

# SERVICIOS IMPLEMENTADOS EN EL ROBOT WIFIBOT PARA SU USO A TRAVÉS DE INTERNET

Oscar Reinoso, Luis Payá, Miguel Juliá, Luis M. Jiménez, Ramón Neco

Universidad Miguel Hernández (Elche, España)  
Departamento de Ingeniería de Sistemas Industriales  
e-mail o.reinoso@umh.es

## **Resumen**

*El presente trabajo recoge las características principales del robot WifiBot como elemento de apoyo para el aprendizaje de robótica en diferentes asignaturas de enseñanzas técnicas. En este sentido, se describe, en primer lugar, tanto el hardware que constituye el sistema, como los diferentes módulos software que se han desarrollado sobre el mismo para dotarlo de funcionalidad. El objetivo final del sistema desarrollado es el empleo para la realización de prácticas docentes a través de Internet por parte de los alumnos.*

*Para concluir, una vez conocidas las funcionalidades básicas de los sistemas integrados, se detallan las experiencias docentes y resultados que se han obtenido una vez que los alumnos han hecho uso de estos sistemas en la realización de las prácticas docentes*

## **1. Introducción.**

La enseñanza en ingeniería requiere de recursos que permitan al alumno poner en práctica los conceptos que va adquiriendo en las clases teóricas. Tradicionalmente, este elemento de apoyo ha sido el laboratorio de prácticas, en el cual, se requiere la presencia física del estudiante supervisada en todo momento por uno o varios profesores, de modo que el estudiante pueda familiarizarse con instrumentos y equipos reales. El contenido de estas sesiones prácticas está fuertemente condicionado por los equipos disponibles, cuyo número es limitado debido al fuerte coste asociado a la adquisición de grandes equipos y su mantenimiento. Debido a estos factores, los centros de enseñanza sólo pueden ofrecer acceso limitado a los equipos.

En este sentido, el rápido desarrollo de las tecnologías basadas en Internet que se ha producido en los últimos años ha posibilitado un gran crecimiento de su uso en el ámbito docente. Estos avances han hecho posible que la educación interactiva a distancia sea una realidad.

El uso de técnicas basadas en Internet supone varias ventajas. Los alumnos pueden acceder libremente a los equipos de laboratorio, pudiendo realizar las prácticas en un horario flexible, desde casa y sin requerir la presencia continua de un profesor. Cada estudiante podrá dedicar el tiempo que estime oportuno para la práctica e incluso podrá realizarla las veces que quiera, posibilitando una mejor asimilación de los conceptos. Por otro lado, es posible implementar un sistema de auto-evaluación en el que el alumno pueda conocer en tiempo real en que medida los conceptos que ha ido adquiriendo se

han afianzado. Estos sistemas incluso permitirían compartir los costes de los grandes equipos con otros centros de enseñanza, implementando un acceso común a los mismos para todos los alumnos.

El presente trabajo recoge una aplicación de la tecnología Internet a la docencia en Robótica. El objetivo es que el alumno pueda acceder remotamente a los robots móviles disponibles en el laboratorio para poner en práctica los algoritmos estudiados en las clases teóricas, pudiendo ver en tiempo real la evolución del robot. En el epígrafe siguiente, se hace una breve descripción de los elementos utilizados en el sistema, haciendo especial hincapié en el robot utilizado. El punto 3 expone los servicios que se han implementado para permitir el acceso remoto al robot, para pasar a comentar, en el apartado 4, el funcionamiento del sistema desde el punto de vista del alumno. Para finalizar, se exponen los resultados obtenidos con la implantación del sistema y las impresiones de los alumnos.

## **2. Descripción Física del Sistema.**

El WiFiBot es un robot de bajo coste y útil en una gran variedad de aplicaciones. Está pensado para cubrir la necesidad existente en el mercado de robots móviles de gama intermedia, técnicamente avanzados y de coste asequible. Estaría ubicado entre la categoría de juguetes y la de los caros robots móviles de características avanzadas para investigación.

Este robot, que podemos ver en la figura 1, se caracteriza por encima de todo por su gran flexibilidad, por lo que puede ser utilizado en múltiples ambientes y situaciones. Su diseño mecánico y su tracción a las cuatro ruedas le permiten desenvolverse sobre superficies irregulares e incluso sobre pequeños obstáculos. Sus pequeñas dimensiones y su bajo peso le hacen ser fácilmente transportable y perfecto para explorar sitios estrechos.



