos. Estos robots se comunican utilizando transceivers de Radiometrix (BIM).

- Robots RugWarrior: Estos robots cuentan con un microcontrolador Motorola 68HC11. Disponen de sensores infrarrojos, con los que detectan la presencia de obstáculos cercanos. Tienen unas capacidades sensoriales muy limitadas.

Cuando se necesitan las medidas de los sensores de un determinado robot, el sistema es capaz de identificar el modelo de robot con el que se quiere establecer una comunicación y leer los datos que este proporciona.



Fig. 1: Robots utilizados en la plataforma. B21r, WiFiBot y EyeBot.

### 2.2 ARQUITECTURA DE LA RED DE COMUNICACIONES

Debido a las características heterogéneas del conjunto de robots móviles, se ha precisado el diseño de una arquitectura de red que integre diferentes

tecnologías de comunicación vía radio y diferentes protocolos de red. La arquitectura se muestra en la figura 2. Se estructura en tres subredes:

- 1) Red cableada Fast Ethernet a la que se conectan los computadores, servidores, routers y cámaras fijas. A esta red se conectará la herramienta de teleoperación que permitirá gestionar todos los agentes móviles del entorno remoto.
- 2) Red WIFI 802.11b/g que comunica los robots de tipo medio (iRobot B21r, y WifiBot).
- 3) Red radio diseñada e implementada específicamente para comunicar los robots de menor capacidad (Eyebot, RugWarrior) utilizando transceivers de Radiometrix (BIM) en la banda UHF de 433MHz.

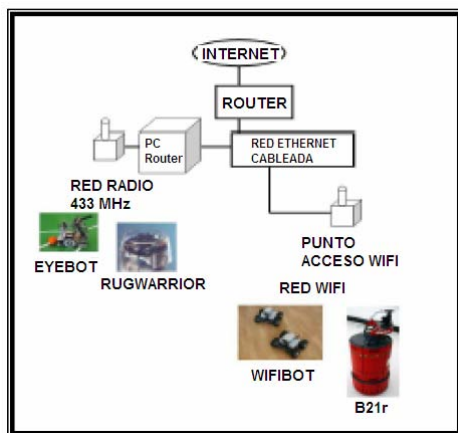


Figura 2: Arquitectura general del sistema de comunicaciones

Las subredes 1 y 2 utilizan protocolos estandarizados de red y enlace (TCP/IP) y sobre ellos se construye el protocolo de control de los robots. El robot iRobot utiliza adicionalmente una interfaz de programación CORBA para la lectura de datos de sensores y control de actuadores.

La subred 3 ha sido diseñada e implementada a partir de los transceivers de Radiometrix BIM y BIM-2 utilizando una interfaz RS-232 tanto en los robots móviles Eyebot y RugWarrior Pro, como en el PC que actúa de router con el resto de la red. Se ha propuesto un protocolo de enlace con paso de testigo y control centralizado que evite las colisiones y permita la comunicación directa entre robots. Adicionalmente se dispone de un servicio de enrutamiento entre la red TCP/IP y la red vía radio.

### 2.3 PROTOCOLOS DE COMUNICACIÓN

Se ha diseñado un protocolo específico de nivel aplicación orientado al control distribuido de robots

móviles para la realización de tareas cooperativas. En la especificación del mismo se ha tenido en cuenta la posibilidad de trabajar de forma transparente con los diferentes dispositivos involucrados en este tipo de aplicaciones: minirobots con capacidad de procesamiento limitada, grandes robots móviles con CPUs embarcadas, cámaras fijas y móviles, computadores externos, servidores de datos, etc.

El carácter heterogéneo de los elementos utilizados ha precisado la utilización de diferentes interfaces de programación pero manteniendo un protocolo común de aplicación. La figura 3 muestra la arquitectura básica diseñada. Los dispositivos más complejos que pueden conectarse a través de la red WIFI o Ethernet mediante TCP/IP utilizan para la comunicación de datos, dependiendo del dispositivo, la librería de sockets o bien una librería que da soporte al estándar CORBA. La arquitectura CORBA proporciona una metodología orientada a objetos bien definida para la implementación de aplicaciones distribuidas y constituye el modelo de referencia utilizado en el diseño general. El estándar CORBA ha sido utilizado para tareas similares de comunicación, por ejemplo en [13], demostrando ser eficiente para intercambiar información en equipos de robots heterogéneos de la liga RoboCup F2000. Ciertos dispositivos están limitados y la arquitectura CORBA es demasiado compleja y lenta, por lo que se dispone de una interfaz adicional con la librería de sockets y la interfaz de enlace de la red radio.

