
Capítulo 4. Ambiente Robótico para el Mapeo de Entornos

Para la elaboración de mapas de entornos interiores de trabajo, se construyó el sistema *Map-Bot*, el cual integra módulos de hardware y software para la navegación, adquisición de información, transferencia de datos y elaboración de mapas. Como se puede apreciar en la figura 4.1, se incorpora un PC en el que corre el software responsable de la construcción de representaciones bidimensionales del entorno (*módulo Vespucci*), un agente robótico móvil que recoge información sensorial del sitio de trabajo y un módulo de comunicaciones inalámbricas para la transferencia de datos entre el PC y el robot.

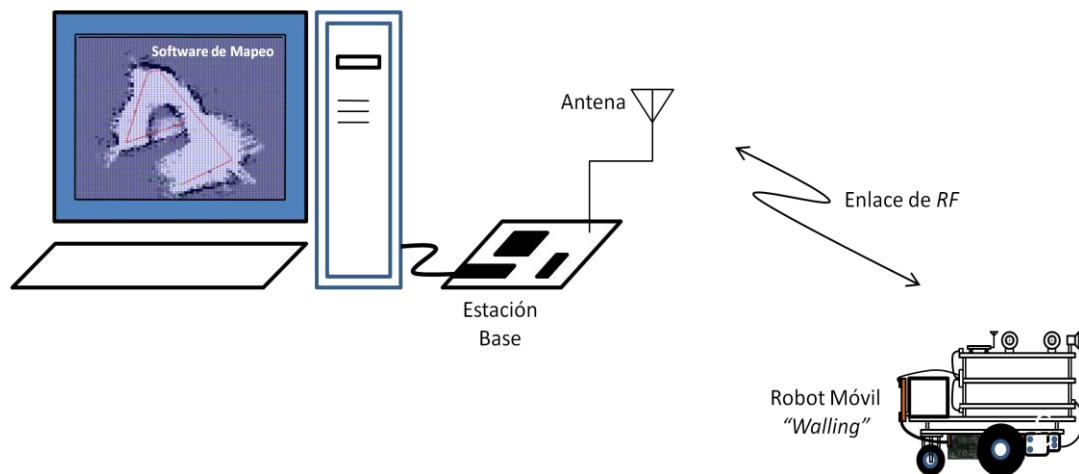


Figura 4.1. Arquitectura *Map-Bot* para el Mapeo de Ambientes Interiores.

4.1 Plataforma Robótica

La exploración del entorno de trabajo se lleva a cabo mediante el robot móvil *Walling*. Su nombre deriva del algoritmo de navegación denominado *Wall Following* (Seguimiento de paredes), una técnica de navegación reactiva ampliamente usada en la navegación robótica en ambientes interiores. *Walling* es un agente robótico que cuenta con subsistemas de percepción, control y actuación que le permiten explorar su entorno de trabajo de forma segura, evitando obstáculos a su alrededor. Durante la exploración, el robot recaba y transmite información de distancias a objetos circundantes, como por ejemplo paredes, sillas, mesas, etc., así como también la longitud de los trayectos recorridos.

Walling cuenta con dos bases en acrílico para la fijación de los efectores y las tarjetas de control y percepción. La inferior, corresponde una base circular de 12.9 centímetros de diámetro a la que se fijan todos los dispositivos que hacen parte del sistema de de efectores, son estos: una caja motorreductora doble con dos motores eléctricos DC, dos ruedas de tracción, una rueda de soporte tipo *caster ball* y un sistema de odometría. El tipo de locomoción corresponde a una configuración diferencial centrada con tercer punto de contacto. De esta manera se cuenta con una estructura mecánica que provee maniobrabilidad y estabilidad al robot.

A la segunda base de acrílico se fijan, modularmente, los circuitos de interfaz de efectores y control, sensórica y comunicaciones. El sistema de control de efectores permite controlar la velocidad y sentido de giro de los motores fijados a la caja de reducción. Se emplean dos puentes H TA7291SG, cada uno de los cuales proporciona una corriente promedio de hasta 400 mA. El sistema de control genera las señales PWM (*Pulse-Width Modulation*) con las que se activan los puentes H.

Las funciones de control de navegación y procesamiento de datos sensoriales se lleva a cabo mediante un sistema de desarrollo DEMOQE-128, cuyo dispositivo principal de control corresponde a un microcontrolador Coldfire MCF51QE128 de 32 bits, el cual corre a una frecuencia interna de bus de 26,6 MHz.

El sistema de percepción exteroceptiva y el de comunicaciones incorpora tres módulos ultrasónicos SRF02 para la medida de distancias. Estos sensores entregan datos de distancia a obstáculos cercanos al frente y al costado derecho del robot. Los trayectos recorridos y las medidas de rango, se transfieren al módulo de mapeo *Vespucci* que corre en el PC, a través de un módulo de comunicaciones inalámbricas *XBee* que soporta el protocolo *IEEE 802.15.4*.

Walling se diseñó y construyó teniendo en cuenta criterios de modularidad, potencia computacional y sensórica avanzada. La plataforma robótica presenta una arquitectura de cuatro niveles expandible y/o reconfigurable. Esto significa que se puede agregar o cambiar uno o varios de sus módulos por otros mejorados o de características diferentes. La arquitectura de *Walling* se observa en la figura 4.2.

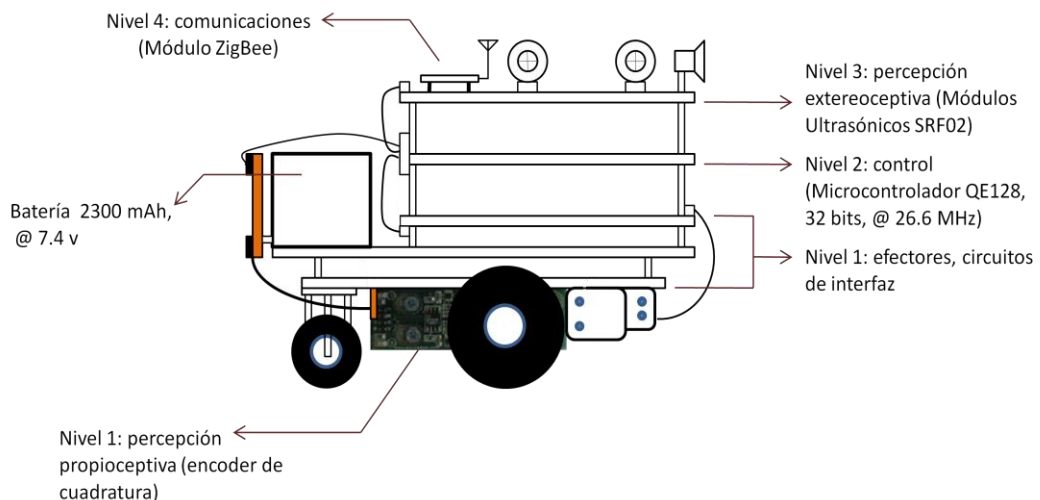


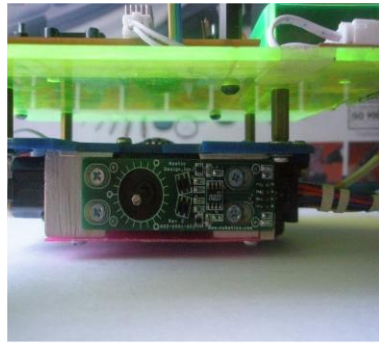
Figura 4.2. Arquitectura de cuatro niveles del Robot *Walling*.

4.1.1 Sensórica Propioceptiva del Robot *Walling*

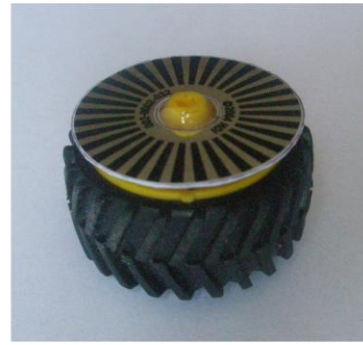
La elaboración automática de mapas del entorno, requiere de robots dotados con dispositivos de percepción que permitan determinar, no solo las distancias a los obstáculos circundantes, sino también las distancias lineales recorridas por el robot y su orientación relativa a un sistema de coordenadas [57], [72]. Las distancias recorridas por *Walling* durante la exploración del entorno, se determinan con base en un sistema odométrico de medida. El algoritmo de navegación del robot permite mantener la distancia a paredes localizadas en su costado derecho, lo que ayuda a conservar la orientación inicial (posición angular) del robot. Se obtienen así trayectorias de navegación casi rectilíneas.

El sistema odométrico de que dispone *Walling* permite determinar el número de vueltas completas o fracciones de vuelta que recorre cada una de sus ruedas. De esta manera es posible conocer la longitud de los desplazamientos que lleva a cabo el robot. El hardware incorpora un disco reflectivo (o disco codificado) con franjas de absorción y reflexión de luz infrarroja, el cual se encuentra acoplado a las ruedas de tracción del robot. Un led emisor de luz infrarroja proyecta un haz sobre el disco y este, refleja o no el haz, sobre un elemento fotodetector. Este fotodetector genera un tren de pulsos eléctricos a medida que la rueda, y el disco reflectivo acoplado a ella, llevan a cabo el desplazamiento angular. El disco cuenta con treinta y dos franjas reflectivas, y un número igual de franjas no reflectivas. Esto que implica que una revolución completa de la rueda genera treinta y dos pulsos eléctricos en el fotodetector.

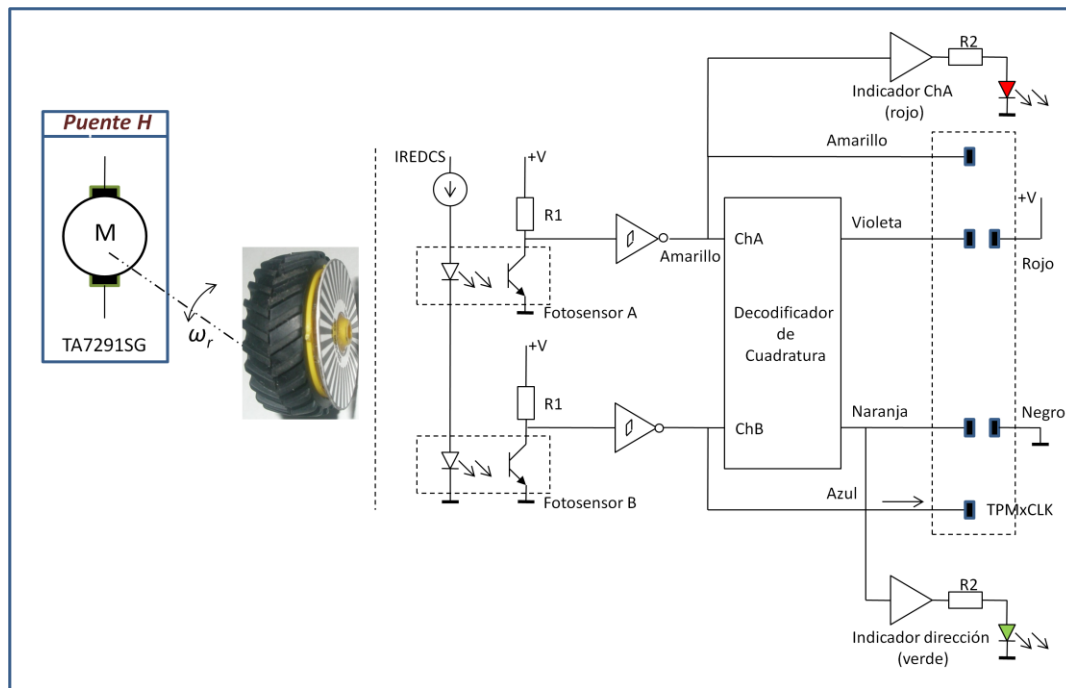
Puesto que el perímetro de las ruedas de tracción es de 11.62 cm, entonces resulta fácil determinar la distancia lineal que recorre el robot al contar los pulsos que genera el sistema de percepción. A esta técnica de medida se le conoce como *odometría* y se puede implementar con base en un dispositivo electrónico denominado *encoder*. El agente *Walling* cuenta con dos encoders de cuadratura para la medida del desplazamiento lineal de cada rueda. La odometría del robot se lleva a cabo con base en el dispositivo con referencia *WW-01* fabricado por *Neotic Design Inc*. El conteo de pulsos y el cálculo de distancias las realiza el procesador central del robot mediante interrupciones de los módulos de conteo y temporización TPM1 y TPM3. Una interrupción se genera cada treinta y dos pulsos. Las rutinas de interrupción correspondientes, actualizan las variables que almacenan los desplazamientos del robot. En la figura A.1 del anexo A se aprecia el diagrama de flujo de estas rutinas. En la figura 4.3 se aprecia el disco codificado acoplado a una de las ruedas del robot, el encoder *WW-01* anclado al chasis de la plataforma móvil y el diagrama electrónico correspondiente.



a)



b)



c)

Figura 4.3. Disco coficado acoplado a una de las ruedas del robot. a) Encoder anclado al chasis del robot y concéntrico al eje del motor, b) disco reflectivo acoplado a una de las ruedas de tracción de *Walling*, c) diagrama electrónico del conjunto driver-rueda-encoder.

En la figura 4.4 se presentan los resultados de las pruebas experimentales efectuadas al sistema odométrico del robot. Se tomaron lecturas de los recorridos lineales realizados por las ruedas derecha e izquierda para trayectos de navegación de 480 cm. Se tomaron dos muestras por tramo lineal de 48 cm demarcado sobre el piso, una muestra por rueda. La desviación respecto a los datos teóricos es mayor en la rueda izquierda que en la derecha. Derecha. Esto se debe a que el esquema de control de navegación efectúa correcciones sobre el efector derecho, por lo que la rueda izquierda “*seguirá*” siempre a la derecha.

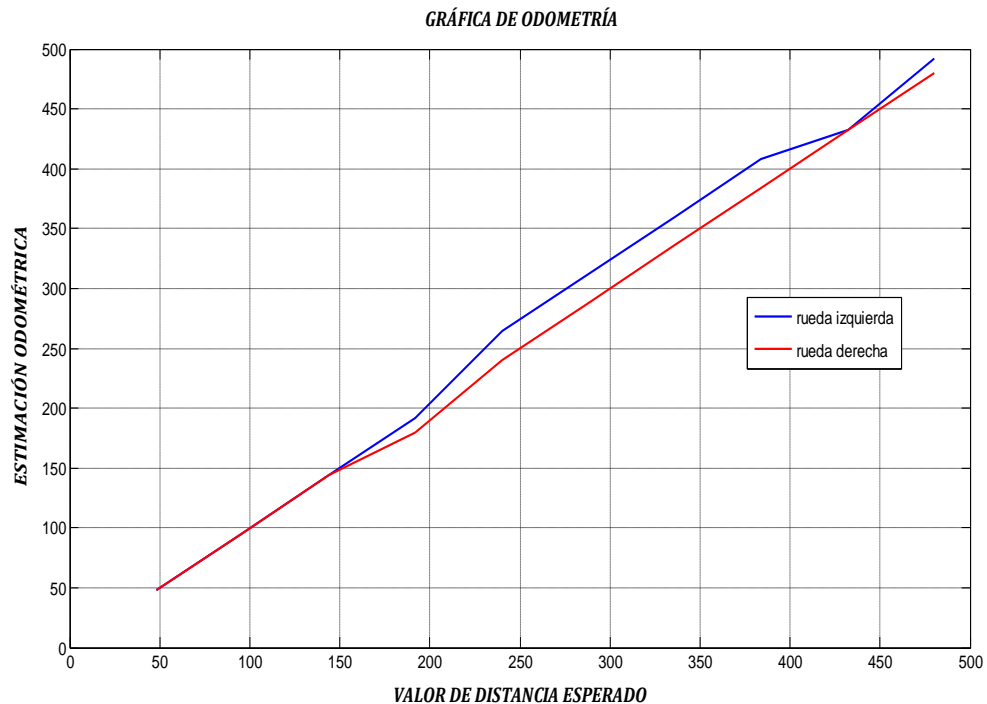


Figura 4.4. Prueba de estimación odométrica para trayectos de navegación de 480 cm.

4.1.2 Caracterización de los Sensores Extereceptivos del Robot *Walling*

La plataforma robótica *Walling* cuenta con un sistema de sonares dispuestos ortogonalmente sobre la placa de circuito *SenCom_Card*. Esta disposición le permite al robot obtener medidas de distancia a obstáculos localizados en su entorno próximo. A partir de estos datos, módulo de navegación *Magalhães* genera las señales de control que permiten una navegación segura del robot en su entorno. Las lecturas proporcionadas por los sensores *S0* y *S1* se emplean para implementar un control de navegación por seguimiento de contornos. En este tipo de navegación el robot bordea el contorno de los objetos a su alrededor y mantiene una distancia respecto a estos. El sensor *S2* se emplea para la evitación de obstáculos localizados al frente del robot.

La sensorica extereceptiva del robot *Walling* se implementó con base en el sonar SRF02, un dispositivo ultrasónico de rango que opera a una frecuencia de 40 KHz. La transferencia de comandos y datos entre el sonar y el procesador central se realiza a través de una interfaz *UART* con formato de trama de 9600, 8, N, 1 y niveles TTL. Este transductor permite el trabajo simultaneo de hasta dieciséis módulos SRF02 pueden conectarse a un bus serial de comunicaciones, siendo necesario configurar una dirección de ocho bits, diferente para cada sonar. El rango de direcciones disponibles va de 0x00 a 0x0F. En la tabla 4.1 se dan las especificaciones técnicas de los módulos SRF02 y en la figura 4.5 se observa el sensor empleado y su patrón de radiación acústica.

Tabla 4.1. Especificaciones técnicas del sensor SFR02.

ESPECIFICACIONES TÉCNICAS DEL SONAR SFR02	
Voltaje	5v
Corriente	4mA (típica)
Frecuencia	40KHz
Rango Máximo	600cm
Rango Mínimo	15cm
Ganancia	Control Automático de 64 pasos
Conectividad	Bus I2C, Serial UART
Ajuste (calibración)	Automática en el encendido
Unidades de la medida	µs, cm, pulgadas
Peso	0.16226 oz.
Dimensiones	24mm w, 20 mm d, 17mm h

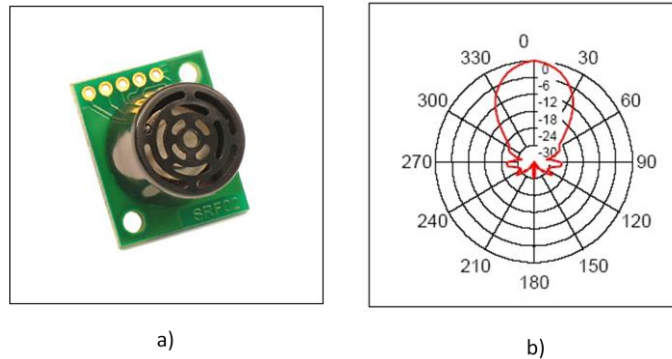
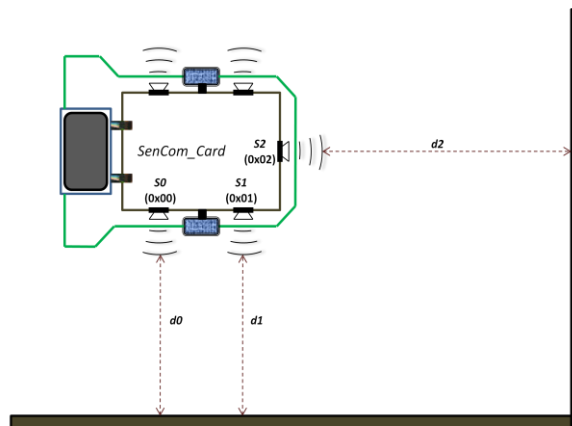


Figura 4.5 Descripción del sensor SRF02. a) Apariencia del sensor ultrasónico SRF02, b) patrón de radiación acústica.

