

e-Laboratorio de robótica: un sistema multirobot distribuido

Josep Fernández Ruzafa¹, Raúl Marín Prades¹

¹Dpto. de Ingeniería de Sistemas, Automática
e Informática Industrial
Universitat Politècnica de Catalunya
c/ Jordi Girona 1-3. - 08034 Barcelona
Josep.fernandez@upc.edu

²Dpto. Ingeniería y Ciencias de los Computadores
Universitat Jaume I
Campus de Riu Sec
08034 Castelló de la Plana
raul.marin@icc.uji.es

Resumen

Fiabilidad, simplicidad y bajo coste son tres requisitos difíciles de combinar en Robótica. A pesar de ello, esta combinación tiene un gran interés para las instituciones educativas. Nuestro trabajo se dirige hacia el desarrollo de un sistema multirobot basado en plataforma de bajo coste.

Tomando como base los simples robots de LEGO®, estos son mejorados agregándoles un módulo de expansión para superar sus principales limitaciones, y que permite integrarlo en un sistema de programación externo al robot (“*off-board*”). A partir de esta plataforma, se ha implementado un e-laboratorio organizado en cuatro niveles de procesamiento distribuidos, que permiten a los usuarios gestionar, controlar y programar un sistema multirobot a través de Internet.

1. Introducción

En los últimos años, un gran número de productos comerciales está ayudando a promover la robótica, y haciendo los robots cada vez más accesibles, tanto para las instituciones educativas, como para el público en general.

Entre estos productos destaca los que permiten la construcción de sistemas robot de una forma flexible, abordando los aspectos más significantes de la robótica: el diseño de la mecánica, la programación y la percepción del entorno. Sin duda, el gran éxito de productos como *LEGO Robotics Inventor System* es debido a su gran flexibilidad y su bajo coste.

El LEGO-RIS se encuentra disponible en centros comerciales y tiendas especializadas, y existe una importante comunidad de usuarios activos que dispersan sus ideas y proyectos en Internet. La cibercomunidad de LEGO-RIS desarrolla herramientas para la creación de proyectos, la programación y simulación de

robots, así como propuestas de nuevas estructuras mecánicas de robots. Además, un número importante de instituciones educativas usan LEGO-RIS, a pesar de sus limitaciones, como una herramienta válida por promover la creatividad del estudiante y poner en práctica los conceptos más importantes de la robótica.

Como es de esperar de un producto de bajo coste, algunas restricciones limitan la complejidad del sistema robot que se puede llevar a cabo usando LEGO-RIS. Entre ellos, podemos destacar cuatro aspectos como son:

1. La memoria para los programas y los datos es muy limitada. Una de las principales características de un sistema robot, particularmente en los robots autónomos, es la adquisición, proceso y almacenamiento de la información del entorno proporcionados por los sensores del robot (sensores externos o de entorno). El RCX (éste es el nombre dado por LEGO al microordenador utilizado por el robot) tiene sólo aproximadamente 5 Kbytes de memoria para los programas de usuario y datos, una vez que se ha cargado el *firmware*. Por consiguiente, la capacidad computacional y de memoria ofrecidos por el RCX son insuficientes para resolver la mayoría de los desafíos que se presentan en el campo de los robots autónomos.
2. Ofrece un número reducido de entradas (3) para la adquisición de información sensorial y salidas (3) para la generación de las señales que permiten controlar la estructura mecánica (ruedas, patas, brazos articulados, pinzas). El número de salidas parece ser suficiente, dado que la simplicidad de los componentes que constituyen el kit de construcción no permite construir robots con estructuras complejas (de más de 3 grados de libertad). Por otro lado, las tres entradas para sensores ofrecidas por el RCX, a las que se han de conectar los sensores

interiores y/o externos, son insuficientes en la mayoría de las configuraciones del robot. Una configuración simple de robot necesita de dos sensores internos, tipo codificadores ópticos incrementales, y sensores externos, tales como los parachoques, sensores de luz o sensores de distancia. Por ejemplo, una configuración del robot móvil con sólo 2 ruedas motrices y que permita la detección de colisión, necesita cuatro entradas para los sensores.