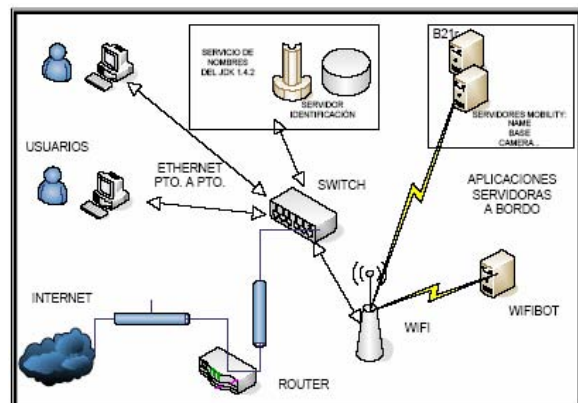


Figura 3: Arquitectura básica diseñada

CORBA (Control Object Request Broker Architecture) [6] proporciona una metodología orientada a objetos en aplicaciones distribuidas. Constituye un modelo de referencia que se ha utilizado para el diseño general del sistema. Dos características fundamentales de CORBA son:

- a) Separación entre interfaz e implementación: Las interfaces de los objetos CORBA se especifican en un lenguaje especialmente pensado para este propósito. Este lenguaje, llamado IDL (*Interface*

*Definition Language*), es parte del estándar CORBA. IDL aísla los servicios que proporciona el objeto de su implementación, ofreciendo una mayor portabilidad e interoperabilidad.

- b) Independencia de la localización del objeto. Un objeto CORBA ofrece una serie de servicios a través de sus operaciones y atributos. Estos servicios pueden ser accedidos de forma transparente por los clientes.

Los métodos de los objetos CORBA pueden ser implementados en lenguajes de programación diferentes, adaptándose a las necesidades del dispositivo en el que se ejecutan.

En nuestro caso, el número de elementos que se encuentran activos en el entorno es variable. Es decir, en un instante dado es posible que uno de los robots móviles se encuentre explorando el entorno y las observaciones que realiza están disponibles y podrán ser consultadas. Por otra parte, también es posible que alguno de los robots móviles deje de funcionar, y, en consecuencia no proporciona ningún servicio a los clientes del sistema. Teniendo en cuenta estas cuestiones se ha diseñado un Servidor de Identidad. Se trata de un servidor encargado de mantener una base de datos actualizada de los dispositivos (agentes) presentes en la red y los recursos que proporciona cada uno de ellos (sensores, actuadores, procesamiento...). De este modo cuando un agente (robot, PC, cámara...) se incorpora a la red, debe solicitar el alta en la base de datos proporcionando los datos de identificación, direccionamiento y los recursos disponibles. Cuando un agente desea utilizar algún recurso distribuido debe realizar una petición al gestor para obtener el estado e información del resto de agentes existentes en la red.

A bordo de cada robot se encuentra un programa servidor CORBA que se encarga de dirigir las peticiones de los clientes a los recursos con que cuenta el robot (movimiento de motores, lectura de datos de imágenes, etc).

Por otra parte, existe un módulo de monitorización/supervisión que centraliza la información recogida por todos los robots. Así, se recogen parámetros, como, por ejemplo, el estado de cada robot, las lecturas de sus sensores, la trayectoria planificada en el entorno o la tarea que está llevando a cabo. Este módulo actúa como cliente CORBA de todos los elementos existentes en el sistema, haciendo peticiones de forma periódica a cada uno de ellos. Además, el módulo de monitorización o supervisión se encarga del control de la misión del equipo de robots, de esta manera, gestiona el reparto de tareas entre el grupo de robots móviles. En la figura 7 se muestra la pantalla principal del

monitorizador. Éste permite visualizar los robots que se encuentran en el entorno, así como la tarea que están llevando a cabo. Este módulo se ha desarrollado en lenguaje Java, hecho que nos permite ejecutarlo en diferentes máquinas, independientemente del sistema operativo.

