

# Mapeo de Laberintos y Búsqueda de Rutas Cortas Mediante Tres Mini Robots Cooperativos

M. Rodríguez\*; S. Sandobalín\*; D. Pozo\*; L. Morales\*; J. Rosero\*; A. Rosales\*

\*Escuela Politécnica Nacional, Facultad de Ingeniería Eléctrica y Electrónica  
e-mail: [mauro\\_drog@hotmail.com](mailto:mauro_drog@hotmail.com); [sant\\_d387@hotmail.com](mailto:sant_d387@hotmail.com); [david.pozo@epn.edu.ec](mailto:david.pozo@epn.edu.ec);  
[luis.moralesec@epn.edu.ec](mailto:luis.moralesec@epn.edu.ec); [jorge.rosero@epn.edu.ec](mailto:jorge.rosero@epn.edu.ec); [andres.rosales@epn.edu.ec](mailto:andres.rosales@epn.edu.ec)

**Resumen:** En el presente documento se desarrolla el diseño y construcción de tres mini-robots, los mismos que se desplazan a través de un laberinto de superficie regular; evitando colisiones y reconociendo rutas, con el fin de realizar el mapeo del laberinto en menor tiempo comparado con el que tardaría un solo robot. Cada mini-robot recoge datos de la exploración realizada y los envía a un dispositivo coordinador (computador). Los datos son procesados usando MATLAB por medio de una interfaz gráfica donde se visualiza las rutas recorridas y la exploración de su entorno de una forma aproximada. Como resultado más sobresaliente se obtiene la ruta más corta de un laberinto, mediante un proceso de eliminación de aquellas que no son válidas hacia la salida.

**Palabras clave:** mini-robot, explorador, cooperativo, mapeo

**Abstract:** This paper develops the design and construction of three mini-robots, which move through a maze of regular surface; avoiding collisions and recognizing routes, in order to perform the mapping of the maze in less time compared to what would take a single robot. Each mini-robot collects data from scanning and sends a coordinator device (computer). All data were processed using MATLAB via a graphical interface where the routes traveled and exploring their environment in an approximate manner is displayed. The most outstanding result of the shortest path maze is obtained by a process of elimination of those that are not valid for the exit.

**Keywords:** mini-robot, explorer, cooperative, mapping

## 1. INTRODUCCIÓN

En los últimos años la robótica ha evolucionado de manera acelerada, teniendo robots cada vez más sofisticados y con un mayor número de aplicaciones. En la actualidad, el campo de los robots cooperativos está cobrando un gran desarrollo, ya que varias aplicaciones pueden resultar física y computacionalmente complejas para ser resueltas por un solo robot, como mover algún objeto voluminoso a través de un entorno con obstáculos, el manejo de un material peligroso, la exploración y cobertura de un campo, entre otros; por lo tanto, la robótica colectiva busca diseñar sistemas compuestos de varios robots con el fin de resolver un problema de manera conjunta. Los sistemas multi-robot son relativamente sencillos en términos de diseño y control, lo que hace que sean menos costosos que sistemas de un solo robot especializado [1].

En el presente trabajo se muestra el diseño y construcción de tres minirobots que son capaces de mapear el entorno, encontrar la ruta más corta de un laberinto y dibujar una

estimación del mapa del lugar en un computador con la ayuda de MATLAB.

## 2. DISEÑO Y CONSTRUCCIÓN DE LOS MINIROBOTS

### 2.1 Prototipo mini-robot

Para el diseño de la estructura de los mini-robots se toma en consideración varios aspectos tales como: el tipo de tracción, las dimensiones y la ubicación de los sensores ultrasónicos que permiten optar por una configuración de tracción diferencial [2], [3], que facilita su implementación mecánica y su sistema de control. Esta configuración ayuda a controlar el cambio de dirección y velocidad de desplazamiento mediante el accionamiento de dos motores de corriente continua (DC), que se logra mediante señales de modulación de ancho de pulso (PWM).

En el diseño de la estructura de los robots móviles se toma en consideración que las dimensiones deben ser lo suficientemente pequeñas para facilitar el recorrido dentro del laberinto, al mismo tiempo, se analiza la mejor ubicación para los componentes de la estructura con el fin de ocupar el menor espacio posible, como se aprecia en la Figura 1.