Figura 1: Fotografía del WifiBot. Izquierda Modelo SC y derecha modelo 4G.

El WiFiBot está abierto a todo tipo de usos y aplicaciones ofreciendo todo un mundo de posibilidades de expansión a diferentes niveles. Existen dos modelos de WiFiBot actualmente, el WiFiBot SC y el 4G.

El WiFiBot SC incluye un controlador de BECK (SC12), con un sistema operativo en tiempo real (RTOS) y la opción de utilizar su gran variedad de interfaces: Ethernet, RS232, I2C y el enlace WiFi mediante un bridge wireless ASUS 330.

Los principales sensores que se incluyen en el robot son una cámara fija modelo D-link DCS 900, dos sensores de infrarrojos situados en la parte delantera con una cobertura de

1,2m aproximadamente y cuatro encoders ópticos para medir la velocidad de cada rueda independientemente, dividida en 300 sectores.

El WiFiBot 4G, que es un modelo posterior, incluye ligeros cambios sobre el modelo SC. En primer lugar el controlador SC12 ha sido sustituido por otro controlador mas potente, el 'Access Cube' de 4G-Systeme (Procesador MIPS AMD, 400 MHz, 64Mb RAM y 32Mb de memoria Flash). Este dispositivo incluye puertos, RS232, USB, I2C y tarjeta ethernet y wlan por lo que ya no es necesaria la presencia de un bridge inalámbrico independiente. La cámara se ha sustituido por otra de mejores prestaciones con la posibilidad mover sus ángulos de pan/tilt.

Además de los componentes que incluye el WiFiBot de serie se ha incluido una placa PC de pequeñas dimensiones con su regulador de tensión en ambos modelos, en la cual se ha instalado un sistema Linux Debian. Este PC dispone de un procesador a 800 MHz, 512Mb de RAM, 950Mb de memoria y red ethernet.

Los cuatro motores del WiFiBot pueden ser controlados en bucle abierto o cerrado dependiendo de las necesidades del usuario. Todas la ruedas tienen una rueda externa dividida 300 sectores como la de la figura 2 cuyo recorrido es leído por un sensor óptico. La señal es filtrada y enviada a cuatro controladores de velocidad PID independientes.

Los microcontroladores trabajan mediante modulación PWM (Modulación por anchura de pulso) para establecer la velocidad mediante una salida digital. El robot está preparado para que podamos regular la velocidad de los motores tanto en bucle cerrado con los PID como en bucle abierto. Para bucle cerrado la velocidad de cada rueda será indicada como número de sectores recorridos por cada 1/25 de segundo mientras que para bucle abierto indicaremos directamente el porcentaje de tiempo de PWM que queremos establecer.

Vamos a ver ahora los sensores que incluye el WiFiBot y sus principales características.



Figura 2: Sensores de WifiBot: a) Cámara MPEG-4 Pan/Tilt. b) Camara DCS-900. c) Sensores de infrarrojos. d) Encoder.

El robot incorpora con una cámara de red que varía en función de la versión del robot siendo esta un complemento del mismo. Este es un periférico independiente que puede ser reemplazado por cualquier otro modelo de cámara o periférico de red.

En nuestro caso, el WiFiBot SC incluía una cámara de D-link modelo DCS-900. Esta cámara, con un sensor VGA Digital CMOS de 1/3", tiene una resolución de 640x480 píxeles, una distancia focal de 6.0 mm y una apertura numérica de 1.8. Emplea un

formato de imagen JPG con diferentes radios de compresión. Se interconecta a través de red Ethernet y se puede configurar y recibir imágenes a través de la interfaz web que incluye, de modo que el alumno puede visualizar la evolución del robot.

Nuestro WiFiBot 4G incluye una cámara de red ethernet MPEG-4 Pan/Tilt de la marca Talitor modelo 3090123. Esta cámara dispone de un sensor CCD de 1/4" y tiene la misma distancia focal y apertura numérica que la anterior. Además permite girarla automáticamente hasta un ángulo de  $-135^\circ$  y  $+135^\circ$  sobre el plano horizontal (pan) y entre  $-45^\circ$  y  $+90^\circ$  en vertical (tilt). Permite formatos JPEG y MPEG a diferentes resoluciones.

