

# USO DE MATLAB EN APLICACIONES DE ROBOTICA MOVIL

WILMER ALEXIS SANDOVAL CACERES

Egresado de la Universidad Pontificia Bolivariana  
Fryethman@yahoo.es

**Resumen:** **Matlab** es un software matemático muy poderoso, comercialmente disponible, desarrollado y distribuido por Mathworks, Inc. Es utilizado ampliamente en la academia y la industria debido a sus capacidades avanzadas, además posee una serie de herramientas que contienen funciones comúnmente usadas en ingeniería. Todas herramientas facilitan el desarrollo de aplicaciones complejas de una forma más versátil.

En este artículo se hará una pequeña descripción del manejo y comportamiento del **puerto serial** del **PC** y el manejo de interfaces graficas, mediante el uso de las herramientas proporcionadas por el software **Matlab**, para desarrollar aplicaciones de robótica móvil.

**Palabras claves:** *Datos, CRC (código de redundancia cíclica, Matlab, microbot, PC (computador personal), puerto serial, radio frecuencia (RF), sensores, toolbox ultrasonido.*

**Abstract:** **Matlab** is a commercially available mathematical software package developed and distributed by The Mathworks, Inc. It is widely used in academia and industry because of its advanced capabilities, it's also has built-in toolboxes that contain commonly used engineering.

In this article, I'm going to do a small description of the handling and behaviour of the computer's serial port and graphic interface handling, by means of the use of the tools provided by the **Matlab** software, to develop Mobile Robot's applications based on this software.

**Key words:** *Data, CRC (Cyclic Redundancy Check Code), microbot, Matlab PC (Personal Computer), radio frequency (RF), serial port, sensor, toolbox, ultrasound.*

## 1. INTRODUCCIÓN

A través de los tiempos, el hombre en su afán por simplificar sus tareas cotidianas, ha desarrollado métodos y herramientas que le han permitido, mejorar su calidad de vida. Como resultado de esa búsqueda constante e incansable, se ha realizado grandes avances científicos y tecnológicos en diversos campos del saber, dentro de los cuales podemos destacar la robótica.

Inicialmente, los robots fueron diseñados para operar en entornos industriales o como aplicaciones de carácter científico, pero con el avance vertiginoso de la tecnología, los requerimientos de las tareas a realizar, se hicieron cada vez, más complejas y exigentes.

La robótica se ha dividido en múltiples ramas, según su campo de acción. Uno de estos campos es la microbotica móvil, donde se construyen robots miniatura para efectuar tareas específicas. Existen una gran diversidad de **microbots**, entre los cuales podemos destacar: bípedos, cuadrúpedos, hexápodos y rastreadores, entre otros.

Toda la información referente al **microbot** ó a su entorno puede ser adquirida, procesada o enviada mediante el uso de los puertos del computador. Existen muchos softwares que permiten manipular los puertos del **PC**, **Matlab** no es la excepción, además de lo anteriormente mencionado, esta dotado con una **toolbox** especializada en robótica.

Quizás una de las aplicaciones de la microbótica móvil mas conocida en el campo de la investigación es el **microbot** SOJOURNER, el cual fue enviado por la NASA en 1997, para la exploración del planeta Marte.

Para comprender de una forma más rápida y concisa, las aplicaciones de robótica móvil, es necesario contar con algunos conceptos básicos de comunicación de **datos**, electrónica, robótica, **sensores** y software de desarrollo. Sin

previo conocimiento de estos temas, se dificultaría un tanto, el entendimiento de lo que se desea plasmar en el presente artículo.

## 2. FUNDAMENTOS TEORICOS.

La robótica es una disciplina, que tiene múltiples campos de acción. Inicialmente se diseñaron robots para uso industrial, pero con el paso del tiempo, se han ampliado sus campos de acción. Hoy en día, es común observar aplicaciones de robots en áreas como: la industrial, la investigación, la salud y el entretenimiento, entre muchas otras.



Tomado de [1].

**Fig. 1. Fotografía Del Robot Sojourner En Su Explotación A Marte.**

Dentro de la robótica móvil podemos destacar los prototipos desarrollados por la NASA, para sus experimentos en el espacio exterior, los cuales efectúan tareas en entornos hostiles para el ser humano, el SOJOURNER es un ejemplo de este tipo de aplicaciones, el cual hizo parte de de la misión espacial Pathfinder en 1997, el cual puede ser observado en la figura 1.