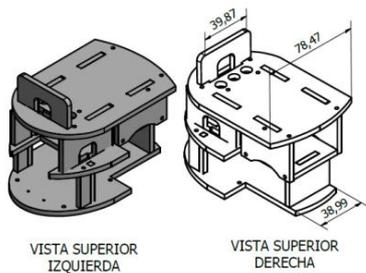


Figura1. Estructura base de los mini-robots

2.2 Ensamblaje

Para el ensamblaje final de los mini-robots se utilizan los siguientes componentes:

- Sensores de ultrasonido HC-SR04 [4]
- Arduino Duemilanove [5]
- Xbee Shield (módulo de comunicación)[6]
- Antenas Xbee S2
- Driver para el manejo de motores L298N
- Kit de encoders POLOLU
- Micromotores DC de relación 150:1 [7]
- Baterías LIPO [8]

El prototipo se diseña a partir de la combinación de componentes electrónicos que forman un sistema microprocesado, el mismo que utiliza software libre para su programación y configuración y que además puede ser adquirido fácil y económicamente en el mercado, dando como resultado los prototipos finales que se muestran en la Figura 2.

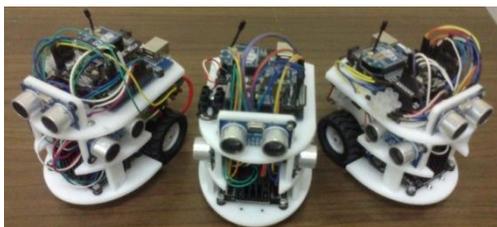


Figura2. Mini-robots

3. DESARROLLO DE LOS ALGORITMOS DE CONTROL

Para desarrollar los algoritmos de control se toma en consideración varios parámetros específicos como son: el desplazamiento de los mini robots, la exploración dentro del laberinto, la obtención de la ruta más corta y el envío de información hacia el computador para realizar su respectivo análisis.

3.1 Desplazamiento de los mini-robots

Dentro del laberinto se tiene que cumplir con movimientos a través de trayectorias rectas y giros en ángulos de 90° en todas las direcciones. Para el control de tracción diferencial se implementa un controlador proporcional – integral (PI) sobre las ruedas de las plataformas robóticas.

$$U(z) = K_p \left[ 1 + \frac{T}{T_i(1 - z^{-1})} \right] E(z) \tag{1}$$

Donde:

- $U(z)$  = señal de control
- $E(z)$  = señal de error
- $K_p$  = constante proporcional
- $T_i$  = constante de tiempo integral
- $T$  = tiempo de muestreo

El algoritmo de control PI [9], [10], se diseña para que el robot avance en línea recta por la parte central de los pasillos del laberinto, siguiendo una trayectoria suave con la menor oscilación posible, reduciendo el error en la adquisición de datos para el mapeo. Para la implementación del controlador PI, se toma como entrada la señal entregada por los sensores ultrasónicos, los cuales dan al robot información acerca de la distancia que existe hasta la pared, dicha información es comparada con una referencia previamente establecida mediante experimentación, es decir, el mini-robot busca compensar el movimiento de las ruedas para permanecer a una distancia constante hacia las paredes del laberinto para evitar colisionar. El error calculado por el controlador se obtiene mediante la diferencia entre el set-point y la medida entregada por el sensor lateral.

Posteriormente la calibración de los parámetros proporcional e integral se realiza aplicando las teorías de sintonización de este tipo de controladores y ajustándolos de manera experimental, dando como resultado que la salida de este controlador varíe el ancho de pulso de la señal PWM aplicada a los motores.

Para el avance de los mini-robots es necesario tomar las señales de los sensores ultrasónicos ubicados al frente y a los costados del mini-robot. En el caso de existir obstáculos al frente, éste se detiene, caso contrario avanza manteniendo su trayectoria mediante el controlador PI.

Por otro lado, los giros que puede realizar el mini-robot están relacionados con la cantidad de pulsos que los encoders, ubicados en las llantas, pueden entregar al microcontrolador, de esta forma se consigue que los giros sean casi exactos.

3.2 Algoritmos de exploración y navegación dentro de un laberinto

Con el fin de realizar la exploración y navegación se implementa el algoritmo denominado de la “mano izquierda” [11], cuya tarea es detectar la pared del costado izquierdo del robot en cada instante de muestreo, permitiendo realizar un seguimiento continuo y efectivo de la trayectoria del laberinto, aún cuando ésta no sea conocida.

