

# Uso de simuladores en docencia de robótica móvil

José María Cañas Plaza, Miguel Ángel Cazorla Quevedo y Vicente Matellán, *Member, IEEE*

**Title**—Use of simulators in the teaching of mobile robotics

**Abstract**—The main goal of this paper is to present the advantages that the new generation of mobile robot simulators has given to the teaching of robotics to computer science students. We present some of their features, as well as some experiences of their use in two Spanish universities. These experiences have taken place at different courses, both at graduate and undergraduate levels. We also discuss about those experiences and introduce how we envision teaching in this field.

**Index Terms**—Simulators, teaching experience, robots

## I. INTRODUCCIÓN

La robótica es una disciplina que aparece escasa pero recurrentemente en distintos niveles educativos, desde la enseñanza secundaria [32] hasta el postgrado. En nuestro país no existe una titulación específica dedicada a ella, sino que sus contenidos aparecen dispersos en varias titulaciones [25]: Ing. Informática, Ing. Industrial, Ing. Electrónica, Ing. de Telecomunicación, etc., siendo habitual en los planes de estudio de estos grados encontrar asignaturas de corte robótico principalmente en los últimos cursos. Igualmente, es posible encontrar algún master de contenido robótico.

Al ser muy variado el entorno de estas asignaturas, también lo son las metodologías docentes con las que se imparte. Mientras unas asignaturas hacen énfasis en la construcción mecánica o la electrónica de sus componentes, otras ponen el acento en la programación y la generación de comportamiento autónomo. Mientras unas se centran en los manipuladores y los usos industriales, otras están enfocadas a los robots móviles. Los contenidos de robótica móvil también aparecen frecuentemente combinados con los de inteligencia artificial y de visión computacional.

La heterogeneidad se manifiesta más en la parte práctica, lo que tradicionalmente ha sido un problema. Las prácticas en este campo requieren el acceso a equipos cuando menos costosos, y en algunos casos peligrosos. La logística tradicional asociada a este tipo de prácticas ha consistido normalmente en asignar a cada alumno un número limitado de turnos de acceso a los equipos, lo que ha hecho que en general la parte práctica de estas asignaturas haya sido bastante deficiente.

Un enfoque académico diferente e interesante es la docencia basada en proyectos. Se fija como objetivo del curso

José María Cañas es miembro del Depto. de Sistemas Telemáticos y Computación de la U. Rey Juan Carlos. Email: jmplaza@gsyc.es; Miguel Ángel Cazorla es miembro del Depto. Ciencia de Computación e Inteligencia Artificial de la U. Alicante. Email: miguel@dccia.ua.es. Vicente Matellán es miembro del Depto. Ingenierías Mecánica, Informática y Aeroespacial de la Universidad de León. Email: vicente.matellan@unileon.es.

DOI (Digital Object Identifier): pendiente

la participación en algún concurso robótico, por ejemplo, y alrededor de la construcción del robot participante se transmite el conocimiento, muy práctico, a los estudiantes. Un ejemplo pionero fue el del robot Xavier [3], creado por estudiantes de la universidad Carnegie Mellon (EE.UU.), bajo la tutela de Reid Simmons, para participar en la AAAI Robotics Competition de 1993. Otro ejemplo más reciente es el robot Stanley, ganador del Grand Challenge de 2005, creado por estudiantes de la universidad de Stanford bajo la tutela de Sebastian Thrun [14].

Sin embargo, esta metodología sólo está al alcance de centros de excelencia con muy pocos alumnos y bastantes medios materiales, lo que generalmente sólo es posible en cursos de postgrado y centros de excelencia. Para los entornos menos elitistas, las plataformas de robótica móvil disponibles han resultado generalmente demasiado costosas, o bien, han sido prototipos construidos por los grupos de investigación. Lo que en ambos casos hacía que fuese muy raro su uso docente hasta principios de este siglo.

### A. Simuladores vs. robots reales

La aparición de los micro-robots a finales de los años 90, siendo el RugWarrior [8] uno de los precursores, hizo que las instituciones con menos medios pudiesen plantearse el uso de plataformas reales para la docencia. Este tipo de robots se usa en la actualidad con bastante frecuencia, incluida España [31], o incluso en otras disciplinas afines [21]. Sin embargo, la experiencia ha demostrado que este tipo de plataformas resulta más útil para la enseñanza de electrónica digital, o de diseño con microprocesadores, que para la robótica móvil propiamente dicha. Esto es así incluso con micro-robots menos orientados a la construcción electrónica, como los LEGO Mindstorms [1], puesto que ofrecen posibilidades muy limitadas para implementar algoritmos sofisticados al disponer de sensores muy simples y de poca capacidad de cómputo [31], aunque han demostrado que se pueden usar para introducir la robótica de manera simplificada en ciclos pre-universitarios [34].

Una alternativa recurrente han sido los *simuladores*. Estas herramientas ofrecen un entorno virtual en el que se emulan las observaciones de los sensores y los efectos de las órdenes a los actuadores. La utilización de prácticas "virtuales" no es una práctica extraña, ni a nivel internacional, donde se pueden encontrar recopilaciones de ejemplos con bastante tradición [24]; ni a nivel nacional, con experiencias variadas con laboratorios virtuales, tanto en universidades no presenciales, como las del grupo del profesor Sebastián Dormido en la UNED, o en universidades presenciales, como por ejemplo

en la de Alicante [22]. Sin embargo, existen pocos estudios de sus implicaciones docentes en el caso de la robótica móvil. Únicamente en las descripciones de entornos concretos de simulación, como por ejemplo Pyro [35], en el caso de plataformas hardware que incluyen simuladores, como [37], o colateralmente en revisiones generales de software robótico [36].

Los simuladores son una opción económica para aquellos centros que no cuentan con la posibilidad de adquirir y mantener hardware real. También son muy interesantes, por ejemplo, en la investigación con grupos numerosos de robots porque abaratan la inversión necesaria. Otra ventaja es la facilidad de obtener ciertos datos, como la verdad absoluta sobre la posición del robot, más complejos o caros de obtener en el mundo real. Por último, los simuladores permiten la repetitividad de los experimentos o prácticas que se realicen, lo que resulta muy difícil con los robots reales, ya que las condiciones siempre cambian en un entorno real, por ejemplo la iluminación, el estado de las baterías, etc.

A pesar de estas ventajas, el uso de simuladores no ha sido históricamente muy popular. Concentrándonos en la docencia de robótica móvil, objetivo de este artículo, los problemas tradicionales de utilizar simuladores se pueden resumir en tres:

- 1) Falta de realismo. Los primeros simuladores únicamente permitían definir mundos planos donde únicamente había obstáculos estáticos bidimensionales.
- 2) Problemas legales y de precio, al tratarse en su mayor parte de software propietario proporcionado por los fabricantes de robots móviles.
- 3) Falta de generalidad. Los alumnos aprendían a utilizar el simulador de una plataforma concreta, que luego difícilmente correspondía con el que utilizarían en su vida profesional.