En el trigésimo aniversario de la exploración robótica de Marte, la NASA selecciono al conocido "SOJOURNER", para ser el primer robot errante en explorar el planeta rojo [1].

Posteriormente, la NASA ha desarrollado otros robots móviles para la exploración espacial como OPPORTUNITY y SPIRIT.

### 2.1 La Microbótica Y Los Microbots:

La microbótica es una tecnología derivada de la robótica industrial que se esta imponiendo en multitud de aplicaciones y que plorifera masivamente en la primera década del siglo XXI. El **microbot** es un dispositivo móvil que reacciona ante el entorno de acuerdo con un plan de acciones programado por el usuario. Para moverse utiliza motores y para reconocer el entorno emplea **sensores** de

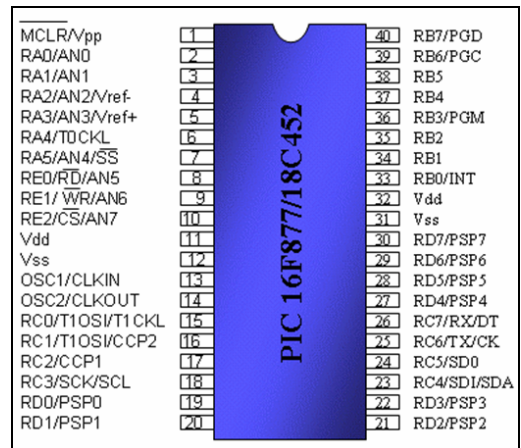
todo tipo. Finalmente para controlar motores, **sensores** y actuadores dispone de un computador en miniatura programado en función de la finalidad a la que se destina [2].

La finalidad de estos **microbots** es realizar tareas, pequeñas con un alto grado de precisión y rapidez. También son utilizados en entornos, los cuales son peligrosos para el ser humano.

### 2.2 Microcontroladores:

Un microcontrolador es un circuito programable que contiene todos los componentes de un computador. Se emplea para controlar el funcionamiento de una tarea determinada, debido a reducido tamaño, suele ir incorporado en el propio dispositivo que gobierna [2].

En nuestro medio, los microcontroladores mas utilizados son los fabricados por Motorola y Microchip, aunque existen otras casas fabricantes como Atmel, Intel, Parallax, Philips y Texas instruments, entre muchas otras. En la figura 2 podemos observar la disposición de terminales de un microcontrolador de microchip.

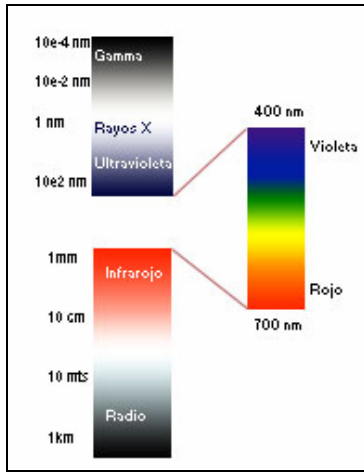


**Fig. 2. Generalidades De Los Microcontroladores de Microchip PIC 16X877/18X452.**

### 2.3 Sensores:

Los **sensores** son elementos que nos permiten establecer un puente entre el mundo físico y el mundo electrónico. También son llamados transductores y tarea es convertir una medida física en una medida eléctrica. A continuación se hará una breve descripción de los **sensores** mas utilizados en el desarrollo de **microbots**.

**2.3.1 Sensores Infrarrojos (IR):** Los **sensores** infrarrojos son un tipo de sensor de luz que utiliza una parte del espectro denominada infrarrojo (como se observa en la Figura 3), cuyo rango fluctúa entre  $3 \times 10^{11}$ Hz hasta aproximadamente los  $4 \times 10^{14}$ Hz.



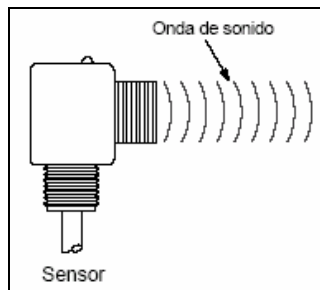
Tomado de [3]

Fig. 3. Espectro Electromagnético.