Todas la ruedas tienen una rueda externa dividida 300 sectores como la de la figura 2d, cuyo recorrido es leído por un sensor óptico. La señal proveniente de estos encoders se emplea para obtener una medida de velocidad de cada rueda que es empleada para el control en bucle cerrado. La velocidad se obtiene contando el número de sectores recorridos en un intervalo de medida de 40ms. No se dispone del signo de este recorrido por lo que asumiremos que se desplazan en el sentido que hemos establecido en el control en velocidad.

El robot está equipado con dos sensores de infrarrojos con un rango máximo de aproximadamente 1,20m. Funcionan por triangulación y dan una salida inversamente proporcional a la distancia al obstáculo. El voltaje de salida es digitalizado sobre 8 bits por un microcontrolador que envía el valor al controlador central. Estos sensores pueden tener diferentes usos como, por ejemplo, activar una parada de emergencia, o esquivar un obstáculo. La salida de los sensores de infrarrojos GP2Y0A02YK que emplea el WiFiBot es no-lineal y sigue una cierta curva.

El robot tiene un autonomía de 1,5 horas, pudiendo variar dependiendo del uso que se hace del robot. Para monitorizar el nivel de batería se ha conectado directamente un convertor A/D de 8 bits a los puertos de entrada/salida del controlador proveyendo esta información. En la figura 3 se puede ver un esquema de las comunicaciones en el WiFiBot para cada uno de los dos modelos.

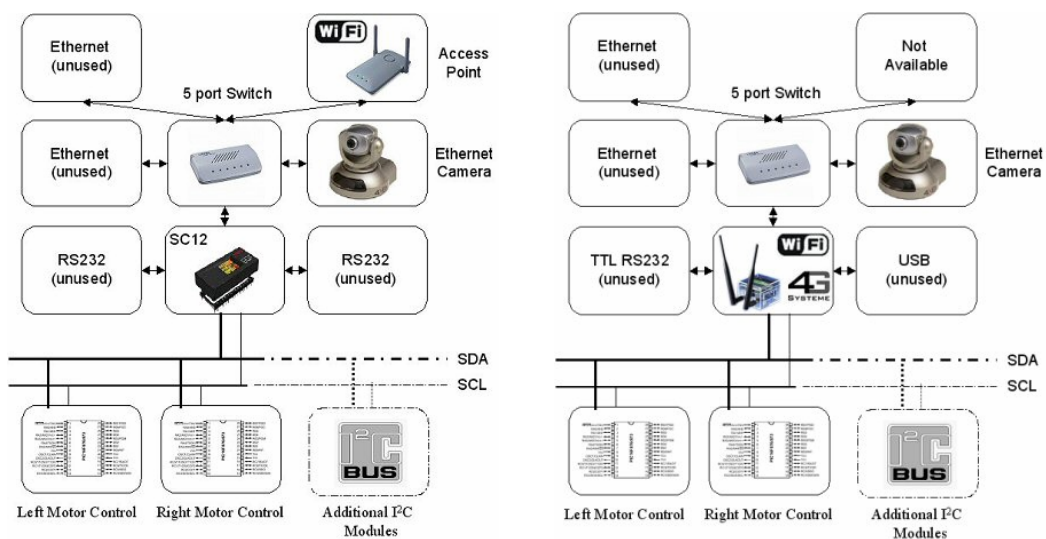


Figura 3: Comunicaciones en el WifiBot. Izquierda SC, Derecha 4G.

Todos los dispositivos de red: Controlador, Cámara, Bridge WiFi para el modelo SC más el PC que se ha incorporado van conectados a un switch de 5 puertos incluido en el robot. Asignamos direcciones IP privadas para todos los dispositivos de red.

El WiFiBot-SC incluye un punto de acceso inalámbrico ASUS wl-330g. Este dispositivo puede ser configurado como punto de acceso, o como puente o adaptador Ethernet dependiendo de la configuración de red deseada.

En nuestro caso lo hemos configurado en el WiFiBot SC como adaptador Ethernet. Este bridge inalámbrico enlaza la red del WiFiBot SC con la red local del laboratorio a través del punto de acceso inalámbrico del laboratorio de forma que se puede trabajar desde los puestos de trabajo de la red local como si estuviésemos conectados directamente al switch del robot perteneciendo todos a la misma subred.