Además, para realizar la navegación del mini-robot dentro del laberinto se debe tener en cuenta las particularidades que

pueden existir en el mismo, como son: camino cerrado, giro obligado a la derecha, avanza pasillo, giro obligado izquierda, cruce en "T", intersección a la izquierda, intersección a la derecha y cruce total, como se aprecia en el Figura 3. El programa de control debe ser capaz de reconocer y decidir la acción a tomar cuando se presenten este tipo de casos y avanzar de acuerdo al algoritmo de la mano izquierda.

```

Switch(sensores)
{
  case (frontal, izquierdo y derecho):
    girar 180°; //camino cerrado;
    break;
  case (frontal e izquierdo):
    girar 90° derecha; //giro obligado derecha
    break;
  case (izquierdo y derecho):
    mantiene mismo sentido; //avanza pasillo
    break;
  case (frontal y derecho):
    girar 90° izquierda; //giro obligado izquierda
    break;
  case (frontal):
    girar 90° izquierda; //cruce en "T"
    break;
  case (derecho):
    girar 90° izquierda; //intersección a la izquierda
    break;
  case (izquierdo):
    mantiene mismo sentido; //intersección a la derecha
    break;
  case (no sensa):
    girar 90° izquierda; // cruce total
    break;
}
    
```

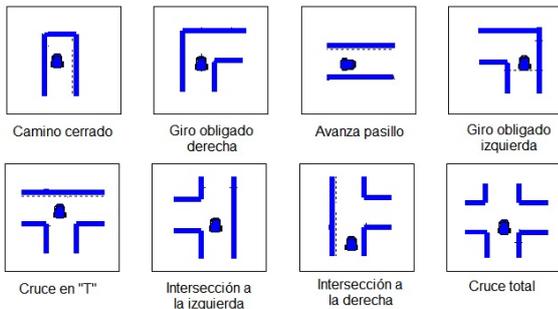


Figura 3. Zonas de operación dentro del laberinto

3.3 Obtención de la trayectoria y ruta corta

La trayectoria del móvil se obtiene por medio del sistema odométrico implementado [12], el mismo que permite estimar la posición y orientación a partir del número de vueltas dadas por sus ruedas mediante el uso de encoders en cuadratura y un sistema microprocesado para el tratamiento de éstos datos. Un robot de tracción diferencial presenta ecuaciones cinemáticas del punto medio del eje, entre las ruedas activas, referidas a un sistema de referencia global, las ecuaciones son:

$$x_k = x_{k-1} + \Delta x_k \tag{2}$$

$$y_k = y_{k-1} + \Delta y_k \tag{3}$$

$$\theta_k = \theta_{k-1} + \Delta \theta_k \tag{4}$$

Donde:

$x_k, y_k, \theta_k =$  posición y orientación actual

$x_{k-1}, y_{k-1}, \theta_{k-1} =$  Posición y orientación anterior

$\Delta x_k, \Delta y_k, \Delta \theta_k =$  desplazamiento lineal y angular actual

Las ecuaciones implementadas en el sistema Arduino son las siguientes:

$$x_k = x_{k-1} + S_{centro} * \cos(\theta_k) \tag{5}$$

$$y_k = y_{k-1} + S_{centro} * \sin(\theta_k) \tag{6}$$

$$\theta_k = \theta_{k-1} + \frac{S_{derecha} - S_{izquierda}}{d_{ruedas}} \tag{7}$$

Donde:

$d_{ruedas} =$  distancia entre las dos ruedas de tracción  $S_{centro}$

$=$  distancia promedio recorrida por las ruedas

Para encontrar la ruta corta es necesario que los giros sean prácticamente exactos y que el recorrido se lo realice por el centro de los caminos dentro del laberinto, esto se logra mediante el uso del algoritmo PI y el algoritmo de la mano izquierda.

Para obtener el camino corto, el mini-robot va almacenando las decisiones que toma durante su recorrido a través del laberinto, luego descarta las rutas que llevan a caminos cerrados, es decir, descarta callejones sin salida.

