



DOCENCIA EN ROBÓTICA MÓVIL

Antonio Falcón Martel¹, Oscar Déniz Suárez²

¹Catedrático de Ciencias de la Computación e Inteligencia Artificial

²Licenciado en Informática. Estudiante de doctorado
Universidad de Las Palmas de Gran Canaria.

Resumen.- Cada vez son más las posibilidades y recursos con los que cuenta el profesorado universitario para impartir conocimientos en temas de contenido disciplinar en continua fase de cambio y adaptación como es el caso de la Robótica Móvil. Son ya muy pocos los centros que no disponen de alguna plataforma móvil más o menos elaborada. En este trabajo presentamos las experiencias derivadas de la impartición de dos cursos de robótica móvil, uno de ellos desarrollado en un ámbito no universitario. En particular, se describen y analizan las posibilidades y limitaciones de dos productos hardware, el kit Lego®

MindStorms y la plataforma Pioneer 2-DX de ActivMedia Robotics, así como uno software, el entorno Saphira, utilizados en el contexto de las actividades prácticas de los mencionados cursos.

1.- INTRODUCCIÓN

En los últimos tiempos ha aumentado significativamente el número de recursos a la disposición del profesorado universitario para el desempeño de las actividades prácticas asociadas a los contenidos teóricos. En el caso de la robótica móvil, muchos centros cuentan ya con plataformas de diseño relativamente elaborado. Estos recursos, antes dedicados casi exclusivamente a tareas de investigación, de forma natural pueden ahora utilizarse en la propia docencia, la cual mejora en calidad.

En este trabajo se exponen las experiencias e ideas extraídas a raíz de la impartición por los autores de dos cursos de Robótica Móvil. Uno de los cursos constituye la asignatura “Sistemas Robóticos Móviles” de la titulación de Ingeniería Informática, en la Universidad de Las Palmas de Gran Canaria, la cual tiene carácter optativo en cuarto curso de carrera [1]. El otro, que contó con la colaboración de la Consejería de Educación, Cultura y Deportes del Gobierno de Canarias y el Museo Elder de la Ciencia y la Tecnología, se impartió a profesores de enseñanza secundaria con el fin de iniciar la introducción de la robótica móvil en la Enseñanza Secundaria Obligatoria (la concepción de este curso seminal tiene como origen trasladar conceptos, principios e ideas a los niveles inferiores del sistema educativo de forma que se generen aptitudes y vocaciones en el ámbito de la Robótica Móvil).

En primer lugar se describirá y analizará las posibilidades del producto Lego® Mindstorms (Lego es una marca registrada del grupo LEGO.) en el contexto reseñado. Seguidamente indicaremos diversas posibilidades de ampliación del producto estándar, tanto a nivel hardware como software. Las actividades prácticas propuestas en los cursos se describen someramente en la sección de Prácticas. A continuación se analizará el entorno Saphira de desarrollo y simulación de aplicaciones de robótica móvil, así como la plataforma móvil Pioneer de ActivMedia Robotics, a la hora de estudiar experimentalmente sistemas basados en comportamientos. Por último, resumiremos las conclusiones más importantes alcanzadas.

2.- LEGO® MINDSTORMS

El producto comercial Lego® Mindstorms [2]-[5] se presenta como un juguete de coste relativamente medio-alto. El producto responde a las expectativas creadas en este mercado en los últimos años. En un contexto menos lúdico, las características del mismo permiten aplicar hasta cierto punto aspectos prácticos de la robótica móvil, lo cual lo convierte en una herramienta atractiva para la docencia. Prueba de ello son los diversos centros que ya hacen uso de él [6]-[10].

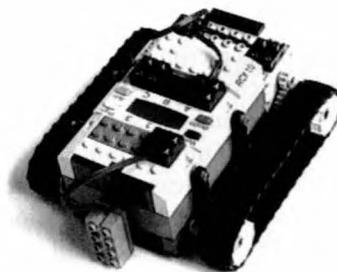


Figura 1. plataforma móvil simple con los sensores del kit.

