



INAOE

# Fusión de Mapas Locales a partir de la Percepción Distribuida de un Sistema Multirobot

por

**José Alberto Méndez Polanco**

Tesis sometida para obtener el grado de **Maestro en Ciencias** en la especialidad de **Ciencias Computacionales** en el Instituto Nacional de Astrofísica, Óptica y Electrónica.

Supervisada por:

Dra. Angélica Muñoz Meléndez

Octubre 2007, Tonantzintla, Puebla

©INAOE 2007

Derechos reservados

El autor otorga al INAOE el permiso de reproducir y distribuir copias de su totalidad o en partes de esta tesis





# Resumen

En este trabajo se presenta el diseño y la implementación de un sistema multirobot compuesto por tres robots móviles homogéneos y un servidor externo. Cada robot fue diseñado y equipado en el Laboratorio de Robótica del la coordinación de Ciencias Computacionales del INAOE. El objetivo principal del sistema multirobot es la actualización del mapa de un ambiente interior a partir de la percepción distribuida de sus miembros. Para cada robot miembro del sistema multirobot se programó un conjunto de comportamientos y un conjunto capacidades que le permite la navegación y localización en un ambiente interior, la comunicación con otros miembros del sistema, la generación de mapas locales y su fusión en un único mapa global, así como de acciones individuales para alcanzar el objetivo global. Finalmente se evaluó el desempeño del sistema en un ambiente en diversas condiciones.



# Abstract

In this thesis, the design and implementation of a multirobot system consisting of three homogeneous mobile robots and an external server are presented. Each robot was designed and equipped at the Laboratory Robotics of the Computational Science Department of the INAOE. The main goal of the multirobot system is the updating of a map or representation of an indoor environment. A set of behaviors and a set of capabilities were programmed for each robot. These behaviors and capabilities enable robots to navigate and self-locate in an indoor environment, communicate to each other, create local maps of their environment to be merged into a global map, and coordinate individual actions in order to the achieve the collective goal. Finally, the performance of the multirobot system was evaluated in various environments.



# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. Antecedentes . . . . .	1
1.2. Definición del problema . . . . .	2
1.3. Objetivos de la tesis . . . . .	2
1.4. Alcances y limitaciones . . . . .	3
1.5. Metodología . . . . .	5
1.6. Organización de la tesis . . . . .	7
<b>2. Fundamentos</b>	<b>9</b>
2.1. Sistemas multirobot . . . . .	9
2.1.1. Interacción . . . . .	9
2.1.2. Comunicación . . . . .	11
2.2. Mapas . . . . .	12
2.2.1. Mapas métricos . . . . .	13
2.2.2. Mapas topológicos . . . . .	14
2.3. Modelo de percepción . . . . .	15
2.4. Localización . . . . .	18
2.4.1. Localización local . . . . .	19
2.4.2. Localización global . . . . .	21
2.4.3. El problema del robot “secuestrado” . . . . .	22
2.4.4. Localización basada en mapas . . . . .	22
2.5. Incertidumbre en la localización . . . . .	23
<b>3. Estado del Arte</b>	<b>25</b>
3.1. Problemática . . . . .	25
3.2. Trabajos relacionados . . . . .	26
3.2.1. Coordinación de los robots . . . . .	26
3.2.2. Estimación de la posición . . . . .	28

3.2.3.	Fusión de mapas . . . . .	32
3.2.4.	Estrategias para la generación de mapas . . . . .	33
<b>4.</b>	<b>SMR para la exploración colectiva de un ambiente</b>	<b>35</b>
4.1.	Requerimientos de sistema . . . . .	35
4.1.1.	Especificación de los tipos de ambientes a explorar por el sistema multirobot . . . . .	35
4.2.	Diseño y equipamiento de los robots . . . . .	37
4.3.	Especificación del Sistema . . . . .	40
4.3.1.	Especificación general . . . . .	40
4.3.2.	Especificación del SMR . . . . .	42
4.4.	Capacidades del sistema (Ca) . . . . .	44
4.5.	Comportamientos (Co) . . . . .	45
4.6.	Organización del sistema . . . . .	48
4.7.	Generación automática del mapa original del ambiente . . . . .	50
4.8.	Mapa basado en rejillas de ocupación . . . . .	50
4.9.	Mapa topológico . . . . .	56
4.10.	Corrección del error odométrico . . . . .	59
4.11.	Estimación de la pose del robot . . . . .	61
4.12.	División y fusión del mapa . . . . .	61
4.12.1.	División, repartición y asignación del espacio . . . . .	62
4.12.2.	Acceso a zonas . . . . .	63
4.12.3.	Exploración colectiva, actualización y corrección de los mapas locales de los robots . . . . .	65
<b>5.</b>	<b>Experimentos y evaluación</b>	<b>67</b>
5.1.	Experimentos individuales . . . . .	68
5.1.1.	Experimento no. 1: entrada al ambiente . . . . .	68
5.1.2.	Experimento no. 2 navegación y localización en el ambiente dado un mapa topológico . . . . .	70
5.1.3.	Experimento 3: seguimiento de pared y estimación de la posición . . . . .	73
5.1.4.	Experimento 4: generación de un mapa local . . . . .	75
5.2.	Experimentos colectivos de coordinación, comunicación, generación y fusión de mapas . . . . .	77
5.2.1.	Experimento 1: Actualización de un mapa del ambiente sin modificaciones . . . . .	79

5.2.2.	Experimento 2: Actualización de un mapa del ambiente con modificaciones . . . . .	82
5.2.3.	Experimento 3: Comunicación y coordinación entre los miembros del sistema multirobot . . . . .	85
5.2.4.	Experimento 4: Actualización de un mapa del ambiente sin modificaciones y sin comunicación con el servidor . . . . .	89
<b>6.</b>	<b>Conclusiones</b>	<b>93</b>
6.1.	Recapitulativo . . . . .	93
6.2.	Aportaciones . . . . .	95
6.2.1.	Materiales . . . . .	95
6.2.2.	Conceptuales . . . . .	96
6.3.	Conclusiones y perspectivas . . . . .	97
<b>A.</b>	<b>Handyboard©</b>	<b>99</b>
<b>B.</b>	<b>Pocket PCs©</b>	<b>103</b>
<b>C.</b>	<b>EWE</b>	<b>105</b>
<b>D.</b>	<b>Modificación de los servomotores</b>	<b>109</b>
	<b>Bibliografía</b>	<b>113</b>