**2.3.2 Sensores Ultrasónicos:** Las ondas ultrasónicas, son ondas mecánicas vibratorias, es decir, para que se propague el **ultrasonido** se requiere que las partículas del medio, ya sea líquido, aire o sólido oscilen alrededor de su punto de equilibrio [4].

Los **sensores** ultrasónicos son detectores que están basados en la emisión-recepción de ondas ultrasónicas. Cuando un objeto interrumpe el haz, el nivel de recepción varía y el receptor lo detecta.

Los detectores ultrasónicos pueden detectar con facilidad objetos transparentes, como cristal y plásticos, materiales que ofrecen dificultades para la detección óptica. Sin embargo, y dado que estos detectores utilizan ondas ultrasónicas que se mueven por el aire, no podrán ser utilizados en lugares donde éste circule con violencia (bocas de aire acondicionado, cercanías de puertas, etc.), o en medios de elevada contaminación acústica (prensas, choques entre metales, etc.).



Tomado de [5]

Fig. 4. Sensor Ultrasónico.

## 2.4 Encoder:

Los encoders son dispositivos formados por un rotor con uno o varios grupos de bandas opacas y traslúcidas

alternadas y por una serie de captadores ópticos alojados en el estator, que detectan la presencia o no de banda opaca frente a ellos. Existen dos tipos de encoders: los incrementales y los absolutos.

## 2.5 Radiofrecuencia:

Las ondas de radio y de luz tienen el mismo origen, electromagnético, por lo que también tienen propiedades físicas comunes, por ejemplo: la velocidad de propagación en el vacío es de  $3 \times 10^8$  m/s y la capacidad de ser reflejadas o refractadas. Las ondas de **radio frecuencia (RF)** van de 0 a  $10^9$  Hz (como se observa en la figura 3), se usan en los sistemas de radio y televisión y se generan mediante circuitos oscilantes.

Las ondas de RF y las microondas son especialmente útiles porque en esta pequeña región del espectro, las señales producidas pueden penetrar las nubes, la niebla y las paredes. Estas son las frecuencias que se usan para las comunicaciones vía satélite y entre teléfonos móviles.

## 2.6 Comunicación De Datos:

En los casos que se debe establecer una comunicación de **datos**, entre dos o más dispositivos, es necesario implementar un sistema que permita detectar errores en la comunicación. Estos errores son introducidos en la información, debido a perturbaciones en el medio de transmisión.

En aquellas situaciones que involucran comunicaciones entre computadoras, y aun dentro de un sistema de computación, existe la posibilidad de que se reciba información con errores, debido a ruidos en el canal de comunicaciones. En realidad, los símbolos binarios adoptan formas físicas, como tensiones y corrientes eléctricas. La forma física está sujeta al ruido que se introduce desde el ambiente, por ejemplo los fenómenos atmosféricos, rayos gamma y fluctuaciones de la alimentación, para nombrar solo algunas de las posibles causas. El ruido puede ocasionar errores, también conocidos como fallas, en las que un cero se convierte en un uno, o un uno se convierte en un cero.

Es posible hacer que el transmisor envíe bits adicionales de verificación junto con los bits de información. El receptor puede examinar estos bits de verificación y, ante ciertas condiciones, no solo puede detectar errores, sino también corregirlos.

## 2.7 MATLAB:

**Matlab** es el nombre abreviado de “MATrix LABORatory”, es un programa especializado para realizar cálculos numéricos [6]. Una de las capacidades más

atractivas es la de poseer un lenguaje de programación propio.

Hoy en día, **Matlab** es usado en una variedad de áreas de aplicación incluyendo procesamiento de señales e imágenes, diseño de sistemas de control, ingeniería financiera e investigación médica.

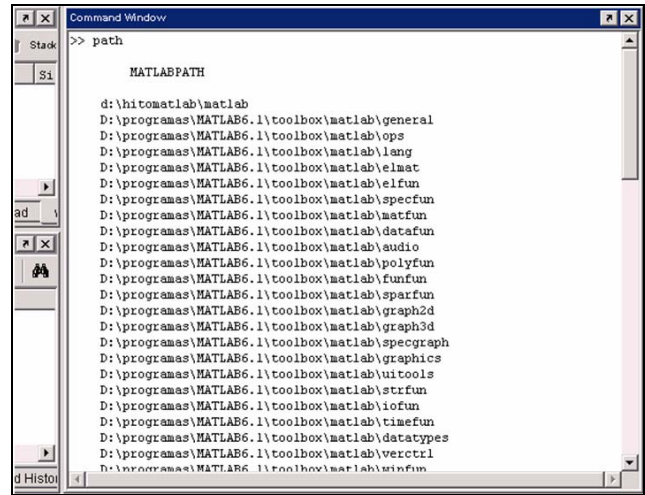
En general, matrices y vectores son el corazón de **Matlab**, todos los **datos** son almacenados como vectores [7]. Además de esto cuenta con herramientas de desarrollo como la GUI (Graphic User Interface), la cual permite generar entornos gráficos para las diferentes aplicaciones.