El elemento central de Lego® Mindstorms es el llamado RCX, que contiene el microcontrolador. El RCX se alimenta con pilas, y alrededor de él se construye el robot con piezas Lego® tradicionales. El RCX dispone de tres

salidas PWM y de 3 entradas a las cuales podemos conectar diversos sensores, incluyendo el kit dos sensores de contacto y un sensor de infrarrojos. El RCX dispone también de un elemento de comunicación por infrarrojos. Lo abierto de las posibilidades de diseño hacen este kit particularmente atractivo. En la siguiente figura se puede ver un ejemplo de plataforma móvil simple con los sensores del kit.

Legó® Mindstorms es un producto versátil basado en el microcontrolador H8/3292. Aunque con ciertas limitaciones en la programación y el hardware accesible externamente, representa una evolución de conceptos similares utilizados con anterioridad, como es el caso del Brick desarrollado en el MIT [11]. La programación del RCX se realiza en un PC. El programa binario se transfiere al RCX mediante una torre emisora/receptora de infrarrojos, que se conecta a un puerto serie. El producto se entrega con un CDROM que contiene dos entornos de programación distintos.

Al utilizar pilas, el robot es autónomo, y puede comunicarse por infrarrojos con el PC o con otro robot.

Uno de ellos es enormemente sencillo, pues permite diseñar el programa en base a bloques gráficos. Sin embargo, para ciertas aplicaciones resulta más flexible programar en el lenguaje NQC [12]. NQC es un lenguaje de sintaxis sencilla, similar al C, que cuenta con un excelente editor, el RCX Command Center [13]. La mayoría de las limitaciones del lenguaje vienen impuestas por el propio hardware del RCX. Entre ellas hay que destacar la falta de números en coma flotante (las únicas variables que existen son los enteros de 16 bits con signo. Además, solo se permite usar un máximo de 32 variables). Por otro lado, la sencillez del lenguaje permite implementar fácilmente comportamientos reactivos básicos. También se puede hacer uso de multitarea, lo cual facilita enormemente la escritura de ciertas aplicaciones de naturaleza muy reactiva [14]. Al utilizar pilas, el robot es autónomo, y puede comunicarse por infrarrojos con el PC o con otro robot. Esto último permite la posibilidad de implementar estrategias básicas de comportamientos de grupo [15]-[16], o de realizar la computación en el PC. Existe un control OCX que puede utilizarse con los lenguajes de programación visual más comunes para comunicarse directamente con el RCX. El sensor de infrarrojos permite implementar aplicaciones sencillas e ilustrativas como seguimiento de líneas, búsqueda de zonas iluminadas o, utilizado conjuntamente con la ventana de comunicación por infrarrojos del RCX, construir un detector de obstáculos.

A pesar de las limitaciones indicadas, creemos que el producto es apropiado para el desarrollo de una parte de

las actividades prácticas asociadas a los contenidos teóricos de una asignatura de introducción a la robótica móvil. Una de las mayores ventajas del producto la representan sus posibilidades de ampliación.

3.- POSIBILIDADES DE AMPLIACIÓN

Las posibilidades de ampliación del Legó® Mindstorms son sustantivas, lo cual ha sido uno de los factores determinantes a la hora de considerarlo positivamente.

Existen numerosas ampliaciones al hardware del RCX que puede construirse uno mismo, entre las que se incluyen diseños para sensores de rotación, temperatura, etc..