Si bien el módulo de control de navegación emplea lecturas de tres sensores, dos laterales ($S0$, $S1$) y uno frontal ($S2$), la tarjeta *SenCom_Card* soporta *hasta* cinco sensores, dos adicionales localizados al costado izquierdo de la plataforma. En la figura 4.6 se aprecia la distribución de sensores sobre la placa de circuito impreso y la dirección asignada a cada dispositivo. La asignación de direcciones en modo serial *UART* se hace mediante una rutina diseñada para tal fin, la cual se denominó *set_addres*. El diagrama de flujo correspondiente se aprecia en la figura A.2 del anexo A.



$d0$, $d1$, $d2$: medidas de rango

Figura 4.6. Sensórica exteroceptiva del robot *Walling*.

En los sensores ultrasónicos los errores de medida se deben básicamente a dos factores que son: la apertura angular del haz acústico de emisión/recepción y la inclinación de la superficie del obstáculo con respecto al plano normal al eje acústico del sonar [53], [54]. En tal sentido se hizo necesario realizar una serie de pruebas que permitieran caracterizar el dispositivo empleado.

4.1.2.1 Pruebas de Distancia

Las pruebas efectuadas consistieron en contrastar medidas de distancia entregadas por el sensor con distancias reales. Se pudieron verificar algunos de los datos proporcionados por el fabricante, como la medida de rango mínima y máxima, y obtener otros no disponibles en la hoja de datos del dispositivo, como por ejemplo el error relativo, el error absoluto y el rango de linealidad del dispositivo.

Pruebas de distancia de 4 a 40cm (cortas distancias)

Se llevaron a cabo una prueba de corta distancia con el fin de establecer las distancias de referencia mínimas para el mapeo y navegación del entorno. En esta prueba se empleó una superficie lisa rectangular de 10cm x 16cm en acrílico, la cual se dispuso de manera ortogonal al eje acústico del sonar, sobre un trípode y a una distancia inicial de 4 cm. El montaje experimental se puede apreciar en la figura 4.7.

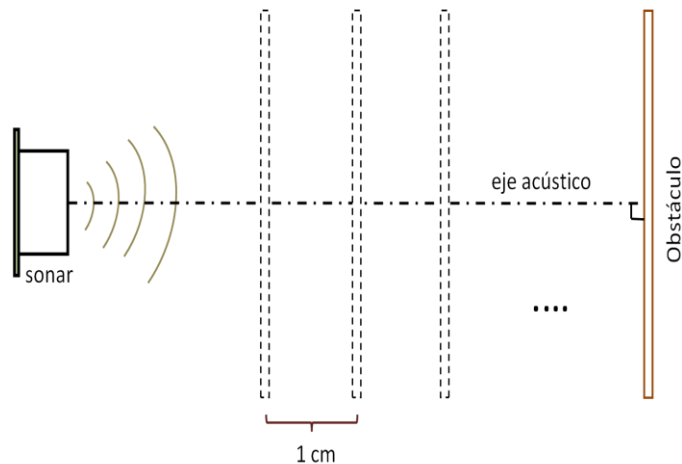


Figura 4.7. Montaje experimental para medidas de distancia con el sonar SFR02.

Se tomaron tres medidas por cada 1 cm de separación entre el sonar y la superficie, hasta llegar a los 40 cm. Se promediaron los tres datos en cada punto de medida obteniéndose las gráficas que se observan en la figura 4.8. De su análisis se desprende que para el rango de distancias comprendido de 4 a 15 cm los errores de medida son significativos. Esto se debe a que el sensor dispone de una sola cápsula de emisión/recepción, no siendo posible disponer de medidas confiables mientras la capsula opera como emisor. A este rango de distancias se le denomina “zona muerta”. En este sentido los datos obtenidos se ajustan a las especificaciones del fabricante. Sin embargo, como se puede apreciar en las graficas, se ha señalado una *zona muerta* que va hasta los 20 cm, siendo a partir de esta distancia que se observa un comportamiento más lineal del transductor con mejores valores menores del error absoluto y relativo. Las pruebas permitieron establecer como nueva *zona muerta* una distancia menor o igual a 20 cm. Siendo así que se deben establecer valores superiores a 20

cm para las distancias de referencia, como por ejemplo distancias de seguridad para la evitación de obstáculos y seguimiento de contornos.

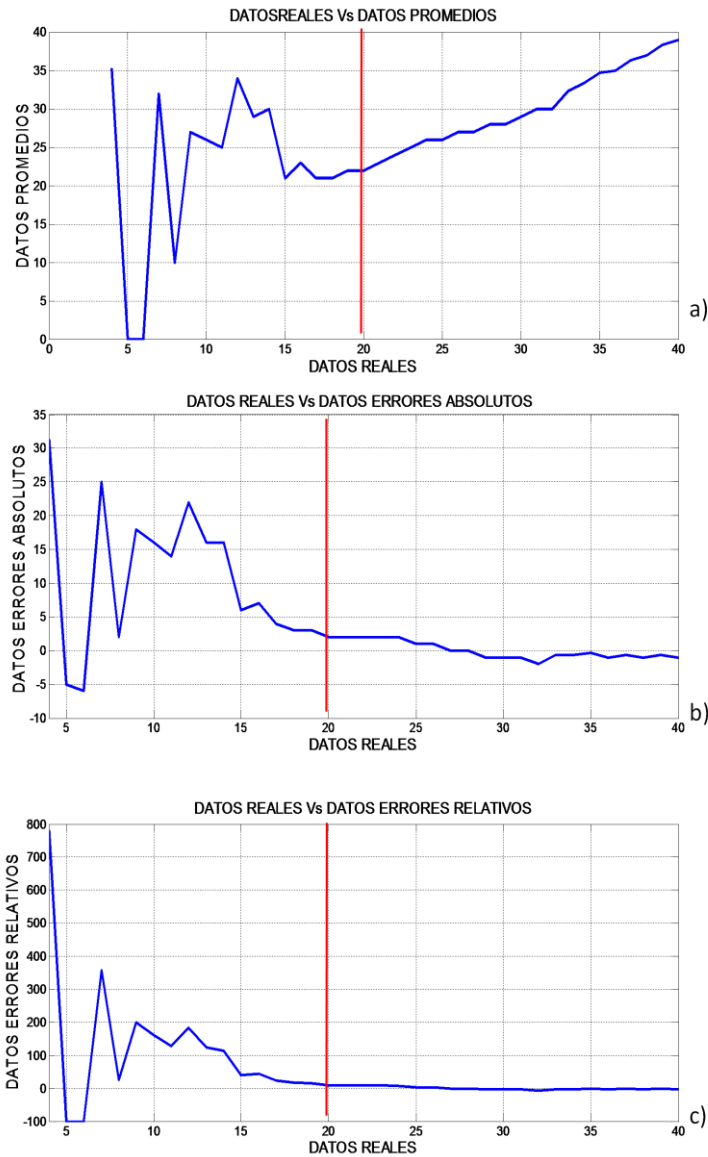


Figura 4.8. Resultados de la prueba de distancia de 4 a 40 cm. a) Distancia medida vs distancia real, b) error absoluto y c) error relativo.

Pruebas con distancias superiores a 40cm (largas distancias)

La prueba de larga distancia permitió establecer el grado de visibilidad de un obstáculo de dimensiones fijas, a medida que se aleja del transductor. La prueba permite establecer distancias de saturación para la implementación de una navegación segura del robot en su entorno.

En esta prueba se empleó la misma superficie reflectora usada en las pruebas de corta distancia. Durante las pruebas se tomaron tres muestras por cada 5 cm de separación de la superficie, iniciando en 40 cm. Se considera el promedio de las tres muestras como dato de

distancia válido. La prueba culmina una vez el sensor no puede establecer la presencia de la superficie de acrílico que se le coloca al frente. Los resultados se presentan en la figura 4.9. En este caso la *visibilidad* de la superficie en acrílico se da hasta una distancia de 240cm, considerando que esta superficie se orientó ortogonalmente con respecto al eje acústico del sensor. También se aprecia un incremento en el error absoluto a medida que el obstáculo se aleja del sensor, normal para este tipo de sensores.

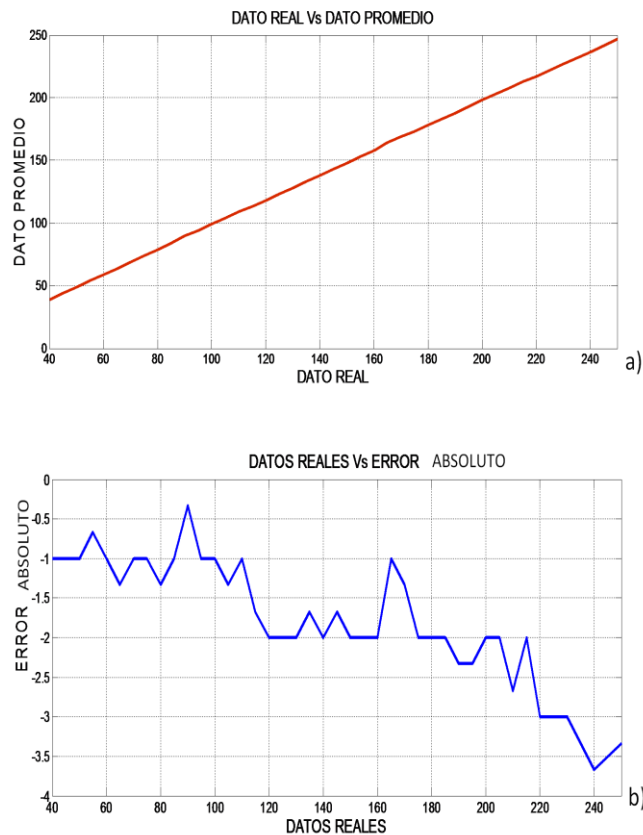


Figura 4.9. Resultados de la prueba de distancias mayores a 40 cm. a) distancia medida vs distancia real y b) error absoluto.

4.1.2.2 Prueba de Reflectividad

Esta prueba permite determinar el ángulo máximo de inclinación de la superficie reflectora para el cual se obtiene medidas de rango confiables, o en otras palabras, el valor límite de inclinación en el que no se presentan reflexiones especulares.

La prueba se realizó con el obstáculo ubicado a una distancia de 50 cm, con cambios de orientación angular de cinco grados por medida. Se consideraron desplazamientos tanto en sentido horario como en sentido contra horario, pudiéndose establecer que el comportamiento de los sensores es simétrico. En la figura 4.10 se ilustra la manera en la cual se llevó a cabo esta prueba.

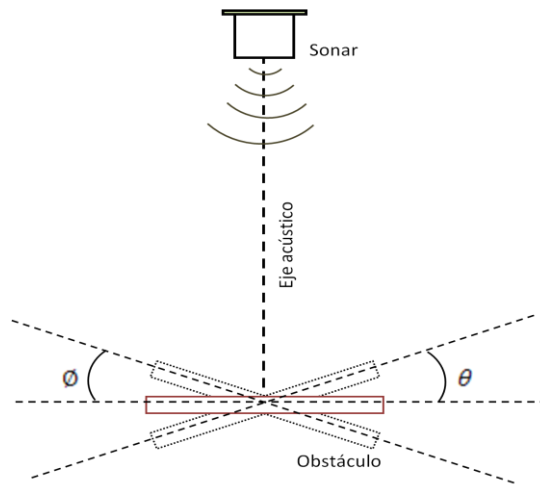


Figura 4.10 Prueba de reflectividad del sonar SFR02.

Para cada posición angular de la superficie reflectiva se tomaron tres lecturas de distancia. Los datos obtenidos se presentan en la tabla 4.2.

Tabla 4.2. Medidas de distancia entregadas por el sonar para diferentes posiciones angulares de la superficie reflectiva.

<i>Desplazamiento angular en sentido horario</i>				<i>Desplazamiento angular en sentido contra-horario</i>			
θ	Lectura 1 (cm)	Lectura 2 (cm)	Lectura 3 (cm)	Φ	Lectura 1 (cm)	Lectura 2 (cm)	Lectura 3 (cm)
0°	49	49	49	0°	49	49	49
5°	49	49	49	5°	49	49	49
10°	49	49	49	10°	49	49	49
15°	50	50	50	15°	50	50	50
20°	193	51	193	20°	190	190	190

Tanto para desplazamientos angulares en el sentido horario, como en el contra-horario, la reflexión especular de la señal sonora se produce a partir de los 20° . Esto significa que solo serán detectados, de manera confiable, aquellos obstáculos con una inclinación no mayor a $\pm 15^\circ$ con respecto al eje acústico del sensor. El fenómeno de reflexión especular tiene una fuerte incidencia en el mapeo y navegación de los entornos de trabajo. Sus efectos negativos se pueden reducir con base en una adecuada distribución y disposición geométrica de los sonares sobre la plataforma móvil, teniendo en cuenta además la geometría de los obstáculos circundantes y el material del cual están hechos.

Las pruebas de mapeo se llevaron a cabo en ambientes estructurados y bajo condiciones controladas, por lo que el fenómeno de reflexión especular no impidió obtener mapas representativos del entorno.

4.1.2.3 Prueba de Apertura Angular del Haz Ultrasónico

La prueba de apertura angular de haz ultrasónico permite determinar el campo de visión del sensor. Las medidas se realizaron sobre una superficie lisa en acrílico de 10cm x 16cm ubicada a una distancia de 34cm. La superficie se desplazó a intervalos de 10° de izquierda a derecha con una orientación orto radial centrada en el sensor, tal como se observa en la figura 4.11.

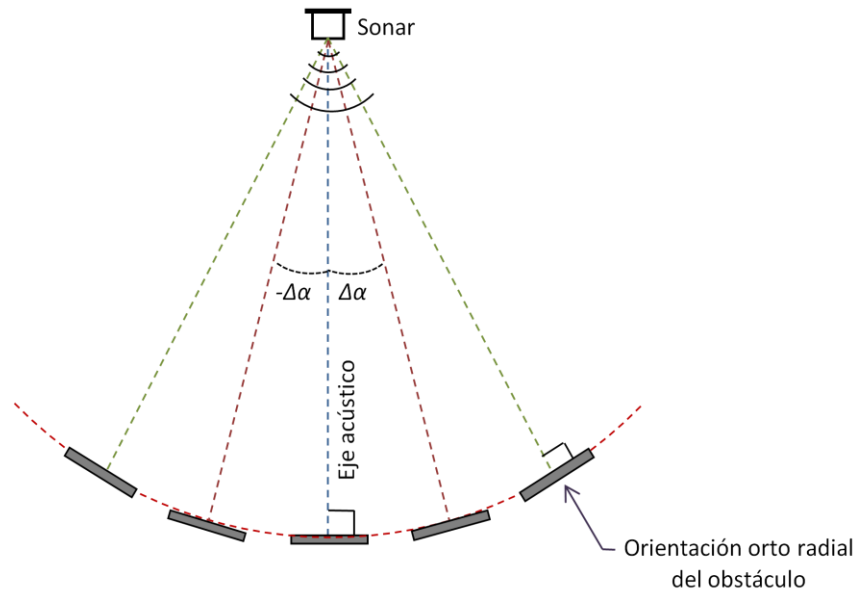


Figura 4.11. Montaje para prueba de apertura angular del haz del sensor ultrasónico.

En la tabla 4.3 se presentan los resultados de la prueba. Los datos obtenidos para desplazamientos angulares positivos indican una apertura de haz que alcanza los 30° , obteniéndose lecturas erróneas de distancia a 40° . Para desplazamientos negativos, la apertura del haz llega hasta los -20° , y a los -30° se comienzan a obtener registros erróneos.

La prueba permite deducir que el lóbulo principal de radiación de la señal sonora, o campo de visión del sensor, es asimétrico, y se extiende de -20° a $+30^\circ$. En las pruebas efectuadas se tomó como referencia la cara posterior del sensor. La prueba permitió asumir como campo de visión aquel comprendido de -20° a $+20^\circ$. Al establecer esta apertura de haz se reducen las incertidumbres en las medidas por reflexiones especulares. Así mismo, un control de seguimiento de contornos que mantenga estable la distancia a las paredes, permitirá restringir la desviación angular del robot en un rango de valores en el cual la pared siempre aparecerá *visible* para el robot.

Tabla 4.3. Datos de la prueba de apertura angular del sensor SFR02.

<i>Desplazamiento angular positivo (sentido contra horario)</i>				<i>Desplazamiento angular negativo (sentido horario)</i>			
α	Lectura 1 (cm)	Lectura 2 (cm)	Lectura 3 (cm)	α	Lectura 1 (cm)	Lectura 2 (cm)	Lectura 3 (cm)
0°	33	33	33	0°	33	33	33
10°	33	33	33	-10°	33	33	33
20°	32	32	33	-20°	32	32	33
30°	34	34	34	-30°	193	234	227
40°	193	234	227	-40°	-	-	-

4.1.2.4 Prueba de Reflexión Angular para Dos Sensores Colineales

Esta prueba se llevó a cabo con la tarjeta *SenCom_Card* montada sobre la plataforma móvil y los sensores en su disposición final sobre la tarjeta en mención. Corresponde a una prueba de medida a la pared de un entorno típico de navegación, en este caso un aula de clase. La prueba se llevó a cabo con el fin de determinar si el desplazamiento angular de la plataforma, con respecto a la pared, originaba reflexiones especulares en ambos sensores a partir de la misma medida angular. Los sensores se ubicaron de forma colineal a una distancia de 5.7cm entre sus centros. Se consideró una separación de 100cm entre la plataforma robótica y la pared para una orientación de 0° , tal como se observa en la figura 4.12.