3. Las limitadas prestaciones del sistema de comunicación. El canal de comunicación entre el RCX y el ordenador esta basado en un puerto de infrarrojo. Las funciones principales del sistema de comunicación son la transmisión de los programas desde el ordenador hacia el robot, la gestión del robot y su supervisión. Aunque es una conexión inalámbrica (usa un transmisor / receptor de luz infrarrojo), su alcance está limitado (aproximadamente 10 metros), y requiere una alineación correcta (los circuitos infrarrojos en ambos lados – del ordenador y del RCX - tiene que estar enfrentado y alineados para que la comunicación sea posible). Estas características junto con su baja velocidad de transmisión hacen necesario la frecuente intervención humana en un sistema que, a priori, se ha diseñado para trabajar de forma autónoma.
4. El kit LEGO-RIS sólo permite construir sistemas con un único robot. Por consiguiente, no permiten la creación de sistemas multirobot.

La presente comunicación describe el diseño e implementación de un sistema que mejora las prestaciones ofrecidas por un robot educacional de bajo coste. Las prestaciones del robot se mejoran en varios aspectos: la complejidad de los sistemas mecánicos y sensoriales del robot, la capacidad de programación, y en el tipo de aplicaciones que puede realizarse, haciendo posible abordar la realización de un sistema multirobot.

2. Especificación

En el proceso de mejora se han establecido algunas premisas básicas:

- La expansión del robot es no destructiva. El módulo de expansión es un módulo autosuficiente que puede agregarse fácilmente para mejorar las capacidades de RCX originales, o extraerse, recuperando así el RCX su función original. En todo momento, se puede recuperar el sistema original.
- Fácil de implementar. Esta expansión se dirige a la comunidad interesada en robótica en general, incluyendo jóvenes y aficionados, como un producto que puede ser construido por uno mismo.
- Uso de componentes estándares y de amplia comercialización. La comunidad de la robótica debe encontrar fácilmente la información sobre los componentes que forman la expansión y de su uso.
- Bajo coste. Para promover su uso, y como consecuencia de las dos decisiones anteriores.
- Uso de tecnologías inalámbricas. Con el propósito de no limitar la maniobrabilidad del robot y aumentar las dimensiones de la zona de movimiento del robot.
- Fácil de programar y de uso amigable. La programación del robot debe sustentarse en un lenguaje de programación de propósito general.
- Soporte para sistemas Multirobot. El sistema tiene que poder controlar simultáneamente varios robots.

2.1. Mejora del hardware

Los principales aspectos a ser mejorados con el módulo de expansión son: 1) aumentar el número y tipo de entradas del sensor, y de salidas para el control de actuadores, 2) mejorar la fiabilidad de las comunicaciones -la distancia entre el robot y ordenador, omni-direccionalidad, tasa de error -, y 3) aumentar el poder de computación -velocidad de cálculo y memoria para programas y datos-.

2.2. Mejora del software

La programación del RCX esta restringida principalmente por sus escasos recursos computacionales, un microcontrolador con 32 Kbytes RAM, de los que una vez cargado el *firmware* del robot, sólo están disponibles aproximadamente 5 Kbytes de memoria para los programas del usuario y los datos. Por lo tanto, las

aplicaciones implementables con estos recursos han de ser relativamente simples.

Se pueden distinguir tres partes principales en un programa de robot, a saber, *percepción*, *planificación* y *acción*. En un programa de robot, estas tres componentes corresponden a los *datos* en los que se guarda la información del entorno, *los algoritmos de decisión* que computan a partir de la información del entorno que acciones tienen que ser realizadas, y finalmente, *las acciones del robot* ejecutadas (instrucciones relacionadas con la actividad del robot, como movimientos del robot o la percepción activa del entorno).

Por lo tanto, un programa de robot se puede dividir en dos partes: 1) acciones que se deben realizar obligatoriamente en la plataforma del robot, las *acciones del robot* o "*on-board*" y 2) acciones relacionadas con el procesamiento de los datos obtenidos mediante los sensores y la planificación, que no tienen que ser necesariamente ejecutadas en la propia plataforma del robot (acciones "*off-board*").

Para garantizar la potencia de cálculo, la simplicidad de uso y la fácil interrelación con aplicaciones orientadas a Internet, se usan Java y LeJos como los lenguajes de programación de apoyo. LeJOS es un *firmware* para el RCX que implementa una Máquina Virtual de Java (JVM).