Hardware

Las posibilidades de ampliación hardware podemos dividir las en dos: productos comerciales de Legó® y ampliaciones de tipo "hágalo usted mismo". De un lado se dispone del *Ultimate Accesory Kit* de Legó®. Este producto contiene un sensor de rotación de 16 pasos por vuelta (que pueden convertirse en más según la relación de engranajes que se utilicen), un mando a distancia, un sensor de contacto y piezas adicionales. Con sensores de rotación, que también pueden adquirirse por separado, es posible realizar aplicaciones con realimentación del movimiento ("cerrar el bucle"), así como poner en práctica nociones básicas de odometría para una plataforma diferencial. En particular, una de las prácticas asignadas en el curso consiste en realizar el experimento del cuadrado bidireccional [17] y comprobar la reducción de los errores de odometría. Otro producto Legó® de expansión lo constituye el *Vision Command*. Este producto incluye una cámara USB que puede captar secuencias a 30 frames por segundo. Con *Vision Command* pueden ponerse en práctica aplicaciones básicas de visión artificial, como detección de zonas de color, etc [18]. No obstante, su uso parece no estar muy extendido aún, y la calidad del proceso es en ciertos aspectos deficiente, en comparación con otros productos de bajo coste en el mercado.

Existen numerosas ampliaciones al hardware del RCX que puede construirse uno mismo, entre las que se inclu-

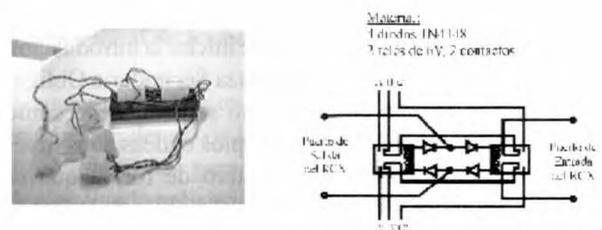


Figura 2. Esquema del multiplexor.

yen diseños para sensores de rotación, temperatura, etc.. Una de las limitaciones del RCX, el tener solo 3 puertos de entrada, puede evitarse fácilmente de varias formas. Una posibilidad es conectar más de un sensor de contacto al mismo puerto de entrada, e incluso conectar un sensor de luz y uno de contacto a un puerto. Para más sensores, existen varias alternativas, como la de construir un multiplexor en el tiempo, como el que mostramos en la figura siguiente. Su funcionamiento se basa en la utilización de uno de los puertos de salida del RCX, y en la capacidad de éstos de adoptar tres estados distintos: polaridad directa, inversa, y desconectado. Con este simple diseño se pueden conectar hasta tres sensores distintos a una entrada del RCX. Cambiando periódicamente el estado del puerto de salida se consigue conectar los distintos sensores al puerto de entrada. Se ha creado un programa para LegOS (ver más adelante) que controla el multiplexor [19]. El multiplexor expuesto destaca por su simplicidad de montaje y bajo coste.

Software

Al igual que en el caso de ampliaciones hardware, existe una gran variedad de herramientas software adicionales, la mayoría de ellas de libre distribución. Es destacable la gran variedad de lenguajes de programación que pueden utilizarse: FORTH, Smalltalk, Java, etc. También hay que destacar la gran cantidad de programas ejemplo y aplicaciones disponibles en código fuente.

Probablemente el esfuerzo más importante de ampliación software del RCX es LegOS [20]. LegOS es un sistema operativo para el RCX, diferente del original de Lego®. La programación para LegOS se realiza en C estándar. Muchas de las limitaciones anteriormente mencionadas desaparecen en LegOS. Entre sus características más atractivas cabe destacar: números en coma flotante, número no limitado de variables, semáforos POSIX y un mayor control sobre el hardware del RCX. Con LegOS se ha podido realizar aplicaciones de control con redes neuronales sencillas. La desventaja fundamental de LegOS estriba en que no es aún un producto muy elaborado, lo cual hace que puedan encontrarse "bugs".

