TALLER DE PROGRAMACIÓN DE MINIROBOTS COMO APOYO PRÁCTICO EN LA ASIGNATURA SISTEMAS INFORMÁTICOS DE TIEMPO REAL

L. M. Jiménez; R. P. Ñeco; O. Reinoso; J.M. Azorín; J.M. Sabater
Universidad Miguel Hernández de Elche.
Dpto. Ingeniería. Div. Ingeniería de Sistemas y Automática
E-mail : luis.jimenez@umh.es

Ref: 076

RESUMEN

En este artículo se describen las experiencias realizadas en el uso de pequeños robots móviles en el estudio de los conceptos de programación de sistemas de tiempo real. Estas prácticas se proponen como una herramienta fundamental no sólo para asimilar los conceptos de diseño de aplicaciones de tiempo real sino también para que el alumno desarrolle diversas habilidades necesarias en su formación como Ingeniero.

Se realiza en primer lugar una revisión de la problemática asociada al diseño de los sistemas de tiempo real, presentando a continuación los equipos utilizados, los objetivos planteados y las experiencias desarrolladas durantes los tres últimos años.

Los resultados obtenidos ha supuesto una mejora importante en la docencia de este tipo de asignaturas incrementando la motivación del alumno ante el desafío planteado. Estas actividades se realizan de forma complementaria con la docencia práctica de los sistemas de tiempo real, donde se plantean aspectos más generales asociados a sistemas operativos.

1. INTRODUCCIÓN

La actualización en la docencia práctica en los estudios de Ingeniería ha llevado en los últimos años al planteamiento de actividades motivadoras para el alumno que permitan estimular y completar su formación. En este artículo se describen las experiencias realizadas por el área de Ingeniería de Sistemas y Automática de la Universidad Miguel Hernández en las prácticas de programación de minirobots, enmarcadas dentro del contexto de la docencia de la asignatura de "*Sistemas informáticos de Tiempo Real*". El resultado de las prácticas es el montaje, diseño y programación de minirobots que resuelvan tareas comunes en las aplicaciones de la robótica móvil.

Estas prácticas se proponen como una herramienta fundamental, no sólo para asimilar los conceptos de diseño de aplicaciones de tiempo real, sino también para que el alumno desarrolle diversas habilidades necesarias en su formación como Ingeniero Industrial. Entre estos aspectos podemos citar:

- Conceptos de multiprogramación
- Implementación de especificaciones temporales
- Gestión de dispositivos físicos mediante sensores y actuadores
- Técnicas de diseño
- Conceptos de inteligencia artificial

Así mismo, la interdisciplinaridad de este trabajo permite también potenciar otros aspectos del aprendizaje en Ingeniería:

- Introducir al alumno en la problemática de un proyecto de ingeniería
- Trabajo en equipo
- Practicar y ampliar los conocimientos de diseño electrónico analógico y digital
- Practicar y ampliar los conocimientos de diseño mecánico.
- Iniciarse en los conceptos de control de sistemas

Adicionalmente, se plantea un objetivo fundamental, y es alcanzar los objetivos anteriores mediante el autoaprendizaje. La capacidad de aprender de forma continuada en base a la experiencia propia es un elemento básico de la formación del Ingeniero y en su futuro profesional.

Esta iniciativa tiene su origen en cursos similares en diferentes universidades de Europa y Estados Unidos, especialmente en el "6.270 Contest Course" del MIT [1], que se desarrolla de forma continuada desde el año 1989. La experiencia y el material, tanto bibliográfico como de hardware y software desarrollado, constituye la base de este tipo cursos el multitud de Universidades en los estudios de Ingeniería. En [2],[3], [4] se presentan cursos similares basados en el uso de pequeños robots.

2. ENSEÑANZA DE LOS SISTEMAS DE TIEMPO REAL EN LOS ESTUDIOS DE INGENIERÍA

La renovación de los planes de estudios en los estudios de ingeniería ha abierto espacio docente a nuevas disciplinas que en los últimos años han alcanzado un nivel de desarrollo muy importante y son demandados en los perfiles profesionales. Entre estas disciplinas se encuentran el estudio de los Sistemas de Tiempo Real, enmarcado anteriormente en cursos de postgrado y líneas de investigación pero que supone actualmente una parte importante en la formación del futuro ingeniero.