## 3 APLICACIÓN AL ROBOT WIFIBOT

### 3.1 DESCRIPCIÓN DEL SISTEMA

El WiFiBot es un robot de bajo coste caracterizado por su flexibilidad, que permite su uso en diferentes entornos y situaciones. Actualmente, existen dos modelos de WiFiBot, el SC y el 4G. El primero incluye un controlador BECK (SC12), con un sistema operativo en tiempo real y varias interfaces de comunicación; Ethernet, RS232, I2C y acceso WiFi a través de un bridge inalámbrico ASUS 330. El modelo 4G tiene un controlador más avanzado, el 'Access Cube' de 4G-Systems (Procesador MIPS AMD, 400 MHz, 64 MB RAM y 32 MB memoria flash). Incluye puertos RS232, USB, I2C, Ethernet y Wlan, de modo que no es necesario un bridge inalámbrico independiente. Ambos modelos poseen una cámara Ethernet, con movimientos pan-tilt en el modelo 4G, dos sensores infrarrojos situados en la parte frontal, con un alcance de 1.2 metros y cuatro encoders ópticos con 300 sectores/vuelta que permiten la medida de la velocidad de cada rueda de forma independiente. Para extender sus posibilidades, se ha incluido en ambos modelos una tarjeta PC 104+ (P4M 1.4 GHz, 512 MB RAM, 2 GB Flash Disk), con el sistema operativo Linux Debian. El uso de estos robots debe ser totalmente transparente para los estudiantes en el entorno remoto. De este modo, se han desarrollado una serie de clases que permiten el uso de ambos de la misma forma a pesar de sus diferencias.

### 3.2 SERVICIOS IMPLEMENTADOS

**Servidor.** Junto con el robot, se proporciona un programa servidor para el controlador SC12 y una aplicación cliente para Windows que actúa como interfaz para monitorizar y controlar el robot. Con el objetivo de incluir la cinemática y la odometría, se han realizado ciertos cambios sobre el servidor. Se han incluido funciones para el cálculo de cinemática y odometría, y se ha incorporado un temporizador para actualizar la odometría. Además, el protocolo ha sido cambiado a modo texto de manera que pueda ser utilizado por cualquier cliente telnet para el manejo del robot. Además, como los sensores infrarrojos no presentan un comportamiento lineal, se ha

implementado una nueva función 'getir' que devuelve la distancia ya linealizada.

**Cliente.** Se ha desarrollado una clase cliente en C++ para que la comunicación sea transparente de cara al usuario, quien únicamente tiene que desarrollar el algoritmo para controlar el robot independientemente del tipo de robot. A través de esta clase, el usuario puede programar cualquiera de las funciones que el servidor ofrece.

Por otra parte, el robot incluye una cámara que proporciona imágenes jpg 640x480. Esta cámara incorpora un servidor al que debe conectarse el cliente para realizar la adquisición de la imagen. El cliente debe crear un socket con la IP y puerto requeridos y enviar una solicitud a la cámara, usando el protocolo HTTP. Esto consistirá en una solicitud GET, seguida por el nombre el fichero requerido, el protocolo y dos caracteres de retorno. Dado que la cámara requiere autenticación, se debe añadir una línea con el login y el password codificados en base-64. Para realizar todas estas tareas, se ha implementado una segunda clase cliente, que incluye un thread independiente encargado de la petición y recepción de la última imagen capturada.

**Servidor puente multi-hilo.** Este servidor se ejecuta en la tarjeta PC montada en el robot, y permite que varios programas accedan simultáneamente a los servicios que proporciona el SC12. Ha sido implementado usando el mismo protocolo que el SC12 de modo que la misma clase cliente puede comunicarse con el servidor del controlador y con el servidor puente, especificando únicamente la IP y puerto correspondientes. De hecho, el servidor puente hace uso de esta clase para enlazar con el SC12. La ventaja de este servidor multihilo es que varios clientes pueden acceder a los servicios a bajo nivel que el robot ofrece. Por ejemplo, podrían estar ejecutándose simultáneamente el programa que controla el robot junto con un programa para monitorizar su evolución. Sin embargo, cuando un cliente está controlando el robot, el resto de clientes que se conecten únicamente pueden monitorizarlo.