Uno de los pioneros fue SRIsim [9], [10] que permitía simular un único robot en un mundo bidimensional estático. Con los años los simuladores han ido ganando en realismo, incorporando ruido en los sensores y en las actuaciones, mejorando el realismo de los entornos, etc. Actualmente disponemos por ejemplo de simuladores tridimensionales, como el que proporciona Microsoft en su Robotics Studio [11] (Figura 1) que son muy realistas y capaces de funcionar en un ordenador personal, lo cual ha aumentado su funcionalidad en la formación en robótica[33]

El uso de simuladores creemos que es particularmente útil en el caso de la docencia en titulaciones informáticas, donde el desarrollo de competencias en el campo de la electrónica y la mecánica es menos necesario. Para plantear de forma más rigurosa esta afirmación, en la siguiente sección describiremos las competencias y perfiles donde creemos que se aplica.

### B. Competencias y perfiles

Dada la mencionada heterogeneidad en la docencia de "robótica" en la oferta educativa española, tanto vertical (se puede encontrar desde secundaria al doctorado), como

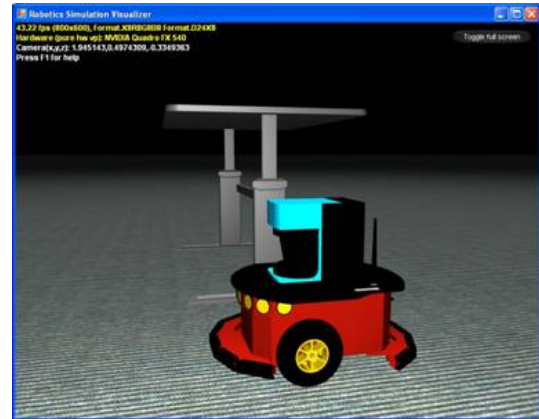


Fig. 1. Simulador de Microsoft Robotics Studio

horizontal (aparece en múltiples enseñanzas técnicas y experimentales), hemos considerado necesario acotar nuestro análisis a un solo nivel, el grado, y a una única titulación, la ingeniería informática. Aún así, la variedad de planes de estudio y las múltiples opciones de situar la robótica en el plan de estudios, nos han hecho dar por supuestos una serie de conocimientos básicos, siendo el más importante el referido a la programación.

- Matemáticas: los alumnos se supone que han adquirido los conceptos básicos de matemática continua: derivación, integración, así como las habilidades básicas para aplicarlos a la solución de problemas.
- Física: manejo de los conceptos básicos de mecánica: fuerzas, velocidades y aceleraciones.
- Geometría: conocimientos básicos de geometría en dos dimensiones, trigonometría.
- Estadística: se supondrán conocimientos de los estadísticos básicos, del manejo del concepto de distribución y de las aplicaciones del teorema de Bayes.
- Programación: supondremos que los alumnos son capaces de analizar problemas y describir la solución algorítmica de los mismos así como de implementarlos en un lenguaje orientado a objetos.

Igualmente, existen múltiples perfiles robóticos que se pueden formar, desde la construcción de manipuladores, al diseño de sistemas cognitivos para robots móviles. En este caso, el perfil objetivo de las experiencias descritas en este artículo es el de "programación de robots móviles". Más en concreto, y una vez descritos los conocimientos que se suponen a los alumnos, las competencias básicas que queremos que los alumnos adquieran son:

- Conocer los problemas clásicos de la robótica móvil.
- Describir los principios de las diferentes arquitecturas cognitivas de control: reactiva, planificada e híbrida.
- Conocer los principios de los algoritmos clásicos para los problemas de navegación y construcción de mapas.
- Comprender los fundamentos de los algoritmos de auto-localización.

- Aplicar los conocimientos anteriores a la resolución de problemas de robótica.

En el resto del artículo desarrollamos esta idea de que los simuladores son especialmente apropiados para la docencia de la robótica en informática, aunque se complementen con el uso de algún robot real. Así, en la segunda sección se describen someramente los simuladores que consideramos a la hora de diseñar las prácticas de nuestras asignaturas. En la tercera se describen las experiencias docentes en la Universidad de Alicante y en la Universidad Rey Juan Carlos. Finalmente, la última sección se dedica a analizar dichas experiencias y a comentar el futuro de la docencia en este campo.

## II. SIMULADORES DE ROBOTS MÓVILES

Como hemos presentado, nuestra tesis en este artículo es que el uso de simuladores constituye una alternativa razonable para el desarrollo de prácticas (ser reproducibles, medibles, etc.). En esta sección describiremos los que consideramos más interesantes. Incluimos SRIsim por motivos históricos, Webots por ser la opción propietaria con más solera (aunque el *Robotic-Studio* de Microsoft haya tenido mucha más repercusión), *Player/Stage/Gazebo* por ser a nuestro juicio la mejor alternativa y *USARsim*, por usarse en varias competiciones internacionales. Hemos preferido dejar fuera simuladores específicos para algún robot (como *Eye-Sim*, para el *Eyebot*) o comparativamente menos conocidos (como *SimRobot*[12], [38]).

### A. SRIsim/Saphira

La arquitectura *Saphira* [10] es un sistema de control de robots desarrollado en SRI International y su distribución incluía el simulador *SRISim*. Esta arquitectura y el robot al que estaba asociado tuvieron un relativo impacto en la comunidad robótica. Pero la notoriedad de su simulador se debe a que fue elegido para distribuirse con los robots fabricados por *MobileRobots* (antigua *ActivMedia*), el mayor fabricante de robots móviles para investigación y docencia (*Pioneer*, *Amigobot*, *PeopleBot*, etc.).

*SRIsim* [9] está orientado a la simulación de un único robot en un mundo bidimensional estático. Permite simular diversos sensores, en particular la simulación del sensor láser está bien conseguida, incluyendo los posibles errores. Este simulador también genera movimientos razonablemente realistas, simulando las lecturas de odometría también con ruido. Su arquitectura, basada en separar el control del simulador usando sockets *TCP/IP*, ha sido también muy influyente.

Sin embargo, aunque se distribuía de forma gratuita y semi-libre, su extraña política de licencias, ha hecho que sea un simulador en franca decadencia. También contribuye a ello la no existencia de una versión tridimensional y el poco soporte para la simulación de varios robots. De hecho, la empresa *MobileRobots* ha dejado de considerarlo su simulador de referencia, que ha pasado a ser *MobileSim* [13] basado en *Stage*.

### B. Webots

La aparición del *Microsoft Robotics Studio* (MRS) [6] ha recibido gran atención mediática. Sin embargo, a nuestro juicio el simulador propietario más relevante es *Webots* [15], que actualmente está en su versión 5. Este simulador se comenzó a desarrollar en el *Laboratoire de Micro-Informatique* (LAMI) del *Swiss Federal Institute of Technology* (EPFL), y aunque las primeras versiones fueron gratuitas, muy pronto pasó a distribuirse en el modelo tradicional por licencias distribuyéndose inicialmente por *K-Team* [16] y posteriormente por *Cyberbotics* [15].