Este algoritmo comprende dos etapas: la primera, almacena sus decisiones representadas por caracteres que toma durante su desplazamiento, siendo las siguientes: **R** de una decisión de avance recto, **V** de una decisión de camino cerrado e **I** de una decisión de giro a la izquierda; la segunda, asigna la letra **D** como decisión a la derecha como proceso de obtención de ruta corta. Para ello se toman en cuenta las particularidades definidas en el algoritmo de exploración, a las cuales se les asignan los caracteres antes mencionados como se resume en la Tabla 1.

Tabla 1. Particularidades y asignación de caracteres

Particularidad	Asignación	Descripción
Camino cerrado	V	Ruta a descartar
Giro obligado derecha	Sin asignación	Acción necesaria para que el robot continúe su trayecto.
Avanza pasillo	Sin asignación	
Giro obligado izquierda	Sin asignación	Decisión: girar izquierda 90°
Cruce en "T"	I	
Intersección a la izquierda	I	
Cruce total.	I	Decisión: avance recto
Intersección a la derecha	R	

A pesar de la designación de caracteres descrita anteriormente, ésta no posibilita determinar la ruta corta, para ello, el algoritmo además verifica los caracteres almacenados e identifica los caminos que no debió tomar. Para esto, dentro de los datos almacenados, según el criterio descrito en la Tabla 2, se ubica el carácter **V** que indica que el mini-robot

entró en un camino sin salida asignándole a  $V$  un valor de 180, mientras que a los caracteres anterior y posterior un valor de:  $R=0$  e  $I=270$  indistintamente [13].

Luego, con el objetivo de reducir la ruta del mini-robot se reemplazan estos tres caracteres por dos posibilidades: la primera, que es  $D$ , la misma que implica un giro de  $90^\circ$  a la derecha, y la segunda, que es  $R$  que como se mencionó es una decisión de avance recto. Ambas se obtienen del residuo de dividir la suma de los caracteres  $I$ ,  $V$  y  $R$  para 360, cuyas posibilidades se resumen en la Tabla 2.

Tabla 2. Asignación de caracter para ruta corta

Posible combinación	$\sum f(V, R, I)$	Residuo	Caracter
R V R	180	No existe esta combinación	No existe esta combinación
R V I	450	90	D
I V R	450	90	D
I V I	720	0	R

Para ilustrar con mayor claridad el algoritmo implementado en la Figura 4 se muestra la trayectoria que tomaría el mini-robot ante una porción de laberinto.

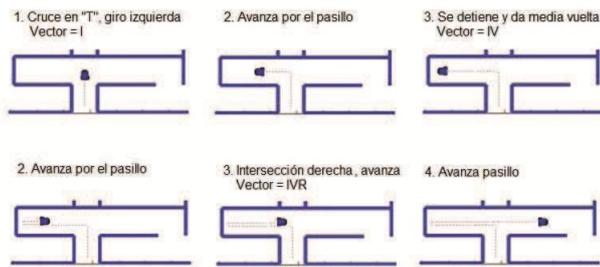


Figura 4. Ejemplo de exploración para obtener la ruta corta

Aplicando el algoritmo de ruta corta se tendría:

Datos almacenados =  $I, V, R$

Tomando en cuenta la secuencia de particularidades y analizando mediante la Tabla 3 se tiene que la ruta corta vendría dada por un giro sencillo de  $90^\circ$  a la derecha simbolizado por  $D$ , dando como resultado la acción presentada en la Figura 5, con lo cual una vez conocido el mapa inicial, es posible conseguir rutas cortas evitando los caminos cerrados.

Ruta corta =  $D$



Figura 5. Ruta simplificada mediante algoritmo de ruta corta.

En la Figuras 6 se muestra la trayectoria del mini-robot sin el algoritmo de ruta corta, mientras que en la Figura 7 se tiene la trayectoria obtenida con la aplicación del algoritmo.