**2.7.1 Características Del Entorno:** **Matlab** provee acceso inmediato a las características gráficas especializadas requeridas en ingeniería y ciencias. Posee una serie de herramientas que facilitan el procesamiento de información, entre los cuales se destacan:

- Representaciones 2-D y 3-D, incluyendo **datos** triangulados y reticulados.
- Representaciones 3-D quiver, ribbon, y stem
- Control de fuentes, letras Griegas, símbolos, subíndices y superíndices.
- Selección expandida de símbolos marcadores de curvas.
- Gráficos de torta, de barras 3-D y gráficos de barras horizontales.
- Gráficos 3-D y sólido modelado.
- Representación de imágenes y archivos I/O.
- Gráficos comentados.
- Leer/Escribir archivos de **datos** Hierarchical Data Format (HDF).
- Presentación de OpenGL software y hardware.
- Animación.
- Soporte de colores verdaderos (24-bit RGB).
- Fuentes múltiples de luz para superficies coloreadas.
- Vista basada en cámara y control de perspectiva.
- Iluminación Plana, Gouraud y Pong.
- Soporte eficiente de imagen de **datos** de 8-bit.
- Control de eje y cámara.
- Propiedades de superficie y match.
- Modelos de iluminación.
- Control gráfico de objetos.
- Impresión y representación de copias.
- Formatos gráficos exportables.

Lo anteriormente descrito se extrapola de [7].

**2.7.2 PATH (DIRECTORIO RAÍZ):** La lista de directorios donde **Matlab** busca sus ficheros se llama MATLABPATH o PATH, y solo los programas que se encuentren allí pueden ser ejecutados correctamente (ver figura 5).



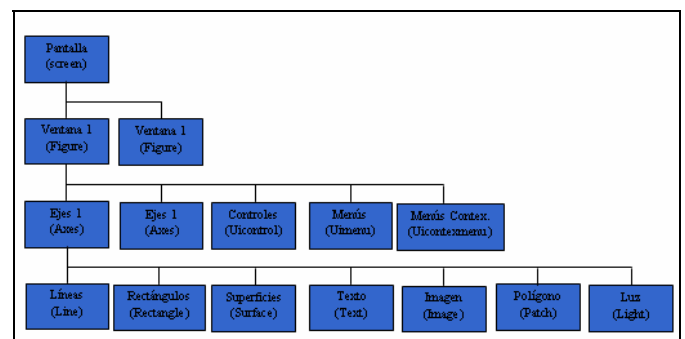
**Fig. 5.** Lista de directorios de MATLAB (Path) desde la ventana de comandos.

**2.7.3 Jerarquización De Objetos gráficos:** El objeto más general es la pantalla (screen). Este objeto es la raíz de todos los demás y sólo puede haber un objeto pantalla.

Una pantalla puede contener una o más *ventanas* (*figures*). A su vez cada una de las ventanas puede tener uno o más *ejes* de coordenadas (*axes*) en los que representar otros objetos de más bajo nivel. Una ventana puede tener también *controles* (*uicontrol*) (tales como botones, barras de desplazamiento, botones de selección o de opción, etc.) y menús (*uimenu*).

Finalmente, los ejes pueden contener los seis tipos de elementos gráficos que permite **Matlab**: líneas (*line*), rectángulo (*rectangle*), polígonos (*patches*), superficies (*surface*), imágenes bitmap (*image*) y texto (*text*).

En **Matlab** hay objetos padres e hijos, como se observa en la figura 6.



**Fig. 6** Jerarquía de los objetos gráficos en MATLAB.

Por ejemplo, todos los objetos ventana son hijos de pantalla, y cada ventana es padre de los objetos ejes, controles o menús que están por debajo. A su vez los elementos gráficos (líneas, polígonos, etc.) son hijos de un objeto ejes, y no tienen otros objetos que sean sus hijos.