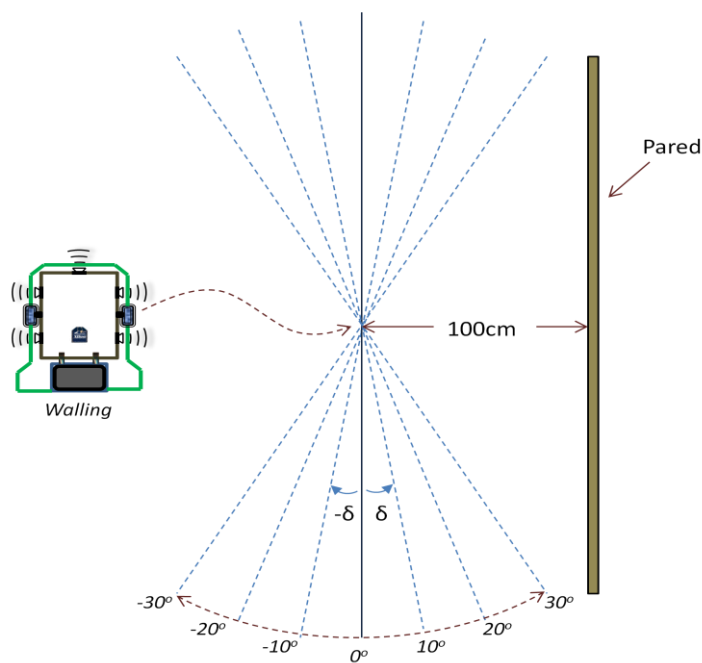


Figura 4.12. Prueba de reflexión especular para dos sensores colineales dispuestos sobre el robot *Walling*.

En la prueba realizada se pudo establecer que el sensor *S1* comenzó a entregar medidas de distancia poco fiables a partir de una orientación de 23° en sentido contra horario. En el sentido horario, las incertidumbres en las medidas de rango se registraron a partir de los -25° para este mismo sensor. Finalmente, con orientaciones de ±30°, o superiores, la reflexión especular no permitió obtener medidas de rango correctas para ambos sensores. La prueba permite seleccionar como seguro, un rango de orientaciones del robot respecto a la pared, de -20° a +20°. Este hecho transfiere la responsabilidad de mantener la orientación del robot al algoritmo de navegación. Lo que significa que, conociendo la posición inicial del robot, correspondiente a una separación lateral respecto a la pared derecha de 50cm y una orientación de 0°, el algoritmo de control deberá mantener no sólo esta separación, sino también la orientación relativa (a la pared), dentro el rango mencionado anteriormente.

Tabla 4.4. Datos de reflectividad para dos sensores colineales.

-25°		-20°		-10°		0°		10°		20°		23°	
<i>S0</i>	<i>S1</i>	<i>S0</i>	<i>S1</i>	<i>S0</i>	<i>S1</i>	<i>S0</i>	<i>S1</i>	<i>S0</i>	<i>S1</i>	<i>S0</i>	<i>S1</i>	<i>S0</i>	<i>S1</i>
103	213	103	102	103	102	102	101	100	101	100	102	101	214
103	0	103	102	103	102	102	102	100	101	100	102	100	109
103	0	103	102	103	102	101	102	100	101	100	102	100	0
103	102	103	102	103	102	101	101	100	101	100	102	100	0
103	102	103	102	103	102	101	101	100	101	100	102	101	219
103	220	103	102	103	102	101	101	100	101	100	102	101	111
103	223	103	102	103	102	102	101	100	101	100	102	101	111
103	110	103	102	103	102	102	101	100	101	100	102	101	110
103	110	103	102	103	102	102	101	100	101	100	102	101	222
103	0	103	102	103	102	102	101	100	101	100	102	101	223

4.2 Interfaz de Comunicaciones

Para la transferencia bidireccional de datos entre el PC y el agente robótico *Walling* se diseñó y construyó una estación base (*Base Station*) de comunicaciones inalámbricas a la cual se le dio el nombre de *SCoI* (*Subsistema de Comunicaciones Inalámbricas*). El módulo *SCoI* consta de:

- Cable de conversión de puerto USB a puerto serial
- Circuito integrado MAX-232
- Jack de alimentación
- Circuito de regulación de tensión
- Módulo de comunicaciones inalámbricas *XBee*
- Led de señalización de alimentación

El componente principal del sistema *SCoI* es el módulo *XBee*, con el cual se consigue implementar un enlace de comunicaciones inalámbricas punto a punto que opera a una frecuencia de 2.4 GHz en la banda de ISM (*Industrial, Scientific and Medical Band*). El módulo cumple con el estándar *IEEE 802.15.4* y permite integridad de datos hasta distancias

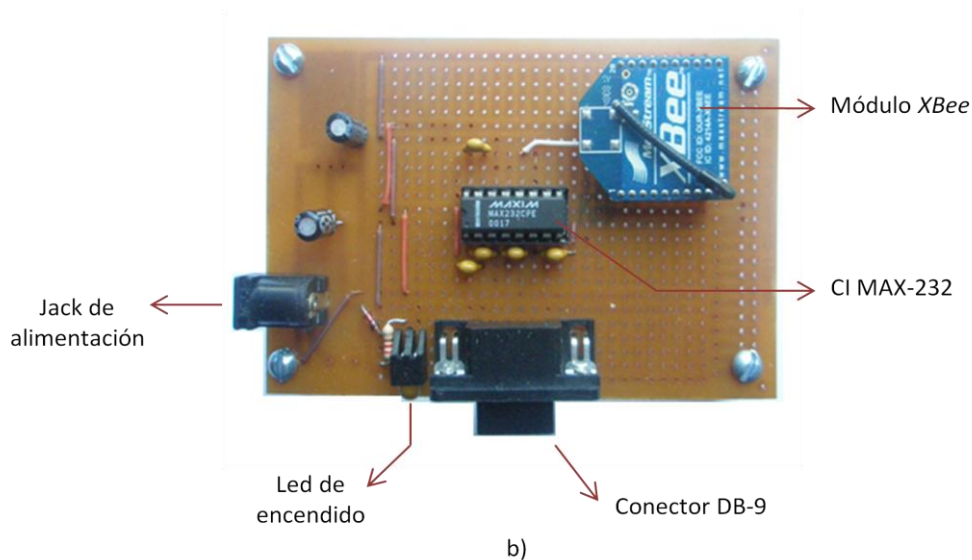


Figura 4.13. Tarjeta del subsistema SCoI. a) Diagrama electrónico del subsistema de comunicaciones inalámbricas SCoI, b) distribución de componentes en la placa de circuito impreso.

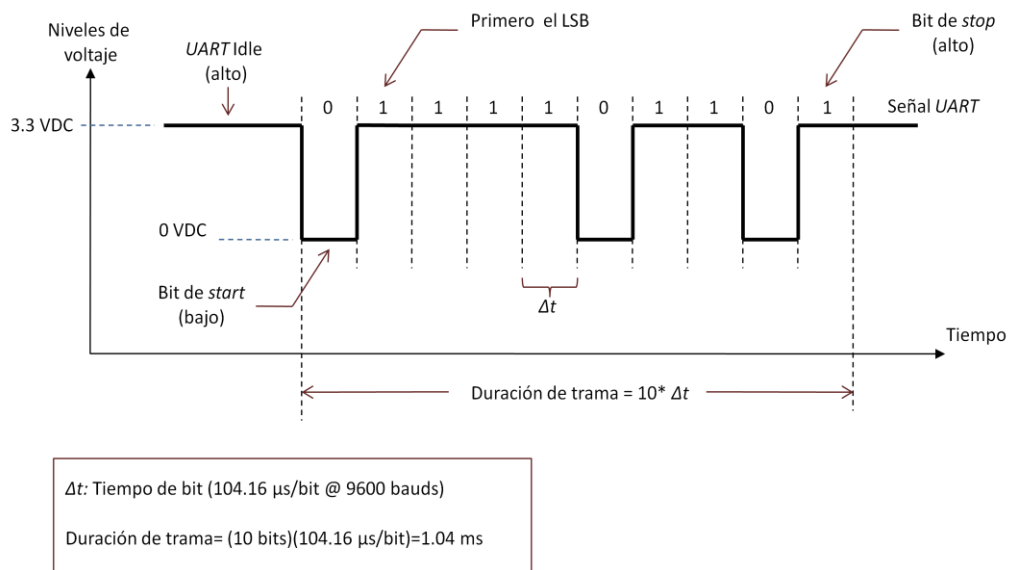


Figura 4.14. Paquete de datos UART que circula por el módulo de comunicaciones SCoI. Se ilustra la transmisión del carácter ASCII "o" (111 decimal, 0x6F hexadecimal).

4.3 Módulo de Navegación Magalhães

El módulo de navegación Magalhães asegura la navegación del agente robótico móvil *Walling* en su entorno. Este módulo de software implementa dos comportamientos reactivos para la navegación. El primero corresponde a un algoritmo de control para el seguimiento de contornos, como por ejemplo paredes. La navegación basada en el seguimiento de contornos es soportada por los sensores ultrasónicos *S0* y *S1*. El controlador

fuzzy implementado permite mantener una distancia de cerca de 50 cm a objetos ubicados en el costado derecho del robot. El otro comportamiento consiste en evitar obstáculos localizados al frente del robot. Este comportamiento se establece a partir de lecturas de sonar proporcionadas por el sensor S3. Una vez a se detecta un obstáculo a una distancia menor o igual a 50 cm, el módulo de control procesa la medida y genera las acciones de control requeridas para evitar el obstáculo detectado. Estas acciones se ejecutan secuencialmente y consisten en: 1) detener ambos motores y 2) activar el motor derecho para conseguir un desplazamiento de un cuarto de vuelta en la rueda derecha, con lo que se consigue efectuar un giro de 90 grados hacia la izquierda para evitar el obstáculo. Una vez se efectúa el giro correspondiente, el control fuzzy de seguimiento de pared reasume el control de la navegación del agente robótico móvil. En la figura 4.15 se ilustra un escenario de navegación típico y una de las posibles trayectorias realizadas por el robot.

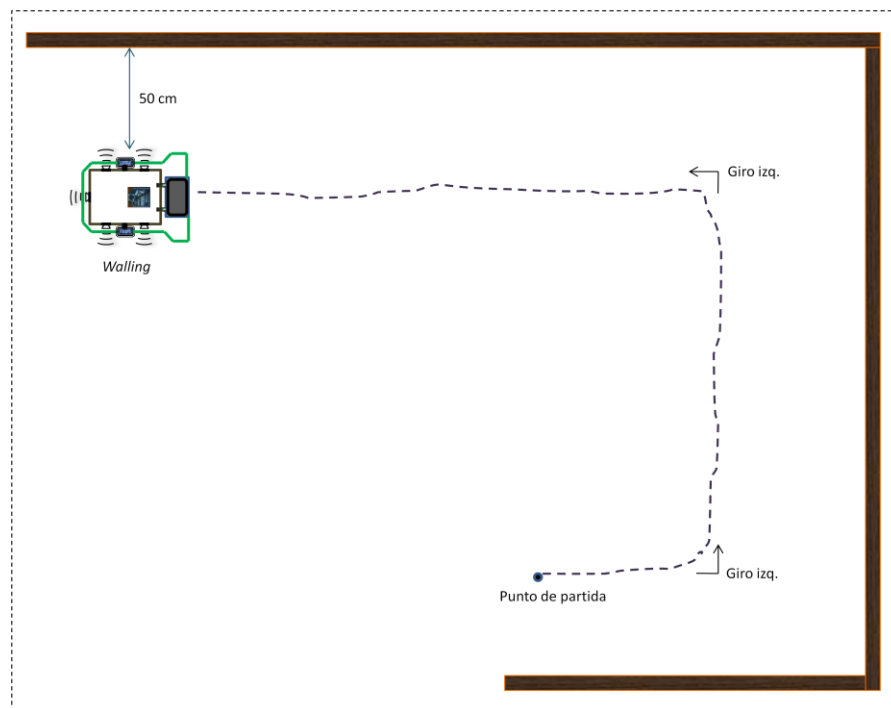


Figura 4.15. Trayectoria de navegación típica del agente robótico *Walling* en un entorno estructurado.

La placa de circuito impreso *SenCom_Card*, localizada en la parte superior *Walling*, incorpora la sensórica exteroceptiva necesaria para la navegación, así como también el módulo de comunicaciones inalámbricas para la transferencia de datos entre el PC y el robot. En la figura 4.16 se aprecia el esquema electrónico de la tarjeta electrónica y la disposición final de componentes en la placa de circuito.

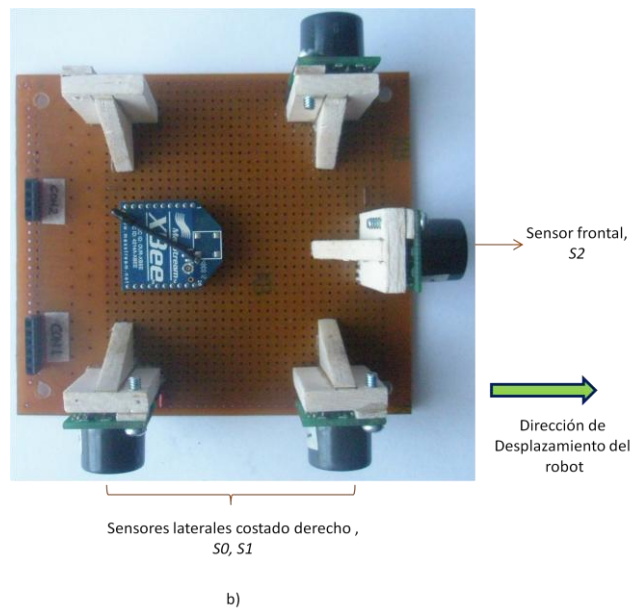
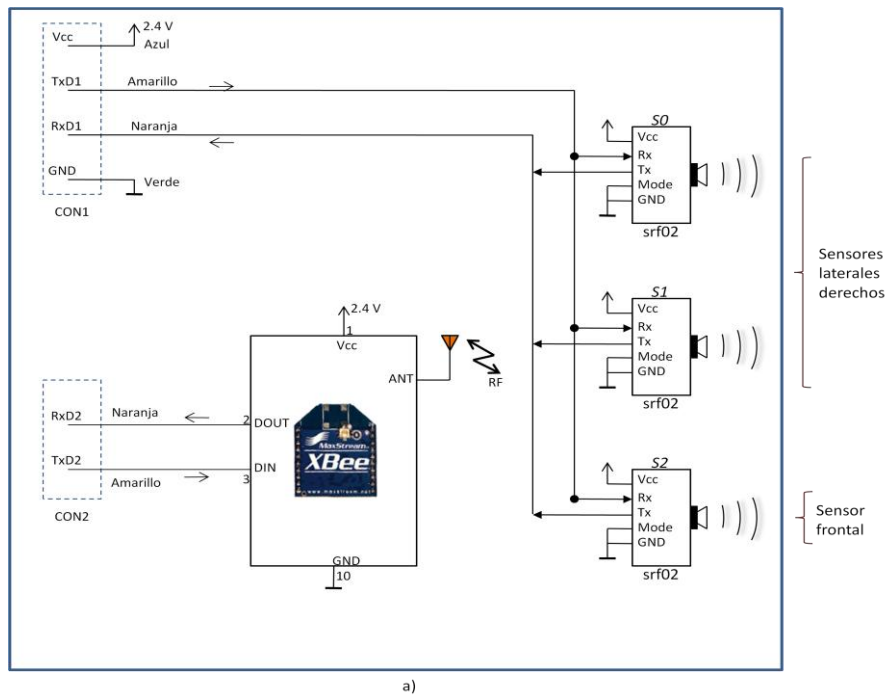


Figura 4.16. Tarjeta del sistema *SenCom_card*. a) Diagrama electrónico y b) placa de circuito de la tarjeta *SenCom_card* para la percepción exteroceptiva y las comunicaciones inalámbricas del robot *Walling*.

4.3.1 Arquitecturas de Control de Plataformas Robóticas Móviles

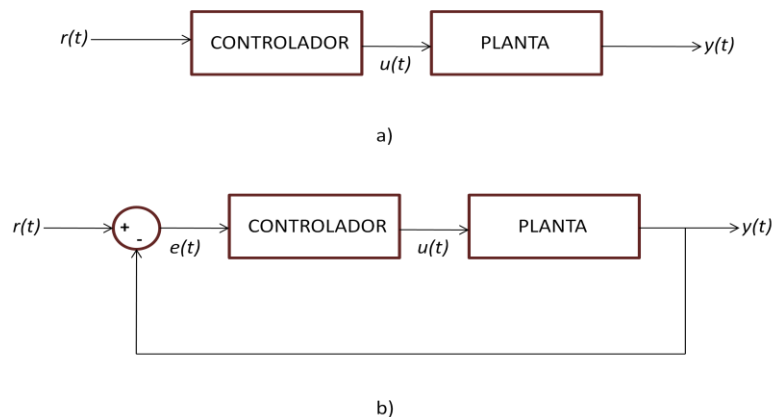
En robótica móvil una arquitectura de control corresponde a una estructura de hardware y software que permite adquirir y procesar información sensorial del entorno, llevar a cabo funciones cognitivas y generar señales para el control de efectores. En tal sentido la

arquitectura representa, organiza e interconecta la percepción, el razonamiento y la actuación [48]. Predominan dos paradigmas de control para robots móviles, uno de bajo nivel y otro de alto nivel. El principal objetivo del control de bajo nivel es provocar que un sistema alcance y mantenga un estado deseado [93]. De otro lado, el control de alto nivel involucra tres primitivas generalmente aceptadas que son: sensado, planeación y actuación (SPA), las cuales son aceptadas como constitutivas de toda acción robótica de alto nivel [97].

4.3.1.1 Control de Bajo Nivel

Los esquemas de control de bajo nivel buscan llevar y mantener una o más variables de interés en unos valores deseados. Por ejemplo para una plataforma robótica provista de dos motores eléctricos en configuración diferencial se puede requerir de una estructura de control que iguale y mantenga constante la velocidad angular de ambos motores con el fin de alcanzar un desplazamiento puramente lineal, es decir sin desviaciones [93], [97].

Para el control de bajo nivel se puede emplear bien sea una configuración de bucle abierto o una de bucle cerrado, tal como se puede observar en la figura 4.17.



$r(t)$: valor deseado de la variable (consigna o set-point)
 $u(t)$: acción o variable de control
 $y(t)$: variable controlada, p. ej: velocidad angular
 $e(t)$: variable de error, $e(t)=r(t)-y(t)$

Figura 4.17. Estructuras de control de bajo nivel en robótica móvil, a) estructura en lazo abierto y b) estructura en lazo cerrado.

En la estructura de control en lazo abierto no se tiene la certeza de que la variable controlada alcance finalmente el valor deseado, esto se debe a que en este tipo de estructura no se incorpora un bloque de medida con bucle de realimentación. En tal sentido no permite corregir desviaciones en la variable controlada debido a perturbaciones externas. El control en lazo abierto requiere, adicionalmente, de un elevado nivel de conocimiento de la planta o proceso que se va controlar [93].

Por otro lado en las estructuras de control en lazo cerrado, se realimenta a la entrada la variable de salida o una función de esta para generar una señal de error $e(t)$. Esta señal permite al controlador generar la acción de control $u(t)$ con la cual se corrige cualquier desviación por perturbaciones entre la variable controlada $y(t)$ y el valor de la consigna $r(t)$. Las estructuras de control en bucle cerrado permiten llevar y mantener el error $e(t)$ en cero,

de tal manera que la variable controlada alcanza el valor de la consigna aún cuando el sistema sea perturbado. Los sistemas de control en lazo cerrado requieren de hardware adicional como sensores y comparadores, pero no exige un conocimiento detallado del comportamiento de la planta o proceso. La realimentación negativa de la salida hace que el sistema sea menos sensible a variaciones en los parámetros de la planta. Así mismo los componentes del sistema no necesitan ser muy precisos y costosos [93], [97].