El WiFiBot-4G no dispone de un bridge inalámbrico independiente como en el modelo SC ya que el propio controlador, Access Cube, incorpora una interfaz wireless. Para nuestro WiFiBot 4G, tendremos una red interna del robot a la que se accederá a través del controlador que actuará como router entre la red interna del robot y la red del laboratorio. Utilizaremos traducción de direcciones de red (NAT) para redireccionar puertos del controlador a la cámara o al PC cuando sea necesario.

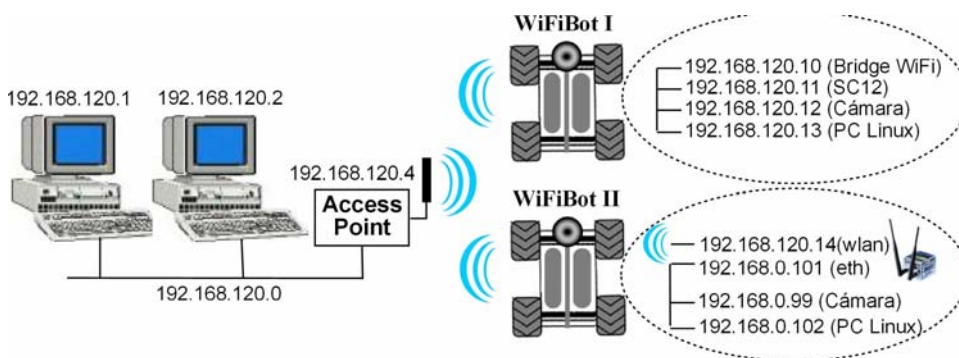


Figura 4: Configuración de la red.

El bus I2C interconecta los microcontroladores encargados de los módulos a bajo nivel (control en velocidad, lectura de infrarrojos y encoders...) con el procesador central que actúa como puente con los niveles superiores de la arquitectura del robot. El controlador central (SC12 o AccessCube) trabaja siempre como maestro y todas las comunicaciones necesariamente pasan a través suyo. Se pueden conectar hasta un máximo de 127 dispositivos y la frecuencia de reloj máxima del bus es de 30 KHz.

El corazón del WiFiBot-SC es el controlador embebido SC12 de BECK. Este controlador es tipo 186 a 20MHz con CPU de 16 bits, tiene 512 Kb de memoria RAM y un sistema operativo en tiempo real (RTOS) con un sistema de ficheros sobre los 512 Kb de memoria Flash de que dispone. Dispone de comunicación via Ethernet, RS232 y bus I2C soportando diversos protocolos: transferencia de archivos a través de los enlaces Ethernet y serie, TCP/IP, PPP, HTTP, FTP, Telnet, POP3, SMTP y DHCP. Este controlador es totalmente programable.

El WiFiBot-4G, que es mucho más avanzado, incluye el controlador 'Access Cube' de 4G-Systeme que dispone de un procesador MIPS AMD Au1500 a 400MHz, 64Mb

RAM y 32MB de memoria Flash. Este controlador incorpora Ethernet LAN, I2C, un puerto RS232 y un puerto USB. Con el Access Cube, el controlador y el punto de acceso inalámbrico se funden en una sola pieza de hardware. Más complicado de usar y de programar que el SC12, este potente controlador bajo Linux puede procesar complicados algoritmos para operaciones autónomas. Además de las configuraciones Wi-Fi estándar permite Mesh.

### **3. Servicios Implementados.**

#### **3.1. Servidor.**

El WiFiBot SC se suministra con dos programas básicamente: el programa servidor para el controlador SC12 y una aplicación cliente para Windows que actúa como interfaz hombre-máquina para manejar y monitorizar el robot (Ver figura 5), mediante el cual el alumno puede mover manualmente el robot y ver el estado de sus sensores. Está pensado para usarlo empleando la red inalámbrica de forma que remotamente el robot pueda ser controlado mediante esta aplicación para Windows que realizará las peticiones al servidor del controlador.