Cuando se borra un objeto de **Matlab** automáticamente se borran todos los objetos que son sus descendientes.

Por ejemplo, al borrar unos ejes, se borran todas las líneas y polígonos que son hijos suyos.

**2.7.4 Creación De Un Objeto Del Puerto Serial: Matlab** ofrece la posibilidad de crear objetos de tipo serial para ello utiliza la función llamada Serial.

La sintaxis para esta función está asociada con el nombre del puerto donde se conecta el dispositivo que se va a controlar.

$$\text{Nombre\_objeto}=\text{serial}(\text{"COM1"}) \quad (\text{Eq. 1})$$

Con esta función podemos crear objetos seriales, definiendo algunos parámetros como son: la velocidad de comunicación, el número de bits de los **datos**, los bits de parada y el de paridad, este último opcional. La forma de manipular el puerto y los comandos utilizados, se explican de forma detallada en un artículo que he escrito y que próximamente publicare, con el nombre: Manejo Del **Puerto Serial**, Mediante **Matlab**.

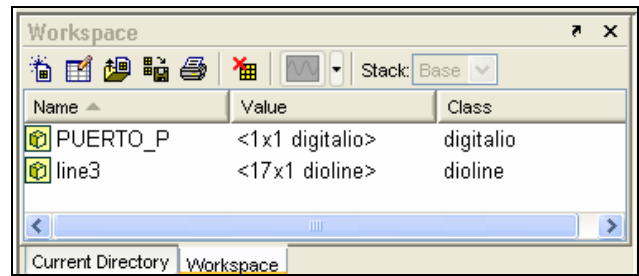
### 3. PRUEBAS REALIZADAS.

Inicialmente se familiarizo con los comandos que permiten manipular los puertos del **PC**, y se construyeron los elementos necesarios para establecer una comunicación bidireccional entre el **PC** y un periférico externo. Para lo anteriormente mencionado se implementaron tanto los cables de comunicación para los puertos, como los circuitos de protección correspondientes para cada caso.

#### 3.1 Puerto Paralelo:

Para observar el comportamiento del puerto paralelo, se hicieron pruebas sencillas de transmisión y recepción de **datos** con el fin de verificar la fiabilidad de la comunicación implementada con el puerto paralelo.

Para este caso específico, se implemento un circuito con LEDs e interruptores miniaturas, de tal manera que el circuito implementado permitiese para manipular los estados de las entradas del puerto y observar las salidas del mismo. En la figura 7, se observa un ejemplo de la creación de un objeto digital tipo paralelo.

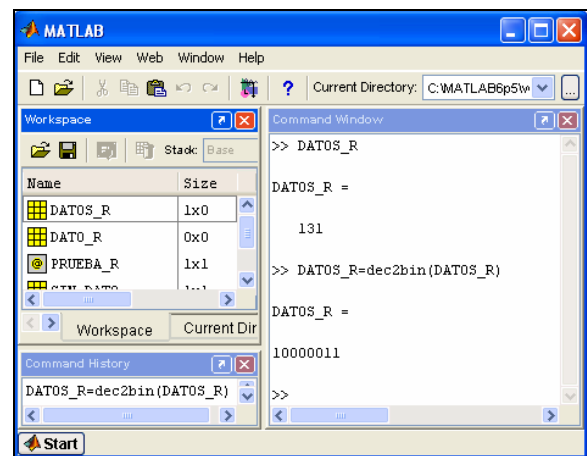


**Fig. 7 Creación del objeto digital de entrada/salida tipo paralelo.**

#### 3.2 Puerto Serial:

Para observar la fiabilidad del **puerto serial**, se realizaron el mismo tipo de pruebas que se describieron en la sección del puerto paralelo. La diferencia fundamental entre estos dos tipos de comunicación radica, en que para la comunicación serial, se debe tener un dispositivo especializado, programado de tal manera que podamos generar o visualizar los **datos** deseados. Para tal fin se es necesario utilizar un microcontrolador con los mismos elementos de señalización y de control que se utilizaron en el circuito del puerto paralelo, es decir, interruptores miniaturas y LEDs.

Para este caso específico se realizaron diferentes pruebas, con el fin de verificar las velocidades de comunicación de **Matlab**, para posteriormente, escoger la rata de baudios, más conveniente según los requerimientos de la aplicación a desarrollar. En la figura 8, se observa un ejemplo de recepción de **datos** mediante el **puerto serial**.