Los sistemas de tiempo real abarca un amplio rango de aplicaciones, pero constituyen un elemento importante en la formación del Ingeniero dada su multidisciplinaridad, con especial énfasis en el especialista en control que tiene que trabajar con variables físicas que son realimentadas para modificar el comportamiento natural de los sistemas, con una necesidad de corrección lógica y temporal en su actuación.

Se presenta en el siguiente apartado un pequeña introducción de los conceptos fundamentales asociados al tiempo real cuyo aprendizaje es el objetivo planteado en este artículo.

3. DEFINICIÓN DE UN SISTEMA DE TIEMPO REAL

Se pueden encontrar diversas definiciones formales de lo que es un sistema de tiempo real, pero en todas ellas se destacan dos aspectos que diferencian y definen tales sistemas [5]:
Por un lado se trata de sistemas en los que el tiempo de respuesta es un aspecto clave. Es decir, no importa solo que sea capaz de generar un resultado correcto sino que éste se produzca en un tiempo determinado. De nada sirve, por ejemplo, que una algoritmo de control de navegación calcule la corrección de trayectoria óptima si esta llega después de que el sistema haya colisionado contra un obstáculo.
El segundo aspecto importante en la definición de un sistema de tiempo real es que debe responder ante estímulos generados por el entorno dentro de un periodo de tiempo finito. Es decir, un sistema de tiempo real interactúa con el entorno (mundo físico real) adquiriendo estímulos y estados del entorno, y generando una acción sobre dicho entorno.
Las dos características antes mencionadas no solo clasifican a los sistemas de tiempo real, sino que determinan también la metodología de diseño. La programación de estos sistemas rompe con el esquema tradicional de programación secuencial, exigiendo por tanto un cambio de mentalidad en la forma de pensar acerca del software. La dificultad no está tanto en la programación de los diferentes componentes, sino en comprender como hacer que trabajen juntos ante la aparición de eventos no predecibles, fallos y la propia concurrencia entre los mismos.
Un sistema de tiempo real es, por tanto, un sistema basado en un computador que debe resolver diferentes aspectos de forma simultanea: rápida respuesta; operación continua; reacción ante estímulos; fallo en los componentes o en sus conexiones; incertidumbre acerca de los retardos de procesamiento y comunicación; posible distribución geográfica; y la posible necesidad de adaptarse a lo largo del tiempo (mientras está operando) ante cambios de requerimientos y circunstancias. Un sistema de tiempo real resuelve estos aspectos por medio de la colaboración entre componentes que individualmente no tienen un conocimiento del sistema en su conjunto.
La utilización de pequeños robots móviles permite plantear la problemática general del diseño de un Sistema de Tiempo Real en una aplicación limitada pero muy estimulante para el alumno que encuentra en el autoaprendizaje un de sus objetivos más importantes.

4. DESCRIPCIÓN DE LOS EQUIPOS UTILIZADOS

Como equipo básico para la realización de las prácticas se optó por el robot RugWarrior Pro desarrollado en el MIT por J. Jones y A. Flyn como un ejercicio práctico de integración de las tecnologías asociadas al diseño de robots de forma asequible, y fácilmente reproducible. Se trata de un kit educacional muy sencillo basado en el microcontrolador de 8 bits Motorota

68HC11 disponiendo de un entorno de desarrollo en C que permite la programación concurrente.
El kit se complementa con un excelente libro [2] que describe de forma detallada los conceptos fundamentales de diseño y programación del robot. Destacar también el libro y los trabajos de Fred Martin [3] que constituyen un manual completo para este tipo de cursos, y que está basado en una plataforma similar.
El robot está constituido por los siguientes elementos (figura 1):

