

# Un sistema didáctico integrado para la enseñanza introductoria de robótica

Jorge Dignani<sup>1,2</sup>

Joaquin Lima<sup>1</sup>

Ricardo Coppo<sup>3</sup>

Renato Mazzanti<sup>1,2</sup>

Rodrigo René Cura<sup>1</sup>

Claudio Delrieux<sup>4</sup>

Gustavo Samec<sup>1,2</sup>

Ricardo Kölln<sup>1</sup>

1 Facultad de Ingeniería - Universidad Nacional de la Patagonia San Juan Bosco, Sede Pto. Madryn

2 Centro Nacional Patagónico – (CENPAT) CONICET

3 Dept. de Ingeniería Electrónica, Universidad Tecnológica Nacional, Facultad Regional Bahía Blanca

4 (Autor para correspondencia) Departamento de Ingeniería Eléctrica y Computadoras - Universidad Nacional del Sur - claudio@acm.org - Parcialmente financiado por la SECyT

## RESUMEN

Se presenta un sistema informático para la enseñanza introductoria de la robótica en escuelas de nivel medio y universitario. Está formado por un robot armado con componentes de bajo costo y por un conjunto de aplicaciones que permite programar y simular situaciones tales como la edición y resolución de laberintos.

**Palabras claves:** robótica educativa, reconocimiento de formas, laberintos.

## CONTEXTO

Esta línea de Investigación forma parte del Proyecto SIRE “Sistema Integrado de Robótica Educativa” de la UNPSJB en donde intervienen también investigadores de la UNS.

## 1. DESCRIPCIÓN DEL PROYECTO

Desde hace varios años se trabaja en la UNPSJB en algunas materias de programación avanzada con software de simulación de robots, guerra de robots y RoboCode de IBM [1] [2] [3]. Dado que

muchos alumnos y profesores tienen una formación en electrónica, se decidió la construcción de un robot telecomandado desde una PC. Así nació el BlueBot [4]. En este proyecto se construyó un robot simple que es controlado en forma inalámbrica desde una PC. Esto posibilita que toda la programación de control resida en esta máquina, y por lo tanto se puede implementar la cantidad necesaria de abstracciones en la misma, de manera que resulte más fácil para los alumnos programar la “inteligencia” de BlueBot. En este trabajo paralelamente se introduce la aplicación SimRo cuyo objetivo es proveer un editor de laberintos y un simulador de estrategias virtual. Esas estrategias una vez testeadas pueden trasladarse al “cerebro” de un robot real como el BlueBot.

### 1.1 Descripción del sistema BlueBot

El sistema está compuesto por los siguientes componentes, que se detallan esquemáticamente en las figuras 1 y 2.

- Robot de desplazamiento autónomo
- Detector de movimiento por imágenes
- Unidad de control
  - Subsistema de telecomando
  - Subsistema HMI.

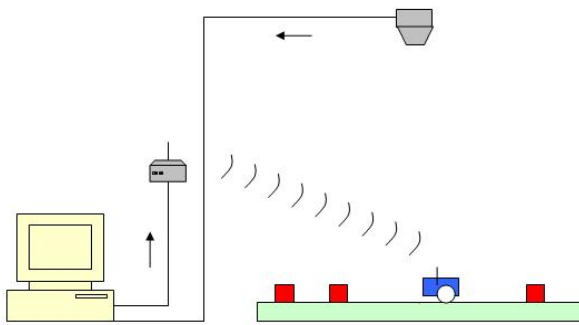


Figura 1: componentes del sistema

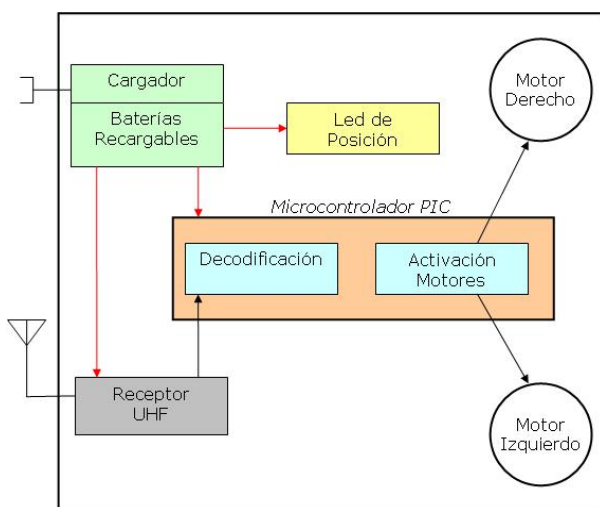


Figura 2: Diagrama de bloques de BlueBot

## 1.2 Estructura del robot

### 1.2.1 Descripción de la estructura

El vehículo telecomandado consta de una plataforma con 2 ruedas de tracción delantera independientes y una rueda trasera de eje giratorio, formando un triciclo. Cada rueda de tracción está fijada directamente al eje de un servomotor (de los utilizados en aeromodelismo modificado).