**Fig. 8. Prueba De Recepción De Datos Mediante El Puerto Serial, Realizada Con El MATLAB.**

#### 3.3 Microbot:

Paralelamente al desarrollo del software y a las pruebas de comunicación entre el **PC** y un dispositivo externo, se realizo la construcción y programación de la

unidad móvil, dotándola con **sensores** especializados para realizar el estudio de su entorno. Como cerebro del **microbot**, se utilizó un microcontrolador de microchip, donde se implementó un programa capaz de adquirir y almacenar la información, obtenida por los diferentes **sensores**, basados en dicha información se manipularon los motores y servomotores de la unidad móvil, para controlar su desplazamiento.

Como resultado final de las pruebas realizadas con la unidad móvil, se optó por utilizar una base circular con dos llantas y una rueda loca como se observa en la figura 9, debido que con esta forma geométrica, el **microbot** utilizaba un área menor para realizar los giros sobre su eje central.

Sobre dicha base, se ubicaron estratégicamente los motores que proporcionan la movilidad al motor, a su vez, se ubicaron **sensores** y sistemas acondicionadores de señales de tal manera que las mediciones de las señales analógicas realizadas, se obtuvieran con la mayor fiabilidad posible.

Una vez construido la unidad móvil, se procedió a realizar pruebas de programación del **microbot** para verificar su funcionamiento básico. Posteriormente se implementó el código necesario para que el **microbot** adquiriera información de su entorno, y basado en dicha información, ejecutara una serie de rutinas previamente programadas.

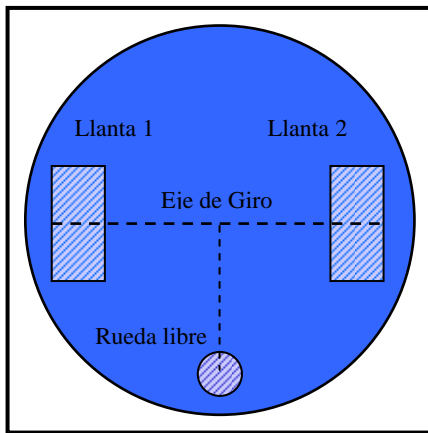


Fig. 9 Base de la Unidad Móvil con dos ruedas libres

### 3.4 Interfaz grafica:

Se diseñó una interfaz grafica híbrida, como resultado de las pruebas realizadas al generar interfaces graficas mediante código y mediante la GUI de **Matlab**.

Como resultado final se obtuvo una interfaz grafica capaz de generar y obtener información del **puerto serial**, con la opción de decodificar la información adquirida, la cual se almacena en un sistema matricial, para su posterior manipulación y estudio.

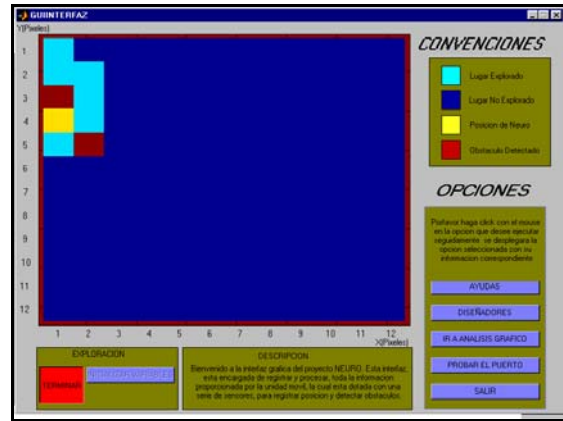


Fig. 10. Interfaz grafica desarrollada.

La interfaz grafica fue diseñada, con la finalidad de ser manipulada fácilmente por un usuario inexperto en la materia, para lo cual fue dotada con una serie de ayudas que guían paso a paso al usuario a largo de toda la aplicación. En la figura 10, se observa el aspecto de la interfaz desarrollada.

### 3.5 Sistema de comunicación inalámbrico:

Se implementó un sistema inalámbrico de comunicación tanto en la unidad móvil como en el periférico externo al **PC**, con la finalidad de enlazar vía radiofrecuencia (RF), al **microbot** con el **PC**. Este tipo de comunicación se implementó con módulos especializados, los cuales están diseñados para codificar **datos** y transmitirlos ó recibirlos y decodificarlos, según sea el caso.