*Webots* es un simulador 3D que utiliza *OpenGL* para visualización y *ODE* [17] (*Open Dynamics Engine*) para el cálculo de la simulación física. En la actualidad simula con realismo multitud de robots, desde los fabricados por *K-Team* (*Khepera*, *Koala*, etc.), hasta humanoides muy recientes como el *Nao* de *Aldebaran Robotics*, o los *Aibo* de *Sony*. Igualmente permite utilizar multitud de lenguajes de programación, desde los más tradicionales, hasta los más modernos y específicos para robots móviles como *URBI*.

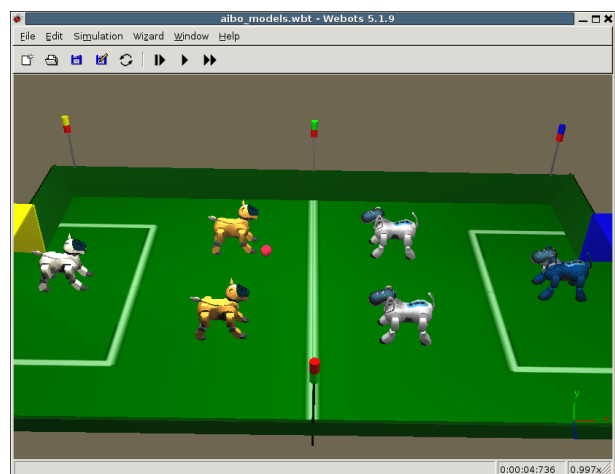


Fig. 2. Webots simulando un partido de la RoboCup

### C. Player/Stage/Gazebo (PSG)

La plataforma *PSG* [19], [7] está formada por los simuladores *Stage* y *Gazebo*, y por el servidor *Player* de acceso al hardware real o simulado. *Stage* (Figura 3) está orientado a sistemas multirrobot numerosos en dos dimensiones, mientras que *Gazebo* lo está a sistemas multirrobot no muy numerosos, pero en entornos tridimensionales (Figura 4).

*Stage* muestra una ventana de visualización desde la que se observa el mundo plano con todos los robots del escenario, y sobrepuesto alguno de sus sensores, como el sensor láser que se aprecia en la Figura 3. *Gazebo* es capaz de simular sensores relativamente complejos como cámaras, usando *OpenGL*, y emplea *ODE* como motor para emular la física del mundo simulado, donde los robots tienen masa, inercia, pueden derribar otros objetos, etc. También permite

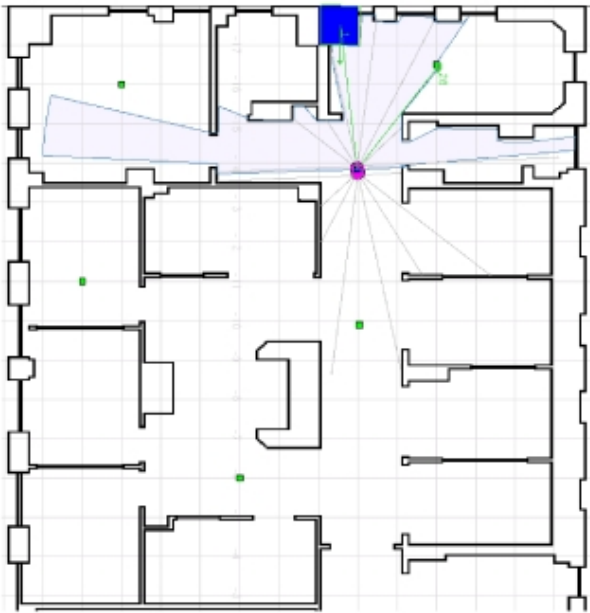


Fig. 3. Simulador de robots Stage

incorporar dentro del mundo cámaras externas para visualizar la escena.

El estilo de programación se basa en una arquitectura cliente/servidor, donde la aplicación se conecta al servidor Player para recoger los datos sensoriales y enviar las órdenes a los actuadores del robot simulado (en Stage o Gazebo) o incluso a un robot real que disponga de un servidor Player. Las aplicaciones establecen de esta forma un diálogo usando *sockets* TCP/IP con el servidor Player, lo que permite el acceso remoto al hardware. Este diseño proporciona a las aplicaciones construidas sobre PSG gran independencia respecto al lenguaje de programación e impone pocas restricciones a la arquitectura del software de control. Así, las aplicaciones pueden escribirse en cualquier lenguaje, y con cualquier estilo, simplemente respetando el protocolo de comunicaciones con el servidor.

PSG se orienta principalmente a ofrecer una interfaz abstracta del hardware de los robots y no a la identificación de bloques comunes de funcionalidad. No obstante, se puede incorporar cierta funcionalidad adicional en forma de nuevos mensajes del protocolo, y servicios añadidos a Player. Por ejemplo, la localización probabilística se ha añadido como una interfaz más que proporciona múltiples hipótesis de localización.

#### D. USARSim

*Unified System for Automation and Robot Simulation* (USARSim [4]) comenzó su desarrollo en 2002 en la universidad de Carnegie Mellon, aunque ahora se trata de un proyecto de software libre autónomo. USARSim es un simulador multi-robot cuya característica más reseñable es que está basado en motor de juegos comercial (*Unreal Tournament*), lo que

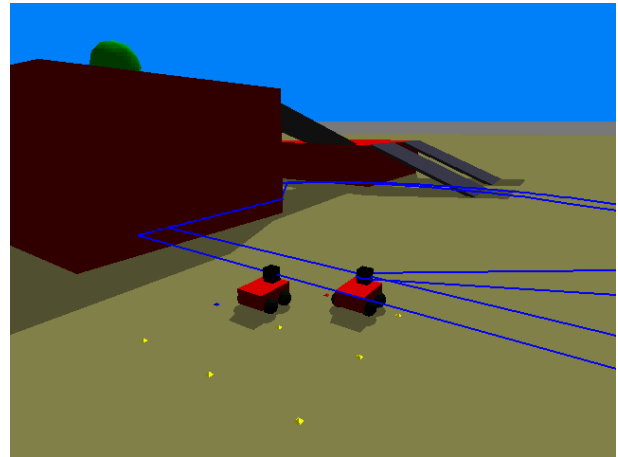


Fig. 4. Simulador de robots Gazebo

concede un elevado grado de realismo a los escenarios y simulación de elementos físicos.

Inicialmente se creó para dar soporte al desarrollo de robots de búsqueda y rescate, pero posteriormente ha demostrado ser una herramienta con un espectro de usos mucho mayor. Así, se ha utilizado en enseñanza, en el diseño de interfaces robot-persona, automatización, cooperación, etc. Además, este simulador es la base para la RoboCup Virtual Robot Competition y para el IEEE/NIST Virtual Manufacturing and Automation Competition (VMAC), lo que le ha dado mucha visibilidad.

#### E. Comparación de simuladores

Existen muchos criterios de clasificación de los simuladores: libres/propietarios, mono-robot/multi-robot, bidimensional/tridimensional, etc. Sin embargo, hemos decidido no presentar ninguna taxonomía en este artículo, que se puede encontrar en otros trabajos, como por ejemplo [5], [20], describiendo simplemente en esta sección los que a nuestro juicio son más relevantes para la docencia. Simplemente, la tabla I resume nuestra opinión sobre las principales características, utilizando una valoración numérica de 0 a 5 (siendo 5 lo mejor).