4.3.1.2 Control de Alto Nivel

Las estructuras de control de alto nivel se clasifican en tres grandes grupos: estructuras deliberativas o jerárquicas, estructuras reactivas y estructuras híbridas, tal y como se puede apreciar en la figura 4.18.

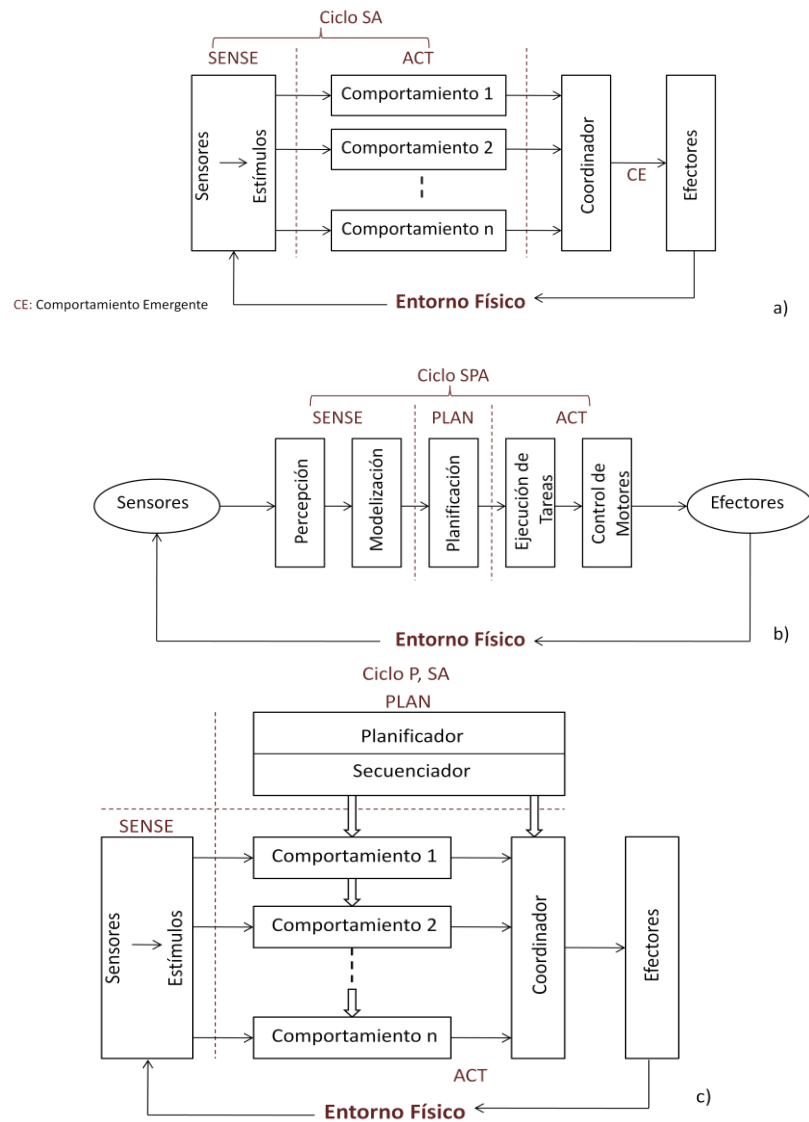


Figura 4.18. Estructuras de control de alto nivel en robótica móvil. a) Arquitectura reactiva b) arquitectura deliberativa y c) arquitectura híbrida (deliberativa-reativa).

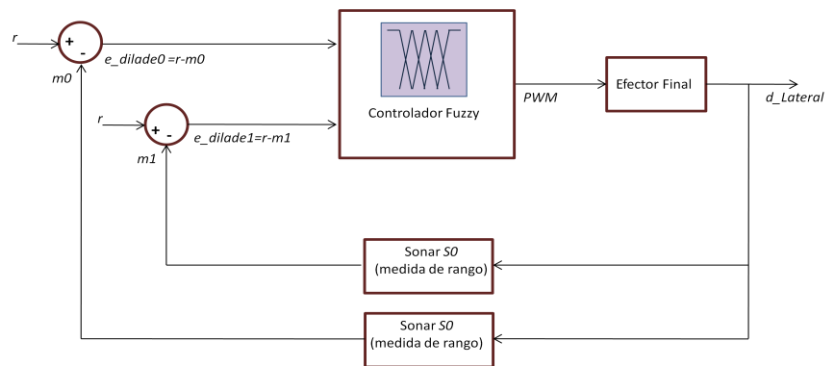
Mientras que las estructuras de control jerárquico la planificación requiere de la construcción de modelos del entorno, en los sistemas de control reactivo se asocia a los estímulos capturados del entorno una acción inmediata. Esto significa que la información sensorial suministrada por los módulos de percepción del robot se procesa lo menos posible antes de generar una acción o comportamiento. El comportamiento reactivo, generalmente se asocia a respuestas reflejas motoras en las que se omite el uso y elaboración de representaciones abstractas del mundo [99], [101]. En este sentido el control reactivo consiste, básicamente, en un mapeo estímulo-respuesta para un estado particular del entorno con el cual se logran satisfacer los requerimientos de bajo nivel impuestos al robot. Por otro lado las estructuras híbridas incorporan una base reactiva de bajo nivel y un planificador en el nivel superior [99].

En el campo de la robótica se reconocen tres primitivas mediante la cual se descompone cualquier acción robótica de alto nivel, estas son: sensado (*sense*), planificación (*plan*) y acción (*act*). Las arquitecturas mencionadas se diferencian básicamente en la forma en que se interrelacionan estas tres primitivas y en como la información sensorial se procesa y distribuye a lo largo del sistema. Mientras que las arquitecturas reactivas permiten tratar con situaciones impredecibles aportando velocidad de respuesta a cambios imprevistos del entorno, las deliberativas permiten generar y mantener representaciones del entorno de gran utilidad en la planificación de desplazamientos [100].

Las arquitectura híbrida busca reunir en sí lo mejor de ambos paradigmas, el reactivo y el deliberativo, incorporando esquemas de razonamiento deliberativo y ejecución de acciones basadas en comportamientos [100], [101].

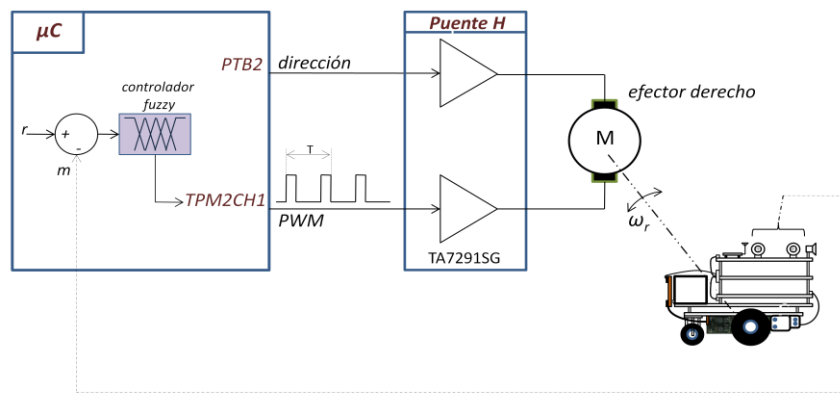
4.3.2 Aspectos Generales Relativos al Algoritmo *Fuzzy* de Seguimiento de Contornos

El diseño de un controlador difuso a partir de la identificación y caracterización las variables lingüísticas que intervendrán en el proceso [100], [101]. Se consideraron dos variables de entrada, nombradas como *e-dilade0* y *e-dilade1*, correspondientes al error en las distancias laterales entregadas por los sensores ultrasonicos *S0* y *S1* respectivamente. Para este controlador se estableció una referencia o *set-point* de 50 cm, lo que significa que el robot deberá mantener esta separación respecto a la pared localizada en el costado derecho. La figura 4.19 presenta la estructura de control de navegación en el robot *Walling*, se puede observar como la señal de control *PWM* se aplica al motor derecho a través de un puente H. Se logra con esto disminuir o aumentar la velocidad angular de la rueda derecha con relación a la izquierda, con lo que se consigue que el robot se aleje o acerque a la pared según se requiera.



r : set-point o consigna (50 cm)
 e_{dilad0} , e_{dilad1} : errores en la distancia lateral de los sensores S0 y S1
 PWM: señal de control aplicada al efector derecho
 $d_{Lateral}$: variable controlada, distancia lateral derecha

a)



M : motor o efector
 ω_r : velocidad angular del efector derecho
 m : medidas de rango sonares S0 y S1
 r : set-point o consigna

b)

Figura 4.19. Estructura de control de navegación en el robot Walling. a) Diagrama de bloques del controlador de navegación *fuzzy* del robot Walling, b) bosquejo de la estructura de control.

Inicialmente se llevaron a cabo pruebas de navegación con base en un solo sensor, sin embargo las trayectorias resultaron ser bastante oscilatorias alrededor del *set-point*. Con el fin de mejorar el desempeño del controlador se agregó un sensor adicional lográndose mejor desempeño en la navegación. Este esquema de redundancia en la medida, permitió también reducir incertidumbres en las medidas debido a reflexiones especulares. Mientras que uno de los sensores puede estar suministrando una lectura errónea, el otro puede proporcionar una lectura correcta, con lo que se logra mantener el control de navegación. Para reducir aún más el impacto negativo provocado por el fenómeno de especularidad, asociado a todo sensor ultrasónico, se incorporó en el módulo de navegación un control de saturación por software. Para medidas iguales a cero o superiores a 100 cm, se ajusta el valor de la variable que almacena la lectura del sensor en 50 cm. En la figura 4.20 se aprecian curvas experimentales de navegación, las pruebas se llevaron a cabo en un aula de

clase sobre un tramo de pared de 560 cm. La pared se localiza al costado derecho del robot, teniendo como referencia la dirección de su desplazamiento lineal.

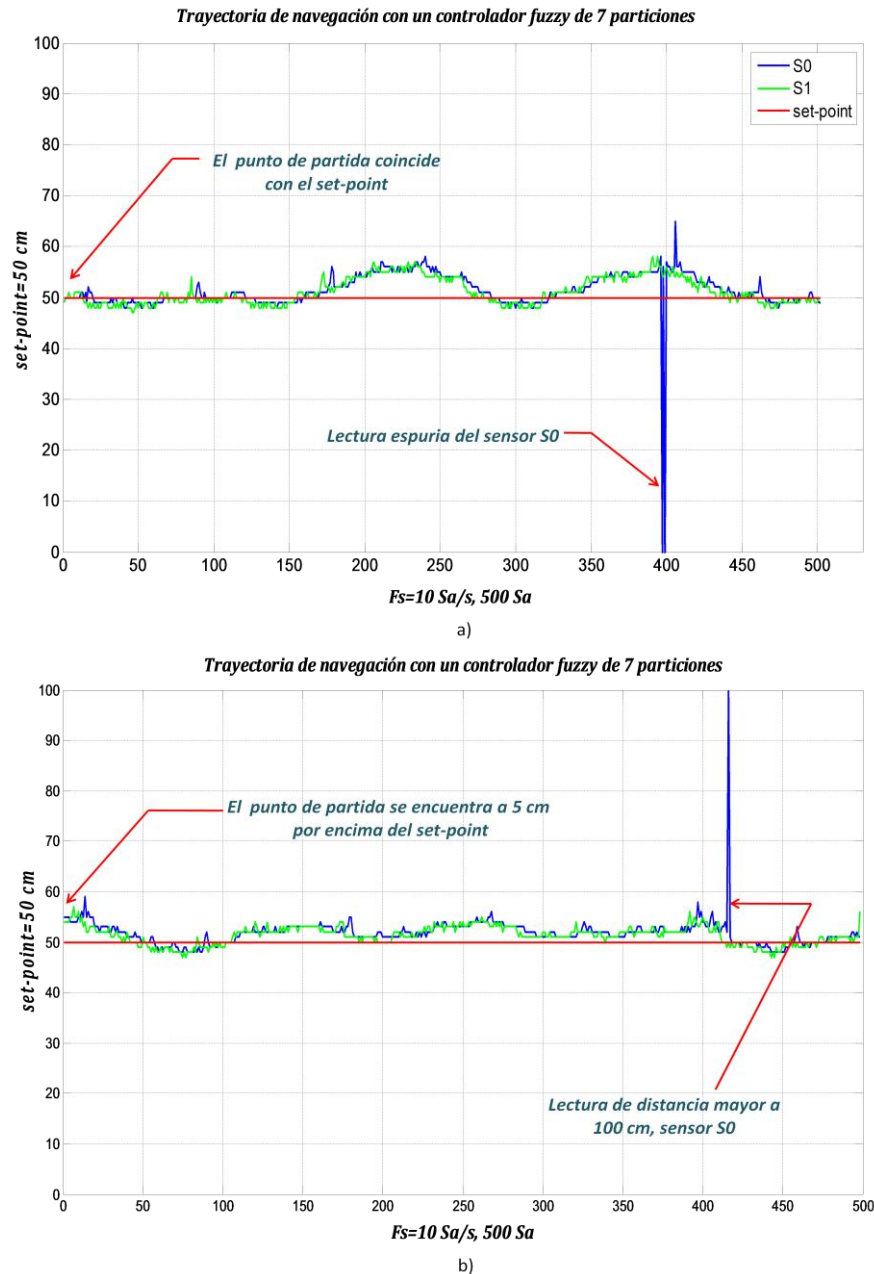


Figura 4.20. Pruebas de navegación en el robot Walling, a) recorrido de 560 cm con lectura espuria de cero centímetros del sensor *S0* después de recorrer 400 cm, b) recorrido de 560 cm con una falsa lectura superior a 100 cm después de 420 cm de recorrido.

Las lecturas se tomaron a una tasa de muestreo de 10 *Sa/s* ó 10 *Hz*. En total fueron adquiridas 500 muestras por recorrido, con un tiempo de navegación dado por:

$$t = \frac{500 Sa}{10 Sa/s} = 50 s \quad (15)$$

Las pruebas se llevaron a cabo a una velocidad lineal promedio de la plataforma de 11.2 cm/s. La distancia lineal recorrida por *Walling* durante las pruebas de navegación se calcula así:

$$d_{nav} = v \times t = 50 \text{ s} \times 11.20 \text{ cm/s} = 560 \text{ cm} \quad (16)$$

Haciendo nuevamente referencia a la figura 4.20, los resultados de la prueba en a) se obtuvieron ubicando el robot a una distancia inicial igual a la del *set-point*, es decir a 50 cm de la pared. Una trayectoria de navegación como estas es satisfactoria, permitiendo la construcción de mapas ajustados a las características de entornos interiores de trabajo. Se aprecia también una lectura espuria del sensor *S0*, éste entrega una falsa medida de distancia de a la pared de cero centímetros. A pesar de lo cual el robot completa el recorrido sin ningún problema. En este caso el control de saturación y una lectura correcta de distancia suministrada por el sensor *S1* permiten le permiten al robot realizar una buena trayectoria de navegación. En b), el robot se localiza, al inicio del recorrido, a una distancia de 55 cm, es decir cinco centímetros por encima del *set-point*. En este caso el controlador de navegación *fuzzy* corrige y mantiene la distancia a la pared de acuerdo a los criterios de diseño establecidos (ver apartado 4.3.2.1). Se observa también una falsa lectura de distancia debida a reflexiones especulares de la señal emitida. Esta vez el sensor *S0* entrega un dato de distancia mayor a 100 cm. De nuevo el controlador mantiene al robot en una trayectoria de navegación segura y acorde a las especificaciones establecidas.

En la figura 4.21 se presenta una prueba de navegación libre de reflexiones especulares en ambos sensores.

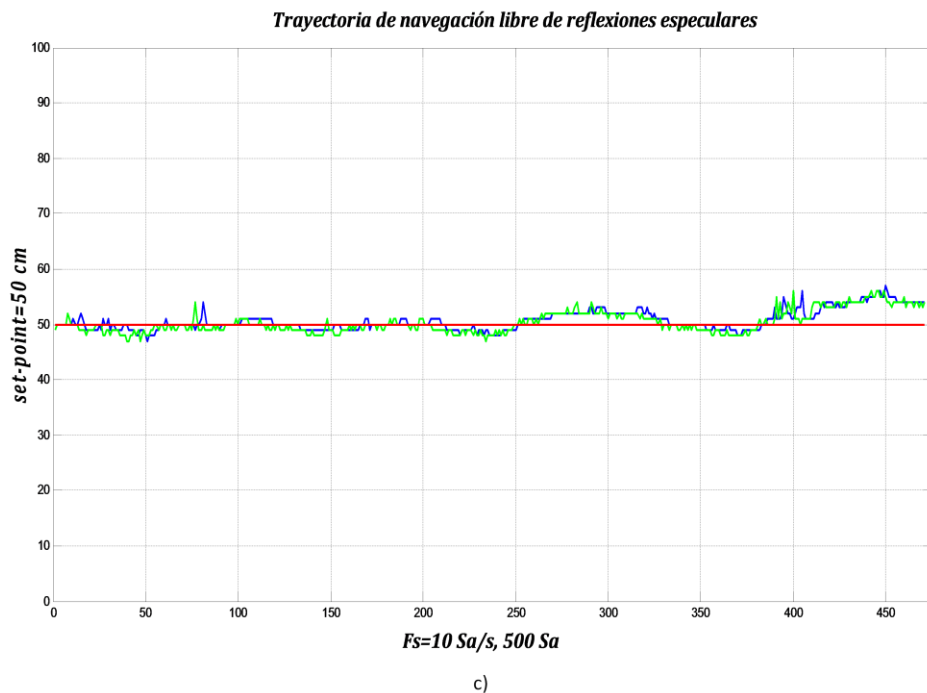
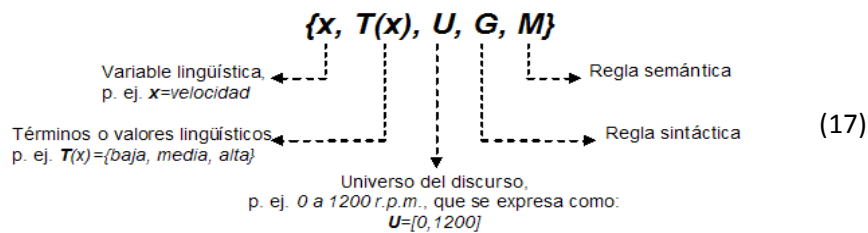


Figura 4.21. Navegación libre de reflexiones especulares en el robot *Walling*.

4.3.2.1 Proceso de Caracterización y Diseño del Controlador de Navegación Fuzzy

El diseño del controlador de navegación mediante lógica difusa inicia identificando las variables lingüísticas de entrada y salida requeridas para el control del proceso. Se consideraron dos variables de entrada, correspondientes al error o diferencia entre un set-point de 50 cm y las medidas de distancia suministrada por los sensores $S0$ y $S1$. Se consideró como variable de salida la señal PWM aplicada al efector derecho del robot. Las variables fueron caracterizadas mediante una tupla de la forma:



En donde:

x : Nombre de la variable lingüística. Por ejemplo: temperatura, velocidad, posición angular.