En la figura 11, podemos observar una representación del sistema de comunicación inalámbrico utilizado.

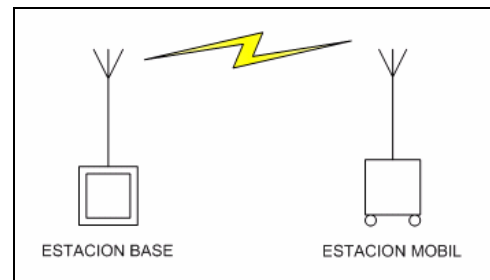


Fig. 11. Sistema inalámbrico implementado.

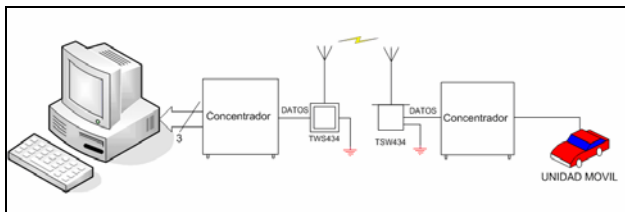
En las comunicaciones inalámbricas es necesario tener en cuenta que el medio introduce errores debido a perturbaciones en él, por lo cual se hace necesario emplear un sistema detector de errores. En nuestro caso específico, se implementó el **código de redundancia cíclica**, para

evaluar los **datos** recibidos que posteriormente se almacenarían en el **PC**.

Para implementar un sistema de comunicación inalámbrico fiable, se hicieron múltiples pruebas de alcance, transmisión y recepción.

### 3.6 Funcionamiento conjunto de la aplicación:

Para verificar el funcionamiento de las partes anteriores descritas, se procedió a acoplar cuidadosamente todas las secciones desarrolladas de forma individual, para conformar una sola aplicación. Con las pruebas realizadas, se detectaron algunas falencias que fueron enmendadas para optimizar el funcionamiento de la aplicación implementada. En la figura 12, podemos observar una representación de toda la aplicación en conjunto.



**Fig. 12. Representación en bloques de la aplicación desarrollada.**

## CONCLUSIONES Y RECOMENDACIONES

Para implementar una aplicación de de robótica móvil, el primer paso es construir el **microbot** teniendo en cuenta aspectos importantes como: seleccionar los **sensores** adecuados, basados en los requerimientos de la aplicación, escoger los sistemas de desplazamiento más adecuados (Motores o servomotores), que no consuman corriente en exceso. Ubicar todos los elementos tanto estratégica como cuidadosamente dentro de la base del móvil. Esta última parte, juega un papel trascendental en el buen desempeño del **microbot** en sus tareas. A manera de experiencia personal, los armazones realizados con poca precaución para robots móviles (que se desplazan sobre ruedas), por lo general presentan un problema muy serio como es la desviación de trayectoria del **microbot** por mala ubicación de sus motores en la armazón. Este problema puede ser corregido en gran medida con una cuidadosa ubicación de los motores, teniendo en cuenta que ambos motores deben estar ubicados sobre un mismo eje central. Otro aspecto muy importante en la construcción de la unidad móvil, es seleccionar un paquete de baterías recargable que suministren buena corriente a toda la unidad móvil. Como recomendación personal es mejor implementar circuitos de alimentación separados tanto para la parte de potencia como para la parte de control.

Otro aspecto importante que hay que tener en cuenta es el sistema de comunicación, el cual debe estar implementado con un sistema detección de errores, para eliminar la posibilidad de adquirir **datos** errados.

**Matlab**, además de permitir la manipulación del **puerto serial**, ofrece la posibilidad de almacenar y efectuar todo el tratamiento matemático de **datos** que el posee, como son: operaciones con vectores, matrices, graficación en dos y tres dimensiones, entre otros. Además de todo esto, **Matlab** ofrece la posibilidad de implementar interfaces graficas de usuario con una herramienta especializada, llamada GUI, cuya función es permitir el diseño de una interfaz grafica de una forma sencilla, rápida, estética y funcional.

El fuerte manejo matemático de **datos**, es una de las características primordiales de este software, acompañado de herramientas como la GUI y las **toolbox** especializadas, hacen que **Matlab** sea una herramienta de desarrollo muy poderosa para implementar aplicaciones de robótica móvil.