2.3. Arquitectura del sistema

El sistema propuesto está compuesto por cuatro componentes (figura 1):

- Ordenador de usuario (o remoto). Se utiliza para el desarrollo del programa, la gestión del robot, la transmisión de rutinas "*on-board*" hacia el robot, el almacenamiento de la

información sensorial, y la ejecución de programa de aplicación del robot (programas "*off-board*", de usuario o externos).

- Ordenador de enlace. Está a cargo de encaminar los mensajes entre el ordenador remoto (usando TCP/IP) y los robots (usando un canal inalámbrico Bluetooth y un protocolo propio).
- Módulo de expansión. Este módulo, que está basado en un microcontrolador, está a cargo de: 1) adquirir la información de los sensores de entorno y enviarla hacia el ordenador remoto, 2) generar las señales de control hacia los actuadores -como pinzas o mecanismos específico-, 3) realizar un puente de comunicación inalámbrica y fiable entre el ordenador de enlace y el RCX, y finalmente, 4) seleccionar los mensajes destinados a un robot específico, dentro del sistema multirobot.
- RCX. Dependiendo del mensaje recibido, el RCX ejecuta las rutinas asociadas, ya sean de movimiento simple (como movimiento adelante, movimiento hacia atrás, giro a la izquierda, etc.), de adquisición y procesamiento de la información de los codificadores incrementales asociados a las ruedas para realizar trayectorias basadas en odometría (p.e. avanzar una distancia especificada, o girar sobre el eje del robot un valor de grados indicados), de generación de las señales PWM para el control de los actuadores asociados a las ruedas del robot, y, de envío de un mensaje al ordenador remoto, vía el módulo de expansión, cada vez que se completa una acción del robot.

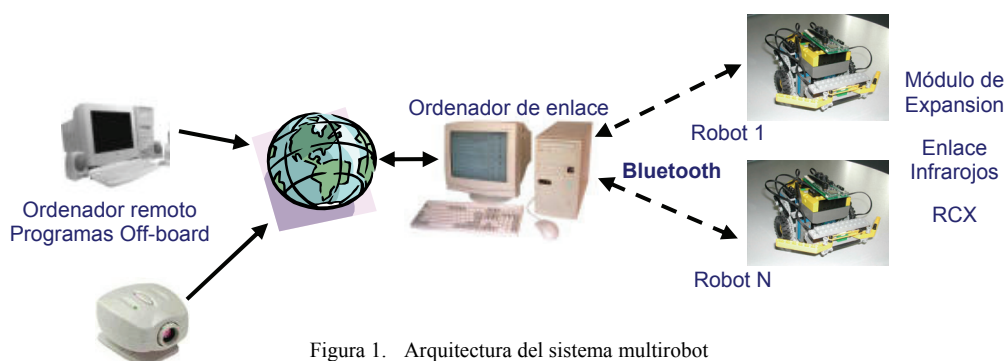


Figura 1. Arquitectura del sistema multirobot

Los mensajes desde el ordenador remoto hacia los robots son, básicamente, órdenes de movimiento, y consultas del estado del robot. En la otra dirección, desde los robots hacia el ordenador remoto, los mensajes contienen actividad de los sensores e información del estado de robot.

Con el propósito de superar las restricciones de memoria del RCX, el software se divide en dos partes. Las rutinas, que se ejecutan “a bordo” del robot (en el RCX), de control y generación del movimiento del robot y de lectura de los codificadores, y los programas, que se ejecutan en el ordenador remoto (el del usuario del sistema), de aplicación del robot. El microcontrolador que se encuentra en el módulo de expansión se dedica a la actividad de los sensores externos y como puente de comunicación con los robots.

3. Implementación del sistema

3.1. Hardware

El objetivo del presente trabajo es proporcionar a los estudiantes una infraestructura de fácil uso para experimentar con un sistema multirobot. Con el propósito de facilitar su difusión se establece como premisa el uso de componentes de amplia difusión, con una buena comercialización y bajo coste, para la implementación del sistema. Una decisión importante es la elección de la solución Bluetooth. Para romper con las limitaciones de una conexión mediante cable, se opta por remplazar el cable por un sistema como el adaptador inalámbrico de Bluetooth a RS-232, el BlueCore o el 3e-250. Estos dispositivos simulan una conexión serie mediante el estándar RS-232 usando la tecnología Bluetooth.