Este servidor establece un enlace con el SC12 usando la clase cliente y se deja un socket a la escucha a la espera de posibles clientes. Cuando un nuevo cliente accede al sistema, se crea un thread independiente para atender a este cliente, como se muestra en la fig. 5. Los diferentes threads se ejecutan en paralelo, compartiendo el mismo espacio en memoria. Para evitar posibles errores de comunicación con el SC12, que podrían producirse cuando varios clientes realizan una petición simultáneamente, debe utilizarse un mecanismo de sincronización. Para la construcción de este servidor, se ha utilizado un mecanismo de exclusión mutua (mutex), que bloquea

los fragmentos de código donde se realizan las peticiones al SC12, de modo que únicamente un thread puede estar ejecutando este fragmento de código al mismo tiempo. El resto de threads son bloqueados hasta que el primero de ellos finaliza el proceso.

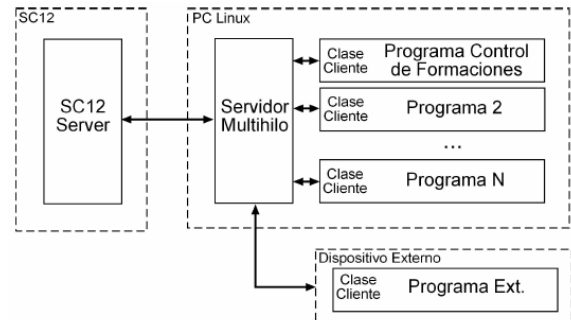


Figura 5: Funcionamiento del servidor multihilo

Para concluir, la figura 6 muestra el modelo de capas desarrollado para la implementación de la plataforma de comunicación basada en CORBA sobre el WiFiBot.

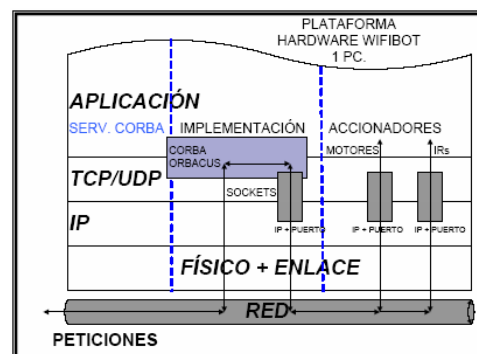


Figura 6: Modelo de capas implementado en el WiFiBot

## 4 USO DEL SISTEMA

Para permitir el acceso remoto a los robots se ha desarrollado una aplicación Java GUI usando el modelo MVD. La interfaz gráfica ha sido implementada usando las clases del paquete javax.swing. El sistema puede generar eventos basándose en tres causas diferentes; cuando el cliente interactúa con los componentes gráficos, a través de un timer que sincroniza los procesos y mediante un thread que, realizando peticiones CORBA al servidor de identificación, controla la presencia de los robots en el sistema. Esta aplicación actúa como un cliente CORBA, de modo que es capaz de requerir servicios a los actuadores y sensores de los robots.

Cuando la aplicación cliente comienza su ejecución, el cliente debe configurar los datos necesarios para el

servidor de identificación (IP y puerto del servidor de nombres e intervalo de tiempo para la sincronización de las peticiones). Dicho servidor realiza una inspección del sistema, en busca de los equipos disponibles, y tras ello, se muestra un árbol que contiene los robots disponibles y los servicios que ofrecen. Además, se asigna a los robots disponibles una interfaz gráfica en caso de que exista. Si no se encuentra ningún robot disponible, la aplicación queda a la escucha de una nueva conexión. La figura 7 muestra la apariencia global de la aplicación. En este caso se muestran los paneles correspondientes a los robots WiFiBot y B21r, que son los robots activos en el sistema en ese momento.