- **La estructura mecánica**: se trata de un esquema mecánico diferencial accionado por dos motores de corriente continua y con sensores de colisión.
- **Tarjeta Controladora**: Se trata de un microcontrolador de 8 bits Motorola 68HC11 equipado con 32K de memoria RAM estática. Dispone de sensores de proximidad, fotocélulas, micrófono, pantalla LCD y encoders. Dispone, así mismo, de dos drivers de 1A para motores de continua controlados mediante modulación PWM. Existen entradas analógicas y digitales para incorporar sensores adicionales.
- **Entorno de programación IC 3.2**: Entorno de programación y depuración de código C para el µC68HC11. Permite el desarrollo de la aplicación en un derivado del lenguaje C sobre un PC, realizando la descarga y depuración de los programas a la tarjeta controladora mediante un puerto serie.

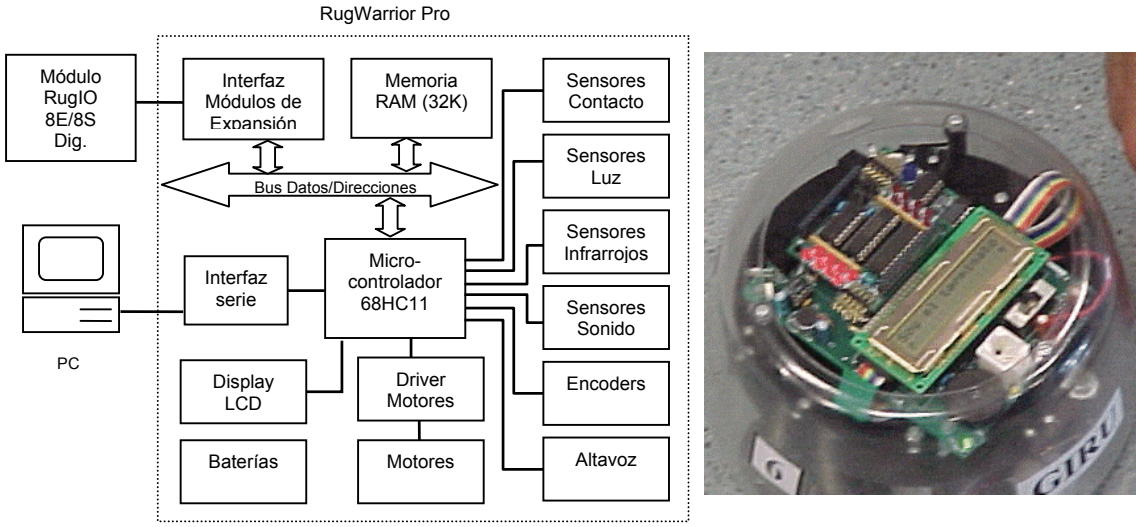


Fig. 1- Diagrama de bloques del robot RugWarrior Pro

4.1. ENTORNO INTERACTIVE C (IC)

Interactive C (IC a partir de ahora) es un entorno de desarrollo para microntroladores de la familia Motorola 68HC11. Se trata de un entorno que se diseñó pensando en el control de pequeños robots, por lo que incorpora de forma nativa muchas funcionalidades necesarias para accionar motores eléctricos, servomotores, leer sensores de proximidad, distancia, etc.
Otro aspecto importante es que incorpora un planificador multitarea, permitiendo ejecutar de forma concurrente varios procesos. IC permite ejecutar código interpretado o la ejecución de código compilado.
IC fue desarrollado por investigadores del Media Lab del MIT con el objetivo de disponer de una herramienta sencilla que permitiera a los estudiantes integrar conceptos de inteligencia

artificial en pequeños robots. Son especialmente famosos los concursos de robots que tuvieron su origen en este centro y que utiliza IC como herramienta básica.
El lenguaje base utilizado por IC es un derivado del **lenguaje C**. El lenguaje C es especialmente potente por su capacidad de gestionar recursos de bajo nivel (necesarios para programar microcontroladores), a la vez de poder realizar una programación estructurada de alto nivel. La implementación de C utilizada en IC no es completa, eliminándose aquellos aspectos que se consideraron innecesarios (recordemos que los microcontroladores tienen pocos recursos de memoria por lo que deben ser bien aprovechados). En todo caso incorpora toda la funcionalidad básica con todas las estructuras de control de alto nivel.
A parte del lenguaje C, el entorno IC permite incorporar de forma sencilla código **ensamblador** para aquellas tareas más críticas (gestión de interrupciones...), permitiendo intercomunicar el código C y el código ensamblador mediante variables globales.
El entorno IC está formado por los siguientes módulos:

- Entorno de desarrollo y depuración (IDE): que se ejecuta sobre un PC (DOS, Windows) o una estación UNIX: incluye un editor, un compilador de C, ensamblador y un interprete para depuración. Incorpora asimismo funciones para descargar el software sobre el microcontrolador por medio de un puerto RS-232 (figura 2).
- Sistema operativo para el microcontrolador 68H11 (pcode): gestiona la comunicación con el PC, integra un depurador interactivo manejado desde el PC, permite la descarga de aplicaciones y, sobre todo, integra las funciones básicas de control de todas las entradas salidas y el planificador multitarea.

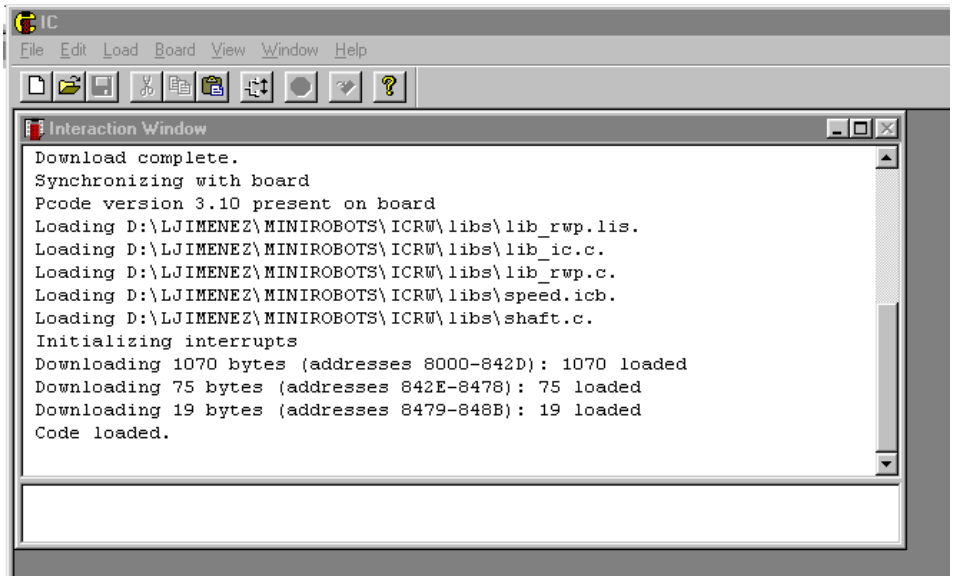


Fig. 2- Entorno de desarrollo IC

5. TALLERES REALIZADOS

La docencia se estructura en una serie de talleres que van introduciendo al alumno en los conceptos básicos de programación de robots y el uso del equipo específico utilizado. Los talleres iniciales tienen fundamentalmente un contenido de prácticas guiadas, para pasar en los talleres finales a la propuesta de objetivos a desarrollar mediante autoaprendizaje.
La docencia se estructura en 4 talleres:

Taller 1: Introducción a IC
Taller 2: Manejo se sensores y actuadores

Taller 3: Control reactivo de robots
Taller 4: Proyectos individuales

En la página web indicada en la referencia [6] se puede encontrar una descripción detallada de cada taller y el material necesario. Se describen a continuación los aspectos más relevantes:

5.1. TALLER 1 : INTRODUCCIÓN A IC

Tiene como objetivo familiarizarse con el robot y el entorno de programación del microcontrolador 68HC11. Se presenta el uso básico de la consola IC, la descarga del sistema operativo, la compilación y descarga de programas y la depuración de los mismos.
Se trata de un taller complemente guiado donde se propone la ejecución de un programa de ejemplo sencillo manejando la pantalla y los motores.