Resumiendo la tabla I, Player/Stage/Gazebo es a nuestro juicio la mejor alternativa y la que por tanto usamos en las experiencias que describimos a continuación.

### III. EXPERIENCIAS DOCENTES

Vamos a comentar las experiencias docentes en robótica realizadas en las universidades de Alicante (UA) y Rey Juan Carlos (URJC). En ambas el enfoque es similar, haciendo énfasis en la programación de la inteligencia de los robots más que en su construcción mecánica o circuitería electrónica.

TABLE I  
RESUMEN DE SIMULADORES

Simuladores	Caract.	Licencia	Complejidad	Fidelidad	Portabilidad	Extensibilidad	Documentación	Modelos simulados	Multirrobot	Total
SRISim	4	4	2	2	3	3	2	1	21	
Webots	2	4	4	4	0	4	5	5	28	
PSG	5	3	4	5	5	5	5	5	35	
USARSim	5	4	4	4	4	3	4	4	32	
No Incluidos en el artículo										
MSRS	5	3	3	3	3	2	1	4	24	
SimRobot	2	4	4	4	0	4	5	5	29	

A. Experiencia docente en la Universidad de Alicante

En los planes de estudios de las ingenierías en informática de 1993 se implantó una asignatura denominada Robótica, optativa para las carreras: Ingeniero en Informática, Ing. Técnico en Informática de Sistemas e Ing. Técnico en Informática de Gestión. En el plan del 2001 se cambió la denominación de la asignatura a Robots Autónomos [26], para reflejar el enfoque hacia la robótica móvil de sus contenidos:

- 1) Historia de la robótica
- 2) Sistemas de coordenadas, locomoción y mapas
- 3) Sensores y visión
- 4) Construcción de mapas y localización. Introducción al SLAM
- 5) Planificación de trayectorias
- 6) Evitación de obstáculos

La asignatura tiene un número de matriculados que ha variado mucho durante cursos, pero que nunca ha descendido de los 50 alumnos. Esto hace inviable el uso de robots reales. Inicialmente se planteó el uso del simulador SRISim. A partir del curso 2003/04 se usa PSG como plataforma de soporte de las prácticas. Esto ha supuesto una mejora sustancial en el desarrollo de las prácticas. Además de soportar los robots disponibles en el departamento, la facilidad de instalación ha supuesto una mayor comodidad para los alumnos. Además, a las mejores prácticas les proponemos que prueben su código con un robot real. Esto es un aliciente más para el esfuerzo del alumno, que puede comprobar el funcionamiento real de un robot. Algunas de las prácticas son las siguientes:

1) *Navegación topológica*: Se pretende guiar a un robot usando información topológica. Partimos de un mapa como el mostrado en la Figura 5. Las etiquetas indican puntos distinguibles del entorno: D para puerta, R para sala, F para fin de pasillo y C para cruce. El estudiante lee un fichero donde se define el grafo con el siguiente formato:

```
R9 D16 Norte 6000
E1 D16 Este 4000
...
```

Cada línea es una arista del grafo. Dentro de cada línea los dos primeros símbolos especifican los nodos de la arista. El siguiente símbolo indica la dirección que relaciona los nodos y el último es el peso de la arista. Por ejemplo, la primera línea es una arista que parte del nodo R9 hacia la D16 en dirección norte y cuyo peso es 6000. Este peso no indica necesariamente distancia métrica, sino que puede contener la “dificultad” para pasar de un nodo al siguiente. El programa del alumno recibe dos nodos del grafo (dónde se encuentra el robot y adónde quiere dirigirse) y se pide calcular el camino más corto dentro del grafo. La salida de este paso será un camino como:

```
R10 Norte D13 Este C2 Norte C1 Este D4
Norte R3
```

que indica que partiendo del nodo R10 debemos ir hacia el norte hacia D13; una vez llegamos allí, debemos ir en dirección este hacia C2, etc.

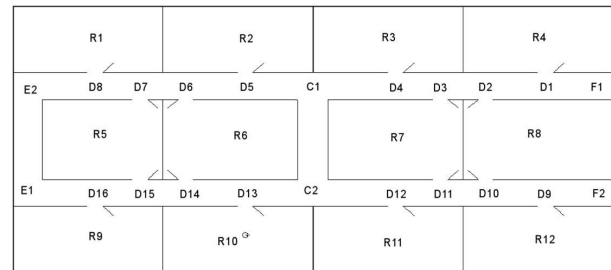


Fig. 5. Entorno donde se va a mover el robot desde R10

Queda recorrer dicho camino usando un planificador. Para ello se emplea una tabla de transición como la mostrada en la Tabla II. Por ejemplo, si partimos del nodo R10 y debemos ir al D13, la entrada de la tabla que parte de un nodo R a uno D tiene el símbolo CP que significa cruzar puerta (SP significa seguir pasillo y ND es que no está definida la operación). El estudiante tiene que crear un método que permita cruzar la puerta usando la información de los sensores y que compruebe cuándo la ha cruzado.

Cuando nos dirigimos hacia una puerta (el símbolo D) sabemos que debemos parar cuando crucemos esa puerta. Cuando vayamos de una puerta a un cruce (de D a C) sabemos que debemos seguir pasillo hasta que encontremos el cruce.

TABLE II  
TABLA DE TRANSICIÓN ENTRE PUNTOS DISTINTIVOS

Desde	A				
	R	E	C	F	D
R	CP	ND	ND	ND	CP
E	ND	SP	SP	SP	SP
C	ND	SP	SP	SP	SP
F	ND	SP	SP	ND	SP
D	CP	SP	SP	SP	SP

2) *Construcción de mapas*: En el curso 2005/06 se introdujo Java, para lo que se usó JavaClient [27], un envoltorio

que crea clases Java que permiten acceder a las clases de PSG. El uso de Java permitió desarrollar fácilmente interfaces gráficas, lo que permitió realizar una práctica de construcción de mapas del entorno, usando segmentos de línea (ver Figura 6). Para ello, se proporciona un módulo construido en Java que visualiza segmentos de línea. El alumno tiene que obtener una lectura del láser, procesar dicha lectura y obtener una lista de segmentos, cuyas coordenadas están definidas con respecto al sistema de coordenadas del robot. Después tiene que convertir esas coordenadas a un sistema de coordenadas absoluto (p.e. con origen en la posición inicial del robot) y emparejar los segmentos obtenidos en un momento dado con los obtenidos previamente. La facilidad que aporta Java en el manejo de las listas (frente a C++) hace que se simplifique este tipo de prácticas.

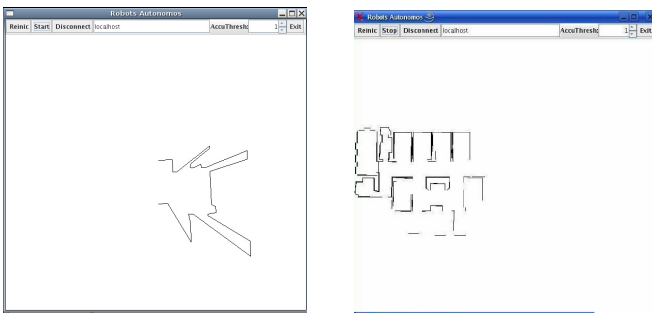


Fig. 6. Entorno de trabajo con una lectura local y mapa construido.