# Índice de figuras

2.1. Rejilla de ocupación creada por Moravec et Elfes (1985) . . . .	14
2.2. Mapa topológico a partir de rejilla de ocupación . . . . .	17
2.3. Modelo del sonar propuesto por Murphy (2000). . . . .	17
2.4. Construcción de mapas con corrección del error odométrico . .	19
2.5. Localización basada en marcas . . . . .	20
2.6. Método de correlación . . . . .	21
3.1. Coordinación de robots basada en fronteras para la construc- ción de un mapa . . . . .	27
3.2. Coordinación de robots mediante un robot líder . . . . .	29
3.3. Problemas en la fusión de mapas . . . . .	32
4.1. Ambientes con distribuidor simple. . . . .	37
4.2. Ambientes con distribuidor con bifurcaciones . . . . .	37
4.3. Ambientes con distribuidor con ciclo. . . . .	38
4.4. Los tres robots miembros del Sistema Multirobot. . . . .	38
4.5. Equipamiento de los prototipos de robot. . . . .	40
4.6. Organización interna de los prototipos de robot. . . . .	41
4.7. Organización del sistema multirobot . . . . .	49
4.8. Sensado de un ambiente. . . . .	51
4.9. Corrección de la odometría. . . . .	60
4.10. Estimación de la posición de un robot mediante características. .	61
4.11. . . . .	62
4.12. Mapa local de un ambiente de pruebas . . . . .	64
4.13. Entrada al ambiente de un miembro del sistema multirobot. .	65
4.14. Mapas locales . . . . .	66
5.1. Experimento individual 1 . . . . .	70
5.2. Experimento individual 2 . . . . .	72

5.3.	Experimento individual 3 . . . . .	74
5.4.	Mapa local en el experimento individual 4 . . . . .	76
5.5.	Mapa métrico inicial . . . . .	80
5.6.	Mapa topológico . . . . .	81
5.7.	Experimento colectivo 1 . . . . .	83
5.8.	Mapas métricos del experimento colectivo 1 . . . . .	84
5.9.	Modificaciones al ambiente de pruebas . . . . .	85
5.10.	Experimento colectivo 2 . . . . .	86
5.11.	Mapas métricos del experimento colectivo 2 . . . . .	87
5.12.	Experimento colectivo 3 . . . . .	88
5.13.	Mapas métricos del experimento colectivo 4 . . . . .	91
A.1.	Tarjeta de control Handyboard© . . . . .	100
A.2.	Tarjeta de expansión de la Handyboard© . . . . .	101
A.3.	Tarjeta controladora Handyboard© y modulo de expansión . . . . .	101
B.1.	Pocket PC IPAQ RX 1950 de Hewlett Packard. . . . .	103
B.2.	Pocket PC IPAQ RX 5550 de Hewlett Packard. . . . .	104
D.1.	Servomotor HiTec. . . . .	110
D.2.	Servomotor modificado. Vista interna. . . . .	111
D.3.	Corte del tope del servomotor. . . . .	111

# Índice de cuadros

2.1. Ventajas y desventajas de los mapas basados en rejillas de ocupación. . . . .	15
2.2. Ventajas y desventajas de los mapas topológicos. . . . .	16
4.1. Estados posibles de las celdas analizadas durante el recorrido virtual del mapa métrico $MM$ realizado por el algoritmo AAMT. . . . .	57



# Capítulo 1

## Introducción

### 1.1. Antecedentes

La autonomía es una de las características más importantes de los robots móviles. Por robot autónomo se entiende un robot que depende para su operación de los sensores y de los actuadores con los que está equipado, así como de los mecanismos de control que se le hayan programado previamente. Un robot móvil no puede ser autónomo si no es capaz de localizarse en su entorno, i.e. capaz de reconocer el área específica del ambiente en el que el robot está ubicado. Para ello, es necesaria una representación del entorno en el cual el robot se encuentra inmerso. El ambiente de un robot se representa en la forma de un mapa que contiene una descripción numérica o cualitativa de algunas características importantes, e.g. dimensión, ubicación de objetos, etc. Los mapas pueden ser locales o globales de manera general, los mapas locales representan la parte del ambiente que rodea a un robot, mientras que los mapas globales representan toda el área de movimiento del robot.

En general, la complejidad en las tareas de construcción o actualización de un mapa dependerá del nivel de detalle que se desea tener para representar el ambiente, y del tamaño de dicho ambiente. Cuando se tiene un ambiente de tamaño considerable que ha sufrido cambios y se dispone de tiempo restringido para explorarlo, el problema se vuelve más complejo y puede llegar a ser complicado de resolver con un único robot. Es en este punto donde toman importancia los sistemas multirobot, sistemas compuestos de más de un robot que colaboran por un objetivo común. El mapa podría ser actualizado en menor tiempo si en lugar de emplear un solo robot, se empleara un

sistema multirobot.

Hoy en día se han desarrollado algoritmos para la creación de mapas por un único robot y dichos algoritmos son cada vez más robustos. No obstante, existe poca investigación en el tema de fusión en único mapa global de mapas locales generados a partir de la percepción de los miembros de un sistema multirobot.

## 1.2. Definición del problema

Este nuevo enfoque de actualización de mapas empleando un sistema multirobot plantea nuevos retos y problemas como la asignación de recursos, la división del mapa en zonas que cada uno de los robots debe explorar evitando en lo posible traslapes entre zonas, la coordinación de los robots para evitar colisión entre ellos, la creación y fusión de los mapas locales tomando en cuenta que puede existir redundancia de información en ellos. Esta propuesta de tesis tiene como propósito crear un sistema multirobot capaz de fusionar mapas locales en un mapa global del ambiente. En concreto, se plantea diseñar e implementar mecanismos de interacción multirobot orientados a:

- la división de un mapa en zonas de exploración.
- la exploración de las zonas del ambiente representadas por cada una de las zonas del mapa y la actualización de los mapas locales de esas zonas.
- la fusión de los mapas locales actualizados en un mapa global del ambiente.

## 1.3. Objetivos de la tesis

El objetivo general de la tesis es diseñar e implementar algoritmos de interacción para un sistema multirobot, orientados a la división de un mapa en zonas, exploración colectiva de un ambiente interior, actualización de mapas locales y su fusión en un único mapa global coherente, i.e. libre de redundancia y de información ruidosa.

Para alcanzar el objetivo propuesto, se han identificado los siguientes objetivos particulares:

- Analizar el equipamiento necesario en sensores y sistemas de comunicación para un robot construido a partir de una plataforma comercial de control diferencial, capaz de comunicarse con otros robots similares y desplazarse de manera autónoma en un ambiente semi-estructurado con características que permitan su representación en forma de un grafo acíclico no dirigido (c.f 4.9) .
- Equipar tres prototipos de robots móviles con base en el análisis anterior.
- Definir un esquema de distribución y asignación de recursos para la exploración de un ambiente representado en un mapa.
- Definir un conjunto de comportamientos individuales para los robots, que les permita el acceso al área a explorar y la actualización de los mapas locales.
- Definir y programar un esquema de fusión de los mapas locales en un único mapa global coherente.
- Validar los diversos esquemas y comportamientos propuestos en la exploración de un ambiente colapsado cuya configuración inicial es conocida.