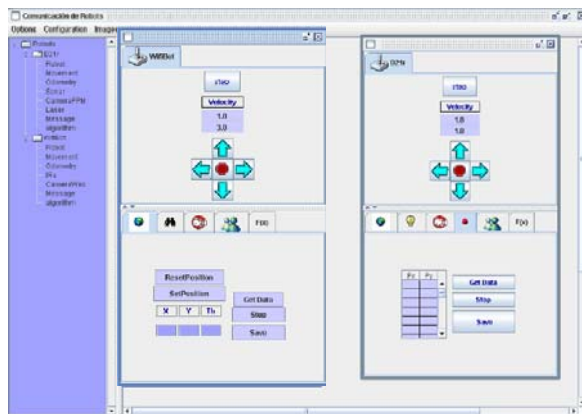


Figura 7. Aplicación para comunicación con los robots

En cuanto a la interfaz gráfica correspondiente al robot WiFiBot, en la parte superior de la ventana se muestran diversos controles para mover el robot manualmente; con los botones de las flechas, las velocidades lineal y angular del robot pueden ser incrementadas o decrementadas, y el botón central detiene al robot. El incremento es configurable. Además, se muestran las velocidades lineal y angular actuales. En la parte inferior de la ventana, hay un componente `javax.swing.JTabbedPane` con varias solapas. Estas solapas dan acceso a los siguientes paneles:

**Panel odometría:** El usuario puede resetear la odometría del robot e inicializarla a un valor predeterminado. Además, se posibilita el ir leyendo y almacenando continuamente el valor de la odometría en un fichero a elección del usuario.

**Panel Infrarrojos.** Se muestran las distancias frente a cada uno de los sensores infrarrojos y dichas distancias también se pueden ir almacenando continuamente en un fichero.

**Panel Cámara Web.** El usuario puede visualizar las imágenes capturadas por la cámara Web. Cuando se pulsa el botón 'Stop', la cámara deja de capturar y la

última imagen tomada se puede almacenar en un fichero jpg.

**Panel Algoritmo.** A través de este panel, el usuario puede cargar el programa que ha desarrollado para controlar el robot de manera que pueda comprobar su funcionamiento. En primer lugar, el usuario tiene que buscar el fichero fuente C++ a enviar. Una vez lo tiene localizado, pulsando el botón 'Compile', el fichero es transferido y compilado en el servidor remoto (el usuario no posee las librerías necesarias para compilarlo en su PC), y aparece una ventana que muestra al cliente los resultados en la compilación. Si no se ha detectado ningún error en el proceso, al pulsar el botón 'Run', el algoritmo cargado comenzará su ejecución. En cualquier momento, el usuario puede detener esta ejecución y parar el robot usando el botón 'Stop'.

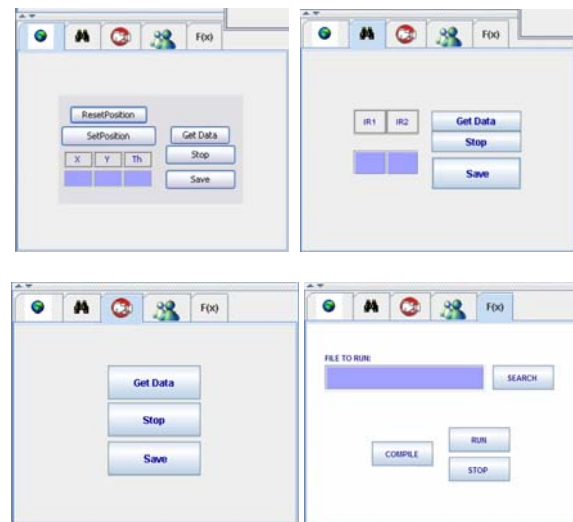


Fig.8. Paneles para monitorización y control de WiFiBot. (a) Posición, (b) Sensores Infrarrojos, (c) Captura de imagen y (d) Descarga de un nuevo programa al robot

Actualmente, el sistema descrito se encuentra a disposición de los estudiantes para que se puedan conectar durante un periodo de tiempo preestablecido. El estudiante necesita una validación para acceder al sistema. De este modo, los estudiantes pueden construir algoritmos de control reactivo básico y probarlos sobre los robots disponibles. Leyendo los valores proporcionados por los sensores y la cámara, y teniendo en cuenta la cinemática de cada robot, el estudiante debe programar su comportamiento de manera que realice una tarea concreta (por ejemplo, el seguir una ruta pregrabada). Durante la navegación del robot, todos los datos relativos a su evolución pueden ser almacenados en varios ficheros de texto de manera que el estudiante, al final del proceso, será capaz de

obtener gráficamente, la evolución de las variables necesarias (trayectoria, velocidad, etc.).