### B. Experiencia docente en la Universidad Rey Juan Carlos

En la URJC se imparten en la actualidad tres asignaturas relacionadas con la robótica móvil: Una asignatura de libre configuración denominada “Introducción a la robótica”, abierta a alumnos de múltiples titulaciones: ingeniería informática, telecomunicación, etc.; Una asignatura optativa para alumnos de Ing. Técnica en Informática de Sistemas, “Robótica”, y una asignatura de postgrado de igual nombre.

Las asignaturas de grado pretenden que los alumnos programen por primera vez en su vida una máquina que no es un ordenador personal, algo que se tiene que comportar autónomamente siguiendo las órdenes de un programa. En este sentido transmite la idea de que la inteligencia del robot reside en su software. Ambas tienen alrededor de 40 alumnos regularmente.

El temario teórico de la asignatura Robótica [28] presenta los componentes esenciales de todo robot (sensores y actuadores) y describe los problemas clásicos de la robótica móvil: navegación autónoma, construcción de mapas, localización, etc.

Inicialmente las prácticas se desarrollaban íntegramente con los robots LEGO RCX. El equipamiento que se prestaba a cada pareja de alumnos contenía un surtido de piezas mecánicas y un ladrillo RCX con un microprocesador al que se le pueden conectar motores eléctricos, sensores de infrarrojo, sensores de contacto y cuenta vueltas, que también se incluían.

Para esta asignatura sustituimos el *middleware* original del fabricante por sistemas operativos de software libre como BrickOS o LeJOS, que permiten su programación en lenguajes potentes como C o Java respectivamente. En ambos casos el entorno de programación es cruzado: se genera el programa en el ordenador y se descarga vía infrarrojos en el propio robot, que es donde se ejecuta.

Con esta plataforma, los alumnos tenían que construir un robot ensamblando piezas, sensores y actuadores, y programarlo para que completara las tareas de las prácticas:

1) *Pañuelo curvo*: Es una variante del clásico siguelíneas (Figura 7). Se tiene una cinta adhesiva negra sobre un suelo blanco y el robot ha de seguir el camino curvo indicado por la línea. La variante consiste en jugar al pañuelo, situando dos robots en los extremos de la línea y en el medio una lata de refresco pintada de blanco. El robot ha de seguir la línea curva, coger la lata y regresar con ella a su punto de partida. Y todo ello antes de que lo haga su oponente.

Las soluciones a esta práctica utilizan uno o dos sensores de infrarrojos apuntando al suelo, los cuales indican al robot si está sobre la línea o se ha salido y hay que corregir el movimiento para retomar la línea. Dependiendo de la calidad de la corrección el avance es más o menos rápido.

Esta práctica permite programar un controlador reactivo basado en información directa de los sensores. Una de las principales enseñanzas es interiorizar que los sensores reales (los infrarrojos) tienen ruido. El programa no sólo ha de estar pendiente de seguir la línea, también de identificar la lata que hace de pañuelo, para agarrarla, por ejemplo con una pinza motorizada, y dar la vuelta con ella.

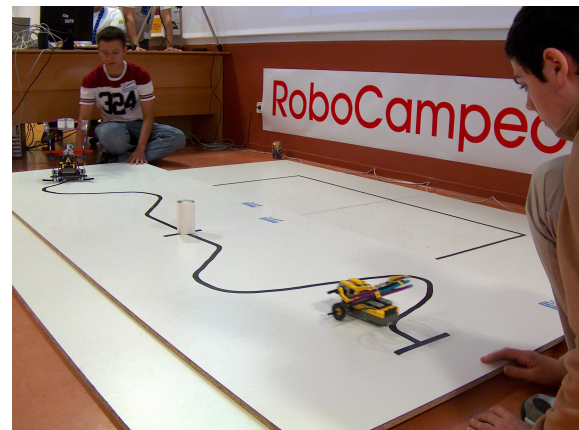


Fig. 7. Práctica del pañuelo curvo.

2) *Navegación real en un laberinto*: En un laberinto de madera como el de la Figura 8 se tiene una zona de salida y otra denominada basurero. Se añaden dos latas de refresco, una en una posición conocida y otra que podría encontrarse en cualquier lugar de la pista. La misión del robot es recoger dichas latas y depositarlas en el basurero. Para hacerlo más entretenido las latas se pintan de blanco o negro y el robot sólo debe llevar al basurero las blancas, dejando las otras en el sitio donde las encuentre.

El robot tendrá que navegar recorriendo el laberinto y a la vez buscar las latas. La mayoría de las soluciones que hemos visto incluyen un barrido sistemático del laberinto, alineándose de vez en cuando contra las paredes para mantener una navegación paralela a ellas, corrigiendo las desviaciones debidas a que los dos motores no giran exactamente a la misma velocidad. A la vez que se barre la zona se está pendiente de la aparición de latas.

La práctica contiene ingredientes reactivos: el robot se puede encontrar en cualquier momento la lata de posición desconocida, debiendo reaccionar ante ella (por ejemplo cogerla con la pinza). También contiene ingredientes deliberativos, puesto que la forma del laberinto y la posición de una de las latas se conoce a priori.



Fig. 8. Laberinto para la búsqueda de latas.

3) *Navegación local con fuerzas virtuales*: Desde el año 2005 hemos complementado las prácticas sobre el robot real con otras sobre simulador. El primer año con SRIsim y desde entonces con el simulador Stage de PSG. Estas nuevas prácticas permiten ilustrar más puntos del temario de teoría que las exclusivamente basadas en el LEGO RCX, cuyo equipamiento sensorial y microprocesador a veces se quedan cortos.

Para facilitar a los alumnos la programación de los robots simulados hemos desarrollado un componente didáctico específico, dentro de la plataforma software que usamos para investigación (llamada *JdeRobot* [29], [2]), que simplifica los detalles de conexión con sensores y actuadores, así como la generación de interfaces gráficas para visualización.

La primera práctica sobre simulador aborda la navegación local segura, basada en información del sensor láser y sensores de odometría. La técnica concreta explorada es el Campo de Fuerzas Virtuales (VFF) [30], donde el destino genera una fuerza atractiva y los obstáculos percibidos con el sensor láser una fuerza repulsiva. La fuerza total es una combinación de ambas, lo que permite una navegación de compromiso capaz de sortear obstáculos mientras se acerca al destino. En la figura 9 se puede apreciar al robot, el punto de destino y las fuerzas virtuales calculadas, todas ellas visualizadas utilizando la herramienta *JdeRobot*. El punto de



Fig. 9. Cálculo de fuerzas virtuales para la navegación local

destino lo especifica el usuario humano a voluntad, picando con el ratón en la interfaz gráfica del programa.

4) *Carreras con navegación híbrida*: La segunda práctica sobre simulador consiste en programar a un robot para que recorra el circuito de la Figura 10 en Stage sin salirse de la pista y en el menor tiempo posible. Los sensores disponibles son nuevamente el sensor láser y la odometría. Los actuadores permiten controlar en todo momento la velocidad de giro y de avance que se ordenan a los motores del robot.