$T(x)$: Conjunto de términos o etiquetas que permiten la asignación de valores a x . Los valores corresponden a conjuntos difusos definidos en el universo del discurso U .

U : Universo del discurso.

G : Regla sintáctica o gramática para la generación de los términos en $T(x)$.

M : Regla semántica que permite asociar cada elemento de $T(x)$ con un conjunto difuso en U .

La caracterización detallada de las variables lingüísticas involucradas en el control de navegación se presenta en la tabla 4.5 y en la figura 4.22.

Tabla 4.5. Caracterización de variables lingüísticas para el controlador de navegación fuzzy del robot Walling.

Variables Lingüísticas de Entrada	
<ul style="list-style-type: none"> ▪ $e-dilade0$: error de distancia lateral pared derecha sensor $S0$. Distancias tomadas en centímetros. ▪ Conjunto de términos: <ul style="list-style-type: none"> - $E-ZERO$: Error igual a cero ($S0$). Robot a una distancia muy cercana al <i>set-point</i> - $E-PPEQ0$: Error positivo pequeño ($S0$) - $E-PMEO$: Error positivo mediano ($S0$) - $E-PGR0$: Error positivo grande ($S0$). Robot muy cerca de la pared derecha - $E-NPEQ0$: Error negativo pequeño ($S0$) - $E-NME0$: Error negativo mediano ($S0$) - $E-NGR0$: Error negativo grande ($S0$). Robot muy alejado de la pared derecha ▪ Universe del discurso: $U=[-6,6]$, en cm 	<ul style="list-style-type: none"> ▪ $e-dilade1$: error distancia lateral pared derecha sensor $S1$. Distancias tomadas en centímetros. ▪ Conjunto de términos: <ul style="list-style-type: none"> - $E-ZER1$: Error igual a cero ($S1$). Robot a una distancia muy cercana al <i>set-point</i> - $E-PPEQ1$: Error positivo pequeño ($S1$) - $E-PMEO$: Error positivo mediano ($S1$) - $E-PGR1$: Error positivo grande ($S1$). Robot muy cerca de la pared derecha - $E-NPEQ1$: Error negativo pequeño ($S1$) - $E-NME1$: Error negativo mediano ($S1$) - $E-NGR1$: Error negativo grande ($S1$). Robot muy alejado de la pared derecha ▪ Universe del discurso: $V=[-6,6]$, en cm

<ul style="list-style-type: none"> Reglas sintácticas: se utiliza la expresión "Error de distancia lateral" y los modificadores "pequeño", "mediano" y "grande". Reglas semánticas: de acuerdo con la gráfica de la figura 4-22. 	<ul style="list-style-type: none"> Reglas sintácticas: se utiliza la expresión "Error de distancia lateral" y los modificadores "pequeño", "mediano" y "grande". Reglas semánticas: de acuerdo con la gráfica de la figura 4.22.
Variable Lingüísticas de Salida	
<ul style="list-style-type: none"> PWM: ciclo de dureza que se aplica al puente H del motor derecho. Conjunto de términos: <ul style="list-style-type: none"> PWM-z: Ciclo de dureza PWM correcto (el robot no se aleja ni se acerca a la pared) PWM-fp: Ciclo de dureza PWM disminuye muy poco (el robot se acerca muy lentamente a la pared) PWM-fm: Ciclo de dureza PWM disminuye medianamente (el robot se acerca a la pared) PWM-fg: Ciclo de dureza PWM disminuye alto (el robot se acerca rápidamente a la pared) PWM-ap: Ciclo de dureza PWM aumenta muy poco (el robot se aleja muy lentamente de la pared) PWM-am: Ciclo de dureza PWM aumenta medianamente (el robot se aleja de la pared) PWM-ag: Ciclo de dureza PWM aumenta alto (el robot se aleja rápidamente de la pared) Universo del discurso: Z=[20,80] Reglas sintácticas: se utiliza la expresión "Ciclo de dureza PWM" y los modificadores "correcto", "disminuye/aumenta muy poco", "disminuye/aumenta medianamente", "disminuye/aumenta alto". <p>Reglas sintácticas: de acuerdo con la figura 4-22.</p>	

En la tabla 4.6 se aprecia la matriz de reglas con las que se establece la interacción *fuzzy* entre las entradas para generar la salida. Estas interacciones se concretan mediante un conjunto de reglas *IF-THEN*. En la figura 4.23 se aprecian las particiones en conjuntos difusos de las variables lingüísticas *e-dilade0*, *e-dilade1* y *PWM*, junto con la superficie de control en la que se señalan valores específicos adoptados por las variables de entrada y el valor que asigna el controlador a la variable de salida. Por ejemplo, para cada variable lingüística en la figura 4.23 se ha resaltado un conjunto *fuzzy*, la selección señalada da origen a la regla:

IF *e-dilade0* IS *E-NME0* AND *e-dilade1* IS *E-PPEQ1* THEN *PWM* IS *PWM-ag*

Tabla 4.6. Tabla FAM simplificada para el controlador de navegación *fuzzy* del robot *Walling*.

		e-dilade0						
		E-ZERO	E-PPEQ0	E-PME0	E-PGR0	E-NPEQ0	E-NME0	E-NGR0
e-dilade1	E-ZER1	PWM-z	PWM-fp	PWM-fm	PWM-fg	PWM-ap	PWM-am	PWM-ag
	E-PPEQ1	PWM-ap	PWM-ap	PWM-fp	PWM-fp	PWM-ap	PWM-ag	
	E-PME1	PWM-am	PWM-am	PWM-am	PWM-z	PWM-ag		
	E-PGR1	PWM-ag	PWM-ag	PWM-ag	PWM-ag			
	E-NPEQ1	PWM-fp	PWM-fp	PWM-fm		PWM-fp	PWM-fm	PWM-fm
	E-NME1	PWM-fm	PWM-fm			PWM-fm	PWM-fp	PWM-fm
	E-NGR1	PWM-fg				PWM-fg	PWM-fm	PWM-fg

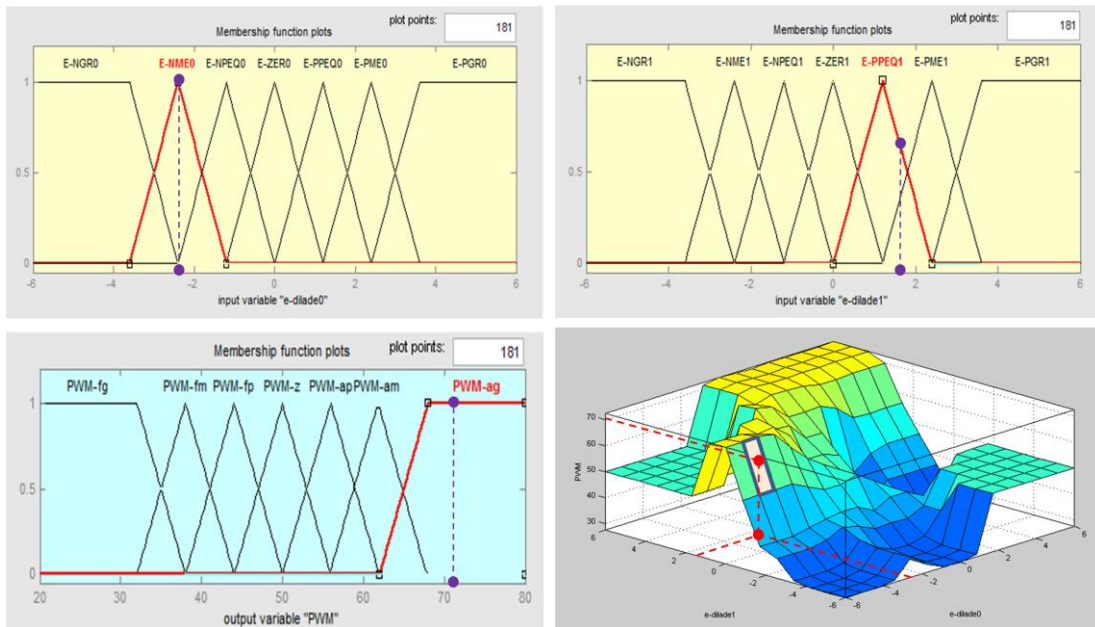


Figura 4.22. Partición de variables lingüísticas en conjuntos difusos y superficie de control del sistema de navegación *fuzzy* del Robot *Walling*.

4.4 Software de Mapeo para Entornos Interiores: Módulo *Vespucci*

Las funciones del módulo de mapeo *Vespucci* son:

- Establecer comunicaciones bidireccionales con el robot para la transferencia de comandos y datos.
- Almacenar y procesar los datos transferidos por el robot.
- Elaborar un modelo gráfico del entorno.

El software fue desarrollado en MATLAB empleando el entorno de programación visual GUIDE. Este permite desarrollar aplicaciones en las que el usuario requiere configurar parámetros operativos del software y/o ingresar datos iniciales.

Una vez se ejecuta el módulo *Vespucci*, se presenta al usuario una interfaz gráfica con una ventana de presentación inicial. Al activar el botón de comando “CONTINUAR” una ventana emergente brinda la posibilidad de seleccionar entre dos modos de operación: uno de adquisición de datos y otro de mapeo (figura 4.24).



Figura 4.23. Presentación del entorno visual.

4.4.1 Modo de Adquisición de Datos

En esta opción (figura 4.25) se pueden adquirir los datos telemétricos enviados por el robot mientras explora su entorno. Los datos se envían en el siguiente orden: 1) distancia lateral derecha sensor S0, distancia lateral derecha sensor S1, 3) distancia frontal sensor S2, 4) odometría rueda derecha y 5) odometría rueda izquierda.

La ventana de adquisición cuenta con cinco botones de comando. El botón de START transfiere la orden de arranque al robot e inicia la recepción de datos, el botón de STOP detiene el robot y la recepción de datos, el botón de GUARDAR permite almacenar en un archivo con extensión *.mat*, los datos capturados y, finalmente el botón CLEAR limpia la base de datos y la caja de despliegue de datos. Una caja de edición de texto con el rótulo **COM#**, permite seleccionar el puerto serial de comunicaciones a través del cual se transfiere información entre el módulo de mapeo *Vespucci* y el agente robótico *Walling*.

Las bases de datos almacenadas, se pueden usar posteriormente en la elaboración de mapas del entorno, al cargar el archivo correspondiente en el modo de operación “*Mapeo*”.

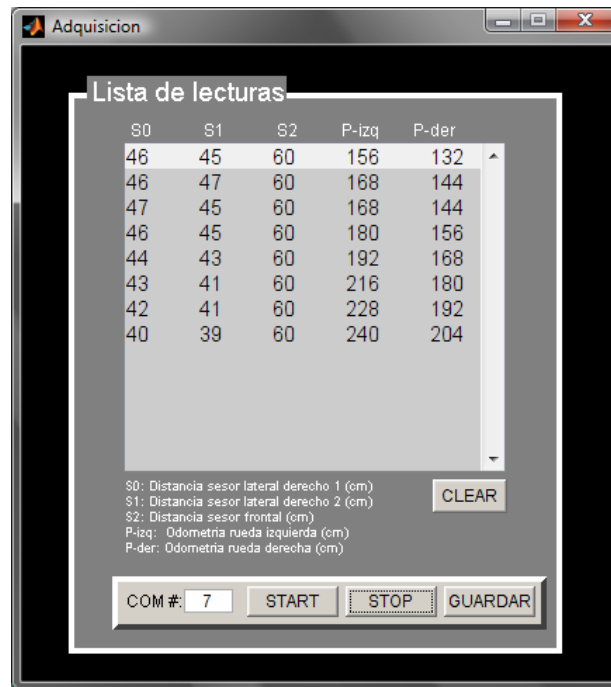


Figura 4.24. Interfaz gráfica para el modo de operación “*Adquisición*”, del módulo de mapeo *Vespucci*.

4.4.2 Modo Mapeo

En este modo de operación se debe ingresar inicialmente el tamaño de celda con el cual se discretizará la representación computacional del entorno, estableciéndose un área de trabajo de 300cm x 300cm, que refleja tamaños habituales en entornos interiores de trabajo como espacios de oficinas.

Al igual que en el modo *Adquisición*, se presentan al usuario los botones de START y STOP con los que se arranca y detiene el robot y se inician y detienen, respectivamente, las transferencias de datos entre el módulo de mapeo y la plataforma robótica. Se incorpora una interfaz gráfica en la que se va elaborando la representación del entorno a medida que el robot explora, captura y transmite información propioceptiva y exteroceptiva. En este modo de operación se pueden elaborar mapas *on-line* o en tiempo real.

También se pueden elaborar mapas *off-line*, al cargar bases de datos en Excel o archivos con extensión *.mat*. En este caso se deberá tener en cuenta la manera cómo están distribuidas las columnas al momento de cargar el archivo para evitar inconsistencias en la elaboración del mapa. Las columnas se disponen de manera secuencial como sigue:

- Distancia lateral derecha: dos medidas de distancia del lado derecho del robot que son suministradas *S0* y *S1*.
- Distancia frontal: una medida de distancia a objetos localizados al frente del robot, la cual es proporcionada por el sensor *S2*.

- Dos distancias odométricas: medidas de distancias correspondientes al desplazamiento lineal de la rueda derecha e izquierda. Esta información la proporcionan los encoders del robot.

La interfaz cuenta también con un modo de prueba o *punto por punto* en donde el usuario puede ingresar las coordenadas de una celda particular y su nivel de ocupación codificado en colores. En la figura 4.26 se observa la ventana que se le ofrece al usuario en el modo de operación *Mapeo*.

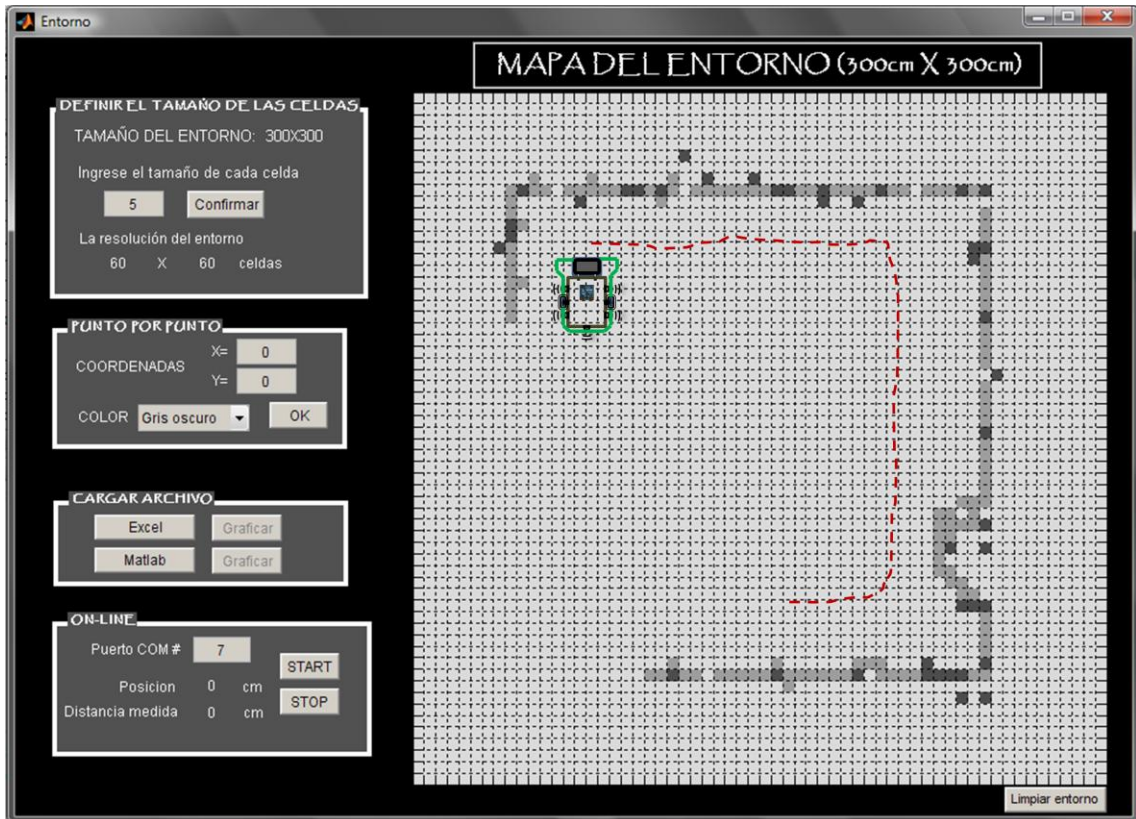


Figura 4.25. Interfaz gráfica en el modo mapeo.

4.5 Conclusiones del Capítulo

En este capítulo se describió de manera pormenorizada la arquitectura robótica propuesta para la construcción autónoma de representaciones computacionales del entorno. Se llevó a cabo la descripción de cada uno de los módulos de hardware y software que hacen parte del sistema, presentando resultados de pruebas experimentales llevadas a cabo para la caracterización de sistemas de percepción propio y exteroceptivos del agente robotico de exploración

Se describe el proceso de diseño y caracterización del controlador *fuzzy* para el seguimiento de contornos. Este controlador hace parte del agente de navegación *Magalhães* e implementa el comportamiento reactivo "*seguir paredes*". El controlador fue sometido a

pruebas de navegación, permitiéndole al robot realizar el seguimiento de paredes localizadas a su lado derecho sobre trayectos de 560 cm. El controlador permitió alcanzar una navegación segura y estable para entornos interiores de trabajo.

Capítulo 5. Pruebas y Resultados

El sistema *Map-Bot* fue sometido a varias pruebas de navegación en las cuales el agente explorador *Walling* recorre su entorno, recaba información y la transmite al módulo de mapeo *Vespucci* que elabora una representación computacional del entorno.

Las pruebas se llevaron a cabo en entornos interiores estructurados sin la presencia de objetos o personas en movimiento (entornos estáticos). Se realizaron trayectos de navegación menores a 300 cm, acorde con las dimensiones establecidas para el entorno virtual sobre el que se elaboran los mapas.

Se consideraron tres escenarios:

- Entorno de oficinas con dos paredes de ladrillo separadas por una puerta de madera.
- Pasillo con dos paredes de ladrillo no colineales y separadas por una columna de concreto.
- Pasillo con dos paredes de ladrillo con puerta abierta que conduce a un baño común con paredes cubiertas en cerámica.

Se elaboraron tres mapas de cada entorno con resoluciones de celda de 3, 5 y 10 centímetros. Así mismo, para cada resolución de celda se elaboró un mapa en tiempo real (*mapeo on-line*) en el que el grado de ocupación que se asigna a cada celda se hace a medida que se reciben datos del explorador, y un mapa *off-line*, en el que primero se adquieren y almacenan los datos y luego se elabora el mapa.

Las representaciones obtenidas se ajustan satisfactoriamente a las características geométricas de los entornos explorados.