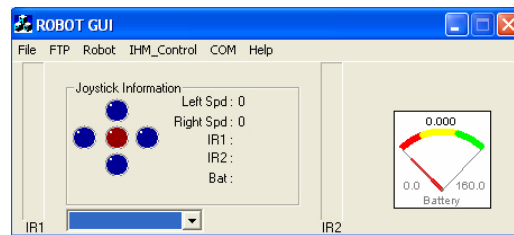


Figura 5: Interfaz Hombre-Máquina

Con objeto de incluir la cinemática y la odometría se han realizado ciertas modificaciones en el servidor original del robot de modo que el alumno pueda acceder a todas las funcionalidades que incorpora el robot, pudiendo hacerlo de forma remota, a través de Internet. Se han incorporado funciones a la librería del WiFiBot para calcular la cinemática y la odometría, se ha incluido un temporizador en el servidor para la actualización de la odometría y se ha cambiado el protocolo por un modo texto pudiendo utilizar cualquier tipo de cliente de telnet para gestionar el robot.

El programa de servidor que se ha incluido en el SC12 escucha en el puerto 5001 del dispositivo. El servidor procesa cadenas de caracteres que contienen peticiones y órdenes de control para el WiFiBot como si se tratara de un intérprete de comandos. Se puede uno comunicar con él desde cualquier programa de telnet. La figura 6 muestra un ejemplo de comunicación con el servidor mediante el conocido cliente de telnet Putty.

```

192.168.120.11 - PuTTY
1 - Unknown Command
reset
0 - Ok
startodo
0 - Ok
ir
29,33
getenc
0,0,0,0
setv 120 0
0 - Ok
getenc
2,2,2,2
getenc
4,4,4,2
setv 0 0
0 - Ok
getpos
913,-30,-6

```

Figura 6: Control del robot desde terminal Telnet

Como nos vemos obligados a trabajar con números enteros en el SC12 al no poder utilizar datos en coma flotante trabajaremos con todos los datos en milímetros y en centésimas de radián, que abreviaremos como crad, para no complicar en exceso el servidor. Los comandos disponibles son los siguientes:

- setv**        Ajusta la velocidad lineal y angular del robot.  
Sintaxis: setv [vlin] [vang] (mm/s crad/s)
- getv**        Devuelve la velocidad lineal y angular vlin,vang (en mm/s y crad/s respectivamente)
- startodo**    Activa la odometría
- stopodo**    Detiene la odometría
- reset**        Pone a cero la odometría
- setpos**      Preajusta la odometría a unos valores de x, y,  $\theta$   
Sintaxis: setpos [x] [y] [ $\theta$ ] en mm y crad
- getpos**      Devuelve la posición y orientación del robot: x, y,  $\theta$  (en mm y crad).
- bat**         Devuelve el nivel de batería
- getenc**      Devuelve los valores de los 4 encoders enc1,enc2,enc3,enc4 (en sectores / 40ms)
- ir**          Devuelve los valores de los sensores de infrarrojos irleft, irright (sin linealizar)

### 3.2. Cliente.

Se ha desarrollado una clase cliente en C++ que realiza la comunicación con el SC12 de manera transparente para el usuario, de modo que el alumno sólo tenga que preocuparse de desarrollar del algoritmo/programa que controle el robot y no de la comunicación. A través de ella podemos realizar cualquiera de las funciones que ofrece el *Nuevo WiFiBot SC12 Server*. Existe un método para cada uno de los comandos permitidos. Además, dado que el sensor de infrarrojos no presenta un comportamiento lineal ha sido necesario linealizar su respuesta, incluyendo un sistema de linealización y de conversión a distancia en la misma clase. La función *getirn()* devuelve los valores del sensor sin linealizar mientras que *getir()* devuelve la distancia ya linealizada y en centímetros.

```

class wbotctrl{
public:

```

```

wbotctrl();
int connection(char *servername, int serverport);
int setv(int v, int w);
int getpos(int *x, int *y, int *a);
int reset();
int setpos(int x, int y, int a);
int startodo();
int stopodo();
int getenc(int *enc1, int *enc2, int *enc3, int *enc4);
int getbat(int *bat);
int getir(double *ir1, double *ir2);
int getirn(int *ir1, int *ir2);
int getv(int *v, int *w);
};

```

Para utilizar la clase simplemente hay que crear un objeto de esta clase, configurar la IP y puerto donde escucha el SC12 mediante el método *setaddress()* y ejecutar el método *connection()*, momento en el cual se establecerá una conexión mediante sockets con el servidor. Al llamar al resto de métodos se envían los comandos correspondientes al servidor y tras recibir la respuesta de éste se devuelven los valores recibidos en los correspondientes punteros o simplemente si la operación se ha realizado satisfactoriamente para los comandos que no tienen datos de respuesta. Todos los métodos de la clase cliente devuelven 0 si no ha habido ningún error. El método *stop()* corta la conexión. De este modo, el alumno debe descargar su programa y seguir los pasos detallados para conocer la evolución del robot según el algoritmo que ha implementado.