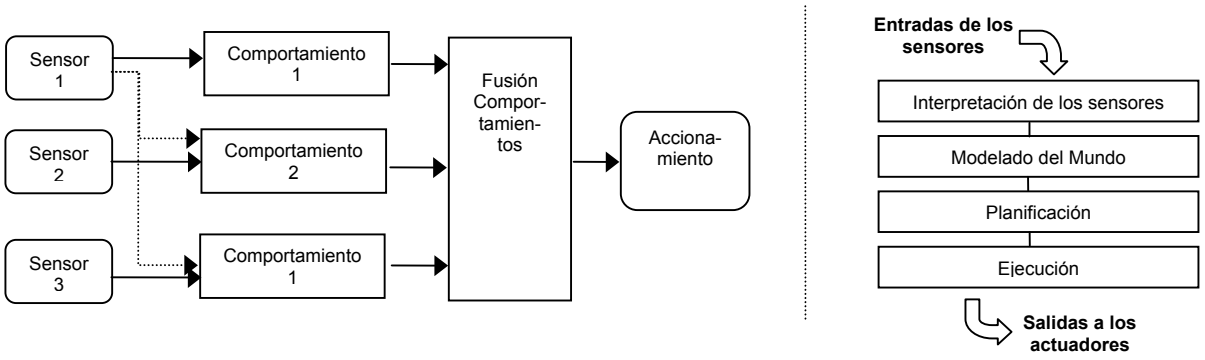
5.2. TALLER 2 : MANEJO DE SENSORES Y ACTUADORES

En este taller se repasan en primer lugar las características más destacadas y específicas de lenguaje IC. Se plantea el uso básico de la librería C para el manejo de sensores y accionamientos del robot. Se completa, como ejercicio práctico, con la calibración de los diferentes sensores y actuadores.

5.3. TALLER 3 : CONTROL REACTIVO DE ROBOTS

Este taller tiene como objetivo presentar una visión general de las estrategias utilizadas en el diseño de sistemas de control de robots, centrándose en el esquema de control reactivo basado en la fusión de comportamientos y su implementación en IC mediante programación concurrente.
Dentro del campo de la robótica existen multitud de algoritmos y estrategias utilizadas en el control del robots. Las diferentes propuestas pueden clasificarse en dos tipos de estrategias de control denominadas [3][7]:

- Control Planificado (control basado en modelos)
- Control Reactivo (control basado en la fusión de comportamientos)



a) b)

Fig. 3- a) esquema de control reactivo, b) esquema de control planificado

El *control Planificado* tiene su origen en las primeras aplicaciones de la robótica y está basado en la obtención de un modelo preciso del entorno en el que se debe desenvolver el

robot. A partir de este modelo del mundo real se realiza una planificación que determina las acciones que tiene que ejecutar el robot en cada instante para alcanzar su objetivo (figura 3.b). Este tipo de esquema *modelado-planificación* tiene la ventaja de obtener un comportamiento óptimo. En cambio puede presentar problemas en ciertas aplicaciones debido a la gran tiempo de procesamiento, la capacidad de almacenamiento necesaria y la exigencia de un modelo preciso y actualizado del mundo. El mundo real es demasiado complejo y tiene demasiados detalles como para obtener modelos exactos de mismo. En muchos casos se reduce esta complejidad considerando el entorno estático lo cual reduce la capacidad de adaptarse cambios no previstos. Como se puede observar, se trata de un proceso secuencial en el cual no es preciso ningún paralelismo.

El *control reactivo* en cambio no utiliza un modelo del mundo para ejecutar las acciones del robot. En este caso cada sensor actúa de forma de forma *refleja* generando uno o varios comportamientos. La fusión de los comportamientos generados por cada sensor determina la acción de control sobre los actuadores y por tanto el comportamiento final del robot. Este tipo de estrategia permite una reacción ante los estímulos procedentes de los sensores en tiempo real (figura 3.a)

Esta estrategia de control se adapta perfectamente a la programación de pequeños robots ya que son equipos que disponen de una limitada capacidad de procesamiento y almacenamiento. Además, en muchos casos no es preciso un modelo del mundo para actuar en él con un comportamiento inteligente.