## 1.4. Alcances y limitaciones

Los alcances y limitaciones de la presente tesis son los siguientes:

- Los ambientes interiores a explorar por un robot móvil son muy diversos, por que será necesario realizar una caracterización de algunos ambientes controlados, en los cuales el sistema multirobot pueda desempeñar su tarea de actualización y fusión de mapas locales. Dichos ambientes deben tener características específicas encontradas en ambientes reales tales como pasillos, habitaciones, paredes etc.

- La comunicación en un sistema multirobot es importante, para una buena coordinación entre los miembros del sistema. Por lo anterior, cada robot miembro del sistema debe equiparse con un mecanismo para comunicación con otros robots y un servidor externo.
- Los robots estarán equipados con sensores de percepción limitados, por lo que debe implementarse un modelo del ambiente discretizado que permita a los robots, entre otras habilidades, navegar y localizarse dentro de un ambiente interior. En particular, cada robot utilizará un par de odómetros que le permitan medir el desplazamiento en un rango de 5 *cm.* a 20 *cm.* con una precisión de 2 *cm.* Cada robot empleará además, un sensor ultrasónico que le permita medir la distancia entre el sensor y los objetos próximos en un rango de 5 *cm.* a 300 *cm.* con una precisión de 1 *cm.* Por último, cada robot empleará un sensor IR que le permita percibir la presencia de objetos en un rango de 10 *cm.* a 150*cm.*
- El trabajo propuesto está orientado al control de  $n$  robots. En esta tesis, principalmente por limitaciones de equipo, nos limitamos a tres robots.
- No se contemplarán soluciones a los problemas que surjan cuando un robot se pierda u obstruya el pasillo principal del ambiente. Estos problemas incrementan la complejidad de los esquemas y algoritmos necesarios para su solución y son, en sí mismos, trabajos de investigación completos.
- Todos los robots tendrán comunicación con un servidor externo, el cual será el encargado de dividir el ambiente en zonas y asignarlas a cada uno de los robots, y de generar los mapas locales y fusionarlos en un mapa global. Sin embargo, cada robot debe ser autónomo, eso quiere decir que recibirá información del servidor, pero la ejecución de tareas, la localización y navegación en sí mismas son realizadas por sus propios medios.
- Se contemplarán los casos en que uno o todos los robots miembros del sistema pierdan la comunicación entre ellos o con el servidor externo. El sistema multirobot deberá ser tolerante a dichas pérdidas de comunicación, y deberá generar un mapa global a pesar de producirse estas pérdidas.

## 1.5. Metodología

Se propone la siguiente metodología para alcanzar los objetivos enunciados anteriormente.

1. **Diseño y equipamiento de los robots.** Consiste en el diseño y equipamiento de tres prototipos de robots a partir de una plataforma comercial de control diferencial.  
Se equiparán dichos robots con los dispositivos de percepción necesarios, e.g. sonar y sensores de proximidad infrarrojos, y con dispositivos de actuación, e.g. motores DC para las llantas y servomotores para el barrido de los sonares, necesarios para una navegación y exploración autónoma. Para la programación cada robot se empleará una tarjeta comercial *Handyboard*© basada en un controlador 68HC11 de Motorola y una *Pocket PC* para la programación en alto nivel y la comunicación inalámbrica entre los robots.
2. **Preparación del ambiente.** Definir las características de los ambientes en los cuales se realizarán pruebas al sistema multirobot. Adicionalmente, se preparará un ambiente físico con dichas características.
3. **Creación de un mapa global basado en rejillas de ocupación.** El mapa global estará representado por rejillas de ocupación y su construcción inicial estará a cargo de un robot autónomo a partir de la información de sus sensores. El ambiente inicial se encontrará en condiciones ideales, i.e. sin derrumbes ni cambios estructurales.
4. **Creación de un mapa topológico.** A partir del mapa basado en rejillas de ocupación se implementará un algoritmo para obtener un mapa topológico, el cual servirá de base para la división del ambiente y la asignación de zonas, así como para la navegación de los robots.
5. **División del mapa.** Para dividir la tarea de actualización del mapa global entre los miembros del sistema multirobot, el mapa global será dividido en zonas disjuntas a partir del mapa topológico obtenido anteriormente.

6. **Asignación de zonas.** Las zonas en las que previamente fue dividido el mapa serán asignadas a los robots por un servidor remoto. Para el presente trabajo se contempla la intervención de tres robots autónomos.
7. **Acceso de los robots a la zona asignada.** Mediante un conjunto de comportamientos individuales, los robots deben acceder al área a explorar. Se pueden emplear algoritmos de planificación de trayectorias para calcular la ruta más adecuada que el robot debe seguir para acceder a la región que le fue asignada. Se pueden también aprovechar propiedades del ambiente que funcionarán como marcas o balizas, las cuales pueden ser utilizadas por los robots como puntos de referencia para acceder a la zona asignada.
8. **Exploración de las zonas y actualización de los mapas locales que las representan.** La actualización de los mapas locales se realizará empleando el mismo algoritmo que fue utilizado para la construcción del mapa global.
9. **Entrega de mapas locales.** La información recolectada por cada robot miembro del sistema será enviada de manera continua o intermitente a un robot servidor o en su defecto a una computadora personal operando como servidor, durante la fase previa de exploración. En el peor de los casos la entrega puede requerir el regreso de los robots al punto de origen.
10. **Fusión de mapas locales.** Los mapas locales serán fusionados por el robot servidor o la computadora operando como servidor en un único mapa global.
11. **Evaluación.** Evaluación de la calidad del mapa aplicando dos tipos de criterios:
  - Cualitativos: Se verifica de forma visual el mapa generado por el sistema multirobot para constatar que refleje las características principales del ambiente. Y en el caso de ambientes con modificaciones, se verifica que dicho mapa refleje los cambios.
  - Cuantitativos: Se emplea una medida de correlación, e.g. correlación cruzada normalizada, para calcular la similaridad entre el mapa creado por el sistema multirobot y un mapa inicial del mismo ambiente generado por un robot que navegó en el ambiente siguiendo una ruta predefinida..

## 1.6. Organización de la tesis

En esta tesis se presenta el diseño, equipamiento y programación de un sistema multirobot con la capacidad de crear, actualizar y fusionar mapas locales en un único mapa global. Se describen también los experimentos obtenidos por el sistema en ambientes con características analizadas y definidas previamente a la realización de los experimentos. La tesis se encuentra organizada en seis capítulos.