### **3.3. Servidor puente multihilo.**

El Servidor Puente Multihilo se ejecuta sobre el PC y permite que varios programas puedan acceder simultáneamente a los servicios que proporciona el SC12. Se emplea el mismo protocolo para el Servidor Puente que para el servidor del SC12 de forma que la misma clase cliente que hemos descrito antes puede comunicarse tanto con el servidor del SC12 como con el servidor puente, tan solo hay que especificar la IP y puertos correspondientes. De hecho, el mismo servidor puente hará uso de esta clase para enlazar con el SC12.

La ventaja de este servidor, como se ha dicho, es que es multihilo de manera que varios clientes finales (alumnos) podrán acceder a los servicios a bajo nivel a la vez. Por ejemplo, se podría querer diseñar un software que realizase algún tipo de control sobre el robot mientras otro programa monitoriza los valores de la odometría o los sensores. Se debe tener en cuenta que el control se debe poder realizar únicamente desde uno de los clientes. El resto sólo podrían monitorizar los resultados.

En la figura 7 podemos ver gráficamente como queda la estructura de servidores y clientes que encontramos en el WiFiBot. Diferentes programas, que pueden estar ejecutándose tanto dentro del PC del robot como otros programas que podrían ejecutarse en otros dispositivos externos, acceden simultáneamente a los servicios básicos haciendo uso de la clase cliente que los conecta con el servidor multihilo el cual conecta a su vez directamente con el servidor del SC12.



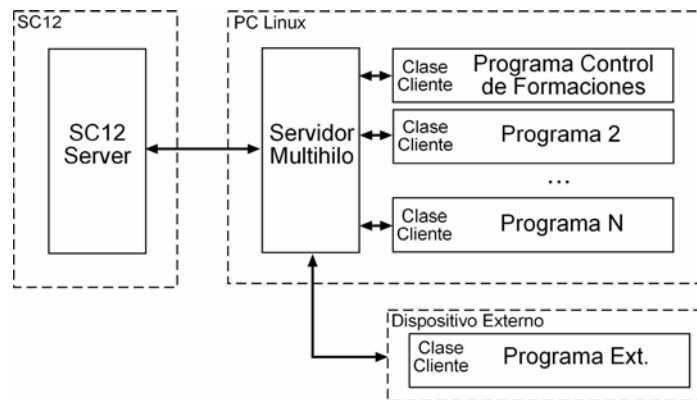


Figura 7. Estructura de servidores del WiFiBot.

Como se ha indicado la principal característica y el motivo por el que se ha diseñado este servidor es por la necesidad de poder atender a varios alumnos a la vez, lo cual no es posible de implementar en el SC12 con todas las limitaciones que tiene. Por ello es necesario programar un servidor multihilo. Los hilos se ejecutan en paralelo compartiendo el mismo espacio de memoria. Este servidor establece una conexión con el SC12 mediante la clase cliente vista en el apartado anterior y deja un socket esperando posibles clientes en el puerto 3500. Para cada nuevo alumno que se conecta se crea un hilo de ejecución independiente que atienda a dicho alumno. Para evitar posibles errores en la comunicación con el SC12 ocasionados por peticiones simultáneas de varios alumnos se emplean mecanismos de sincronización. Hemos utilizado un mecanismo de exclusión mutua (mutex) que nos permite bloquear los fragmentos de código correspondientes a las peticiones al SC12 de forma que sólo puede haber un único hilo ejecutando dicho fragmento en un determinado momento. El resto de hilos quedan bloqueados hasta que son liberados por el hilo que se encontraba ejecutando dicho código.