Cada uno de los comportamientos activados por los sensores operan temporalmente en paralelo por lo que es preciso poder ejecutar tareas concurrentemente. El problema de los conflictos entre los datos generados por diferentes sensores se maneja como un problema de conflicto entre comportamientos por lo que debemos establecer un mecanismo de fusión de la red de comportamientos. La figura 3.a muestra un esquema de este tipo de control.

Como se puede observar este esquema de control establece un conjunto de *capas de comportamiento* y un mecanismo de prioridades en el acceso a los accionamientos, el cual regula la fusión de los comportamientos. Se trata de una arquitectura inherentemente paralela que puede ser implementada mediante sistemas multitarea (mono o multiprocesador). No precisa de un modelo del mundo por lo que el consumo de memoria es pequeño. El código de cada comportamiento puede ser sencillo y la arquitectura altamente modular, pudiendo ser extendida fácilmente para incorporar nuevos comportamientos y sensores.

5.4. EJEMPLO: IMPLEMENTACIÓN DEL CONTROL REACTIVO EN EL ROBOT RUGWARRIOR

Se presenta a continuación un primer ejemplo que deben realizar los alumnos de implementación de los conceptos de control reactivo utilizando procesos concurrentes. Siguiendo el esquema visto anteriormente se detallan los bloques de entrada de los comportamientos (sensores) y los bloques de salida (actuadores)

Sensores	Actuadores
3 Sensores de Contacto 2 Fotocélulas 2 Sensores Infrarrojos 2 Encoders Incrementales 1 Micrófono	2 Motores Pantalla LCD Altavoz

Tabla 1.- Sensores y actuadores del Robot.

En la figura 4 se muestra un ejemplo del esquema de control reactivo propuesto para el robot RugWarrior Pro. En el disponemos de cinco comportamientos asociados a diferentes sensores.

- Comportamiento **Crucero**: el propósito de mantener en movimiento el robot.
- Comportamiento **Seguimiento**: dirige el robot hacia la zona más luminosa.
- Comportamiento **Evasión**: evita obstáculos detectados mediante los sensores infrarrojos.
- Comportamiento **Colisión**: actúa frente a posibles choques.
- Comportamiento **Sonido**: detecta ordenes sonaras externas.

La fusión de todos los comportamientos genera un comportamiento global complejo, de forma que el robot busca la luz rodeando o evitando obstáculos, activándose o desactivándose con palmadas. El mecanismo de fusión es un simple supresor multinivel que inhibe el comportamiento inferior si se activa el superior.
El alumno debe implementar el código de cada uno de los comportamientos como un proceso que se ejecuta indefinidamente que genera una acción ante la medida obtenida de un sensor. La fusión es implementada mediante un proceso que establece el mecanismo de prioridades y se comunica con el resto de procesos mediante variables globales.