Los recorridos son cronometrados y se premia la competición directa entre los robots de diferentes alumnos, que se sitúan simultáneamente en la pista y realizan carreras entre ellos. Esto permite probar la capacidad de sortear obstáculos dinámicos de los programas y da mayor aliciente a las prácticas. En la Figura 10 se puede apreciar a dos robots, azul y rojo, y las trayectorias que han descrito durante su carrera.

Para poder introducir varios robots en el mismo mundo si-mulado tuvimos que cambiar el simulador monorobot SRIsim por el Stage, que sí tiene capacidad multirrobot.

El objetivo académico de esta prueba es profundizar en la navegación híbrida autónoma, que combina elementos reactivos para sortear obstáculos con elementos deliberativos para seguir una trayectoria deliberativa que lleva al destino remoto, en este caso la meta del circuito. Para evitar diferencias entre grupos y no sobreadaptar la navegación a este circuito concreto dábamos la trayectoria deliberativa requerida instantes antes del examen y era igual para todos.

5) *Navegación basada en visión*: La tercera práctica sobre simulador, en este caso con Gazebo, consiste en programar a un robot para que siga una línea pintada en el suelo empleando únicamente visión monocular. El equipamiento sensorial es sólo una cámara; los actuadores controlan la velocidad de giro y de avance del robot.

Se preparó un mundo virtual en Gazebo con una línea roja en el suelo que recorre un circuito cerrado. En la Figura 11 se puede apreciar el escenario y, sobrepuesta en la parte

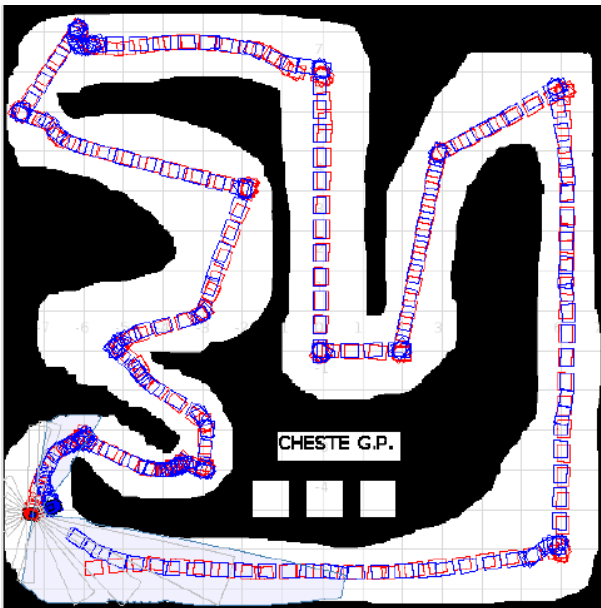


Fig. 10. Circuito de carreras en Stage para que dos robots naveguen con técnicas híbridas

central, la imagen instantánea subjetiva de la cámara a bordo del robot. Nuevamente los recorridos son cronometrados y se fomenta la competición entre los alumnos. Se valoran la rapidez en completar el circuito, que no pierda la línea en ningún momento y la suavidad de movimientos.

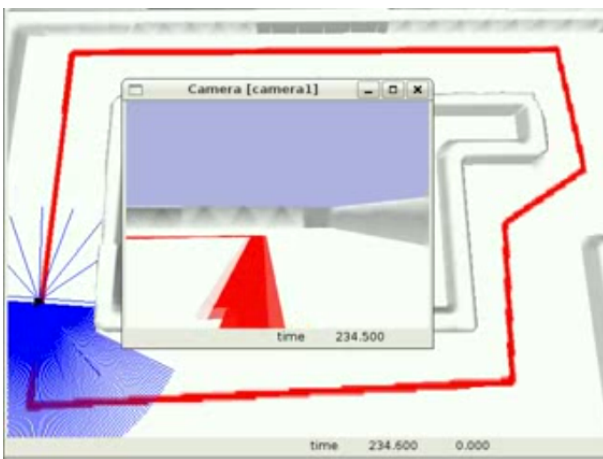


Fig. 11. Escenario en Gazebo para que un robot siga la línea roja usando visión

El objetivo académico de esta práctica es la familiarización con técnicas de navegación basada en visión, en las que la cámara es un sensor más, aunque algo más complejo. También se persigue el ejercicio con técnicas de control realimentado, típicamente controles PID, para navegación reactiva. Una solución habitual es analizar una o varias filas horizontales de la imagen, quedarse con la media de los píxeles rojos y utilizar su distancia al centro de la imagen

como error a minimizar, actuando sobre la velocidad de giro.

#### IV. DISCUSIÓN

El empleo de simuladores de última generación aporta en nuestra opinión varias ventajas académicas a la hora de impartir asignaturas de robótica móvil:

- 1) Extienden el abanico de prácticas posibles. Con grupos numerosos de alumnos, si se quieren hacer prácticas con robots reales, estos tienen que ser necesariamente baratos y por lo tanto con sensores y actuadores limitados. Con ese hardware hay puntos del temario de teoría difíciles de probar. Por ejemplo, con los sensores de infrarrojo y cuenta vueltas del LEGO RCX es complicado practicar la construcción autónoma de mapas, o la navegación deliberativa métrica. El simulador permite utilizar sensores más ricos, como el láser y la odometría más precisa, que serían muy caros en realidad y que son necesarios para una navegación híbrida realista. El empleo del simulador también abre la puerta, por ejemplo, a practicar la construcción de un mapa de obstáculos sin necesidad de resolver la localización, que se toma directamente del simulador.
- 2) El empleo del simulador y un entorno de programación más avanzado que el simple programa empotrado ejecutando en el robot amplía las posibilidades de visualización y depuración. Por ejemplo, la pantalla disponible en el LEGO RCX es muy limitada, y la depuración se hace típicamente con pitidos y escribiendo texto corto en la pantalla. Al utilizar Stage se simplifica la visualización (Figuras 6 y 9). Esto agiliza la depuración de las prácticas enormemente y hace más comprensible el comportamiento del software.
- 3) El empleo de simuladores potentes ha acercado el entorno docente al de investigación. Los simuladores actuales permiten abordar desde las cuestiones más sencillas de iniciación hasta las actualmente activas en la comunidad investigadora (SLAM, etc.). En este sentido, el simulador empleado en la docencia de grado es exactamente el mismo que el usado en la asignatura de postgrado y el que empleamos en la investigación. De este modo los alumnos se familiarizan con herramientas que puede que le sean directamente útiles más allá de su licenciatura.
- 4) Al ampliar el tipo de problemas abordables, permite diseñar un repertorio de prácticas de dificultad variable y con ello ayuda a adaptar su complejidad al perfil heterogéneo del alumnado, siguiendo la metodología de Bolonia. Esto ha sido de utilidad, por ejemplo, en la asignatura "Introducción a la Robótica", al ser de libre elección nos encontramos con un alumnado muy dispar, con estudiantes entre el primer curso y el quinto.
- 5) Permiten el trabajo a distancia. Un problema a la hora de trabajar con robots reales es que no es posible, en la mayoría de los casos, hacer que el alumno disponga de



un robot todo el tiempo que necesite. Con el simulador se permite el trabajo en su casa, no necesariamente en el aula de prácticas. En el caso de trabajos en grupo, permite un desarrollo individual de cada miembro del grupo, para una posterior puesta en común.