En el capítulo 2 se presentan los fundamentos teóricos para la navegación, construcción de mapas y localización de un robot móvil autónomo. En el capítulo 3 se realiza un estudio del estado del arte en sistemas multirobot para la construcción de mapas que involucra otras tareas del área de robótica móvil como la localización, coordinación y asignación de recursos en un sistema multirobot. En el capítulo 4 se presenta el sistema multirobot propuesto en esta tesis para la exploración colectiva de un ambiente. En el capítulo 5 se dan a conocer los experimentos y la evaluación realizada al sistema multirobot y finalmente, en el capítulo 6 se dan las conclusiones y se plantean las perspectivas de esta tesis.



# Capítulo 2

## Fundamentos

### 2.1. Sistemas multirobot

**Definición 1.** *Un **sistema multirobot** es aquel compuesto por dos o más robots diseñados para realizar tareas conjuntas y alcanzar objetivos comunes.*

Los miembros de un sistema multirobot requieren de ciertos aspectos para el alcance de la tarea global del sistema. Estos aspectos se describen a continuación.

#### 2.1.1. Interacción

La interacción es la relación entre los miembros de un sistema multirobot, y entre estos y su medio ambiente, que hace que las acciones de unos tengan influencia sobre los otros. Los miembros de un sistema multirobot pueden ejecutar tareas individuales o colectivas, no obstante, el objetivo final del sistema es uno mismo. La interacción dentro del sistema multirobot es importante para alcanzar objetivos comunes. Para lograr una adecuada interacción, se deben tomar en cuenta los siguientes aspectos:

- **Estructura del sistema multirobot.** La estructura del sistema multirobot se refiere a la morfología y control del sistema, los cuales pueden ser homogéneos o heterogéneos. Su elección dependerá de las tareas que el sistema multirobot deba ejecutar. Por ejemplo, cuando los miembros

de sistema son homogéneos, la comunicación entre ellos se puede facilitar debido a que son iguales en hardware y software. Una estructura homogénea puede ser adecuada cuando se requiera una alta comunicación entre los miembros del sistema multirobot, como en el caso de tareas tales como transporte y manipulación de objetos, coordinación de movimientos, etc. Por otro lado, los sistemas heterogéneos se conforman de robots con diferente hardware y software, e.g. estructura física distinta, diferente equipamiento, diferente forma de comunicación, etc. En este caso, se pueden tener robots con características específicas especializados en ciertas tareas, por ejemplo, para la construcción de un mapa del ambiente se pueden tener robots con la capacidad de localizarse y navegar en ambientes no estructurados, complementados con un segundo tipo de robot con las capacidades de sensado adecuadas para crear el mapa del ambiente. Si estos dos tipos de robot forman parte de un mismo sistema multirobot, los robots del primer tipo pueden guiar a los robots del segundo tipo hasta determinadas zonas del ambiente cuya representación se desea capturar en un mapa.

- **Compartición de recursos.** Los miembros del un sistema multirobot interactúan en un mismo ambiente, por lo que en ciertas ocasiones será necesario que compartan recursos para la ejecución de una tarea en común. Por ejemplo, si los robots deben entrar a un mismo ambiente para su exploración y únicamente existe un acceso, en tal escenario más de un robot puede intentar acceder al ambiente al mismo tiempo, por lo que deben organizarse para evitar el conflicto que se produce. Algunas formas de resolver este conflicto pueden ser por votación, i.e. los miembros del sistema votan por quién de los miembros del sistema debe seguir en el orden de uso del recurso; de manera aleatoria, i.e. a azar elige quién de los miembros es el siguiente en turno para emplear el recurso; o bien por prioridad o jerarquía, i.e. cada miembro puede tener asignada una prioridad que indica el orden en que el recurso será empleado.
- **Obtención de la información.** Cada miembro del sistema debe ser programado para la toma de decisiones y la ejecución de las tareas de manera individual o colectiva. Por ejemplo, en la presente tesis cada miembro del sistema multirobot debe recibir información *a priori* del mapa métrico (c.f. 2.2.1), así como de la zona que debe actualizar y un

mapa topológico (c.f 2.2.2) necesario para la navegación en el ambiente.

- **Dependencia entre los módulos.** Las acciones de un miembro del sistema pueden afectar el comportamiento de otros miembros del sistema. Un ejemplo de esto se presenta cuando varios robots navegan en un ambiente con pasillos estrechos en los que únicamente un robot puede navegar a la vez. En este caso, si un robot ya está navegando por el pasillo, debe notificarlo a los demás miembros del sistema para que éstos no intenten cruzar por el mismo pasillo y se evite un problema de colisión entre los robots.
- **División y asignación de tareas.** La realización de una tarea en conjunto puede requerir la división de la tarea en subtareas y la asignación de las subtareas entre los miembros del sistema multirobot. Sin embargo, surgen preguntas como ¿quién va a decidir cómo deben dividirse las tareas?, ¿cuándo y cómo se realiza esta división y asignación, por mencionar algunos de los aspectos a abordar en este rubro.

### 2.1.2. Comunicación

**Definición 2.** *La comunicación en un sistema multirobot es la capacidad que poseen los miembros del sistema para transmitir y recibir información entre ellos. La información es codificada empleando un lenguaje que es conocido por todos los miembros del sistema multirobot.*

La comunicación entre los miembros del sistema multirobot es importante si se desea una coordinación puntual y precisa entre los miembros del sistema multirobot. La comunicación puede ser síncrona o asíncrona. En la comunicación síncrona un robot envía un mensaje a otro robot y espera respuesta de él. Por otro lado, en la comunicación asíncrona el robot que envía el mensaje no espera respuesta del robot que lo recibe.

De acuerdo con Pereira et al. (2002), dependiendo de la forma en que la información es transmitida la comunicación puede clasificarse en:

- **Comunicación directa.** Es el intercambio de mensajes entre los miembros del sistema multirobot, donde un robot envía información a otros robots de manera específica y éstos la reciben.

- **Comunicación indirecta.** Consiste en proporcionar información a otro robot no específico, por ejemplo, la realización de marcas en el ambiente o la portación de marcas por los mismos robots las cuales pueden ser percibidas por ellos.

## 2.2. Mapas

Para que un robot pueda ejecutar ciertas tareas de manera eficiente y autónoma, a menudo es necesario dotarlo de la capacidad de localizarse en el ambiente en el que se encuentra inmerso. Para la localización, el robot debe tener un modelo del ambiente también conocido como mapa del ambiente. Un mapa puede ser empleado por un robot al menos para tres objetivos:

- identificar el espacio libre, i.e. sin obstáculos.
- reconocer zonas específicas en el ambiente.
- reconocer por su ubicación, objetos en el ambiente.