Prácticas diseñadas

En el contexto de los cursos reseñados, el diseño de las prácticas con el kit Lego® Mindstorms se orientó a alumnos cuyos conocimientos en cinemática y dinámica son escasos. Las actividades prácticas diseñadas fueron las siguientes:

- Construcción de un vehículo tipo tanque con dos sensores de contacto y uno de luz y desarrollo de un programa NQC para seguimiento de líneas y evitación de obstáculos. Se construye un circuito con cinta adhesiva negra.
- Desarrollo de un programa NQC que ante la detección de un obstáculo asuma que se ha colisionado con otro

robot e inicie un intercambio de mensajes a través del puerto IR para establecer un protocolo de continuación de movimientos.

- Desarrollo de un programa en NQC o para LegOS para hacer que el robot se acerque a una fuente luminosa, a la vez que evite obstáculos.
- Desarrollo de un vehículo de Braitenberg [21] que cumpla los requisitos de la práctica anterior. La computación necesaria se realiza mediante un perceptrón.
- Realización del test del cuadrado bidireccional. El test permite corregir errores de odometría sistemáticos. En este caso se hace uso de los sensores de rotación ya mencionados, uno por cada rueda.

Con el fin de ilustrar la diferencia entre robots móviles con ruedas y robots con patas, en el curso impartido a profesores de enseñanza secundaria se construyó un robot móvil con 6 patas no independientes. El diseño de este robot puede consultarse en [22]. Con respecto a las prácticas en la asignatura "Sistemas Robóticos Móviles", se formaron 5 grupos, cada uno compuesto de 4 alumnos. Cada grupo se hizo cargo de un kit, responsabilizándose (a principios de curso) por escrito de cuidar todas su piezas y las pilas entregadas. Es necesario señalar que, si bien son comparativamente más caras, las pilas recargables resultan imprescindibles. La actividad práctica de la asignatura correspondía a un total de 30 horas, a razón de dos horas semanales. Las únicas características requeridas por el laboratorio son ordenadores PC con un puerto serie libre, mesas amplias y cargadores de pilas.

Los criterios de evaluación utilizados en la asignatura fueron:

- Un cuaderno de experimentos, donde se recojan los diseños, planteamientos, problemas y estrategias adoptadas, con un recorrido cronológico de la actividad realizada.
- Vídeo explicativo que muestre el funcionamiento correcto del robot móvil según las prácticas propuestas. Con el fin de grabar las secuencias de vídeo se instaló una Web Cam en el laboratorio de prácticas.
- Memoria sintética del trabajo (Descripción del hardware final, tareas que lleva a cabo el robot, software documentado...)

5.- SAPHIRA

Saphira [23] es un entorno de desarrollo y simulación de aplicaciones de robótica móvil. Saphira ya ha sido utilizado con éxito como herramienta educativa en el ámbito universitario [24] y en el investigador. Existen versiones de libre distribución, para diversos sistemas operativos,

con la única limitación de no poder usarlas con un robot físico.

Para poder utilizar Saphira para desarrollar aplicaciones es necesario tener instalado algún compilador de C. Creemos que las posibilidades de simulación de Saphira, así como la posibilidad de utilizar su librería C para crear aplicaciones complejas lo hacen un recurso de gran valor en el contexto que nos atañe. Esta librería permite utilizar funciones de lectura de sonars, sensores de contacto y un sistema de visión opcional. Saphira permite simular y ejecutar las aplicaciones desarrolladas de forma transparente. Existe una versión multiagente de Saphira, capaz de controlar varios robots. Utiliza la arquitectura *Open Agent* (de libre distribución para propósitos no comerciales) que, entre otras posibilidades, permite el control de los agentes a través de Internet.