- 6) Los simuladores permiten una gran independencia de la tecnología. Así en Alicante ha sido fácil, por ejemplo, cambiar de lenguaje de programación (el mencionado cambio a Java), que hubiese sido más complejo en el caso de usar robots reales (existencia de bibliotecas, licencias, etc.).

Evaluar de forma objetiva estas ventajas es complejo. En el caso de la URJC ha sido posible hacer algunos análisis, puesto que hasta el curso 2005-06 no se empleaban simuladores, se usaban en las prácticas únicamente los robots LEGO Mindstorms.

Un primer indicador a estudiar es el éxito de matriculación. No se observan diferencias significativas por el hecho de usar simuladores: En la URJC se han seguido cubriendo sistemáticamente todas las plazas, que estaban históricamente limitadas a 40 alumnos por el número de robots LEGO disponibles. En la UA, donde el número de optativas es elevado, se mantiene un número no inferior a 50 alumnos por curso académico.

Otro indicador objetivo es la percepción de los alumnos sobre la relación de los contenidos teóricos de la asignatura con las prácticas. Esta relación, de acuerdo con las estadísticas numéricas que realiza la universidad Rey Juan Carlos, ha aumentado en un 20% aproximadamente de los cursos 2003-04 y 2004-05, a los cursos siguientes en los que se emplean nuevos simuladores. Es decir, los alumnos consideran ahora que las prácticas están más relacionadas con el corpus teórico.

Un tercer indicador es el número de temas de teoría directamente aplicados en las prácticas. El temario de la asignatura de la URJC ha incluido tradicionalmente 9 temas: introducción, sensores, actuadores, navegación local, global, construcción de mapas, auto-localización, arquitecturas y visión en robótica. Utilizando únicamente robots reales sólo se realizaban prácticas con los 3 primeros (sin contar la introducción). Con el uso de simuladores se ha pasado a poder realizar prácticas en todos los temas.

Somos conscientes de que los simuladores tienen ciertas limitaciones que sólo los robots reales pueden suplir. Por ejemplo, el ruido y la incertidumbre de los sensores reales, que aunque se pueden simular, no pasa de ser eso, una simulación. Por ello creemos que lo más sensato es una aproximación mixta a las prácticas. Por ejemplo en la URJC se realizan prácticas básicas con un robot real para aprender que los sensores son ruidosos y ver en acción la idea del control autónomo y luego se pasa a un simulador con sensores más ricos (p.e. láser) y comportamientos más elaborados. En la UA se utiliza una estrategia diferente: se realizan las prácticas con simulador, y los alumnos con mejores calificaciones tienen la ocasión de probar alguna de ellas en un robot real (tipo Pioneer).

Otra ventaja de la aproximación mixta es que los alumnos trabajan con dos entornos robóticos diferentes, pero pueden darse cuenta de que en el fondo son similares. En ambos casos hay un interfaz de programación para acceder a los sensores y a los actuadores, y entre medias tienen que escribir su software, que es el que dota de inteligencia al robot, ya sea real o simulado.

#### A. Mirando al futuro

Resulta interesante observar la evolución de los precios de plataformas robóticas. Actualmente se puede conseguir a precio razonable un LEGO NXT (250 euros), un iRobot Create Roomba con *gumstick* (450 euros) o un MobileRobots Amigo (3.000 euros). Adicionalmente los sensores y los procesadores están bajando de precio. En este contexto, ¿tiene sentido trabajar con simuladores o sería mejor usar únicamente plataformas reales?

Atendiendo a las ventajas que hemos comentado: independencia tecnológica, prácticas remotas, variedad de prácticas, etc. nuestra predicción es que el uso de simuladores será generalizado, incluso más allá de los grados en informática. Y podemos usar el argumento de que incluso gigantes de la informática como Microsoft piensan lo mismo que nosotros y por eso han desarrollado su propio simulador de robots.

En cuanto al futuro inmediato, en la URJC este año mantendremos el perfil mixto de las prácticas, pero renovando las plataformas. El antiguo LEGO RCX ha sido reemplazado por el LEGO NXT, que es más robusto y cuenta con un procesador más potente. Las prácticas con simulador seguirán utilizando Stage, pero tenderán a utilizar Gazebo cada vez más, aprovechando su potencia con escenarios tridimensionales y su capacidad para simular cámaras. En particular incluiremos alguna práctica adicional alrededor de navegación guiada por visión y reconstrucción 3D desde un par stereo de cámaras a bordo del robot. En cuanto a la UA, el planteamiento es también seguir con el mismo esquema mixto: prácticas principales con el simulador y prácticas optativas para grupos reducidos con robots reales.

#### AGRADECIMIENTOS

Los autores quieren expresar su reconocimiento al resto de profesores de las asignaturas descritas en este artículo, en particular a Otto Colomina en la UA, y a Carlos E. Agüero y Francisco Martín en la URJC.

Este trabajo ha sido parcialmente financiado por el Ministerio de Educación (DPI2007-66556-C03), por la Comunidad Autónoma de Madrid (S-0505/DPI/0176 y CCG07-URJC/DPI-1694) y por la Generalitat Valenciana (GV06/134).

#### REFERENCES

- [1] **Dave Baum.** *Dave Baum's Definitive Guide to LEGO Mindstorms.* Apress, USA, 1999
- [2] **J. M. Cañas, V. Gómez, P. Barrera and V. Matellán.** *JDEc: arquitectura basada en esquemas para aplicaciones robóticas.* Proceedings of RoboCity2030 1st Workshop on Arquitecturas de Control para Robots, pp 219-232, Madrid, 2007.