Inicialmente se estudian tres alternativas, para la aplicación en el sistema:

- a. Un ordenador de enlace que usa un dispositivo Bluetooth (un circuito cerrado), y un módulo de expansión basado en un chip Bluetooth de cierta complejidad (con puertos de I/O, USART, etc.), y con la capacidad de comunicar con el RCX vía un puerto de infrarrojos.
- b. Un ordenador de enlace con un dispositivo Bluetooth, y un módulo de expansión con un microcontrolador y un chip Bluetooth (que debe ser integrado en un circuito de comunicación).

- c. Un ordenador de enlace que usa un dispositivo Bluetooth, y un módulo de expansión que usa ambos, un microcontrolador y un dispositivo Bluetooth.

La opción b ofrece un mejor compromiso entre el coste, prestaciones, flexibilidad y facilidad de integración con el RCX. Por tanto, los principales componentes del módulo de expansión serán un microcontrolador y un chip Bluetooth. Los requisitos básicos para el microcontrolador son:

1. Una alta frecuencia de reloj con el propósito de reducir la latencia en las comunicaciones.
2. Bajo consumo.
3. Un número elevado de entradas y salidas, incluyendo recursos como entradas para interrupciones externas y entradas analógicas.
4. Disponer de como mínimo dos USART (Universal Síncrono & el Receiver & Transceiver Asíncrono), con el propósito de simplificar la comunicación entre el chip Bluetooth y con el módulo de infrarrojos.

Podemos encontrar varios microcontroladores con estas características, por ejemplo, AN2131QC o AN2131SC de Cypress, Z80180088VSC o Z8018110FEC de Zilog y PIC18F6520, PIC18F6310 o PIC17C752 de Microchip.

En una primera versión del sistema se selecciono el PIC17C752 dado su bajo coste, el gran número de entradas y salidas que ofrece, y por formar parte de una gama de microcontroladores que goza de gran popularidad y de fácil obtención. Las dificultades al combinar estos componentes en un circuito integrado fiable hicieron reconsiderar esta decisión, orientándonos hacia placas de desarrollo comerciales. En este caso, un criterio adicional a tener en cuenta es que la placa de expansión se sitúa sobre el robot, en concreto sobre el RCX, y por lo tanto han de ser de pequeña dimensión y peso reducido.

Entre la gran oferta de placas de desarrollo, pocas son las que ofrecen conectividad Bluetooth con bajo consumo, coste y dimensiones reducidas. Nuestra elección ha ido hacia la propuesta de Gumstix, que propone unas plataformas con un microprocesador Intel XScale® PXA255, con 64MB de RAM, 16MB de memoria Flash y un módulo Infineon Bluetooth. Asimismo, es posible conectar un sistema de comunicación por infrarrojos (el utilizado por el RCX) utilizando

uno de los puertos serie que ofrece la plataforma. La figura 2 muestra la arquitectura del módulo de expansión.

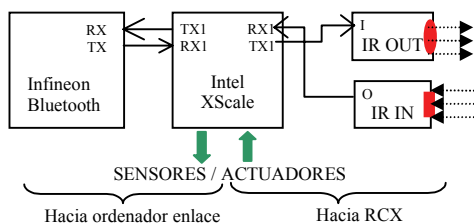


Figura 2. Componentes del módulo de expansión

3.2. Firmware

Como hemos indicado anteriormente, las principales funciones realizadas en el robot (módulo de expansión + RCX) son la supervisión de la actividad de los sensores externos, el control del movimiento del robot y comunicar cada cambio del estado del robot al ordenador remoto. El módulo de expansión lleva a cabo una parte significativa de este trabajo. La información del entorno es adquirida por sensores conectados al módulo de expansión, y los codificadores asociados a los motores se conectan al RCX. El RCX sólo se encarga de generar y controlar las trayectorias del robot (genera las señales de PWM para los motores y lee los codificadores que permiten determinar la velocidad de giro de las ruedas y obtener información odométrica).

Para identificar un robot dado en el sistema multirobot, se asigna un código numérico (el identificador de Robot) a cada robot. Todos los mensajes tienen en su cabecera el identificador del robot al que va dirigido el mensaje o del robot que envía el mensaje. Cuando el ordenador remoto envía un mensaje a través del ordenador de enlace, esta información va directamente al módulo de expansión de todos los robots activo en el sistema a través del canal Bluetooth. Cuando el identificador de robot que aparece en el mensaje recibido corresponde al identificador de robot el microprocesador del módulo de expansión, entregan el mensaje al RCX a través del canal IR con pequeñas modificaciones. Finalmente el RCX seleccionado ejecuta la acción indicada.