El control del robot se ha implementado con un microcontrolador PIC que recibe comandos de un módulo receptor de datos en UHF transmitidos desde la estación de control del usuario. El microcontrolador luego los decodifica y genera las señales de comando para los servomotores.

### 1.2.2 Descripción del sistema de visión

Se ha dotado al sistema de una cámara cenital de tipo web-cam estándar que se fija a un brazo cuya altura se puede ajustar entre 1,5m y 2 m. El ángulo de visión de la misma es de  $52^\circ$  lo que determina un área de cobertura de 195cm x 146cm. El robot que es de color azul se mueve sobre un piso verde y se programa para resolver un laberinto que está formado por bloques de color rojo. BlueBot en la parte superior tiene 3 leds blanco dispuestos en forma de V que son usados para determinar su posición y rumbo.

Cada 500 mseg se realiza la captura de una imagen de 640x480 pixels. El primer paso es aplicar un algoritmo de categorización por vecino más cercano para etiquetar cada píxel en uno de los cuatro categorías posibles: piso, laberinto, robot, LED. Este proceso también elimina distorsiones por defectos de iluminación, variaciones de foco, ruido, y otros efectos ópticos producidas en las imágenes tomadas por cámaras de bajo costo.

Con los pixels identificados como LED se realiza un proceso de clustering para determinar el centro de gravedad de cada uno, obteniéndose así un punto en coordenadas del robot de cada foco luminoso. Como la geometría de los estos puntos en el robot es fija, se puede calcular las coordenadas x, y, del centro de gravedad del triángulo y su orientación espacial. De esta manera queda determinada la posición y el rumbo del dispositivo.

### 1.2.3 Descripción de la unidad de control.

Toda la aplicación de visión y control de robot fue escrita en lenguaje Delphi por ser un lenguaje compilado de alto nivel de eficiencia (tanto desde el punto de vista del desarrollo de aplicaciones como desde la velocidad de procesamiento de los ejecutables) y por contar con abundante soft de procesamiento de imágenes. El código generado es compacto, corre aún en máquinas de baja performance,

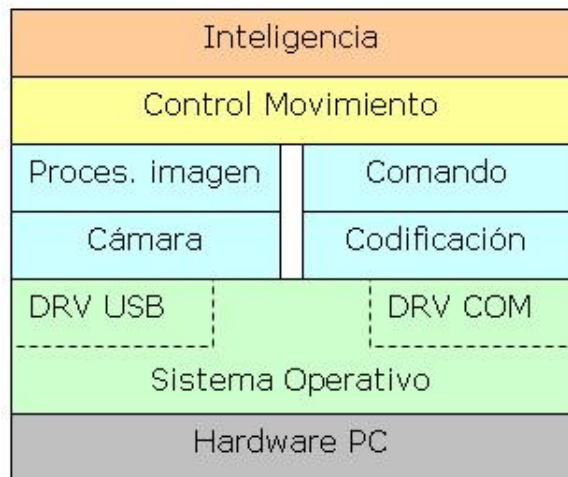


Figura 3: Diagrama de capa de soft. de la PC

el ejecutable es portable sin requerir bibliotecas runtime. En la figura 3 se detallan las capas de la aplicación.

La ventaja de la cámara cenital y el telecontrol desde una PC es que se puede construir un robot de muy bajo costo ya que este no necesita sensores.

### 1.2.4 Interfase hombre-máquina

El software contempla el ajuste de la escala píxel/mm de acuerdo a la altura de la cámara mediante una figura patrón que también es usada para ajuste cromático. Luego se opera directamente usando los elementos provistos por la GUI (ver figura 4).

### 1.3. Programa SimRo: Simulador Robot

Otra área de trabajo, en el marco del mismo proyecto, es el desarrollo de un software llamado SimRo, que edita y simula el recorrido de laberintos aplicando alguna estrategia de recorrido. Es una aplicación escrita en lenguaje Java que tiene las siguientes características:



Figura 4: Captura de pantalla de la aplicación mostrando la GUI

- Editor gráfico
  - Creación formas de laberinto
  - Almacenamiento en disco
  - Recuperación de disco
  -
- Simulador
  - Recorrido paso a paso
  - Recorrido total
  - Gráfico de la traza de recorrido

En la versión actual, el código de estrategia de solución se escribe en una clase Java y luego se debe compilar todo el código de la aplicación para poder ejecutar la simulación de dicha estrategia. Este diseño es adecuado para alumnos que están en el nivel universitario y cursan materias de programación pero es excesivamente complejo para alumnos de nivel medio.

Se está dotando al ambiente de SimRo de una ventana de programación [6] que use un lenguaje sencillo donde se expresen comandos y decisiones que permitan simular la estrategia sin necesidad de conocimientos avanzados de programación.