El que un robot pueda construir un mapa a partir de su percepción sensorial es importante debido a que normalmente no existen mapas adecuados a los requerimientos de un robot en particular, y hacerlos un operador manualmente es un proceso tardado y costoso. Por ello, si el robot construye su propio mapa, este podrá ser empleado de una manera más eficiente para las tareas que el propio robot tenga que realizar. Sin embargo, la elaboración de un mapa es una tarea compleja, principalmente debido a la naturaleza de los sensores con los que el robot está equipado, los cuales pueden proporcionar información incompleta o ruidosa. A continuación definimos lo que se entiende en esta tesis por un mapa de un ambiente.

**Definición 3.** *Un mapa de un ambiente es una abstracción con la que se representan únicamente las características del ambiente que se consideran útiles para una determinada tarea del robot. Estas características usualmente comprenden, paredes, pasillos, marcas en el ambiente etc.*

En general, en la tarea de construcción de un mapa de un ambiente se parte de una mapa vacío y el robot es colocado en un lugar del ambiente cuya

posición en el mapa es conocida, y usualmente expresada en coordenadas específicas. Al iniciar el proceso, el robot toma información de sus sensores, esta información es empleada para crear la parte del mapa cercana a su posición, i.e. un mapa local. Después, el robot se mueve hacia zonas del ambiente que vayan a ser incluidas en el mapa y nuevamente vuelve a tomar información de sus sensores para procesarla y actualizar el mapa. Este proceso se repite sucesivamente hasta que todas las zonas del ambiente hayan sido incluidas en el mapa.

Ahora bien, existen diversas maneras de representar un mapa, las dos más importantes son mapas métricos y mapas topológicos, los cuales se detallan a continuación.

### 2.2.1. Mapas métricos

Un mapa métrico es aquel que se basa en un marco de referencia absoluto y en estimaciones numéricas de dónde se encuentran los objetos en el ambiente.

La representación más conocida de mapas métricos son las rejillas de ocupación (ver Figura 2.1). Los mapas basados en rejillas de ocupación fueron propuestos originalmente por Moravec et Elfes (1985).

**Definición 4.** *Un mapa basado en rejilla de ocupación discretiza el ambiente en celdas de igual dimensión, cada una de las cuales mantiene un estado de ocupación (1) o de no ocupación (0) de la zona del ambiente asociada a dicha celda.*

En este tipo de mapas se asocia la lectura de los sensores directamente a las celdas para actualizar su estado, de forma que la rejilla se va actualizando mientras el robot explora el ambiente.

El valor de cada celda se puede extender utilizando una probabilidad de ocupación en vez de un valor binario, la cual se va actualizando con las lecturas de los sensores del robot y aplicando la regla de Bayes (c.f. 4.8).

La principal desventaja de los mapas basados en rejillas de ocupación es la cantidad de espacio requerido para su almacenamiento, el cual dependerá del grado de resolución del mismo. No obstante, lo anterior puede ser una ventaja al tenerse una representación detallada del ambiente principalmente para problemas de localización debido a que contine información detallada del

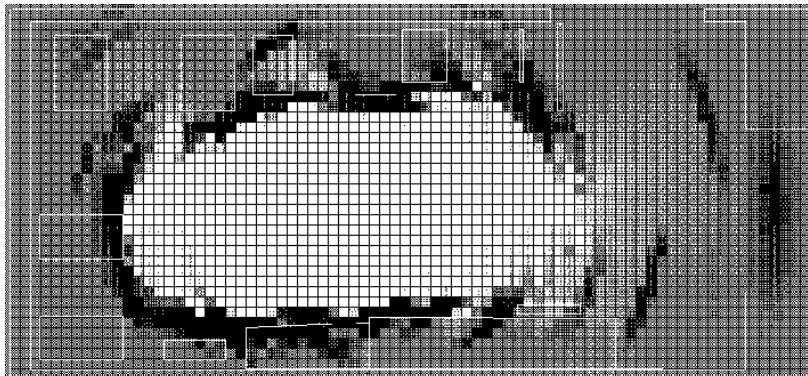


Figura 2.1: Ejemplo de una de las primeras rejillas de ocupación creadas en el trabajo realizado por Moravec et Elfes (1985). Las celdas en gris representan zonas del ambiente no exploradas, las celdas en negro representan zonas del ambiente ocupadas, mientras que las celdas en blanco indican zonas del ambiente libres de obstáculos.

ambiente que hace posible una localización más precisa . En el cuadro 2.1. se escriben las ventajas y desventajas de los mapas basados en rejillas de ocupación.

### 2.2.2. Mapas topológicos

**Definición 5.** *Un **mapa topológico**, también conocido como **mapa relacional**, es aquel que representa mediante un grafo lugares distintos del ambiente y las relaciones que existen entre ellos.*

En los mapas topológicos los nodos corresponden a lugares distintivos del ambiente, y los arcos son los caminos entre esos lugares determinados mediante alguna estrategia de control local, e.g. “seguir centro de pasillo”, “seguir pared a la derecha” o “seguir pared a la izquierda. El robot debe ser capaz de distinguir, por medio de los sensores con los que cuenta, entre diversos lugares del ambiente.

En el cuadro 2.2 se mencionan las principales ventajas y desventajas de los mapas topológicos.

Para explotar las representaciones de los mapas métricos y los mapas topológicos, a menudo es conveniente construir la representación topológica

Ventajas	Desventajas
<ul style="list-style-type: none"> <li>• Son útiles para la construcción de mapas empleando sensores ruidosos como el láser o el sonar</li> <li>• Se pueden modelar ambientes con formas irregulares</li> <li>• Describen de forma detallada el ambiente del robot</li> <li>• Permiten fusionar distintas observaciones realizadas por el robot</li> </ul>	<ul style="list-style-type: none"> <li>• Requieren una gran cantidad de espacio para su almacenamiento</li> <li>• Son sensibles a los errores de odometría.</li> </ul>

Cuadro 2.1: Ventajas y desventajas de los mapas basados en rejillas de ocupación.

empleando las observaciones realizadas en la representación métrica, por lo cual estos dos tipos de mapas no necesariamente deben ser excluyentes. En la Figura 2.2(b) se puede apreciar un mapa topológico creado a partir de un mapa basado en rejilla de ocupación ilustrado en la Figura 2.2 (a).

## 2.3. Modelo de percepción

Para que un robot pueda construir un modelo de su ambiente, es necesario contar con un modelo de los sensores con los que está equipado con el objeto de caracterizar con precisión el desempeño de los instrumentos de medición para construir dicho modelo. Dependiendo del tipo de sensor y de la función de estos en la creación del mapa y navegación, los modelos de los sensores se pueden agrupar en dos tipos: modelo de percepción y modelo dinámico. En esta tesis nos interesamos por el primero.