5.1 Diseño de Pruebas de Validación

Para la realización de las pruebas de navegación y mapeo, se seleccionaron espacios con características típicas de entornos estructurados de trabajo, con dimensiones ajustadas al entorno virtual de trabajo definido por el módulo de mapeo *Vespucci* (3m x 3m). Los espacios de prueba deberán incorporar uno o varios de los siguientes elementos:

- Muros de obra con o sin acabado enlucido o empapelado.
- Columnas de hormigón de sección transversal cualquiera.

- Puertas de aglomerado con acabado en Railite o Fórmica.

Las medidas se llevan a cabo en condiciones normales de presión y temperatura, cuyos valores de referencia son 1 atm y 20° C respectivamente. Para la exploración del entorno se programa el robot *Walling* con los siguientes parámetros:

- Velocidad lineal promedio de 12 cm/s.
- Toma de dos distancias laterales para la navegación por seguimiento de paredes, una distancia frontal para la evasión de obstáculos presentes en el entorno y distancias recorridas por la rueda derecha e izquierda. La toma de datos se hace a razón de diez muestras por segundo.

Se debe establecer también un parámetro cuantitativo que permite determinar cuantitativamente la confiabilidad del mapa elaborado.

5.2 Confiabilidad de la Representación Computacional del Entorno

Puesto que todo sensor incorpora algún grado de incertidumbre en la medida, se hace necesario establecer un parámetro que permita cuantificar el grado de aproximación de la representación computacional obtenida con respecto a las dimensiones reales del entorno físico. Para tal fin se define el parámetro: *Nivel de Aproximación del Mapa al Entorno Real*, que se denota como *NAMER* y que viene dado por:

$$NAMER = \frac{D_{map}}{D_{real}} \quad (18)$$

Donde:

D_{map} : Distancia calculada sobre el mapa.

D_{real} : Distancia real medida sobre el entorno.

Haciendo referencia a la ecuación (18), mientras más se acerque el parámetro *NAMER* a la unidad, mayor será el grado de correspondencia entre la distancia real y la distancia virtual (calculada sobre el mapa). Valores inferiores a la unidad indican una distancia menor a la distancia real, y valores superiores, una distancia mayor. En este trabajo el *NAMER* se ha calculado para los mapas elaborados con resolución de celda de cinco centímetros y sobre tres distancias representativas del entorno, las cuales, se señalan sobre la respectiva figura. La distancia D_{map} se calcula con base en el número de celdas libres entre la trayectoria seguida por el robot y un obstáculo o entre dos obstáculos. El grado de confiabilidad final corresponde al promedio de los tres valores de *NAMER* calculados.

5.3 Entorno 1: área de oficinas con dos paredes y una puerta

En esta prueba se consideró un entorno interior de oficinas con dos paredes de ladrillo separadas por una puerta en madera. En el experimento el agente explorador realizó un recorrido de 237 centímetros, y adquirió y transmitió un total de 1220 muestras de distancia (ultrasonido y odometría) a una frecuencia de muestreo de 10 Hz.

En la figura 5.1 se aprecia: a) el entorno físico en el que se realizó la prueba, y b), c) y d) mapas del entorno con resoluciones de celda de 5, 3 y 10 centímetros respectivamente.

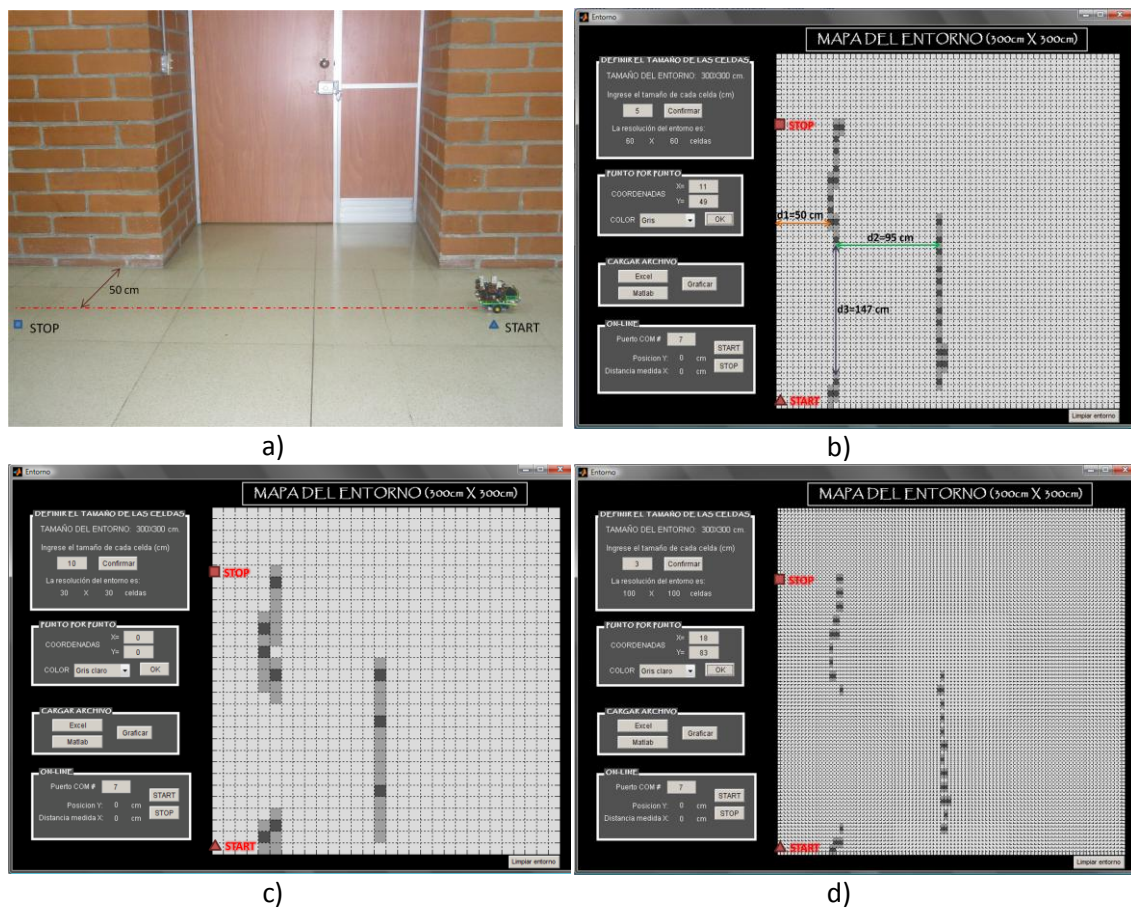


Figura 5.1. Mapeo de un entorno de oficinas, a) área de prueba; b), c) y d) mapas obtenidos con resoluciones de celda de 5, 10 y 3 centímetros respectivamente. Se señalan las distancias reales d_1 , d_2 y d_3 en la figura b).

En la tabla 5.1 se aprecia el cálculo del *Nivel de Aproximación del Mapa al Entorno Real (NAMER)* para el mapa de la figura 5.1 b). Se consideraron las siguientes distancias:

- d_1 : distancia de la trayectoria seguida por el robot respecto al muro.
- d_2 : profundidad a la que se encuentra la puerta respecto al muro.

- d3: apertura de la hendidura de la puerta.

Tabla 5.1. Cálculo del *NAMER* para el mapa de la figura 5.1 b).

	d1	d2	d3
Distancia Medida en el Mapa (D_{map}), en cm	45	85	105
Distancia Real (D_{real}), en cm	50	95	104
<i>NAMER</i> (parámetro adimensional)	0.9	0.895	1.01
<i>Promedio NAMER: 0.935</i>			

d1=9 celdas x 5 cm/celda=45 cm, d2=17 celdas x 5 cm/celda= 85 cm, d3=21 celdas x 5 cm/celda=105 cm.

5.4 Entorno 2: pasillo con dos paredes y una columna

Este experimento se llevó a cabo en un entorno de pasillo con dos paredes no colineales en ladrillo y entre las cuales se encuentra una columna de hormigón con sección transversal rectangular.

El recorrido llevado a cabo por el explorador fue de 250 cm, adquiriéndose un total de 1060 muestras a una tasa de muestreo de 10 Sa/s.

En la figura 5.2 se aprecia el entorno físico en el que se realizó la prueba, y los mapas elaborados con diferentes resoluciones de celda.



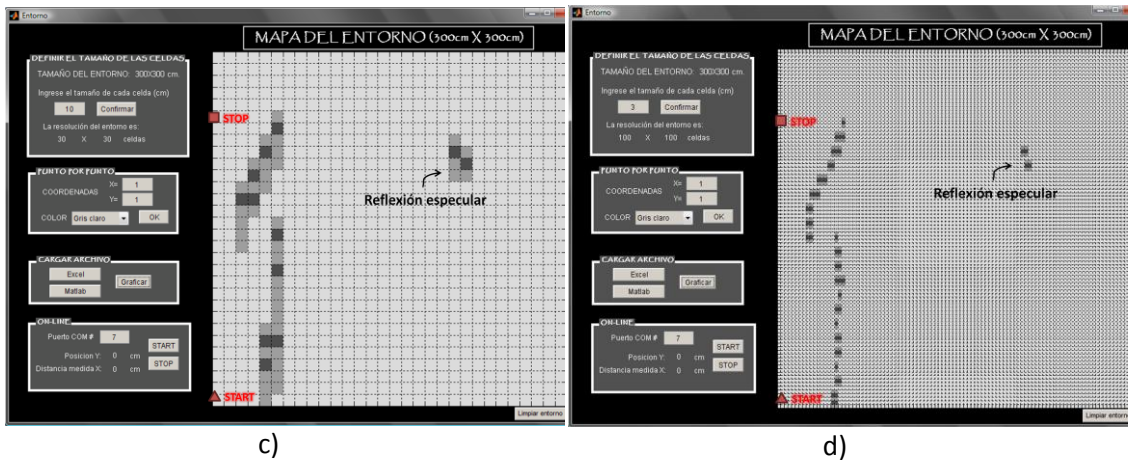


Figura 5.2. Mapeo de un tramo de pasillo con paredes no colineales y columna, a) área de prueba; b), c) y d) mapas obtenidos con resoluciones de celda de 5, 10 y 3 centímetros respectivamente. Se señalan las distancias reales d_1 , d_2 y d_3 en la figura b).

En la tabla 5.2 se puede apreciar el cálculo del *Nivel de Aproximación del Mapa al Entorno Real (NAMER)* para el mapa de la figura 5.2 b). Se consideraron las siguientes distancias:

- d_1 : profundidad del muro con respecto a la cara visible de la columna que da al pasillo.
- d_2 : distancia de la trayectoria seguida por el robot respecto al muro.
- d_3 : ancho de la columna de sección transversal rectangular que da al pasillo.

Tabla 5.2. Cálculo del *NAMER* para el mapa de la figura 5.2 b).

	d_1	d_2	d_3
Distancia Medida en el Mapa (D_{map}), en cm	20	45	50
Distancia Real (D_{real}), en cm	25	50	40
<i>NAMER</i> (parámetro adimensional)	0.8	0.9	1.25
Promedio <i>NAMER</i> : 0.983			

$d_1=4$ celdas \times 5 cm/celda=20 cm, $d_2=9$ celdas \times 5 cm/celda= 45 cm, $d_3=10$ celdas \times 5 cm/celda=50 cm.

5.5 Entorno 3: pasillo con dos paredes y puerta cerrada

Esta prueba se realizó en un entorno de pasillo con dos paredes de ladrillo y una puerta cerrada que conduce a un cuarto de aseo.

Para este recorrido fueron adquiridas 1165 muestras a una frecuencia de muestreo de 10 Hz en un trayecto de 290 centímetros.

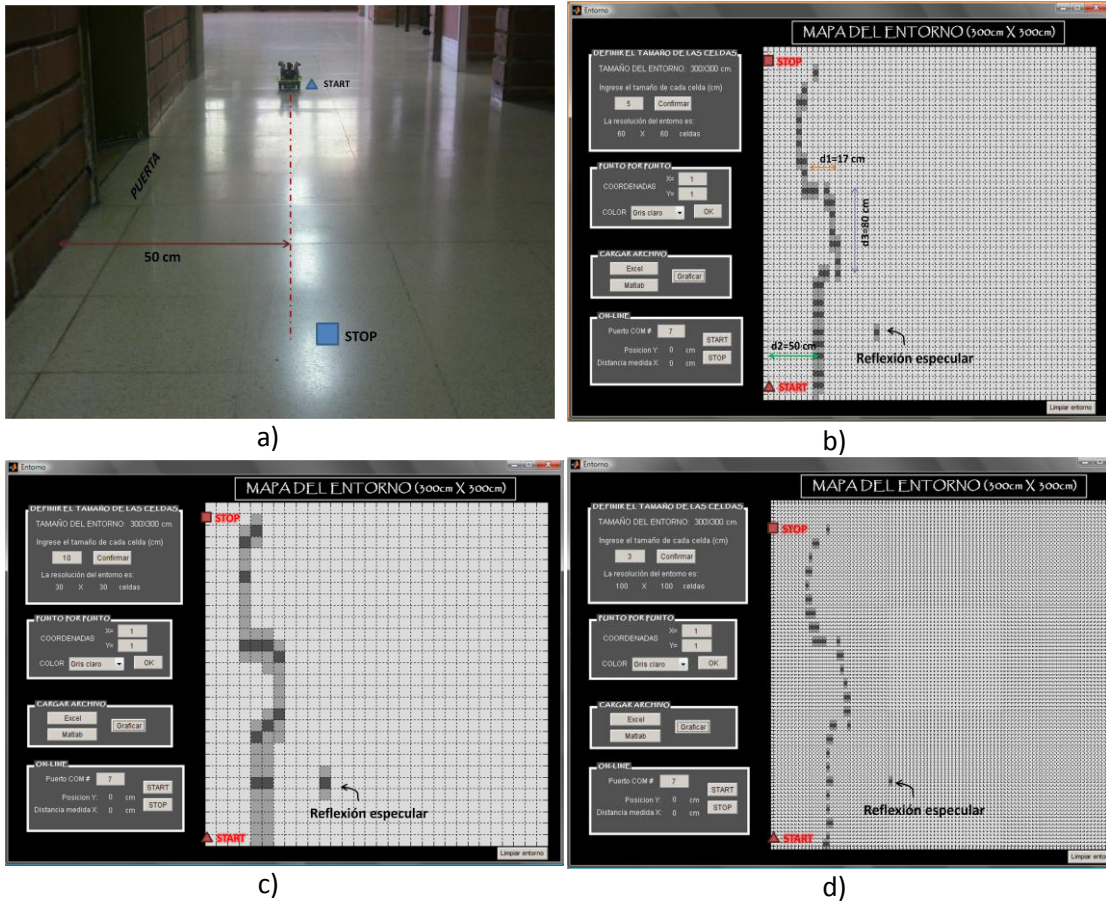


Figura 5.3. Mapeo de un tramo de pasillo con puerta cerrada entre dos paredes, a) área de prueba; b), c) y d) mapas obtenidos con resoluciones de celda de 5, 10 y 3 centímetros respectivamente. Se señalan las distancias reales d_1 , d_2 y d_3 en la figura b).

En la tabla 5.3 se presenta el cálculo del *Nivel de Aproximación del Mapa al Entorno Real (NAMER)* para el mapa de la figura 5.3 b). Se consideraron las siguientes distancias:

- d_1 : profundidad a la que se encuentra la puerta respecto al muro.
- d_2 : distancia de la trayectoria seguida por el robot respecto al muro.
- d_3 : apertura de la hendidura de la puerta.

Tabla 5.3. Cálculo del *NAMER* para el mapa de la figura 5.3 b).

	d1	d2	d3
Distancia Medida en el Mapa (D_{map}), en cm	20	45	70
Distancia Real (D_{real}), en cm	17	50	80
<i>NAMER</i> (parámetro adimensional)	1.176	0.9	0.875
Promedio <i>NAMER</i> : 0.984			

$d_1=4$ celdas \times 5 cm/celda=20 cm, $d_2=9$ celdas \times 5 cm/celda= 45 cm, $d_3=14$ celdas \times 5 cm/celda=70 cm.

5.6 Conclusiones del Capítulo

Las pruebas realizadas sobre entornos reales de oficinas permitieron obtener representaciones computacionales que se ajustan en buena medida a las características del entorno, reflejando sus propiedades geométricas más relevantes. Se definió y aplicó el parámetro *NAMER*, que permite determinar cuantitativamente el grado de aproximación entre la representación computacional y el entorno físico explorado. Se calculó el parámetro para los mapas con resolución de celda de 5 cm, que corresponde al valor promedio de los mapas elaborados durante las pruebas.

Para cada uno de los mapas con resolución de celda de 5 cm se calculó el *NAMER* para tres distancias representativas del entorno. Finalmente se tomó como valor característico el promedio de los tres cálculos. Se obtuvieron los siguientes valores promedio: *NAMER entorno1=0.935*, *NAMER entorno2=0.983*, *NAMER entorno3=0.984*. En el peor de los casos se obtuvo un *NAMER* de 0.935, valor que resulta ser bastante cercano a la unidad, esto indica que la representación computacional se aproxima a las características reales del entorno en un 93.5%, comprobándose de esta forma la consistencia, confiabilidad y robustez del desarrollo propuesto.

Capítulo 6. Evaluación

En este capítulo se da respuesta a las preguntas de investigación formuladas y se establece la contribución de esta tesis campo del mapeo autónomo de entornos de trabajo.

6.1 Respuestas a las Preguntas de Investigación

Enmarcados en el problema identificado, relativo a la construcción de representaciones computacionales de entornos estructurados de trabajo, se plantearon las siguientes preguntas de investigación:

- *¿Cómo configurar un ambiente multi-agente robótico para la navegación y mapeo de entornos estructurados de trabajo?*

R/. Una buena estrategia consiste en trabajar modularmente tanto a nivel macro como a nivel micro. Se configuró un ambiente multi-agente robótico constituido por:

- Un agente robótico (*Walling*) para la exploración del entorno, adquisición de información y transmisión de datos a través de una interfaz de comunicaciones inalámbricas.
- Un agente de navegación (*Magalhães*) que exhibe dos tipos de comportamientos reactivos emergentes y soporta la navegación segura del agente explorador en el entorno de trabajo.
- Un módulo de software para la representación bidimensional de entornos estructurados mediante la técnica de celdas de ocupación. Este módulo se codificó en Matlab y ofrece al usuario del sistema una interfaz gráfica simple y fácil de usar. Este módulo hace parte del sistema pero no corresponde a un agente.

- *¿Qué arquitectura de Sistema Multi-Agente será adecuada para solucionar el problema de la elaboración de mapas de entornos estructurados de trabajo?*

R/. El control de alto nivel se llevó a cabo con base en una arquitectura reactiva cuya principal característica es proveer una relación estímulo-respuesta casi inmediata. Esta característica se traduce en un bajo costo computacional, poco demandante en recursos para el sistema de cómputo de la plataforma robótica. El control de bajo nivel se llevó a cabo mediante un controlador *fuzzy* en lazo cerrado, del tipo *Mamdani*. El esquema de control difuso permitió tratar de manera simple y eficiente con las incertidumbres propias de los sistemas ultrasónicos de percepción. Para el agente explorador *Walling* se estableció una arquitectura modular de cuatro niveles:

- Nivel 1: efectores, drivers de potencia y sensórica propioceptiva (odometría)
- Nivel 2: control microcontrolado (μ C MCF51QE128 de 32 bits)
- Nivel 3: sensórica exteroceptiva (sonares SFR02)
- Nivel 4: comunicación inalámbrica (módulo XBee)

Para el agente explorador se estableció una arquitectura basada en dos comportamientos reactivos emergentes:

- Comportamiento “seguir paredes”.
- Comportamiento “evitar obstáculos”.