En el ámbito de los cursos impartidos por los autores, se utilizó Saphira para desarrollar actividades prácticas relacionadas con el control borroso de robots móviles y los sistemas basados en conductas. Entre otras posibilidades, Saphira permite crear comportamientos y combinarlos mediante reglas borrosas para dar lugar a actividades más complejas [25]-[26]. Además, Saphira incluye determinados comportamientos básicos, como detección de obstáculos o movimiento a velocidad constante. Los comportamientos se escriben en una mezcla de C y un lenguaje de especificación

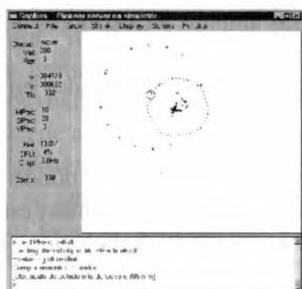


Figura 3. Entorno de desarrollo Shaphira.

de comportamientos borrosos. Saphira guarda las lecturas de los sensores en un modelo de los datos llamado LPS (*Local Perceptual Space*), a partir del cual se computan determinadas variables difusas a utilizar en los comportamientos. Las reglas difusas hacen uso de estas variables para dar como resultado un conjunto difuso de control de los motores del robot. Por último, este conjunto se *defuzzifica* para obtener un valor concreto a enviar a los motores [27]. Al tratarse de un curso introductorio se propuso una actividad práctica muy sencilla que consistía en hacer que el robot merodease y cuando encontrase un obstáculo (utilizando los sonars) empezara a orbitarlo, esto es, a girar a su

alrededor a una distancia relativamente fija. Una vez diseñados los programas, se procede a su simulación en Saphira y, una vez comprobada su validez, se ejecutó con una plataforma móvil física Pioneer (ver figura siguiente). En este nivel es importante que los alumnos tengan conocimientos básicos de lógica difusa.

La plataforma diferencial Pioneer, producto comercial de ActivMedia Robotics, está preparada para funcionar con Saphira y, además de un anillo de sonars y sensores de contacto, dispone de una amplia variedad de accesorios como módem radio, cámaras, láser, etc. Es de destacar la precisión de los codificadores de las ruedas, así como su capacidad de carga.

Además de la posibilidad de escribir aplicaciones de control borroso, Saphira incluye un lenguaje llamado Colbert. Colbert es un lenguaje de control reactivo, de sintaxis parecida a la de C, que permite emplear concurrencia y el paradigma de los autómatas de estados finitos en la programación. La utilización conjunta de Saphira y una plataforma móvil permite poner en práctica conceptos más avanzados como actividades complejas, realimentación visual, construcción de mapas, etc.

CONCLUSIONES

En virtud de lo expuesto en los apartados anteriores, donde se analizan las posibilidades docentes de una solución hardware y una software, se plantea la validez de dichas soluciones para el desempeño de las actividades prácticas asociadas a los contenidos teóricos de un curso universitario de introducción a la robótica móvil.

La solución hardware expuesta no supondría una inversión económica excesiva. Por otro lado, como se ha explicado las limitaciones del producto pueden superarse de diversas formas. En general la respuesta de los alumnos resultó positiva, en tanto en cuanto la actividad práctica se hizo más amena. Es de destacar el entusiasmo que el kit consiguió imprimir a los alumnos.

Actualmente se está estudiando la posibilidad de diseñar actividades prácticas más elaboradas para cursos posteriores, especialmente relacionadas con comportamientos de grupo. La solución software descrita destaca por ser de libre distribución para fines educacionales, así como por la potencia de sus capacidades de simulación.

Debido a la falta de tiempo en la asignatura, no se trató con la profundidad deseada. No obstante, actualmente se utiliza como herramienta fundamental en dos Pro-

yectos de Fin de Carrera relacionados con navegación, lo cual es a su vez un indicador de sus posibilidades.