Por otro lado, un robot inicia la comunicación con el ordenador remoto en dos situaciones. El

módulo de expansión toma la iniciativa de enviar un mensaje cuando un sensor del robot cambia su estado o cuando se completa un movimiento limitado. En este último caso, el RCX envía un mensaje al módulo de expansión y, si es posible, el módulo de expansión envía un nuevo paquete a través del canal Bluetooth hacia el ordenador remoto.

Resulta importante destacar que no todas las comunicaciones van desde el ordenador remoto hasta el RCX (o viceversa). Por ejemplo, los mensajes relacionados con la actividad de los sensores externos van entre el ordenador remoto y el módulo de expansión. Así es, a cada cambio del estado de un sensor, el microprocesador comunica el nuevo estado del sensor al ordenador remoto, y por lo tanto, el RCX no recibe información relacionada a la actividad de los sensores externos. Esto simplifica el funcionamiento, ya que el microprocesador verifica todos los eventos, y determina la prioridad de la comunicación teniendo en cuenta la actividad sensorial. El segundo caso a contemplar, se produce al finalizar un movimiento limitado (avanzar D cm. o girar G grados). Ya que el RCX es el encargado de generar los movimientos del robot, cuando un movimiento finaliza, el RCX envía un mensaje al ordenador remoto vía el módulo de expansión para comunicar la finalización. En la sección 4 se incluye más información sobre el protocolo de comunicación.

Las rutinas “a bordo” y programas se reparten entre el microprocesador del módulo de expansión y el RCX. Teniendo en cuenta el hardware presente en cada nivel, el microprocesador del módulo de expansión ejecuta las rutinas para el encaminado de mensajes (incluida la conversión requerida entre Bluetooth y el medio IR) y la adquisición de la información relacionada con los sensores de entorno del robot. Cada vez que un mensaje llega al módulo de expansión proveniente del ordenador remoto, determina, analizando el identificador de operación presente en la cabecera del mensaje, si la acción tiene que ser ejecutada por el microprocesador del módulo de expansión o por el RCX. En el último caso, el mensaje recibido se reenvía al RCX.

Cuando el valor generado por un sensor externo cambia, el módulo de expansión envía un mensaje al ordenador remoto donde se indica el identificador de robot, el identificador del sensor y

su nuevo valor. Esta información es utilizada por los programas de aplicación (“*off-board*”) para actualizar la representación del entorno del robot. Cuando el módulo de expansión recibe un mensaje del RCX, este reenvía el mensaje al ordenador remoto al que se agrega una cabecera con el identificador de robot que origina el mensaje.

La otra parte de las rutinas a bordo corre en el RCX. La rutina principal del RCX espera hasta la recepción de un mensaje a través del puerto de IR. Entonces desempaqueta el mensaje y ejecuta la acción del robot indicada. Cuando la acción del robot es un movimiento limitado (una trayectoria) basada en odometría, el RCX envía un mensaje de confirmación al finalizar la acción.

4. Diseño del software

La arquitectura del software esta basada en una estructura maestro-esclavo, y se organiza en 3 niveles:

1. El nivel de aplicación. Éste es el nivel más alto que se encarga de controlar los robots enviando mensajes de movimiento y control hacia el nivel de expansión del robot(s) deseado(s).
2. El nivel de expansión. Este nivel junto con el nivel de RCX, implementa al esclavo del sistema (el robot). Dependiendo del mensaje recibido, diferencia si la acción tiene que ser ejecutada en este nivel (gestión de sensores de entorno) o tiene que ser reenviado hacia el nivel RCX (comandos de movimiento y estado).
3. El nivel RCX. Es el nivel más bajo y sólo ejecuta comando de movimiento usando un juego de rutinas primitivas que se describen usando el lenguaje de programación LeJos.