Actualmente, hay tres robots disponibles en el laboratorio (WiFiBot, B21r y Eyebot). Teniendo en cuenta el hecho de que la plataforma desarrollada permite controlarlos de una forma común y transparente para el usuario, y que además, dicha plataforma permite la comunicación entre robots, el estudiante también tendrá la posibilidad de programar algoritmos para llevar a cabo tareas que supongan colaboración entre los diferentes robots del equipo (por ejemplo, mantenimiento de formaciones).

## 5 RESULTADOS ALCANZADOS

Los servicios descritos en los apartados anteriores han permitido diseñar un nuevo conjunto de prácticas y experimentos a realizar por parte de alumnos de la titulación de Ingeniería Industrial en la Universidad Miguel Hernández de Elche. En concreto y dentro de la asignatura de Control de Robots y Sistemas Sensoriales se han venido utilizando durante el último curso académico este tipo de servicios lo que ha posibilitado la realización de dos nuevas prácticas a distancia por parte de los alumnos.

Para el acceso a las prácticas vía Internet, el alumno únicamente debe registrarse y validarse dentro de la base datos que el sistema remoto dispone para darle permiso de acceso. Una vez validado el alumno puede realizar las dos sesiones de prácticas diseñadas hasta el momento. Mientras un usuario se encuentra con el uso a distancia de uno de los dispositivos móviles que se ubican en el laboratorio otros posibles usuarios pueden hacer lectura de todos los servicios que proporcionan estos dispositivos y descritos con anterioridad pero no es posible que comanden el sistema físico. De esta manera se previene que diferentes usuarios interactúen al mismo tiempo con un mismo sistema físico. Por último a cada usuario se le da permiso de acceso al Wifibot durante un periodo de tiempo determinado de forma que no restrinja el uso de este dispositivo a los posibles usuarios en espera. Este tiempo es función del número de peticiones que hay en ese momento para la conexión o en acceso de lectura.

La realización de prácticas a distancia permite que los alumnos puedan realizar este tipo de experimentos en cualquier momento del día y cualquier día que los dispositivos físicos se encuentren operativos, incluidos los fines de semana sin atender a un horario previamente planificado a principios de curso. En el gráfico de la figura 9 se observan los datos acerca del número de horas promedio por sesión dedicado por parte de los

alumnos en cada una de las dos prácticas diseñadas y configuradas. Como se observa, el valor promedio del número de horas es muy superior al de realización de estos alumnos en el laboratorio de forma presencial durante el curso previo. Esto se justifica dado que los alumnos se encuentran más predispuestos a la realización de las prácticas sin atender en demasía al tiempo que le dedican.

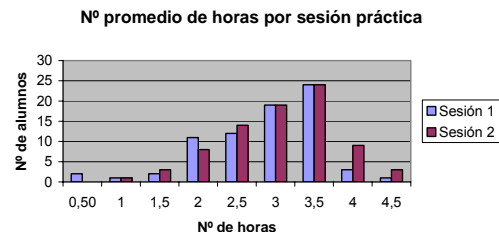


Figura 9: Número promedio de horas dedicadas por parte de los alumnos a cada una de las sesiones

En el gráfico de la figura 10 se observa el número de sesiones de conexión que han realizado los alumnos para completar una práctica. Esto permite evidenciar que los alumnos (en promedio) prefieren realizar la práctica en diferentes intervalos de tiempo en lugar de dedicarle un tiempo excesivamente largo. Obviamente estos datos pueden verse alterados en función del número de conexiones realizadas a la vez dado que el sistema acota el tiempo de acceso al usuario en estas condiciones.

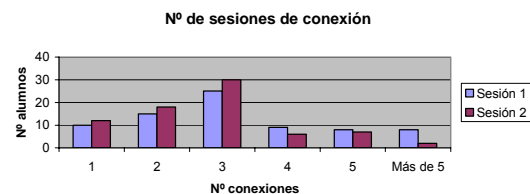


Figura 10.- Número de sesiones de conexión realizadas por los alumnos en cada práctica.

Si bien no es posible constatar a largo plazo la ventaja de realizar prácticas a distancia por parte de los alumnos dado que únicamente se disponen de datos acerca del último curso académico, sí que se observa una mayor predisposición por parte de los alumnos para la realización de las prácticas docentes. Además las encuestas periódicas realizadas a los estudiantes muestran la alta predisposición por parte de los alumnos a la realización de este tipo de prácticas a distancia.