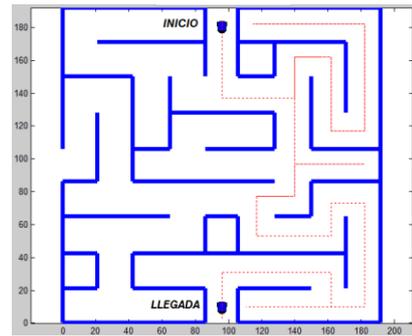


Figura 6. Simulación de la exploración del laberinto sin el algoritmo de ruta corta

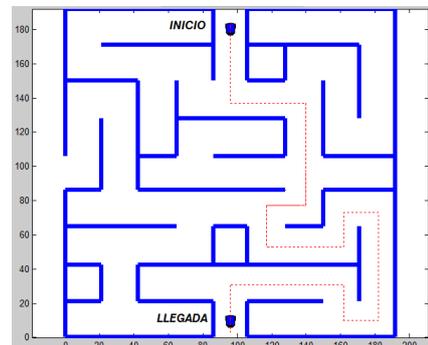


Figura 7. Simulación del algoritmo de ruta corta

### 3.4 Tratamiento de datos

Los datos que envían los mini-robots hacia la PC se transmiten utilizando una red Zigbee formada por un coordinador y tres routers, los cuales son configurados mediante un software especializado de la compañía DIGI, los mini-robots envían los datos adquiridos: posición en X, posición en Y, orientación “ $\emptyset$ ” y las medidas de las distancias de los sensores ultrasónicos del frente y de los lados, izquierdo y derecho. El coordinador recibe los datos enviados por los minirobots y con estos datos se realiza la visualización de la ruta recorrida por las plataformas (mediante odometría), mientras que con las medidas de distancia obtenidas con los sensores se dibuja las paredes detectadas y se realiza una animación con la ayuda de MATLAB.

## 4. PRUEBAS Y RESULTADOS

### 4.1 Calibración del Controlador PI

La calibración de los mini-robots es de gran importancia para la ejecución correcta de sus movimientos ya sea cuando se desplazan en línea recta o cuando giran. A partir de un correcto desplazamiento se obtiene una exploración óptima dando lugar a la adquisición de datos con menor error los cuales permitan obtener una buena resolución del laberinto y el camino más corto.

Un robot móvil para su desplazamiento necesita de un sistema de percepción de entorno mediante sensores; para obtener una trayectoria de avance recta es necesario que los mini-robots se guíen por las paredes que observan a los costados compensando su posición con la medida del sensor de la izquierda o derecha, dependiendo de la zona en la que se encuentre dentro del laberinto.

Realizando pruebas mediante métodos heurísticos se ha determinado los parámetros para la implementación del control PI, siendo estos:

$$\begin{aligned}
 K_p &= 0.5 \\
 T_i &= 0.02 \\
 T_m &= 0.01
 \end{aligned}$$

4.2 Giros de los mini-robots

Los giros a realizarse deben ser de 90° y 180°, para estos se utilizan encoders de cuadratura, los cuales brindan una cantidad específica de pulsos por cada giro completo de cada una de las ruedas, con este principio se las condiciona a girar un número de pulsos determinado antes de detenerse con la finalidad de llegar a la posición deseada (Figura 8). La Figura 9 muestra el giro incorrecto de la plataforma robótica, resultado de una mala medición de los pulsos obtenidos a través del uso de los encoders.

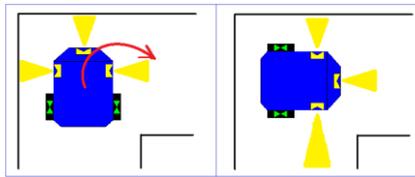


Figura 8. Giro correcto a la derecha, lectura de sensores adecuado

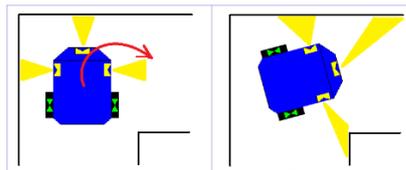


Figura 9. Giro incorrecto a la derecha, medida de sensores errónea

4.3 Pruebas de exploración y mapeo

Para las pruebas de exploración y mapeo se trabajó con los mini-robots a distintas velocidades, estableciéndose una velocidad máxima de 7.5 cm/s en rectas. Con esta velocidad se garantiza evitar colisiones con las paredes del laberinto y entre los mini-robots, así como también una correcta identificación de las diferentes particularidades que presenta el laberinto, mediante la ejecución del algoritmo de la mano izquierda.

Los resultados de este procedimiento se muestran en la Figura 10, 11, 12.