4.1. El nivel de aplicación

Los usuarios desarrollan sus programas de aplicación de robot usando el lenguaje de programación Java. De esta forma, los programas del robot pueden aprovecharse de todas las ventajas de Java, como manejar grandes estructuras de datos y/o la implementación de algoritmos complejos. El sistema presentado soporta el uso de un sistema de robots en red. Por tanto, uno o más programas de aplicación, ejecutándose en uno o más ordenadores, pueden

controlar una colonia del robot. La única restricción es que un robot es controlado por un único programa de aplicación. La comunicación entre los programas de aplicación de robot esta basada en el mecanismo RMI de Java. La figura 3 muestra una configuración típica de este sistema. Al nivel de aplicación, los programas en Java están a cargo de recopilar la información sensorial, guardarla, y basándose en esta información, desarrollar un plan para resolver las tareas del robot. Es en este nivel que se integran los robots, creando así un sistema multirobot.

Como resultado de la ejecución de los programas de usuario, se envían mensajes, que indican las acciones del robot, hacia el nivel de expansión de uno o más robots. La comunicación hacia el nivel de expansión pasa por el ordenador de enlace que entrega el mensaje al robot, usando el enlace Bluetooth. Cada mensaje produce una acción del robot, ya sea en el módulo de expansión o en el RCX. La tabla 1 muestra las relaciones entre los mensajes enviados desde los programas de aplicación y las acciones realizadas en el robot. En la otra dirección, los programas de usuario reciben los mensajes provenientes de los robots. Estos mensajes contienen información de los sensores que permiten construir y/o actualizar una descripción del entorno del robot en el (los) ordenador(es) remoto(s).

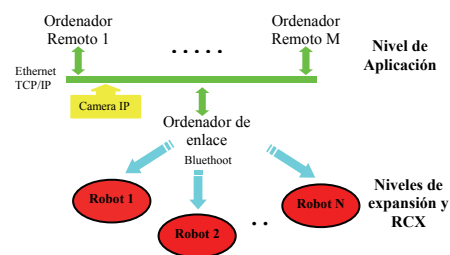


Figura 3. Configuración del sistema

4.2. El nivel de expansión

Este nivel, junto con el nivel de RCX, implementa la parte esclavo del sistema (el robot). En este nivel se ejecutan las instrucciones primitivas de lectura de los sensores de entorno y control de los actuadores de mecanismos. El resto de primitivas (de movimiento y consulta del estado del robot) se ejecutan en el nivel de RCX.

El nivel de expansión ejecuta el programa descrito a la sección 3.2. La comunicación entre los niveles de expansión y RCX se realiza mediante infrarrojos para respetar la premisa de no destructividad. El software de módulo de expansión está escrito en C, y se ejecuta en el microprocesador de la plataforma Gumstix.

Nivel Aplicación	Nivel Expansión	Nivel RCX
// Op's Movimiento		
Stop(rid)	→	setPow(motor,0)+ setPow(motor,0)
forward(vel,dist,rid)	→	setPow(motor,pow)+ forward(pulses)
Bacward(vel,dist,rid)	→	setPow(motor,pow)+ Backward(pulses)
forward(vel,rid)	→	setPow(motor,pow)+ forward(-1)
Bacward(vel,rid)	→	setPow(motor,pow)+ Backward(-1)
TurnRight(vel,deg,rid)	→	setPow(motor,pow)+ TurnRight(pulses)
TurnLeft(vel,deg,rid)	→	setPow(motor,pow)+ TurnLeft(pulses)
setEncoder(nenc,rid)	→	setEncoder(nenc)
GetEncoder(nenc,rid)	←	getEncoder(nenc)
SetRobotId(rid)	→	setId(rid)
Reverse(rid)	→	reverse()
//Op's Entrada-Salida		
getPort(npt,rid)	←	GetPort(npt,value,rid)
setPort(npt,value,rid)	→	SetPort(npt,value)
ActivePort(npt,mc,rid)	→	ActivePort(npt,mc)
getVar(nvar,rid)	←	GetVar(nvar,rid)
setVar(nvar,rid)	→	SetVar(nvar)
NewS(npt,value,rid)	←	NewS(npt,value,rid)
//Op's Timer		
setTimer(ntimer,rid)		
startTimer(ntimer,rid)		
stopTimer(ntimer,rid)		
// Op's Estatus		
getState(rid)	←	getState(value)
setState(value,rid)	→	setState(value)
NewState(value,rid)	←	NewState(value)
MovComp(pulses,rid)	←	MovComp(pulses)

Tabla 1. Correspondencia entre comandos y primitivas de bajo nivel, y su nivel de ejecución.