#### **4. Funcionamiento del sistema.**

Para acceder a todos los servicios comentados, el usuario debe validarse mediante su login y contraseña. Una vez que el usuario ha entrado al sistema se posibilita una interacción plena con el robot. En primer lugar, el alumno debe descargar el programa que ha implementado y que quiere probar sobre el robot. Una vez que el robot tiene cargado el programa de control, se pasará a la fase de experimentación, en la cual, el alumno tiene acceso a todos los datos suministrados por las funciones descritas. Además cabe destacar que el alumno puede tener acceso a las imágenes captadas por la cámara del robot, según se ve en la figura 8. De este modo, al conocer en cada momento que es lo que está viendo el robot, el alumno puede hacerse una idea de si su algoritmo está funcionando correctamente de un solo vistazo.



Figura 8. Acceso a la imagen captada por la cámara del robot.

Teniendo en cuenta los sensores disponibles y los servicios implementados, las utilidades que se pueden dar al sistema desde el punto de vista del usuario remoto (alumno) son múltiples. Entre ellos, podemos destacar:

- Uso como equipo de experimentación en visión por computador. Por ejemplo, el sistema se puede utilizar para el reconocimiento de objetos situados en el entorno del robot. El alumno movería manualmente el robot mediante la interfaz de la figura 5 a la vez que visualizaría las imágenes captadas por la cámara del robot. El algoritmo desarrollado por el alumno tendría que ser capaz de reconocer ciertos objetos en la escena.
- Prácticas de ingeniería de control avanzada. Conociendo la cinemática del robot, el usuario puede proponerse la implementación de ciertas tareas a realizar por el robot utilizando técnicas de control avanzado, como el control difuso. Por ejemplo, una posible tarea sería que el robot se mueva hasta haberse situado a cierta distancia y posición respecto a un objeto de la escena que debe reconocer mediante la cámara. El alumno tendrá que implementar varios comportamientos (por ejemplo, ir al objetivo, que haría uso de las imágenes de la cámara y evitar obstáculos, que haría uso de los datos de los infrarrojos) y combinar dichos comportamientos haciendo uso de distintas técnicas de control.
- Prácticas de robótica móvil. El alumno podría probar algoritmos sencillos de control de un robot móvil y planificación de trayectorias, observando en tiempo real cuales son los resultados de su algoritmo.

## **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 1 se observa 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.

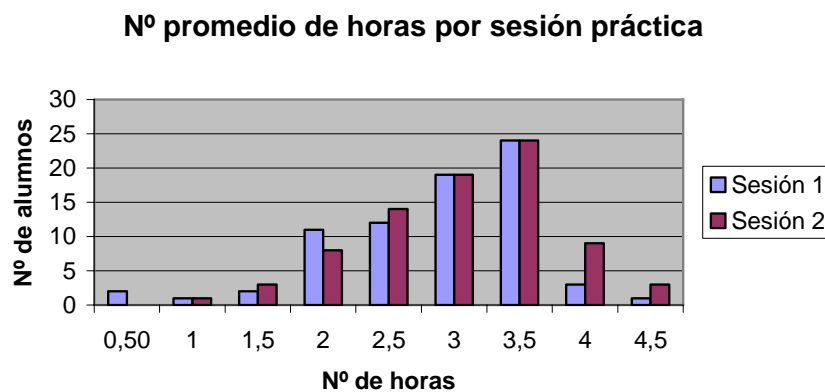


Figura 9: Número promedio de horas dedicadas por parte de los alumnos a cada una de las sesiones

En el gráfico de la figura 2 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.

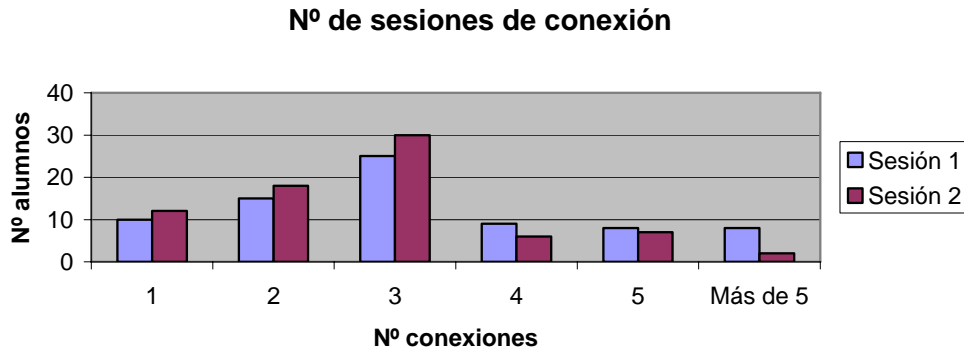


Figura 10.- Número de sesiones de conexión realizadas por los alumnos en cada práctica.