## 4. AGRADECIMIENTOS

Agradezco de manera muy cordial al ingeniero Jesús Antonio Vega, por su gran colaboración y amistad, a los ingenieros: Cristian Jhon Ruiz, Carlos Gerardo Hernández, Juan Carlos Villamizar, Claudia Leonor Rueda por sus comentarios y apoyo. A mi directora de tesis, la Ing. y a mi compañero de tesis, el Ing. Iván Mauricio Salcedo.

A la Dra Ivonne Twigg Sandoval Cáceres y Tilcia Cáceres de Sandoval por su gran colaboración y apoyo incondicional.

## 5. REFERENCIAS

- [1] NASA. Pathfinder. Visitado en Marzo de 2005. URL: <http://mpfwww.jpl.nasa.gov/MPF/index1.html>
- [2] ANGULO USATEGUI, José María. ANGULO MARTINEZ, Ignacio y ROMERO YESA, Susana. Microcontroladores PIC "Diseño práctico de aplicaciones". 2 Ed. Madrid: McGraw Hill / Interamericana de España, S.A. 2000. Pág 1-6.
- [3] CARVAJAL T. Carlos Andres. Espectro electromagnético Visitado en Mayo de 2004. URL: <http://almaak.tripod.com/temas/espectro.htm>
- [4] ESCUELA de Ingeniería de Antioquia. Visitado en octubre de 2004. URL: <http://materiales.eia.edu.co/laboratorios/ultrasonido.htm>.

- [5] URL: [www.ab.com/catalogs](http://www.ab.com/catalogs)
- [6] MATHWORKS. Serial Port I/O. Visitado en Febrero de 2005.  
URL: <http://www.mathworks.com/access/helpdesk/help/techdoc/matlab.html>.
- [7] HANSELMAN Duane, LITTLEFIELD Bruce. Mastering Matlab 5. 7 Ed. New Jersey: Prentice Hall / Pearson Educación, S.A. 1998. Pág 654-664, 861-871.
- [8] BALCELLS, Joseph y ROMERAL. José Luis. Autómatas Programables. México D.F. Alfaomega grupo editor S.A. 1998. Págs. 119 y 126.
- [9] SANDOVAL C. Wilmer Alexis, SALCEDO R. Iván Mauricio. Tesis de grado "Estudio de Detección De Obstáculos mediante sensores ultrasónicos e infrarrojos para la realización de mapa bidimensional con representación gráfica en computador". Universidad Pontificia Bolivariana. Bucaramanga, Marzo de 2005.
- [10] STALLING William. 6 Ed. Madrid: Prentice Hall / Prentice Hall Hispanoamericana, S.A. 2000. Pág. 165-168.
- [11] STREMLER, Ferr el G. Introducción a los sistemas de comunicación. 3 Ed. Wilmington: Addison-Wesley Iberoamericana S.A. 1993. Pág. 608.
- [12] TOMASI Wayne. Sistemas De Comunicaciones Electrónicas 4 Ed. México: Prentice Hall / Pearson Educación, S.A. 2000.. Pág 535-566.
- [13] URRIZA MACAGNO, Roberto Ángel. Ingeniería del conocimiento "Robótica e Infraestructuras Inteligentes". Buenos Aires: Editores S.R.L 2003. Pág. 109-110.
- [14] ZITRAX, RS232 Cables, Wiring and Pinouts. Visitado en Enero de 2005.  
[http://www.zytrax.com/tech/layer\\_1/cables/tech\\_rs232.htm#db9](http://www.zytrax.com/tech/layer_1/cables/tech_rs232.htm#db9).

representación gráfica en **computador**", proyecto el cual fue validado como tesis de grado. Un año de experiencia en el desarrollo de aplicaciones con el puerto serial bajo plataforma Matlab.

**C.V.: Wilmer Alexis Sandoval Cáceres:** Ingeniero electrónico egresado de la UPB (Universidad Pontificia Bolivariana) seccional Bucaramanga, 3 años de experiencia en el campo de la robótica móvil, miembro 2002 del semillero de robótica de la UPB, miembro 2004-2005 del grupo de investigación GIDE de la Universidad Pontificia Bolivariana seccional Bucaramanga (co\_investigador), en el desarrollo del proyecto de investigación denominado "Estudio de la detección de obstáculos, mediante sensores ultrasónicos e infrarrojos para la realización de mapa bidimensional con