La arquitectura planteada le permitió al sistema *Map-Bot* explorar y representar el entorno de trabajo. Los mapas obtenidos reflejaron las características geométricas del entorno.

➤ *¿Cuales deberán ser las características de hardware y software del sistema de comunicaciones de tal forma que se garantice un intercambio eficiente de información entre agentes?*

R/. La tarea principal del agente *Walling* consiste en realizar la exploración del entorno de trabajo, adquirir información sensorial y transmitirla al módulo de mapeo. Puesto que el agente explorador realiza desplazamientos permanentes en su entorno de trabajo, entonces fue necesario agregar un sistema de comunicaciones con el cual se pudiera garantizar la movilidad del robot. Para tal fin se incorporaron al sistema dos módulos de comunicaciones con las siguientes características:

- Bajo consumo de energía.
- Un área de cobertura suficiente para garantizar la transmisión y recepción de datos en entornos urbanos interiores.
- Facilidad de conexión con sistemas microcontrolados a través de una interfaz tipo UART.
- Fácil configuración del sistema mediante rutinas de software simples.
- Baja potencia de transmisión
- Alta sensibilidad

Se emplearon los módulos de comunicaciones *XBee*, que operan a una frecuencia de 2.4 MHz en la banda ISM (*Industrial, Scientific and Medical Band*). Esta interfaz cumple con el estándar *IEEE 802.15.4* y con los requerimientos establecidos para la aplicación.

➤ *¿Con que características de percepción, locomoción y control deberá contar la plataforma robótica con el fin de garantizar una exploración segura del entorno?*

R/. A la plataforma robótica móvil se incorporaron dos tipos de sistemas sensoriales con las siguientes características:

- Un sistema propioceptivo de percepción (odometría) que provee datos de desplazamientos lineales de la rueda derecha e izquierda. Estos datos son empleados por el mapeador *Vespucci* para actualizar el nivel de ocupación de las celdas respectivas en el mapa del entorno.

- Un sistema de percepción exteroceptivo (sonares) que permite determinar la distancia a paredes y obstáculos. Estos datos soportan la navegación segura del robot en su entorno y permiten también actualizar el nivel de ocupación de las celdas con las que se discretiza el entorno.

Para la locomoción se emplea una caja motorreductora con dos motores DC y dos ruedas en configuración diferencial. La razón de reducción es de 203:1, obteniéndose una velocidad de 60 rpm en las ruedas. Se cuenta con una tercera rueda de apoyo tipo *caster ball* que brinda estabilidad a la plataforma. La potencia mecánica obtenida es suficiente para que el robot se pueda desplazar en entornos interiores.

El sistema de control basado en una arquitectura reactiva incorpora un controlador *fuzzy* tipo *Mamdani* que permitió obtener una navegación segura del entorno.

Las pruebas experimentales realizadas permitieron obtener representaciones computacionales que reflejaron las propiedades geométricas del entorno. En tal sentido las características de los subsistemas seleccionados fueron las correctas para soportar la aplicación de mapeo autónomo de entornos de trabajo con base en un sistema multi-agente robótico.

- *¿Cuáles estrategias de exploración resultarán ser las más eficientes para la elaboración de mapas en entornos estructurados de trabajo?*

R/. En robótica móvil se cuenta con una amplia gama de estrategias de navegación, un robot puede realizar desplazamientos aleatorios en su entorno (vagabundeo), seguir líneas o marcas en el piso (Line Following), seguir fuentes de luz, como por ejemplo luminarias en el techo, o desplazarse de un sitio a otro empleando el principio “*campos de atracción*” y “*repulsión*” de manera análoga a los campos eléctricos. Una estrategia de navegación fácil de implementar con base en sensores ultrasónicos y que a la vez resulta eficiente es el seguimiento de paredes o contornos (Wall Following). Las pruebas de navegación realizadas al agente explorador *Walling*, arrojaron resultados positivos cuando se empleó esta estrategia de exploración, pudiéndose obtener mapas que reflejaron las características geométricas de los entornos de prueba.

- *¿En qué medida resulta ser más eficiente y precisa la construcción de modelos de entornos estructurados con base en un sistema multiagente en comparación con la aproximación tradicional basada en un único agente robótico?*

R/. El costo computacional asociado al mapeo de entornos resulta ser bastante alto. En tal sentido, un agente robótico en el que se concentra toda la sensórica, capacidad de cómputo y la adquisición y procesamiento de datos, requerirá de sistemas microprocesados de características avanzadas que elevarían los costos del sistema, incluyendo también los costos en recursos energéticos. Sería también un sistema más propenso a fallos por la complejidad misma del sistema y la concentración de tareas y subsistemas.

La distribución de tareas entre un agente robótico explorador, un agente de navegación y un módulo de software, simplifica notoriamente el hardware de cada uno de los integrantes del sistema y facilita la ejecución de tareas, el agente explorador, por ejemplo, respondería

por las tareas de exploración, adquisición de información y transmisión. De esta forma se libera al explorador del procesamiento complejo de datos y la actualización de representaciones computacionales del entorno. La computación distribuida se traduce también en una reducción del tiempo requerido para la construcción de mapas. Un único sistema al que se asigna toda la responsabilidad de elaborar mapas, deberá reducir la velocidad de sus desplazamientos para poder ejecutar tareas relativas al mapeo, por lo tanto la obtención de mapas requerirá de tiempos mayores.

6.2 Contribuciones de esta Tesis

Uno de los principales aportes de este trabajo consistió en la construcción de un sistema multi-agente robótico totalmente operativo orientado a la elaboración de mapas de entornos de dimensiones reducidas, a partir de una conceptualización completa basada en la metodología MAD-Smart propuesta por Jiménez en [102].

Otro aporte fue la incorporación de técnicas de control Fuzzy en la implementación de comportamientos reactivos emergentes para el control de navegación del agente Walling, específicamente el algoritmo de seguimiento de paredes.

Se logró diseñar un software de mapeo en Matlab en el que se incorpora la incertidumbre asociada a los sistemas ultrasónicos de percepción con base al modelo bidimensional Gaussiano, adoptado en este trabajo. Tal incertidumbre se presenta de forma gráfica mediante la asignación de colores en la escala de grises a las celdas cubiertas en cada lectura del sonar.

Capítulo 7. Conclusiones y Trabajo Futuro

7.1 Conclusiones

Después de analizar los paradigmas dominantes en la construcción de representaciones computacionales del entorno, se optó por emplear la técnica de mapeo por celdas de ocupación. Si bien resulta ser más demandante en recursos computacionales que las representaciones topológicas, reproducen de manera explícita la estructura del entorno, lo que facilita poder determinar la localización del robot en el modelo a partir de su localización en el mundo físico.

Precisamente esta correspondencia geométrica del modelo con el entorno, hace de los mapas de celdas descripciones fáciles de construir y mantener. Una de sus desventajas es que son bastante sensibles a errores odométricos acumulativos y a fenómenos de reflexión especular cuando se emplean sensores de ultrasonido. No obstante, la estrategia de navegación por seguimiento de contornos con evitación de obstáculos frontales, permitió reducir la incertidumbre odométrica y el efecto negativo de la reflexión especular de la onda ultrasónica.

La selección de un controlador fuzzy para el seguimiento de contornos, en lugar de un esquema tradicional de control (p. ej. un PID discreto), se debió a que no se hace necesario contar con un conocimiento detallado de la dinámica del sistema representada en expresiones matemáticas. Las pruebas realizadas en entornos reales permitieron validar esta técnica de control al obtener trayectos seguros de navegación para el robot.

El sistema robótico se pudo materializar más fácilmente gracias al trabajo previo de conceptualización del sistema mediante la metodología de diseño MAD-Smart, lográndose reducir los tiempos de construcción de cada agente y de los módulos requeridos por la aplicación. El empleo de una herramienta metodológica como MAD-Smart permitió un mejor control del proceso de construcción del sistema reduciendo las probabilidades de fallo. El resultado final fue un sistema complejo para la construcción de modelos del entorno que resultó ser confiable y robusto.

El sistema fue puesto a prueba en diferentes tipos de entornos interiores como fueron pasillos, oficinas, laboratorios y aulas de clase. Los modelos obtenidos se ajustaron tanto a las características morfológicas de los entornos de prueba, como a las características métricas.

7.2 Trabajo Futuro

- Evaluar otros esquemas de control de navegación tradicionales y no tradicionales basados en técnicas de inteligencia artificial, pueden ser estos control PID, control *neurofuzzy* y control adaptativo. Un análisis comparativo permitiría acceder a mejores estrategias de control de navegación.
- La arquitectura del sistema permite experimentar diversas alternativas en la construcción de mapas por celdas de ocupación, por ejemplo la forma en que se asigna el grado de ocupación de las celdas. Una alternativa interesante sería la discretización del entorno con celdas cuadrangulares no uniformes. Un área del entorno con mucha presencia de obstáculos se podría discretizar con celdas de área reducida para obtener una mejor resolución de esa parte. Áreas con baja densidad de obstáculos podría discretizarse con celdas de mayor tamaño ya que se trata de espacios libres. Este tipo de discretización puede dar lugar a una reducción en los tiempos de mapeo de entornos, sin sacrificar la calidad del mapa.
- Para mejorar los mapas obtenidos se puede hacer una exploración redundante del entorno (con varias pasadas o recorridos sobre una misma área) lo que permitiría:
 - Identificar objetos en movimiento de forma tal que el sistema estaría en capacidad de mapear tanto como entornos estáticos como dinámicos.
 - Filtrado de datos espurios por reflexiones especulares en los objetos.
- Al sistema se pueden incorporar agentes exploradores adicionales que examinarían en simultaneidad otras zonas del entorno. Una configuración tal permite explorar entornos de dimensiones mayores sin incrementos en los tiempos de mapeo.

Referencias

- [1] Swain, R.; Devy M.; Jonquières S. (1998) Navegación de un Robot Móvil por Medio de Control Visual en Ambiente Estructurado. En: Computación y Sistemas Vol.1 No. 3. pp. 161-169
- [2] Labidi, S.; Lejoudad W. (1993) De l'Intelligence Artificielle Distribuée aux Systèmes Multi-Agents.
- [3] Mäkelä, H. (2001) Outdoor Navigation of Mobile Robots. University of Technology. Tesis Doctoral.
- [4] Souza, J. (2002) Cooperação entre Robôs Aéreos e Terrestres em Tarefas Baseadas em Visão. UFMG. Proceeding of SPG'2002.
- [5] Jiménez, J. (2006) Un Modelo de Planificación Instruccional usando Razonamiento Basado en Casos en Sistemas Multi-Agente para entornos integrados de Sistemas Tutoriales Inteligentes y Ambientes Colaborativos de Aprendizaje. Tesis de Doctorado. Universidad Nacional de Colombia.
- [6] Russell, S.; Norving, P. (2003) Intelligent Agents. En: Artificial Intelligence: A Modern Approach. Second Edition. Prentice Hall Series in Artificial Intelligence.
- [7] Schalkoff, R. J. (1990). Artificial Intelligence: An Engineering Approach. McGraw-Hill. New York.
- [8] Kurzweil, R. (1990). The Age of Intelligent Machines. MIT Press, Cambridge, Massachusetts.
- [9] Charniak, E. and McDermontt, D. (1985). Introduction to Artificial Intelligence. Addison-Wesley, Reading, Massachusetts.
- [10] Ferber, J. (1999) Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence. Addison-Wesley.
- [11] Wooldridge, M.; Jennings, N.; Kinny, D. (1999) A Methodology for Agent-Oriented Analysis and Design. Disponible en: <http://www.csc.liv.ac.uk/~mjw/pubs/agents99.pdf> Fecha de acceso: Abril de 2007.
- [12] Hewitt, C. (1977) Viewing Control Structures as Patterns of Passing Messages. En: Artificial Intelligence, Vol. 8 No 3, 323-364.
- [13] Minsky, M. (1986) The Society of Mind. New York: Simon and Schuster.
- [14] Castrillón, D. (2000) Un Modelo de Interfaz Inteligente basado en Agentes para Ambientes de Aprendizaje Colaborativo. Universidad Nacional de Colombia.
- [15] Pérez, F. (2000) Meta-Modelo de Coordinación de Tareas para Sistemas Multi-Agentes en Ambientes de Trabajo Cooperativo. Universidad Nacional de Colombia.
- [16] Marcenac, P.; Giroux, S. (1998) GEAMAS: A Generic Architecture for Agent-Oriented Simulations of Complex Processes. En: Applied Intelligence, Vol. 8, No. 3.
- [17] Sycara, K.; Pannu, A.; Williamson, M.; Zeng, D.; Decker, K. (1996) Distributed Intelligent Agents. IEEE Expert: Intelligent Systems and Their Applications, Vol. 11, No. 6.
- [18] Gómez, J. (2003) Metodologías para el Desarrollo de Sistemas Multi-Agente. En: Revista Iberoamericana de Inteligencia Artificial No. 18, pp 51-63.

- [19] Green, S.; Hurst, L.; Nangle, B.; Cunningham, P.; Somers, F.; Evans, R. (1997) Software Agents: A Review. En: Distributed Agent Technology: An Overview.
- [20] Gomes, E. (2004b) Intelligent Learning Objects: an Agent based Approach of Learning Objects. Universidade Federal do Rio Grande do Sul.
- [21] Wooldridge, M.; Jennings, N. (1995) Intelligent Agents: Theory and Practice. En: The Knowledge Engineering Review, Vol. 10 No. 2.
- [22] Chiariglione, L. (2001) Fipa: Foundation for Intelligent Physical Agents. En: FIPA Board of Directors, Resolutions of the Osaka.
- [23] Jennings, N.; Wooldridge, M. (1998) Applications of Intelligent Agents. En: Agent Technology. Foundations, Applications, and Markets. Springer-Verlang.
- [24] Nwana, H. (1996) Software Agents: An Overview. En: Knowledge Engineering Review, Vol. 11, No. 3.
- [25] Wooldridge, M. (1999) Intelligent Agents. En: A Modern Approach to Distributed Artificial Intelligence. Gerhard Weiss (Eds.) MIT Press.
- [26] Maes, P. (1996) Intelligent Software Easing the Burdens that Computers Put on People. MIT Media Laboratory. IEEE Expert Intelligent Systems, Vol. 11 No 6.
- [27] Nwana H.; Ndumu, D. (1998) A Brief Introduction to Software Agent Technology. En: Agent Technology: Foundations, Applications, and Markets. Springer-Verlang.
- [28] Franklin, S.; Graesser, A. (1996) Is it an Agent, or just Program?: A Taxonomy for Autonomous Agents. En: Proceedings 3rd International Workshop on Agent Theories, Architectures, and Languages Institute for Intelligent Systems.
- [29] Lalo (1999) LALO, Centre de Recherche d'Informatique de Montreal. CANADA URL: <http://www.crim.ca/sbc/english/lalo/introduc.html>, 1999.
- [30] Quintero, A.; Rueda, S.; Ucrós M. (1998) Agentes y Sistemas Multi-Agente: Integración de Conceptos Básicos. En: Informe Grupo de Investigación Hidra, Universidad de los Andes.
- [31] Jennings, N.; Faratin, A.; Lomuscio, S.; Sierra, C.; Wooldridge, M. (2001) Automated Negotiation: Prospects, Methods and Challenges. International Journal of Group Decision and Negotiation Vol. 10 No. 2.
- [32] Deshpande, U.; Gupta, A.; Basu, A. (2004) Coordinated Problem Solving Through Resource Sharing in a Distributed Environment. En: IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics, Vol. 34, No. 2.
- [33] Xu, P.; Deters, R (2004) MAS & Fault-Management. En: Proceedings of the 2004 International Symposium on Applications and the Internet (SAINT'04) IEEE.
- [34] Sugawara, T.; Kurihara, S.; Akashi, O. (2003) Learning Implicit Resource Relationships from Past Plans in Multi- Agent Systems. En: Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems.
- [35] Huhns, M.; Stephens, L. (1999) Multiagent Systems and Societies of Agents. En: Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence, editado por Gerhard Weiss, The MIT Press.
- [36] Flores, C. (2005) Negociação Pedagógica Aplicada a um Ambiente Multiagente de Aprendizagem Colaborativa. Universidade Federal do Rio Grande do Sul. Tese de Doutorado.
- [37] RAE (2007) Real Academia de la Lengua Española. [En línea]: <http://www.rae.es> Fecha de acceso: Abril de 2007.
- [38] Jiménez, J.; Acosta, G.; Londoño, L.; Castro, C. (2006) Ambientes Colaborativos en Robótica Móvil. En: Jornadas de Investigación Universidad de Medellín, Medellín, 2006.

- [39] Dorneich, M. (2002) A System Design Framework-Driven Implementation of a Learning Collaboratory. En: IEEE Transactions on Systems, Man and Cybernetics—Part A: Systems and Humans, Vol. 32, No. 2.
- [40] Grudin, J. (1994) Computer-Supported Cooperative Work: History and Focus. En: IEEE Transactions Vol. 27 No. 5.
- [41] Johnson, D.; Johnson, R. (1978) Cooperative, Competitive and Individualistic Learning. Journal of Research and Development in Education, Vol 12.
- [42] Ortega, M.; Bravo, J. (1998) Groupware and Computer-Supported Collaborative Learning.
- [43] Carroll, J.; Rosson, M.; Convertino, G.; Ganoë, C. (2005) Awareness and Teamwork in Computer-Supported Collaborations. En: Interacting with Computers, Vol. 18, Nro. 1, pags 21-46.
- [44] Mohammed, S.; Dumville, B. (2001) Team Mental Models: Expanding Theory and Measurement Through Cross-Disciplinary Boundaries. En: Journal of Organizational Behavior 22, pp. 89–106.
- [45] Neisser, U. (1976) Cognition and Reality. San Francisco: Freeman.
- [46] Dourish, P.; Bellotti, V. (1992) Awareness and Coordination in Shared Workspaces. Proceeding ACM Conference on Computer Supported Cooperative Work (CSCW'92), Toronto, ACM Press.
- [47] Navarro-Serment, L.L.; Paredis, C.; Khosla, P. (1999) A Beacon System for the Localization of Distributed Robotic Teams. En: International Conference on Field and Service Robots, Pittsburgh, PA, pp. 232–237.
- [48] Habib, M.K. (2007) Real Time Mapping and Dynamic Navigation for Mobile Robots. En: International Journal of Advanced Robotic Systems, Vol. 4, No. 3, pp. 323-338, 2007.
- [49] Thrun, S.; Burgard, W.; Fox, (2000) A Real-Time Algorithm for Mobile Robot Mapping with Applications to Multi-Robot and 3d Mapping. En: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA2000), Vol. 1, pp. 321–328, 2000.
- [50] AL-Taharwa, I.; Sheta, A.; AL-Weshah, M. (2008) A Mobile Robot Path Planning Using Genetic Algorithm in Static Environment. En: Journal of Computer Science, Vol. 4, No. 4, pp. 341-344, 2008.
- [51] Shiguemi, L.; Corrêa, A.; Silveira, G.F. (2004) Simultaneous Localization and Map Building by a Mobile Robot Using Sonar Sensors. En: ABCM Symposium Series in Mechatronics, Vol. 1, pp. 115-223, 2004.
- [52] Andrade-Cetto, J.; Sanfeliu, A. (2001) Learning of Dynamic Environments by a Mobile Robot from Stereo Cues. En: International Conference on Multisensor Fusion and Integration for Intelligent Systems, Baden-Baden, Germany, pp. 305-310, 2001.
- [53] Aboshosha, A.; Zell, A. (2003) Robust Mapping and Path Planning for Indoor Robots Based on Sensor Integration of Sonar and 2D Laser Range Finder En: IEEE 7th. International Conference on Intelligent Engineering Systems, March 4-6, 2003, Assiut-Luxor, Egypt, 2003.
- [54] Dufourd, D. (2005) Des Cartes Combinatoires pour la Construction Automatique de Modèles D'environnement par un Robot Mobile. PhD. Tesis, Institut National Polytechnique de Toulouse, France, 2004.
- [55] Lee, D.; Chung, W.; Kim, M. (2002) Probabilistic Localization of the Service Robot by Map Matching Algorithm. En: International Conference on Control, Automation, and Systems ICCAS, Korea, pp. 1667-1677, 2002.