Asimismo, la plataforma implementada ha sido utilizada en una tarea de mantenimiento de formaciones en un equipo de varios robot WiFiBot, basándose entre comportamientos. Para esta tarea,

que implica colaboración entre robots, la plataforma ha resultado altamente eficiente, proporcionando muy buenos resultados [9].

## 6 CONCLUSIONES

En este trabajo se ha presentado una plataforma distribuida para la comunicación con y entre los miembros de un equipo heterogéneo de robots. Esta plataforma permite la monitorización y control de estos robots, de forma remota, a través de Internet, y de forma totalmente transparente para el usuario, que maneja todos los robots usando la misma interfaz gráfica. Además, se posibilita la realización de tareas que impliquen colaboración entre robots, lo cual requiere que se comparta información entre los miembros del equipo.

El sistema implementado esta siendo utilizado actualmente para la realización de prácticas de robótica móvil por parte de estudiantes de ingeniería. Los estudiantes pueden probar los algoritmos que diseñan sobre equipos reales y puede acceder a todos los servicios que ofrecen los robots, aprovechando además las ventajas que tiene la educación a través de Internet.

### Agradecimientos

Este trabajo ha sido subvencionado por el Ministerio de Educación y Ciencia a través del proyecto DPI2004-07433-C02-01 'Herramientas de Teleoperación Colaborativa. Aplicación al Control Cooperativo de Robots', y por el proyecto PCT-G54016977-2005 'Robots Cooperativos para la vigilancia e inspección de edificios e instalaciones industriales'

### Referencias

- [1] Berger, K. A. and Topol, M. T. (2001). Technology to enhance learning. Use of a web site platform in traditional classes and distance learning. *Marketing education review.*, vol. 11, pp 15-26
- [2] Candelas, F. A., Puente, S. R., Torres, F., Ortiz, F. Gil, P., Pomares, J. (2003). A virtual laboratory for teaching robotics. *International Journal of Engineering Education*, vol. 19, No 3, pp. 363-370
- [3] Casini, M., Prattichizzo, D., Vicino, A. (2003). The automatic control telelab: a user-friendly interface for distance learning. *IEEE Transactions on Education*. vol. 46, No 2. pp. 252-257
- [4] Jiménez, L. M., Puerto, R., Reinoso, O., Fernández, C., Neco, R. (2005). RECOLAB: Laboratorio remoto de control utilizando Matlab y Simulink. *Revista Iberoamericana de Automática e Informática Industrial*, vol. 2, No 2, pp 64-72
- [5] Khamis, A., Rivero, D. M., Rodríguez, F., Salichs, M. (2003). Pattern-based architecture for building mobile robotics remote laboratories. *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 3, pp 3284-3289
- [6] OMG, Object Management Group. (1995). Common Object Request Broker: Architecture and Specification. Revision 2.0.
- [7] Pastor, R., Sánchez, J., Dormido, S. (2003). A XMLFramework for the Development of Web-based Laboratories focused on Control Systems Education. *International Journal of Engineering Education*, vol. 19, No 3. pp. 445-454
- [8] Payá, L., Gil, A., Reinoso, O., Riera, L., Jiménez, L.M. (2006). Distributed platform for the control of the WiFiBot robot through Internet. *ACE 2006. 7<sup>th</sup> IFAC Symposium on Advances in Control Education*
- [9] Payá, L., Juliá, M., Reinoso, O., Gil, A., Jiménez, L.M. (2006). Behaviour-based multi-robot formations using computer vision. Aceptado para publicación en *6<sup>th</sup> IASTED International Conference on Visualization, Imaging and Image Processing*
- [10] Siegwart, R. and Saucy, P. (1999). Interacting mobile robots on the Web. *Workshop Proceedings of the IEEE International Conference on Robotics and Automation*
- [11] Stafford, R. F.. (2005). Understanding motivations for Internet use in distance education. *IEEE Transactions on Education*, vol. 49, No 2. pp. 301-306
- [12] Thamma, R., Huang, L. H., Lou, S. J., Diez, C. R. (2004). Controlling robot through Internet using Java. *Journal of Industrial Technology*, vol. 20, No 3
- [13] Utz, H., Stulp, F., Moeld, A. (2004). Sharing belief in teams of heterogeneous robot. *Proc. of RoboCup-2004 symposium*, pp. 508-515