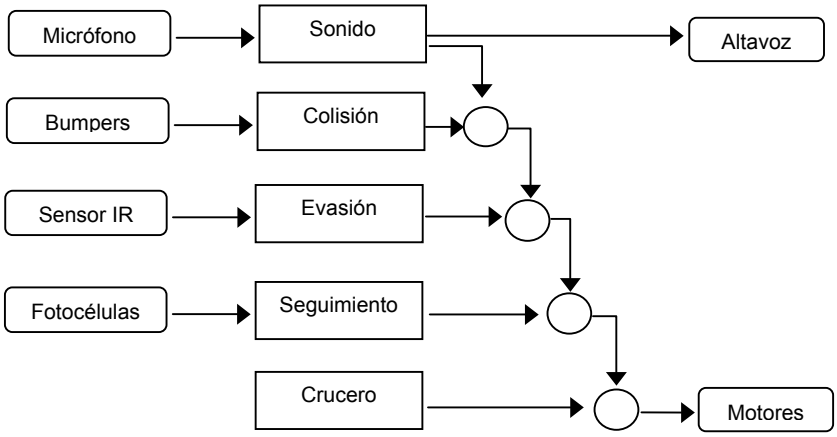


Fig. 4- Ejemplo de arquitectura de control reactivo para el robot RugWarrior Pro

5.5. TALLER 4: PROYECTOS INDIVIDUALES

En este último taller se plantean diversos desafíos a los alumnos que exige la puesta en práctica los conocimientos adquiridos tanto en el manejo del robot, como en los conceptos de diseño de una aplicación de tiempo real, enfrentándose a un proyecto de ingeniería. Los proyectos incluyen aspectos de programación y también de diseño electrónico por lo que

supone un estímulo adicional al coexistir en el tiempo con los estudios de esta materia. Entre las propuestas planteadas en los diferentes cursos cabe destacar [6]:

- **Transmisión de comandos entre robots mediante modulación infrarroja y sonora**: se ha utilizado la transmisión de comandos sonoros en juegos de persecución implementando un protocolo sencillo de asentimiento. Los alumnos desarrollaron elementos auxiliares para mejorar la sensibilidad del micrófono incorporado (figura 5.b). Nuevos trabajos han sido realizados utilizando el sensor y emisor infrarrojos permitiendo una comunicación más compleja.
- **Búsqueda de la salida de un laberinto evitando obstáculos**: este es una de los trabajos iniciales que les permite a los alumnos asimilar los conceptos de control reactivo para la navegación autónoma. Sobre esta base los diferentes grupos añaden la detección de marcas o la generación de mapas (figura 5.d)
- **Generación de mapas de obstáculos** : tiene como objetivo que el robot "cartografíe" el laboratorio. Constituye un proyecto individual como introducción a las técnicas de planificación de trayectorias y la utilización de sensores odométricos (encoders).
- **Juego entre varios robots. "tu la llevas"**: actúa como desafío entre varios grupos generando diferentes comportamientos de huida y persecución, exigiendo la adición de las posibilidades de comunicación comentadas previamente para identificar los obstáculos y los contrincantes (figuras 5.b, 5.c y 5.e).
- **Juego de Fútbol**: este proyecto supone el diseño hardware y software por parte de los alumnos. Por un lado se diseñó una pelota que genera señales infrarrojas, la integración de sensores infrarrojos adiciones en los robots y el diseño de estrategias de juego para dirigir la pelota a la portería. En la figura 5.a se muestra el diseño realizado por los alumnos durante e el curso 2001.
- **Seguimiento de líneas**: este proyecto es un clásico en los concursos de robots y supone la integración de sensores adicionales (fotoreflectores, fotocélulas) para guiar el robot a elevada velocidad por un camino marcado en el suelo mediante una línea negra. (figura 5.f).
- **Campeonato de sumo**: otro concurso clásico del cual se celebra anualmente un concurso oficial en la UMH.

Adicionalmente a los proyectos anteriores, que son los más atractivos para los participantes, se han propuesto en los diferentes cursos otros desafíos en los que los alumnos se han encontrado con la dificultad de llevar a la práctica su ideas iniciales debido a la complejidad de los objetivos planteados. En todo caso, ha supuesto para los alumnos una experiencia importante al enfrentarse a problemas de inteligencia artificial cuya resolución práctica constituye en mucho casos temas de investigación. Cabe destacar los siguientes proyectos:

- Navegación en entornos no estructurados
- Comportamientos coordinados entre dos robots
- Aprendizaje mediante premios y castigos