- [56] Beeson, P.; Provost, J.; MacMahon, M.; Savelli, F.; Modayil, J.; Kuipers, B. (2003) Exploiting Local Perceptual Models for Topological Map-Building. En: IJCAI-2003 Workshop on Reasoning with Uncertainty in Robotics (RUR-03), Acapulco, Mexico, pp. 15–22, 2003.
- [57] Milagros, M. (2004) Contribuciones a la Construcción de Mapas para la Navegación de Robots, con Información Procedente de Sensores de Ultrasonidos. PhD. Tesis, Universidad Politécnica de Valencia, España, 2004.
- [58] Jefferies, M.E.; Yeap, W.K.; Smith, L.; Ferguson, D. (2001) Building a Map for Robot Navigation Using a Theory of Cognitive Maps. En: IASTED International Conference on Artificial Intelligence Applications, 2001.
- [59] Cañas, J. (2003) Jerarquía Dinámica de Esquemas para la Generación de Comportamiento Autónomo. PhD. Tesis, Universidad Politécnica de Madrid, España, 2003.
- [60] Burgard, W.; Fox, D.; Henning, D.; Schmidt, T. (1996) Estimating de Absolute Position of a Mobile Robot using Position Probability Grids. En: Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI-96), pp. 896–901, 1996.
- [61] Dudek, G.; Jenkin, M. (2000) Computational Principles of Mobile Robotics. Cambridge University Press, Cambridge, United Kingdom, 2000.
- [62] Ratner, D.; McKerrow, P. (2001) Lanmark Recognition with CTMF Ultrasonic Sensing. En: Australian Conference on Robotics and Automation ACRA2001, Sydney, Australia, pp. 104-111, November 2001.
- [63] Leonard, J.J.; Durrant-Whyte, H.F. (1991) Simultaneous Map Building and Localization for an Autonomous Mobile Robot. En: Proceedings of the International Workshop on Intelligent Robots and Systems, Osaka, Japan, Vol. 3, pp. 1442–1447, 1991.
- [64] Aufrère, R.; Mertz, C.; Thorpe, C. (2003) Multiple Sensor Fusion for Detecting Location of Curbs, Walls, and Barriers. En: Proceedings of the IEEE Intelligent Vehicles Symposium (IV2003), June 2003.
- [65] Lakaemper, R.; Latecki, L. J.; Sun, X.; Wolter, D. Geometric Robot Mapping. En: 12th International Conference on Discrete Geometry for Computer Imagery (DGCI 2005), Poitiers, Francia, Abril 2005.
- [66] Aguirre, E.; González, A. (2002) Integrating Fuzzy Topological Maps and Fuzzy Geometric Maps for Behavioral-Based Robots. En: International Journal of Intelligent Systems, Vol. 17, No. 3, pp. 333–368, 2002.
- [67] Armingol, J.M.; Moreno, L.E.; de la Escalera, A.; Salichs, M.A. (2002) A Genetic Algorithm for Mobile Robot Localization Using Ultrasonic Sensors. En: Journal of Intelligent and Robotic Systems, Vol. 34, No. 2, pp. 135-154, 2002.
- [68] Crowley, J. L. (1989) World Modeling and Position Estimation for a Mobile Robot Using Ultrasonic Ranging. En: Proceedings of the 1989 IEEE International Conference on Robotics and Automation ICRA89, Scottsdale, Arizona, USA, Vol. 3, pp. 674-680, May 14-19, 1989.
- [69] Caramés, C.F. (2007) Técnicas de Navegación de Robots Basadas en Sistemas de Medición por Láser. Trabajo de Grado, Universidad de Salamanca, Salamanca, España, 2007.
- [70] Premebida, C.; Nunes, U. (2005) Segmentation and Geometric Primitives Extraction from 2D Laser Range Data for Mobile Robot Applications. En: Proceedings of the 5th National Festival of Robotics, Scientific Meeting (ROBOTICA2005), Coimbra, Portugal, 2005.

- [71] Moravec, H.P.; Elfes, A. (1985) High Resolution Maps from Width Angle Sonar. En: Proceedings of the 1985 IEEE International Conference on Robotics and Automation.
- [72] Winters, N. (2001) A Holistic Approach to Mobile Robot Navigation using Omnidirectional Vision. PhD. Tesis, University of Dublin, Trinity College, Dublin, Ireland, 2004.
- [73] Krebsbach, K.; Willard, B. (2006) Map-Making with a Four-Legged Mobile Robot. En: Proceedings of the AAAI Spring Symposium on Robots and Robot Venues: Resources for AI Education, Stanford University, Palo Alto, CA, March 26-28, 2007.
- [74] Acosta, G.; Flórez, C.; González, J.; Rodríguez, L. (2007) Sistema Ultrasónico de Percepción para Sistemas Multi-Agente Robótico. En: Memorias del XXI Congreso Panamericano de Ingeniería Mecánica, Eléctrica, Industrial y Áreas Afines COPIMERA2007, Lima, Perú, 2007.
- [75] Fonseca, J.; Martins, J.S.; Couto, C. (2001) An Experimental Model for Sonar Sensors. En: ITM 2001 Proceedings of the 1st. International Conference on Information Technology in Mechatronics. 1-3 October 2001, Istambul-Turkey, pp. 203-208.
- [76] Maita, F.; Nunes, U. (2001) Multi-Echo Technique for Feature Detection and Identification Using Simple Sonar Configuration. En: IEEE/ASME International Conference on Advanced Intelligent Mechatronics Proceedings, 8-12 July, 2001, Como, Italy.
- [77] O'Sullivan, S.; Collins, J. J.; Mansfield, M.; Eaton, M.; Haskett, D. (2004) A Quantitive evaluation of sonar models and mathematical update methods for Map Building with mobile robots. En: Proceedings of the Ninth International Symposium on Artificial Life and Robots, January 2004, Oita, Japan, 2004.
- [78] Fukuda, T.; Sekiyama, K. (1994) Communication Reduction with Risk Estimate for Multiple Robotic System. In IEEE ICRA, pages 2864–2869.
- [79] Iglesias, C.; Garijo, M.; González, J.,; Velásquez, J. (1998) Analysis and Design of Multiagent Systems using MAS-CommonKADS. Disponible en: <http://citeseer.ist.psu.edu/iglesias98analysis.html> Fecha de acceso: Abril de 2007.
- [80] DeLoach, S.; Matson, E.; Li, Y. (2002) Applying Agent Oriented Software Engineering to Cooperative Robotics. Proceedings of the Fifteenth International Florida Artificial Intelligence Research Society Conference.
- [81] Odell, J.; Parunak, H.; Bauer, B. (2000) Extending UML for Agent. Disponible en: <http://www.jamesodell.com/ExtendingUML.pdf> Fecha de acceso: Abril de 2007.
- [82] Parker, L. (1998) Toward the Automated Synthesis of Cooperative Mobile Robot Teams. Disponible en: <http://citeseer.ist.psu.edu/parker98toward.html> Fecha de acceso: Abril de 2007.
- [83] Rueda, S.; Ucrós, M.; Quintero, A. (1999) Agentes y Sistemas Multiagente: Integración de Conceptos Básicos. Memos de investigación. Universidad de los Andes no. 319, pp.1-14.
- [84] Cao, Y.; Fukunaga, A.; Kahng, A.; Meng, F. (1995) Cooperative Mobile Robotics: Antecedents and Directions. Proc. 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '95) Pittsburgh, PA, Vol. 1, pp 226-234.
- [85] VanLehn, K. (1991) Architectures for Intelligence: the 22nd Carnegie Mellon Symposium on Cognition. Laerence Elrbaum Associates.
- [86] Dudek, G.; Jenkin, M.; Miliós, E.; Wilkes, D. (1993) A Taxonomy for Swarm Robots. Proc. 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Yokohama Japan, pp. 441-447.

- [87] Kendall, E. (1998) Agent Roles and Role Models. Intelligent Agents for Information and Process Management (AIP'98).
- [88] Iglesias, C.; Garijo, M.; González, J., (1999) A survey of Agent-Oriented Methodologies. Disponible en: <http://citeseer.ist.psu.edu/iglesias99survey.html> Fecha de acceso: Abril de 2007.
- [89] Iglesias, C. (1998) Definición de una metodología para el desarrollo de sistemas multi-agente. Tesis doctoral. Universidad Politécnica de Madrid, pp. 284.
- [90] Chandrasekaran, B.; Josephson, J.; Benjamins, V. (1999) Ontologies of Tasks and Methods. IEEE Intelligent Systems, 14(1): 20-26.
- [91] Nilsson, N. (2001) Inteligencia Artificial. Una Nueva Síntesis. Mc-Graw Hill.
- [92] Rekleitis, I.; Sim, R.; Dudek, G.; Millos, E. (2001) Collaborative Exploration for the Construction of Visual Maps. Disponible en: <http://citeseer.ist.psu.edu/449413.html> Fecha de acceso: Abril de 2007.
- [93] Evans, J.; Krishnamurthy, B. (1998) HelpMate, The Trackless Robotic Courier. Lectures Notes in Control and Information Sciences. Springer Verlag.
- [94] Mataric, M. (1994) Issues and Approaches in Design of Collective Autonomous Agents. En: Robotics and Autonomous Systems, 16(2-4), Dec 1995, pp. 321-331
- [95] Donald, B.; Jennings, J.; Rus, D. (1993a) Experimental Information Invariants for Cooperating Autonomous Mobile Robots. Proceedings International Symposium on Robotics Research.
- [96] Drogoul, A.; Zucker, J.-D. (1998) Methodological Issues for Designing Multi-Agent Systems with Machine Learning Techniques: Capitalizing Experiences from the RoboCup Challenge. LIP6 – Université Paris 6.
- [97] Jones, J., Seiger, B. ; Flynn, A. (1998) Mobile Robots. Inspiration to Implementation. Second Edition. A K Peters Natick, Massachusetts.
- [98] Mataric, M. (1997) Behavior-Based Control: Examples from Navigation, Learning, and Group Behavior. Disponible en: <http://citeseer.ist.psu.edu/mataric97behaviorbased.html> Fecha de acceso: Abril de 2007.
- [99] Vernon, C. (2005) A Principle Design Methodology for Minimalist Multi-Robot System Controllers. Ph.D. Thesis. University of Southern California.
- [100] Szabo, R. (2004) Topological Navigation of Simulated Robots using Occupancy Grid. En: International Journal of Advanced Robotic System, Vol. 1, No. 3, pp. 201-206, 2004.
- [101] Parasuraman, S.; Ganapathy, V.; Shirinzadeh, B. (2005) Behavior based Mobile Robot Navigation Technique using AI System: Experimental Investigations. En: International Conference on Automation, Robotics and Autonomous Systems (ARAS 05), Cairo, Egypt, 2005.
- [102] Jiménez, J., Ovalle, D., Branch, J. W. (2008) Comunicación en Sistemas de Múltiples Robots desde la Metodología MAD-Smart. En: Ingeniería e Investigación, Vol.28, Nro.2, Agosto de.2008.

Anexos A. Diagramas de Flujo de programas y subrutinas

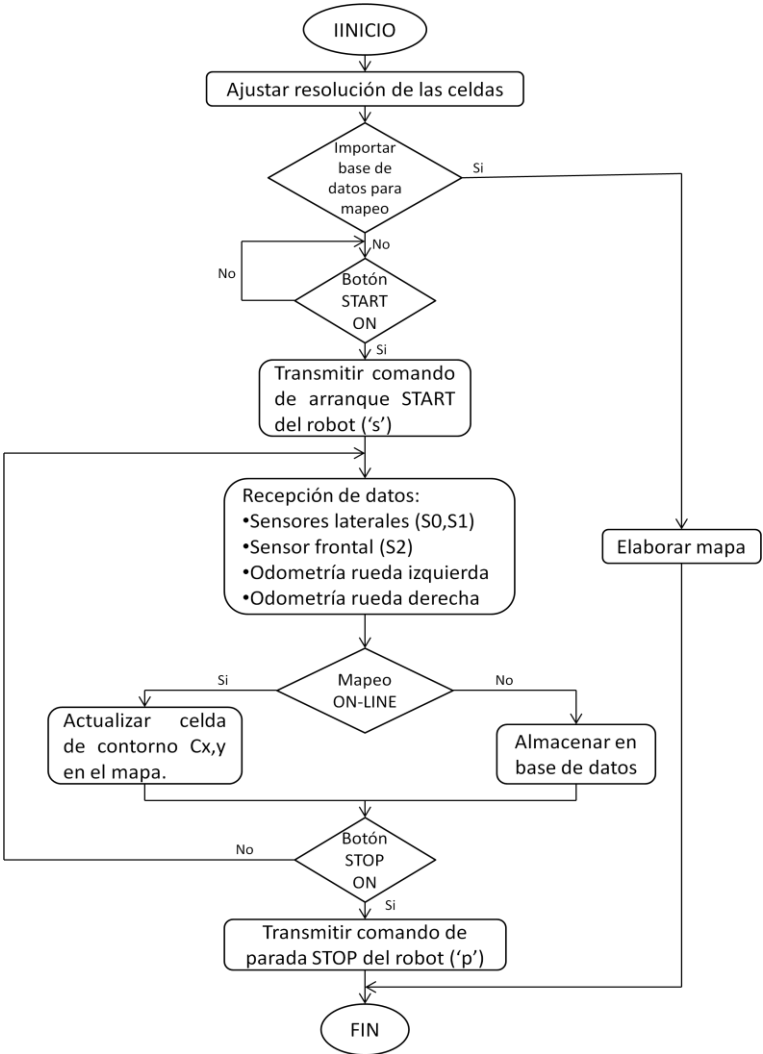


Figura A1. Diagrama de flujo del módulo de mapeo Vespucci.

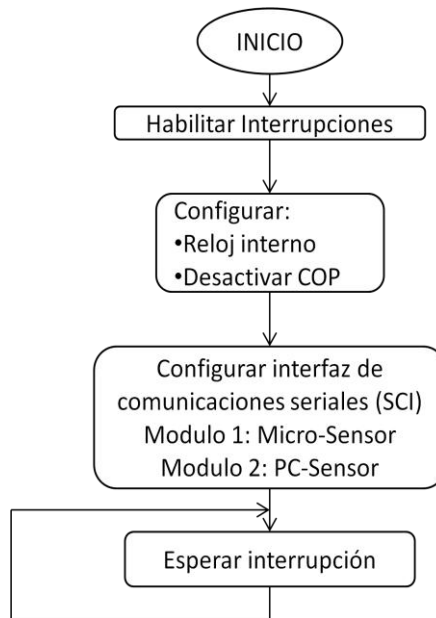


Figura A2. Diagrama de flujo del programa principal del módulo de control del robot *Walling*.

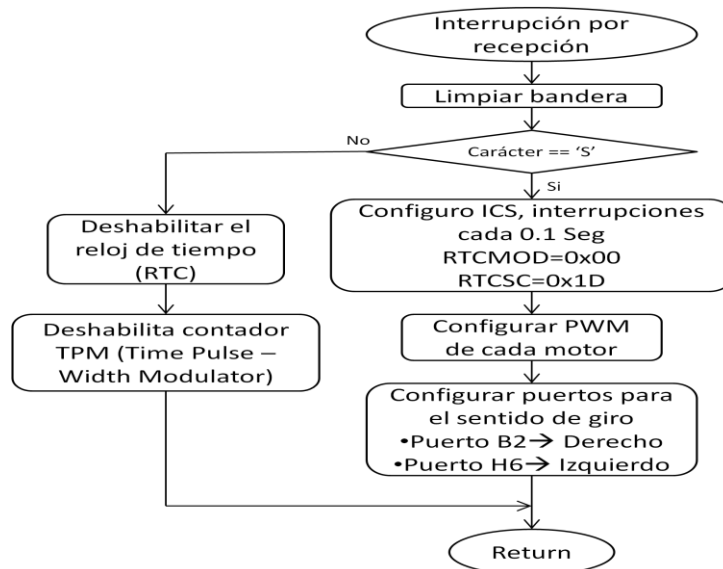


Figura A3. Rutina de atención a interrupción por recepción de datos del puerto serial en el robot *Walling*.

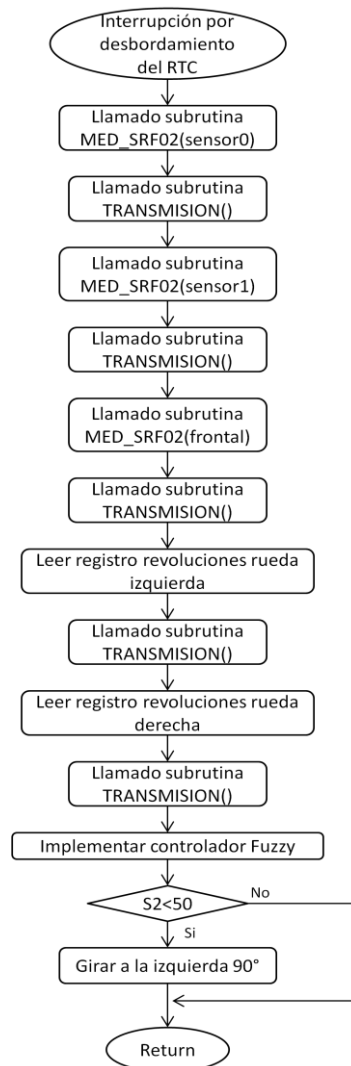


Figura A4. Rutina de atención a interrupción por desbordamiento del reloj de tiempo real.