REFERENCIAS Y BIBLIOGRAFÍA

- [1] Sistemas Robóticos Móviles, Ingeniería Informática, Universidad de Las Palmas de Gran Canaria, <http://serdis.dis.ulpgc.es/~ii-srm>
- [2] The LEGO MindStorms official site, <http://www.legomindstorms.com>
- [3] Dave Baum. «Dave Baum's Definitive Guide to LEGO MINDSTORMS». Apress, 1999
- [4] Dave Baum, Michael Gasperi, Ralph Hempel, Luis Villa. «Extreme Mindstorms: an Advanced Guide to Lego Mindstorms». Apress, 2000
- [5] Jonathan B. Knudsen. «The Unofficial Guide to LEGO MINDSTORMS Robots». O'Reilly, 1999
- [6] Página de la asignatura de robótica, Ingeniería Técnica en Informática de Sistemas, Universidad Rey Juan Carlos, <http://gsyc.escet.urjc.es/docencia/asignaturas/robotica/>
- [7] Lego Lab en la Universidad de Aarhus (Dinamarca), <http://legolab.daimi.au.dk/>
- [8] Curso del departamento de informática de la Universidad de Utrech (Holanda), <http://www.cs.uu.nl/people/markov/lego/index.html>
- [9] Curso de robots LEGO en SUNY/Utica, <http://gozer.sunyit.edu/classes/CSC490/csc490.html>
- [10] Curso de robótica basado en LEGO en la Rice University, <http://www-brazos.rice.edu/elec201/>
- [11] The MIT Programmable Brick, <http://lcs.www.media.mit.edu/groups/el/projects/programmable-brick/>
- [12] Not Quite C, <http://www.enteract.com/~dbaum/nqc/index.html>
- [13] Lego Robots: RCX Command Center, <http://www.cs.uu.nl/people/markov/lego/rcxcc/>
- [14] R. Brooks, «Cambrian Intelligence: The Early History of the New AI». MIT Press, 1999
- [15] R. Arkin, «Behavior-Based Robotics». MIT Press, 1998
- [16] R. Murphy, «Introduction to AI Robotics». MIT Press, 2000
- [17] J. Borenstein, H.R Everett, L. Feng, «Navigating Mobile Robots: Sensors and Techniques». A.K. Peters, Ltd., 1996
- [18] MindStorms RCX Sensor Input Page, <http://www.plazaeearth.com/usr/gasperi/lego.htm>
- [19] Programa de control de multiplexor para LegOS, http://serdis.dis.ulpgc.es/~ii-srm/MatDocen/notas_practicas/legOS_Ejemplos/sensor_mux.c
- [20] LegOS Official Web Page, <http://www.noga.de/legOS/>
- [21] V. Braitenberg, «Vehicles. Experiments in Synthetic Psychology». MIT Press, 1984
- [22] Mindstorms Info Center, <http://www.mi-ra-i.com/JinSato/MindStorms/index-e.html>
- [23] Saphira Robot Control System, <http://www.ai.sri.com/~konolige/saphira/index.html>
- [24] CS224 - Real World Autonomous Systems at Stanford University, <http://www.ai.sri.com/~konolige/CS224/index.html>
- [25] A. Saffioti, E.H. Ruspini, K. Konolige, «Blending reactivity and goal-directedness in a fuzzy controller». Procs. IEEE Intl. Conference on Fuzzy Systems, pp. 134-139, San Francisco, CA, 1998
- [26] D. Kortenkamp, P. Bonasso, R. Murphy (Eds.), «Artificial Intelligence and Mobile Robots. Case Studies of Successful Robot Systems». MIT Press, 1998
- [27] ActivMedia Robotics, «Saphira Software Manual - Saphira version 6.1». ActivMedia Robotics, 1997

AUTORES



Antonio Falcón Martel: Catedrático de Ciencias de la Computación e Inteligencia Artificial. Profesor de las asignaturas de Visión por Computador, Sistemas Robóticos Móviles y Teoría de Sistemas, en el Departamento de Informática y Sistemas de la Universidad de Las Palmas de Gran Canaria. Director del programa de doctorado "Sistemas Inteligentes y Aplicaciones Numéricas en la Ingeniería", en la misma Universidad.



Oscar Déniz Suárez: Licenciado en Informática. Estudiante de doctorado y becario de investigación adscrito al Departamento de Informática y Sistemas de la Universidad de Las Palmas de Gran Canaria. Sus intereses investigadores se centran en robótica, interfaces percepto-efectores, y reconocimiento de caras.