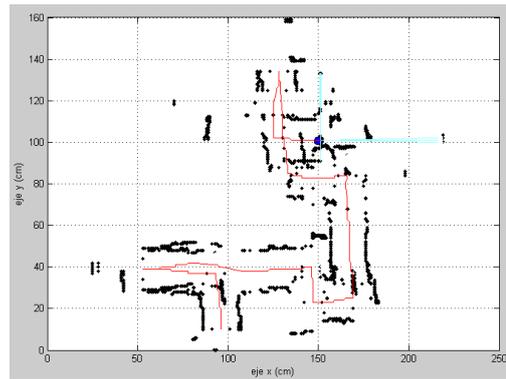
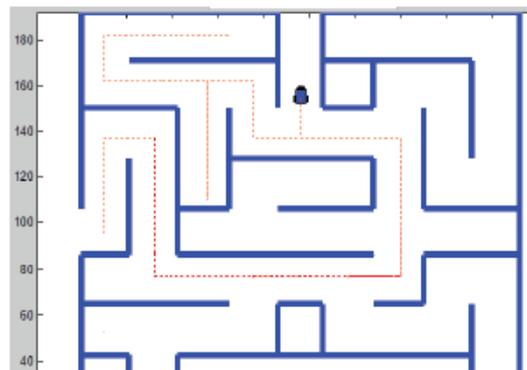
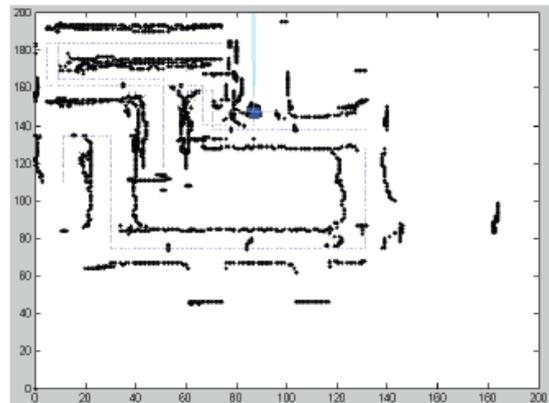


Figura 10. Exploración y visualización de la trayectoria y del entorno con un minirobot

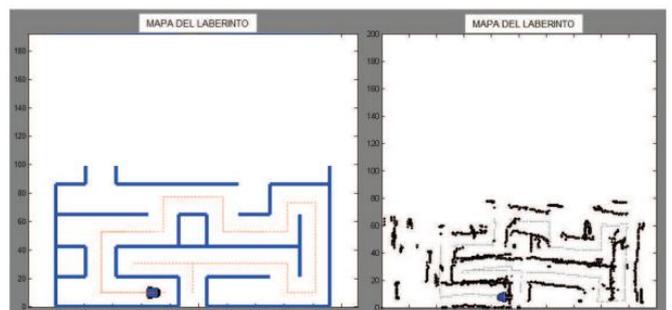


(a)



(b)

Figura 11. Mapeo del laberinto. (a) Entorno real. (b) Entorno experimental



**Figura 12.** Comparación entre el entorno real (izquierda) y el obtenido por el sistema de mini-robots (derecha)

## 5. CONCLUSIONES

La exploración que realiza un solomini-robot es aceptable, por su tamaño y velocidad, pero no es eficiente, debido al tiempo que emplea, por esta razón se ha dotado de cooperativismo al sistema, logrando disminuir los tiempos de exploración de un entorno mediante el funcionamiento de las tres plataformas robótica simultáneamente.

La trayectoria recorrida por los mini-robots se obtiene a partir de la odometría, la cual posee un error inherente en la medición, debido a esto, es necesario realizar ajustes en los mini-robots y también correcciones por software, ya que la acumulación de errores en la orientación causa grandes desviaciones en la posición, las cuales se incrementan exponencialmente con la distancia recorrida del mini-robot.

Se ha realizado la exploración y obtención de la ruta corta de un laberinto, resolviendo el problema de hallar el camino menos costoso (en tiempo) hacia la salida.

Se ha implementado la comunicación punto – multipunto para realizar el cooperativismo, de manera que se configuró una red zigbee que proporciona una comunicación entre un único coordinador y sus routers (mini-robots), los datos son enviados hacia el coordinador para que a través del mismo se comuniquen hacia la PC o se de una orden a cualquiera de los mini-robots.

Se puede incorporar a los mini-robots sensores de tipo laser, los cuales permitan detectar con menor rango de error las particularidades del laberinto, logrando obtener mayor detalle en la reconstrucción del mismo, trabajando a la par con los sensores ultrasónicos aplicando lo que se conoce como “*fusión sensorial*”.