Un modelo de percepción proporciona las probabilidades  $P$  del comportamiento conocido del sensor  $s$ , dado que se ha (*ocupado*) o no detectado (*noocupado*) un obstáculo, i.e.  $P(s|ocupado)$  y  $P(s|no ocupado)$ .

El modelo del sonar propuesto por Murphy (2000), basado en un trans-

Ventajas	Desventajas
<ul style="list-style-type: none"> <li>• Proporcionan una base a partir de la cual es posible realizar directamente planificación de trayectorias.</li> <li>• Facilitan la interfaz con planificadores simbólicos.</li> <li>• Proporcionan una interfaz natural para la interacción humano-robot.</li> </ul>	<ul style="list-style-type: none"> <li>• El robot debe ser capaz de distinguir entre diferentes zonas del ambiente, empleando únicamente la información que le proporcionan los sensores con los que cuenta, lo que dificulta la representación de zonas abiertas de gran extensión.</li> <li>• No existe ningún criterio formal para la definición de nodos y las conexiones entre ellos.</li> <li>• Los elementos presentes y no modelados inicialmente pueden provocar que el robot pierda su localización.</li> </ul>

Cuadro 2.2: Ventajas y desventajas de los mapas topológicos.

ductor ultrasónico de Polaroid©, consiste en representar el cono de alcance del sonar por un campo de visión  $\beta$  y un alcance máximo de detección de obstáculos  $R$ . De esta manera, si se cuenta con el modelo del sensor y una rejilla de ocupación, el campo de visión del sensor puede ser proyectado sobre la rejilla y actualizar las celdas dentro del alcance del sensor. El campo de visión es dividido en las siguientes tres regiones:

- **Región I.** La probabilidad de ocupación de las celdas es mayor.
- **Región II.** La probabilidad de que las celdas estén desocupadas es mayor.
- **Región III.** Se desconoce el estado.

Dichas regiones se encuentran representadas en la Figura 2.3

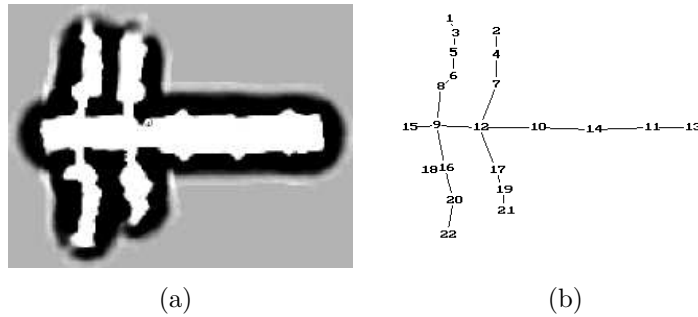


Figura 2.2: A partir de un mapa basado en rejilla de ocupación (a) se obtiene un mapa topológico (b) Thrun et al. (2005)

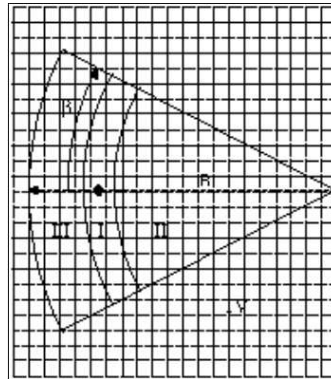


Figura 2.3: Modelo del sonar propuesto por Murphy (2000). La punta del cono representa el punto de emisión del sonar.

La probabilidad de no ocupación de las celdas de la **Región I**, al estar más cercanas al origen del sonido, será mayor a la probabilidad de que las celdas que pertenecen a la **Región II** se encuentren ocupadas. Adicionalmente, las probabilidades anteriores disminuirán conforme las celdas estén más alejadas de la mitad del cono del sonar, llamado eje acústico.

De esta manera, puede generarse una función de probabilidad para el modelo del sonar, de la siguiente forma:

Para cada elemento dentro de la Región I:

$$P(s|ocupado) = \frac{\left(\frac{R-r}{R} + \frac{\beta-\alpha}{\beta}\right)}{2} \times Max_{occupied} \quad (2.1)$$

$$P(s|no\ ocupado) = 1 - P(s|ocupado) \quad (2.2)$$

donde  $R$  representa el alcance máximo de detección de obstáculos del sonar modelado,  $r$  la distancia a la celda a ser actualizada,  $\beta$  el campo de visión del sonar y  $\alpha$  el ángulo al cual se encuentra la celda a ser actualizada con respecto al eje acústico.

## 2.4. Localización

En diversas tareas que realizan los robots móviles, estos deben ser capaces de determinar su posición ya sea para realizar planificación de trayectorias, o bien para determinar si el lugar en el que se encuentran es adecuado para ejecutar ciertas operaciones.

**Definición 6.** *La localización de un robot móvil es el problema de determinar la posición relativa de un robot respecto a un mapa de un ambiente, dadas las percepciones que se tienen del ambiente y los movimientos del robot.*

A menudo, a la localización también se le conoce como estimación de la posición. La localización se encuentra ligada al problema de generación de mapas, debido a que si lo que se desea es tener un mapa preciso de un ambiente, es necesario conocer la posición del robot. Un ejemplo del problema que existe si un robot no es capaz de localizarse dentro del mapa de un ambiente se puede apreciar en la figura 2.4. En la figura 2.4(a) se muestra un mapa creado a partir de la información cruda de los sensores del robot, al cual se le aplicarán específica y únicamente la información de los odómetros que miden la distancia recorrida por el robot, la cual es empleada para localizarse. Por otro lado, en la Figura 2.4(b) se muestra el mismo mapa al cual se le aplicaron diversas técnicas de localización las cuales permiten corregir la posición del robot y de esta manera generar un mapa global coherente .

La localización puede ser “fuerte” o “débil”. La localización “fuerte” involucra estimar la posición del robot, ya sea en términos cuantitativos o cualitativos, con respecto a algún tipo de representación global del ambiente. Por otro, lado la localización “débil” involucra únicamente el conocer si la posición actual del robot ha sido o no visitada anteriormente.