- [3] [url] **Xavier** <http://www.cs.cmu.edu/Xavier> Visitado 1 octubre 2008
- [4] **S. Carpin, M. Lewis, J. Wang, S. Balarkirsky, C. Scrapper.** *USARSim: a robot simulator for research and education.* Proceedings of the IEEE 2007 International Conference on Robotics and Automation, pp. 1400-1405, 2007
- [5] **Jeff Craighead, Robin Murphy, Jenny Burke, Brian Goldiez.** *A survey of Commercial Open Source Unmanned Vehicle Simulators.* Proceedings of the IEEE 2007 International Conference on Robotics and Automation, pp. 852-857, 2007
- [6] **Bill Gates.** *A robot in every home.* Scientific American, 296:11, pp. 58-65, Scientific American, 2007.
- [7] **Gerkey, Brian P. and Vaughan, Richard T. and Howard, Andrew.** *The Player/Stage project: tools for multi-robot and distributed sensor systems.* Proceedings of the 11th International Conference on Advanced Robotics ICAR'2003, pp. 317-323, Coimbra (Portugal).
- [8] **J. L. Jones, A. M. Flynn y B. A. Seiger.** *Mobile Robots: Inspiration to Implementation (2nd Edition).* A. K. Peters, Wellesley, Massachusetts (USA), 1998.
- [9] [url] **SRIsim** <http://www.ai.sri.com/konolige/saphira> Visitado 1 octubre 2008
- [10] **K. Konolige, K.L. Myers, E.H. Ruspini, and A. Saffiotti.** *The Saphira architecture: A design for autonomy.* Journal of experimental and theoretical artificial intelligence: JETAI, 9(1):215-235, 1997.
- [11] [url] **Microsoft Robotics Studio** <http://msdn2.microsoft.com/es-es/robotics> Visitado 1 abril 2009
- [12] [url] **SimRobot** <http://www.informatik.uni-bremen.de/simrobot> Visitado 1 abril 2009
- [13] [url] **MobileSim** <http://robots.mobilerobots.com/MobileSim> Visitado 1 abril 2009
- [14] **Thrun, Sebastian.** *Teaching challenge.* Robotics & Automation Magazine, IEEE, vol. 13(4), pp 12-14, 2006.
- [15] [url] **Webots, Cyberbotics** <http://www.cyberbotics.com> Visitado 1 abril 2009
- [16] [url] **K-Team** <http://www.k-team.com> Visitado 1 abril 2009
- [17] [url] **ODE** <http://www.ode.org> Visitado 1 abril 2009
- [18] [url] **URBL, Gostai** <http://www.gostai.com> Visitado 1 abril 2009
- [19] [url] **Player, Stage, Gazebo** <http://playerstage.sourceforge.net/> Visitado 1 abril 2009
- [20] **James Kramer and Matthias Scheutz** *Development environments for autonomous mobile robots: A survey* Autonomous Robots, Vol 22(2), pp. 101-132, 2006.
- [21] **Josep A. Ramon, Josep Ll. de la Rosa, Albert Figueras e Israel Muñoz** *Utilización de los robots móviles en la docencia de control por ordenador XXIII Jornadas de Automática, Universidad de la Laguna, 2002*
- [22] **Flexible virtual and remote laboratory for teaching Robotics F.A. Candelas, F. Torres, P. Gil, F. Ortíz, S. Puente y J. Pomares** *Current Developments in Technology-Assisted Education, pp.1959-1963, (2006)*
- [23] **O. Déniz y A. Falcón** *Robótica móvil: recursos para la docencia* Workshop de Agentes Físicos, WAF-2001.
- [24] **Webcams: An Introduction to Online Robots** K. Goldberg, R. Siegwart. Beyond The MIT Press, (2002)
- [25] **CEA Comité Español de Automática Libro Blanco de la Robótica** CEA Comité Español de Automática y Ministerio de Educación y Ciencia, 2007
- [26] [url] **Curso Robots Autónomos, Univ. Alicante** <https://moodle.ua.es/moodle/course/view.php?id=4> Visitado 1 abril 2009
- [27] [url] **Java Client** <http://java-player.sourceforge.net> Visitado 1 octubre 2008
- [28] [url] **Curso Robótica, Univ. Rey Juan Carlos** <http://gysc.es/moodle/course/view.php?id=11> Visitado 1 abril 2009
- [29] [url] **JdeRobot** <http://jderobot.org> Visitado 1 abril 2009
- [30] **Y. Koren y J. Borenstein** *Potential Field Methods and Their Inherent Limitations for Mobile Robot Navigation* Proceedings of the IEEE Conference on Robotics and Automation, Sacramento, California, April 7-12, 1991, pp. 1398-1404
- [31] **S. Galvan, D. Botturi, A. Castellani, P. Fiorini** *Innovative Robotics Teaching Using Lego Sets* Proceedings of the 2006 IEEE International Conference on Robotics and Automation Orlando, Florida - May 2006
- [32] **J.M.Cañas, C. Agüero, P. Barrera, V. Matellán y R. Morales** *Robótica Móvil y Programación en Educación Secundaria, RoboCampeones 2007* ISBN-978-84-690-9401-3
- [33] **Stewart Tansley** *Microsoft Robotics Studio in Education* RSS Workshop on Research in Robots for Education, 2007
- [34] **Tucker Balch et al.** *Designing Personal Robots for Education: Hardware, Software, and Curriculum* IEEE Pervasive Computing, Vol.7(2), pp.5-9 Apr-Jun 2008.
- [35] **Douglas Blank, Deepak Kumar, Lisa Meeden, Holly Yanco** *Pyro: A python-based versatile programming environment for teaching robotics* Journal on Educational Resources in Computing, Vol 3(4), 2003
- [36] **James Kramer, and Matthias Scheutz** *Development environments for autonomous mobile robots: A survey* Autonomous Robots, Vol. 22(2), pp. 101-132, 2007
- [37] **Thomas Brauml** *Robotics Education using Embedded Systems and Simulations 2007 AAI Spring Symposium.* Robots and Robot Venues: Resources for AI Education, pp. 16-20
- [38] **T. Laue, K. Spiess and T. Röfer** *SimRobot - A General Physical Robot Simulator and Its Application in RoboCup* In A. Bredenfeld, A. Jacoff, I. Noda, Y. Takahashi (Hrsg.), RoboCup 2005: Robot Soccer World Cup IX, Nr. 4020, S. 173-183, Lecture Notes in Artificial Intelligence. Springer 2006



**José María Cañas** es Ingeniero de Telecomunicación (1995) y Doctor Ingeniero de Telecomunicación por la Univ. Politécnica de Madrid (2003). Trabajó durante cinco años en el Instituto de Automática Industrial (CSIC). Ha realizado numerosas estancias de investigación en Carnegie Mellon University y en Georgia Institute of Technology. Actualmente es profesor de la Universidad Rey Juan Carlos, donde dirige el grupo de robótica. Sus intereses giran alrededor de la robótica móvil y la visión computacional.



**Miguel Ángel Cazorla** es Ingeniero en Informática por la Universidad de Alicante (1995) y Doctor Ingeniero en Informática (2000) por la misma Universidad. Desde 1995 es profesor de esta Universidad, ocupando actualmente una plaza de profesor titular de Universidad en el área de Ciencia de la Computación e Inteligencia Artificial. Es director de varios proyectos de investigación y ha publicado multitud de artículos tanto en congresos como en revistas de prestigio. Sus áreas de investigación se centran en la visión por ordenador y la robótica móvil, especialmente en el punto de unión de estas dos áreas. También realiza actividades de divulgación de la ciencia e investigación en docencia.



**Vicente Matellán Olivera** enseñó e investigó desde 1999 hasta enero de 2008 en la Universidad Rey Juan Carlos de la que fue profesor titular de universidad en el área de arquitectura y tecnología de computadores. Anteriormente trabajó durante 6 años también como profesor en la Universidad Carlos III de Madrid. Actualmente es profesor de la Universidad de León. Es licenciado y doctor en informática por la Universidad Politécnica de Madrid. Sus intereses científicos cubren una gran variedad de asuntos, casi todos ellos relacionados con la robótica móvil. Es autor de más de 90 publicaciones científicas en esas áreas y ha participado y dirigido numerosos proyectos en este campo. Es miembro del IEEE (Robotics and Automation Society).