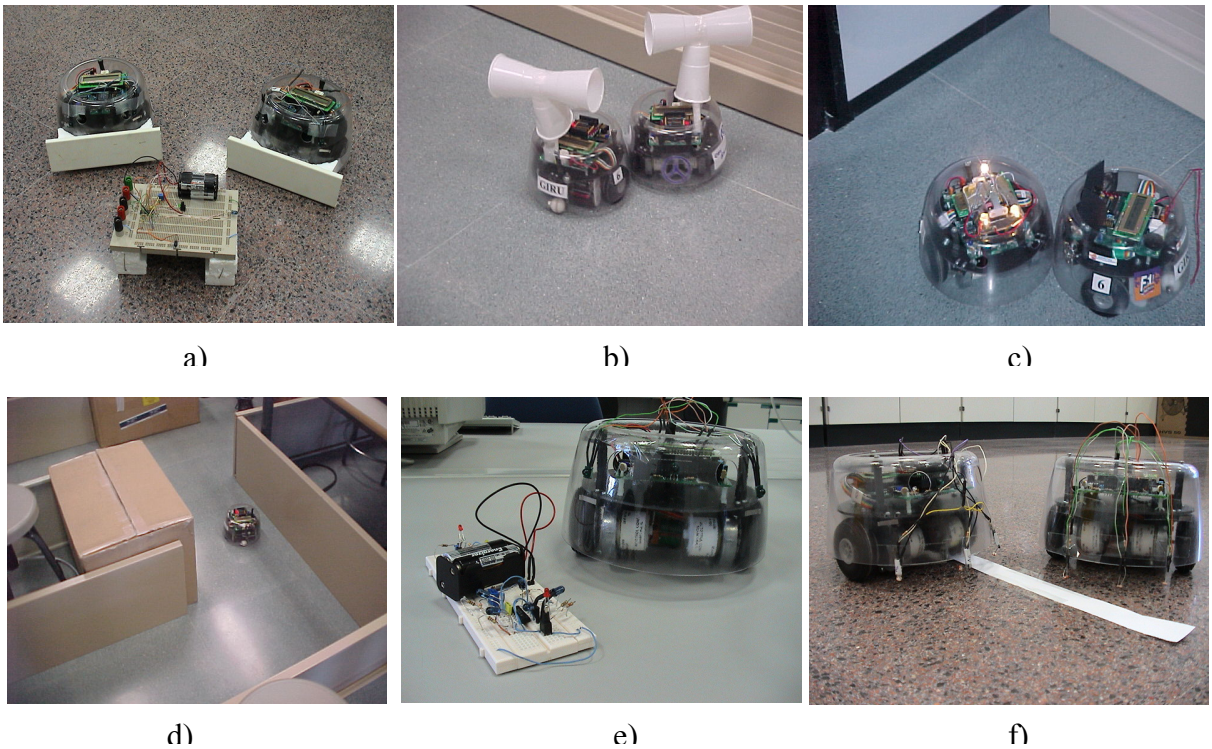


Fig. 5- Ejemplos de proyectos realizados por los alumnos

6. CONCLUSIONES

El uso de pequeños robots supone un desafío para el alumno de ingeniería que permite motivarlo en unas prácticas cuyo objetivo es asimilar los conceptos fundamentales de los sistemas de Tiempo Real. Esta docencia actúa de complemento a las prácticas puramente software sobre sistemas operativos.

Supone un punto de partida importante para el alumno que se adquiere una formación básica en los aspectos de programación, control, inteligencia artificial, diseño electrónico, etc. Estos conocimientos le permiten al alumno la participación más provechosa en los concursos abiertos de diseño de robots que están proliferando en las diferentes Escuelas de Ingeniería.

Por último destacar la motivación que supone para el alumno este tipo de prácticas estimulando el autoaprendizaje.

7. REFERENCIAS

[1] Martin, F. "6.270 Contest Course", Massachusetts Institute of Technology, Cambridge, MA, USA (fredm@media.mit.edu) (*http://www.mit.edu:8001/courses/6.270/home.html*)
[2] J. Jones, A. Flynn, "Mobile Robots: Inspiration to Implementation" Ed. A.K. Peters.
[3] Martin, F. "Robotic Explorations. A hands on introduction to engineering", Prentice Hall.
[4] Angulo, J.M., "Microbótica" Ed. Paraninfo
[5] Burns, A. Wellings, A., (1997) "Real time systems and Programming Languages", Addison Wesley.
[6] Servidor Web curso Minirobots ISA-UMH (http//lorca.umh.es/isa/es/temas/minirobots)
[7] Arkin, R., "Behavior-Based Robotics" MIT Press