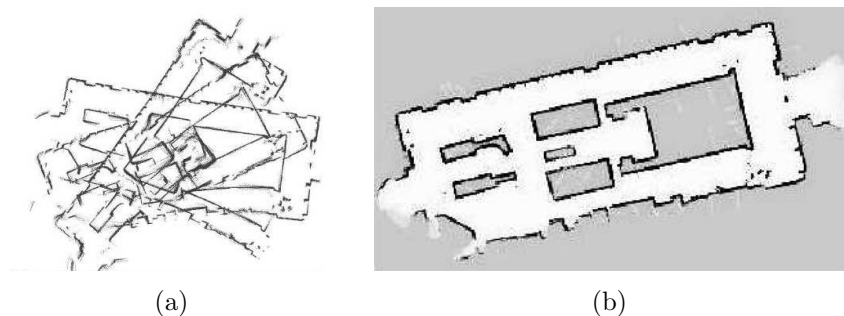


Figura 2.4: (a) Mapa construido a partir de la información odométrica (b) Mapa del mismo ambiente construido tomando en cuenta la corrección de la posición del robot.

Thrun et al. (2005) identifican tres tipos de problemas de localización dependiendo de la información que se tenga disponible al inicio y durante el proceso de localización: seguimiento de la posición, localización global y el problema del robot “secuestrado”. A continuación se describen brevemente dichos problemas y los métodos propuestos para solucionarlos.

### 2.4.1. Localización local

En el problema de localización local, también conocido como seguimiento de la posición, se asume que la posición inicial del robot es conocida. Empleando las observaciones del ambiente se puede realizar un seguimiento del estado del robot, el problema radica en mantener constantemente su localización en el mapa a pesar de los errores en la odometría que por lo general son acumulables, i.e. conforme el robot se mueve los errores se incrementan de manera acumulativa y son mayores para las rotaciones. Si se emplea la odometría, la posición es estimada integrando los movimientos, e.g. traslación y rotación, desde una posición conocida. Para solucionar este problema puede utilizarse la odometría en combinación con información de sensores externos. Los métodos para resolver el problema de la localización local son los siguientes:

1. **Métodos basados en marcas en el ambiente (*landmarks*)**. La localización del robot se realiza con ayuda de marcas en el ambiente cuya posición es conocida *a priori* y a partir de las cuales, se puede corregir la posición del robot. Bajo una definición genérica, casi todos

los métodos de localización pueden verse como basados en marcas. Las marcas pueden ser artificiales o naturales. Las marcas artificiales pueden ser colocadas en lugares estratégicos del ambiente, de modo que el robot al localizarlas puede corregir su posición. El problema radica principalmente en que si la marca se encuentra obstruida, el robot no podrá localizarse. Por otra parte, las marcas naturales son aquellas que no son colocadas explícitamente a la intención del robot, sino que se basan en características del ambiente, e.g. esquinas, pasillos, puertas, etc. En la Figura 2.5 se puede ver un esquema de localización basado en marcas en el ambiente, en este caso el robot emplea seis marcas en el ambiente para localizarse.

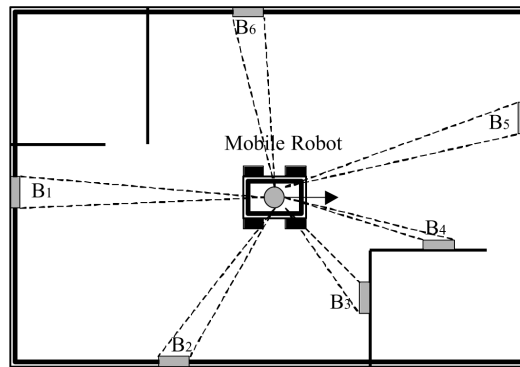


Figura 2.5: Ejemplo de localización basado en las marcas  $B_1$ ,  $B_2$ ,  $B_3$ ,  $B_4$ ,  $B_5$  y  $B_6$  en un ambiente realizado por Hu et Gu (2000).

2. **Métodos basados en correlación de características.** Primeramente se estiman características locales basándose en las últimas observaciones del ambiente. Con base en las características estimadas, se encuentra la mejor correlación a la posición actual del robot maximizando la correlación entre las características locales y el mapa del ambiente. Romero et al. (2006) emplean el método de correlación para estimar la posición de un robot móvil asumiendo que los mapas generados antes y después de pequeños desplazamientos son similares. De esta manera se puede calcular la orientación  $\theta$  del robot (ver Figura 2.6 (a)) y el desplazamiento  $x, y$  (ver Figura 2.6(b)) rotando o trasladando el mapa local generado hasta encontrar la máxima correlación con el mapa previamente almacenado.

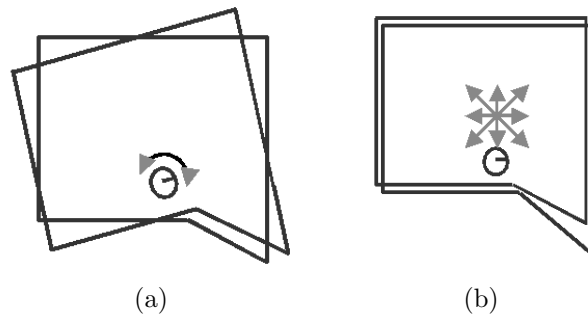


Figura 2.6: Método de correlación para la estimación de (a) la orientación  $\theta$  y (b) el desplazamiento  $x, y$  de un robot móvil empleado por Romero et al. (2006).

3. **Métodos basados en el filtro Kalman.** Se estima la posición del robot en el entorno mediante una distribución normal. La covarianza de esta distribución representa la incertidumbre local de la posición estimada. Cada vez que el robot se mueve, su posición es actualizada con base en las lecturas de odometría. Las observaciones hechas a partir de los sensores se emplean para actualizar la distribución de probabilidad de la localización, buscando la nueva distribución que maximiza la verosimilitud de las lecturas.

### 2.4.2. Localización global

En la localización global la pose inicial del robot es desconocida y los errores en la odometría pueden afectar la localización. Tradicionalmente se ha propuesto que para solucionar el problema de la localización global son necesarias técnicas basadas en búsquedas, i.e. se crea un mapa local del ambiente y se compara con distintas zonas de un mapa global para encontrar la zona que más se asemeje al mapa local. Esto se debe a que, dadas las lecturas de los sensores, normalmente existen varias posibles localizaciones para el robot correspondiente a las mismas lecturas. El enfoque de búsquedas fue abandonado por la comunidad de robótica móvil y el problema de localización global ha permanecido sin solución hasta la utilización recientemente, de enfoques bayesianos que ofrecen alternativas promisorias de solución (Thrun (2001)).

### 2.4.3. El problema del robot “secuestrado”

El problema de robot “secuestrado” es una variante más complicada del problema de localización global. Durante su operación, el robot puede ser “secuestrado” y transportado a otra ubicación en el ambiente, por lo que estima saber dónde está cuando en realidad su ubicación ha variado. Thrun et al. (2005) argumentan que la importancia del problema del robot “secuestrado” radica en que ninguno de los métodos de localización existentes es infalible, por lo que la capacidad de recuperarse de fallas es esencial para lograr una autonomía real.