Para obtener una posición más exacta del mini-robot dentro del laberinto se puede utilizar otro método para estimar la posición del minirobot además del sistema odométrico utilizado. Puede ser una estimación de la posición mediante balizas, que permiten determinar la posición de un robot móvil en un entorno delimitado. La posición de los mini-robots se determinaría en base al principio de triangulación, a partir de medidas de distancias, ángulos o de la combinación de los dos. La precisión y fiabilidad de este tipo de estimación dependerá del tipo de señal utilizada; infrarrojos, laser, radio, ultrasónicos, de las características del sensor y del número de balizas utilizadas.

## REFERENCIAS

- [1] L. d. S. Rodrigo, Seminario de Diseño y Construcción de Microrobots.
- [2] A. Ollero Baturone, Robótica: manipuladores y robots móviles, Barcelona: MARCOMBO, 2001.
- [3] N. Cardenas y G. Simbana, DISEÑO Y CONSTRUCCIÓN DE UN ROBOT MOVIL DIDACTICO PARA ESTUDIOS EXPERIMENTALES DE ROBOTICA COOPERATIVA, Cuenca: Universidad Politecnica Salesiana, 2007.
- [4] Cytron Technologies, «HC - SR04 User's Manual,» Noviembre 2012. [En línea]. Available: [https://docs.google.com/document/d/1Y-yZnNhMYy7rwhAgyL\\_pfa39RsB-x2qR4vP8saG73rE/edit?pli=1](https://docs.google.com/document/d/1Y-yZnNhMYy7rwhAgyL_pfa39RsB-x2qR4vP8saG73rE/edit?pli=1).
- [5] Arduino, «Arduino Duemilanove,» 2010. [En línea]. Available: <http://arduino.cc/es/Main/ArduinoBoardDuemilanove>. [Último acceso: Marzo 2012].
- [6] Gravitech, «Electronic Experimental Solution,» 2006. [En línea]. Available: <http://www.gravitech.us/arxbsh.html>. [Último acceso: Noviembre 2012].
- [7] Pololu Corporation, «Pololu Robotics & Electronics,» [En línea]. Available: <http://www.pololu.com/catalog/product/2368/specs>. [Último acceso: 10 Febrero 2013].
- [8] HobbyKing, «HobbyKing,» 2012. [En línea]. Available: [http://www.hobbyking.com/hobbyking/store/\\_9494\\_\\_Turnigy\\_1300mAh\\_4S\\_20C\\_Lipo\\_Pack.html](http://www.hobbyking.com/hobbyking/store/_9494__Turnigy_1300mAh_4S_20C_Lipo_Pack.html). [Último acceso: Enero 2013].
- [9] Grupo de Investigación en Tecnologías de la Información y las Comunicaciones, «MÉTODO BÁSICO PARA IMPLEMENTAR UN CONTROLADOR DIGITAL PID EN UN MICROCONTROLADOR PIC PARA DESARROLLO DE APLICACIONES A BAJO COSTO,» [En línea]. Disponible: [http://www.edutecne.utn.edu.ar/microcontrol\\_congr/industria/mtodob~1.pdf](http://www.edutecne.utn.edu.ar/microcontrol_congr/industria/mtodob~1.pdf)
- [10] P. H. Analia, «Universidad Nacional de San Juan,» [En línea]. Disponible: <http://dea.unsj.edu.ar/control1/apuntes/resumen%20controladores.pdf>. [Último acceso: Octubre 2012].
- [11] P. E. Acuña Carlos, «DISEÑO Y CONSTRUCCIÓN DE UN ROBOT MOVIL QUE PUEDA DESPLAZARSE DENTRO DE UN LABERINTO,» Escuela Politecnica Nacional, Quito, Agosto 2006.
- [12] Cuentos Cuánticos, «Robótica: Estimación de posición por odometría,» 15 Diciembre 2011. [En línea]. Available: <http://cuentos-cuanticos.com/2011/12/15/robotica-estimacion-de-posicion-por-odometria/>. [Último acceso: 12 Diciembre 2012].
- [13] Pololu Corporation, «Robot Pololu 3pi Guía de usuario,» [En línea]. Available: <http://www.pololu.com/file/0J137/Pololu3piRobotGuiaUsuario.pdf>. [Último acceso: 10 Febrero 2013].