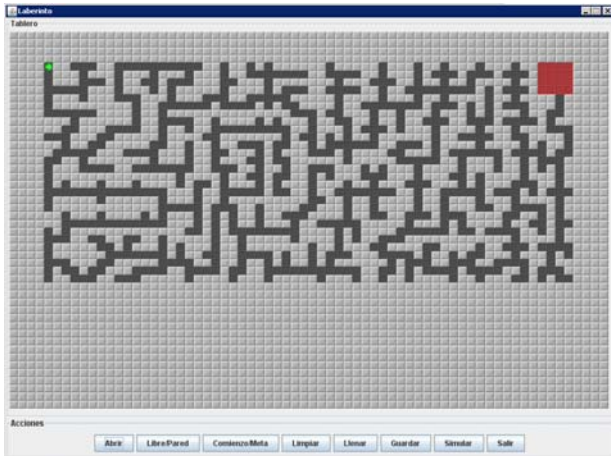


Figura 5: pantalla de SimRo

## 2. LINEAS DE INVESTIGACION Y DESARROLLO

Las áreas de investigación en relación a este proyecto son:

- Detección de movimiento por imágenes
- Reconocimiento de objetos
- Métodos de comunicación Wireless
- Robots telecomandados
- Simulación y algoritmos de inteligencia artificial
- Programación Orientada a Objeto
- Fabricación de Robots para el área educativa

## 3. RESULTADOS OBTENIDOS/ESPERADOS

Hay varias iniciativas a nivel internacional para introducir los conceptos de robótica y resolución de laberintos aplicados a la educación de nivel medio y universitario. [5] [9] Los resultados de este proyecto se están usando en cátedras de programación de la UNPSJB, Sede Puerto Madryn para motivar a los alumnos en los aspectos de programación y robótica.

Se hicieron demostraciones en escuelas secundarias de la ciudad con el fin de divulgación tecnológica. Se estudia la realización de competencias de resolución de laberintos en forma virtual y real incluyendo la construcción/utilización del robot.

## 4. FORMACION DE RECURSOS HUMANOS

Tres integrantes de esta línea de investigación están realizando su Tesis de Licenciatura en Informática en áreas relacionadas con este proyecto y otros tres desarrollarán su tesis de Maestría en la UNLP.

Se está realizando cursos para capacitar a docentes y alumnos de un colegio técnico de nivel medio de la ciudad de Puerto Madryn, en la construcción y aplicación educativa de este tipo de Robot.

## 5. CONCLUSIONES Y TRABAJO FUTURO

El robot puede ser estudiado, analizado y construido en todas sus etapas por alumnos de escuelas técnicas a un costo muy inferior a cualquier kit comercial. En otros establecimientos se puede trabajar en el aula con sólo el software SimRo e introducir los conceptos básicos de robótica y resolución de laberintos. Es posible usar el SimRo para probar estrategias y trasladarlas directamente a un robot autónomo.

Como trabajo a futuro podemos mencionar

- Estudiar estrategias de resolución de laberintos
- Disponer de una biblioteca de estrategias.
- Perfeccionar el manejo de hilos para manejar de manera concurrente la captura de imagen, el control básico del robot y la estrategia de resolución

- Introducir los conceptos de robótica, solución de laberintos y programación en escuelas de nivel medio.

Secondary Education”, LNCS, Springer, 2005.

- [10] H Tamada, A Ogino, H Ueda, “Robot Help Teachers of Education of the C Language Beginners ”, Human Computer Interaction Part II, Springer-Verlag, 2009.

## 6. BIBLIOGRAFIA

- [1] Sing Li, “Rock ‘em, sock ‘em Robocode!, Learning Java programming is more fun than ever with this advanced robot battle simulation engine”, IBM Developer Network, <http://www.ibm.com/developerworks/java/library/j-robocode/>, 2002.
- [2] Sichel Yohanatan, Eran Ziserman, Moshe Sipper, “GP: Robocode, Using Genetic Programming to Evolve Robocode Players, [http://www.cs.bgu.ac.il/~sipper/papabs/eur\\_ogprobo-final.pdf](http://www.cs.bgu.ac.il/~sipper/papabs/eur_ogprobo-final.pdf), 2005
- [3] Robocode Project, <http://robocode.sourceforge.net/>, autor original: Mathew Nelson, Developers: Flemming Larsen, version utilizada: 1.7.1.6 Enero, 2010.
- [4] Jorge Dignani, Renato Mazzatti, Fernando Tidona, Francisco Paez, Rodrigo René Cura, Ricardo Coppo, Claudio Delrieux, “Bluebot: Una Propuesta Accesible de Robótica Móvil para Propósitos Didácticos”. WICC 2008. General Pico. La Pampa. 2008.
- [5] Mohawk College, “A-Maze-Bot Programming Competition”, [http://www2.mohawkcollege.ca/events/am\\_azebot/](http://www2.mohawkcollege.ca/events/am_azebot/), 2010
- [6] Don Libes, "Obfuscated C and Other Mysteries" John Wiley & Sons, 1993,
- [7] Eneko González Benito “Introduction al API reflection de Java” <http://www.javahispano.org/contenidos/archivo/85/reflection.pdf>
- [8] Stéphane Ducasse, “Squeak: Learning Programming with Robots”, Apress, 2005.
- [9] Sartatzemi, M. “Teaching programming with Robots: A case study on Greek