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.

## **6. Conclusiones.**

En este artículo se han mostrado los servicios implementados en un sistema físico como un robot móvil (Wifibot), de forma que éste pueda ser utilizado por los alumnos para la realización de prácticas docentes mediante Internet. Como se desprende de las características físicas y de los servicios implementados, este dispositivo físico cuenta con un enorme potencial dentro de este ámbito. Los servicios implementados posibilitan la lectura de los diferentes sensores disponibles en el dispositivo de forma que éste pueda ser monitorizado a distancia por parte de un conjunto elevado de usuarios, así como el control a distancia por parte de un usuario.

Basándonos en el uso de este tipo de dispositivos y servicios implementados se han diseñado dos prácticas de manera que los alumnos puedan realizar éstas en cualquier banda horaria sin las restricciones habituales de los laboratorios docentes. La realización de prácticas docentes con dispositivos físicos a distancia permitirá aprovechar mucho mejor los recursos disponibles dado que un sistema físico es utilizado por un conjunto muy elevado de usuarios y en franjas horarias cualesquiera.

## **Bibliografía.**

[1] José Miguel Alonso. *TCP/IP en Unix programación de aplicaciones distribuidas*. Ra-ma, 1998.

[2] ASUS. *ASUS Wireless 330g Access Point manual*.

[3] T. Balch and R. Arkin. Avoiding the past a simple but effective strategy for reactive navigation. *IEEE International Conference on Robotics and Automation*, may 1993.

[4] T. D. Barfoot and C. M. Clark. Motion planning for formation of mobile robots. *submitted to Robotic and Autonomous Systems*, january 2003.

- [5] BECK. *SC12 Documentation*.
- [6] J. Borenstein, H. R. Everett, and L. Feng. Where am i? sensors and methods for mobile robot positioning. Technical report, University of Michigan, 1996.
- [7] Jean-Yves Bouguet. *Camera Calibration Toolbox for Matlab Documentation*.
- [8] R. Brooks. A robust layered control system for a mobile robot. *IEE Journal of Robotics and Automotion*, RA-2(1);14, 1986.
- [9] Y. U. Cao, A. S. Fukunaga, and A. B. Kahng. Cooperative mobile robotics: Antecedents and directions. *IEEE/TSJ International Conference on Intelligent Robots and Systems*, 1995.
- [10] D-Link. *Camera D-Link DCS 900 Manual*.
- [11] Intel. *Open Source Computer Vision Library (OpenCV) Documentation*.
- [12] Miguel Juliá Cristóbal. Seguimiento de robots mediante visión artificial, aplicación al mantenimiento de formaciones basada en comportamientos. Proyecto fin de carrera de Ingeniería de Telecomunicación en la Universidad Miguel Hernández de Elche, 2005.
- [13] Robin R. Murphy. *An Introduction to AI Robotics (Intelligent Robotics and Autonomous Agents)*. MIT Press, 2000.
- [14] Lauro Ojeda and Johann Borenstein. Reduction of odometry errors in over-constrained mobile robots. *Proceedings of the UGV Technology Conference at the 2003 SPIE AeroSense Symposium, Orlando, FL*, pages 21–25, april 2003.
- [15] G. Pajares and J. M. de la Cruz. *Visión por Computador*. RA-MA, 2001.
- [16] L. E. Parker. Current state of the art in distributed autonomous mobile robotics. *In Distributed Autonomous Robotic Systems*, 2000.
- [17] J. Rosenblatt. Damn: A distributed architecture for mobile navigation. *AAAI Spring Symposium on Software Architectures for Physical Agents*, 1995.
- [18] A. Saffioty. The uses of fuzzy logic in autonomous robot navigation. *Soft Computing*, pages 180–197, 1997.
- [19] R. Siegwart and I. Nourbakhsh. *Introduction to Autonomous Mobile Robots*. Massachussets Institute of Technology, 2004.
- [20] WIFIBOT. *WiFiBot 4G and SC Guide*.