4.3. El nivel de RCX

Éste es el nivel más bajo y sólo ejecuta las instrucciones de movimiento utilizando un conjunto de primitivas implementadas mediante LeJos. El nivel de RCX está básicamente a cargo de generar el movimiento del robot. En este nivel, las rutinas primitivas generan las señales PWM que controlan la velocidad de los motores y leen los codificadores ópticos incrementales asociados

a las ruedas, con el fin de generar y controlar las trayectorias realizadas por el robot.

La recepción de un mensaje de movimiento implica la ejecución de una o más rutinas "on-board". Asimismo, el RCX envía un mensaje hacia el nivel de aplicación cada vez que se completa un movimiento limitado.

Los 32 Kbytes de memoria RAM disponibles en el RCX se usan para el *firmware* de LeJos, y para guardar las rutinas que llevan a cabo las primitivas de movimiento. Los comandos de movimiento (resumidos en la Tabla 1) puede ser clasificado en dos categorías: los movimientos limitados basados en información de odometría con el propósito de controlar velocidad del robot y la distancia recorrida (avanzar adelante o girar una distancia o grados indicados) y movimientos ilimitados que usan la información de odometría para controlar sólo velocidad del robot. Las principales acciones realizadas por el RCX son la activación de motores a un sentido y velocidad específico, la lectura de los codificadores asociados a los motores para controlar trayectorias avance y giro, y algunas instrucciones para la configuración de los codificadores y la inicialización. La tabla 1 muestra la relación entre los comandos a nivel de aplicación y las rutinas primitivas a nivel del RCX.

4.4. Estructura de los mensajes

Todos los mensajes que circulan entre el ordenador remoto y el módulo de expansión son empaquetados en una palabra de tamaño finito (en la implementación actual 24 bits). Los mensajes están compuestos por los siguientes campos:

- El Identificador del robot. Define el robot al que el mensaje tiene que ser entregado, o el robot que envía el mensaje hacia el nivel de aplicación. Se utiliza un valor de identificador de robot igual a cero para transmitir un mensaje a todos los robots activos en el sistema (todos los robots procesan un mensaje "broadcast"). El identificador del robot se codifica en 3 bits, y por lo tanto, el sistema puede manejar un máximo de 7 robots.
- El identificador de función/sensor. Indica la acción que debe ser ejecutada por el robot(s) seleccionado(s), o el código de sensor/estatus en un mensaje de respuesta. El identificador

de función se codifica en 5 bits, y por lo tanto, se pueden definir un máximo de 32 acciones.

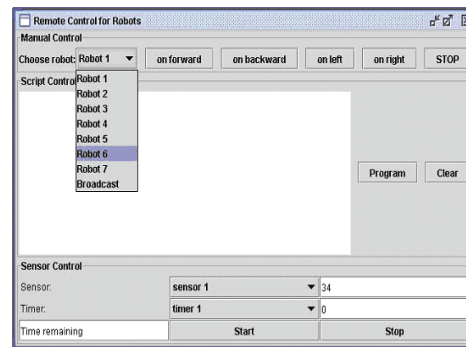
- Parámetros de operación. Se utilizan para indicar los valores de aquellas funciones que tienen asociados parámetros. Este campo del mensaje (16 bits) puede ser subdividido si es necesario. Por ejemplo, en un movimiento de avance limitado -forward(vel,dist,rid)-, se requieren dos parámetros. En este caso se usan tres bits para definir la velocidad del movimiento (8 velocidades posibles) y 13 bits para indicar la distancia a recorrer -hasta 8192 pulso del codificador(es)-. Los parámetros son extraídos de la palabra del mensaje en el nivel apropiado cuando son necesarios.

5. Experiencias y aplicaciones

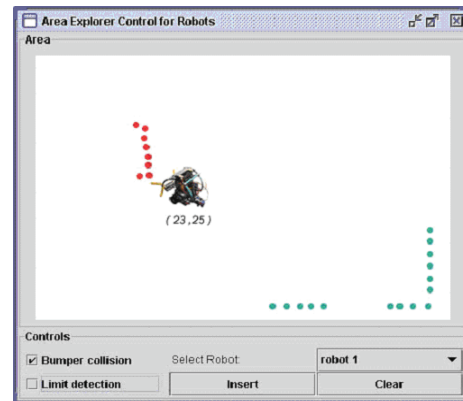
El sistema multirobot presentado, basado en robots para el ocio y la educación de bajo coste, ha sido diseñado para promover la creatividad e interés de los estudiantes. En la versión más simple del sistema, un grupo de robots es controlado por un único programa de usuario. En la figura 4.a se muestra una interfaz gráfica de usuario (GUI) simple para el control de un grupo de 7 robots, usando dos alternativas diferentes: 1) Control manual - una lista desplegable permite seleccionar el robot a operar, y cinco botones permiten realizar movimientos simples con el robot seleccionado -, 2) Control mediante *scripts* - la interfaz ofrece una área de texto para escribir programas simples, usando la sintaxis de LeJos, y los transmite al robot seleccionado para su ejecución. Además la interfaz dispone de una zona donde se visualiza el valor de los sensores del robot seleccionado. En este caso, la interfaz se usa para escribir programas de LeJos que finalmente se ejecutan por completo en el RCX, y para supervisar la actividad de los sensores de robot.

Una aplicación más compleja es el robot explorador (Figura 4.b). En este ejemplo la tarea del robot consiste en obtener un mapa de un área delimitada utilizando odometría y los sensores de entorno del robot. Los puntos rojos del mapa corresponden a las posiciones en las que el parachoques del robot ha detectado una colisión. Los puntos verdes indican los límites del mundo en el que evoluciona el robot, marcados por líneas de color en el suelo que son detectadas mediante un sensor de luz infrarroja por reflexión. Por

último, la aplicación muestra la posición actual estimada del robot. En este ejemplo, los algoritmos de planificación y los datos de la descripción del mundo, se encuentran en el ordenador del usuario.



(a)



(b)

Figura 4. Ejemplos de Aplicaciones

- Sistema multirobot centralizado.
- Robot explorador. Interfaz de usuario

En una primera experiencia, para probar el sistema propuesto y verificar sus prestaciones y posibilidades, se crearon 4 equipos de trabajo cada a los que se les proporcionó un robot. Se propusieron dos tareas a resolver: Un sistema multirobot colaborativo (la exploración de un mundo desconocido y el desplazamiento cooperativo de un conjunto de objetos) y un sistema multirobot competitivo (localizar el máximo número de objetos dentro de un recinto en un tiempo fijo). Las dos aplicaciones requieren cuatro programas de usuario ejecutándose en el

nivel de aplicación. En la primera tarea, todos los equipos tienen que compartir la información de sensores y del estado del robot para obtener cuatro planes (uno para cada robot) compatibles. Por el contrario, en la segunda tarea los programas de usuario no comparten información, pero es necesario resolver conflictos, como por ejemplo determinar si una colisión se ha producido contra un obstáculo o contra otro robot.

6. Conclusiones

El sistema propuesto permite realizar experimentos en un sistema multirobot de bajo coste (incluyendo el equipo *Robotics Inventor System de LEGO* más el módulo de expansión) su coste es inferior a los 400€ por unidad. Este hecho hace fácil su uso en las instituciones educativas. Con este sistema los estudiantes pueden experimentar en los campos de los sistemas multirobots, los enjambres de robots de inspiración biológica (*swarm robots*), las técnicas de la inteligencia artificial aplicada a la robótica o la planificación multiagente.

Los trabajos futuros se orientan a mejorar la fiabilidad y disponibilidad del sistema. Estos dos aspectos son de gran importancia para hacer factible el uso remoto del sistema multirobot a través de Internet, sin las intervenciones continuas de un operador humano.

Agradecimientos

En la realización de este trabajo han participado Joan Oliva y Pedro López. La realización de este trabajo está financiada por el Ministerio de Educación y Ciencia a través del proyecto CYCIT TSI2004-05165-C02-02.

Referencias

- [1] J. Fernandez and A. Casals, "Open Laboratory for Robotics Education" Proc. 2004 IEEE Int. Conf. on Robotics and Automation, pp. 1837-1842.
- [2] J. Fernandez, J. Aranda, A. Casals, "Some Reflections on Robotics Curricula" Workshop on Robotics Education & Training 2001, pp 131-136.
- [3] LEJOS official website, <http://lejos.sourceforge.net/>
- [4] LEGO Internal website, <http://www.crynwr.com/lego-robotics/>
- [5] J. Oliva and J. Fernandez "Entorn de Programació de Robots Mòbils amb Java", PFC FIB 2005.
- [6] Servidor Gumstix Inc. <http://www.gumstix.com/>