Como se mencionó al inicio de esta sección, la localización se basa en un mapa del ambiente, los dos tipos de mapas comúnmente empleados son los mapas basados en rejillas de ocupación y los mapas topológicos descritos en la sección 2.1. A continuación veremos cómo son empleados estos dos tipos de mapas en la localización.

### 2.4.4. Localización basada en mapas

Un mapa métrico del ambiente permite que los algoritmos de localización determinen la posición absoluta del robot, i.e. coordenada  $x,y$  y orientación  $\theta$  del robot. Un modelo topológico por su parte, permite la localización de forma no tan exacta, e.g. al inicio pasillo o final del pasillo. En estos casos bastará con saber que el robot se encuentra, por ejemplo, en un pasillo y no exactamente en qué lugar del pasillo.

Cuando un robot emplea mapas basados en rejillas de ocupación, puede emplear la información que le proporciona dicho mapa junto con la información de su posición geométrica dentro de un marco de referencia global para distinguir entre diferentes lugares del ambiente. La posición del robot es estimada de manera incremental, basándose en la información de sus odómetros y en las lecturas de otros sensores, e.g. sonares o láseres. Por lo tanto, se requiere un gran número de lecturas de los sensores para determinar la localización del robot.

Si por el contrario, se emplean mapas topológicos, la localización del robot por lo general se realiza empleando métodos basados en marcas en el ambiente. El problema que existe en este tipo de esquemas de localización se debe a que, si el robot atraviesa por dos lugares similares, i.e. indistinguibles a partir de las lecturas de los sensores del robot y no dispone de marcas o no las ha percibido recientemente, será incapaz de determinar si dichos lugares

son o no el mismo. Sin embargo, debido a que los métodos basados en mapas topológicos no requieren determinar la posición geométrica exacta del robot, a menudo son utilizados por ser menos sensibles a los errores de odometría.

## 2.5. Incertidumbre en la localización

Actualmente los robots son empleados en numerosas aplicaciones que van desde el hogar, la industria, la exploración de lugares peligrosos, hasta misiones espaciales. Un robot percibe su ambiente mediante sensores, los cuales tienen como característica ser susceptibles al ruido en sus lecturas. Además, en mayor o en menor grado el robot interactúa con el mundo real, el cual tiene un alto grado de incertidumbre. Es importante mencionar esto ya que el hecho de que el mundo real sea parcialmente observable y presente incertidumbre, origina la mayor parte de los problemas en robótica.

Thrun et al. (2005) identifican cinco factores principales que contribuyen a la incertidumbre en la adquisición de datos y ejecución de acciones en los robots:

1. **El ambiente del robot.** Los robots industriales por lo regular operan en ambientes estructurados y controlados. Sin embargo, los robots que ejecutan tareas en ambientes humanos, tienen que tomar en cuenta que los modelos que se tienen de dichos ambientes pueden variar sobre todo por la presencia de personas.
2. **Los sensores.** Los sensores son dispositivos electrónicos, y por lo tanto están sujetos a ciertas limitaciones físicas. Además, dependiendo de su tipo, los sensores presentan ruido que es difícil de modelar y que provoca que la información que se obtenga de ellos sea limitada e imprecisa.
3. **La actuación del robot.** El robot ejecuta acciones en su ambiente mediante efectores o activadores, los cuales tampoco están exentos de incertidumbre, i.e. las acciones no se ejecutan tal y como se programaron. La incertidumbre en los efectores puede deberse a ruido en el control, a fallas mecánicas o eléctricas, o a factores externos del ambiente.
4. **Modelos internos.** Los modelos son abstracciones del mundo real y por lo tanto son aproximaciones de éstos.

5. **Algoritmos aproximados.** Los robots son sistemas que suelen operar en tiempo real, en cuyo caso, sus cálculos deben realizarse en el menor tiempo posible. Lo anterior provoca que muchos algoritmos sean simplificados sacrificando precisión.

El grado de incertidumbre depende de la aplicación. Sin embargo, como se verá en la siguiente sección, la incertidumbre aumenta cuando se tiene un sistema multirobot en lugar de un sistema monorobot.

# Capítulo 3

## Estado del Arte

### 3.1. Problemática

El problema de la construcción de mapas por un robot, ha sido abordado por los investigadores en robótica móvil con especial interés en los últimos 20 años. Mientras que en la versión para un único robot las técnicas para la construcción de mapas están bien estudiadas, en los sistemas multirobot surgen nuevos problemas los cuales se describen a continuación.

**Coordinación.** Incrementar la eficiencia es una de las principales razones para desarrollar sistemas multirobot en lugar de emplear un único robot. Sin embargo, conforme el número de robots crece, también crece la complejidad para coordinarlos. La tarea de coordinación se puede facilitar si los robots conocen sus posiciones relativas y además comparten un mapa del ambiente que será explorado. En este caso, la coordinación se puede lograr guiando a los robots a distintas áreas del ambiente que no se traslapen. No obstante, la tarea se dificulta si los robots no comparten un mapa o marco de referencia.

**Fusión del mapa.** Para generar un modelo consistente del ambiente, cada uno de los mapas locales creados de manera individual por los miembros del sistema multirobot debe ser fusionado en un único mapa global coherente. La fusión es una tarea compleja ya que involucra tareas diversas como la localización de los robots. Una integración consistente de los datos, cuando no se conocen las poses relativas de los robots, es difícil debido a que no es claro cómo se deben complementar.

**Limitantes en la comunicación.** La comunicación en los sistemas multirobot se puede tratar de dos maneras: comunicación entre los robots miem-

bros del sistema, o bien comunicación de los robots a una estación central. No obstante, cualquiera de los dos esquemas de comunicación anteriores puede fallar. Una forma de conseguir una comunicación eficiente fue propuesta por Konolige et al. (2002) y consiste en considerar a cada uno de los robots del sistema multirobot como un nodo de una red. De esta manera, la información es transmitida a cada uno de los nodos que se puedan comunicar directamente. Un nodo puede funcionar como un *ruteador* que establece comunicación entre nodos que no se pueden comunicar. La comunicación en este caso es más robusta debido a que no se depende de un nodo central, por lo que si un nodo se cae, los demás seguirán funcionando.

## 3.2. Trabajos relacionados

En los últimos años se han realizado diversas investigaciones en las tareas de localización, creación de mapas y exploración de ambientes interiores empleando un único robot autónomo; sin embargo, su aplicación a sistemas multirobot es relativamente nueva. Adicionalmente, muchas investigaciones en el área de sistemas multirobot se pueden ver como extensiones de algoritmos existentes desarrollados para un único robot, en lugar de emplear un nuevo algoritmo fundamentalmente distribuido.

Las investigaciones en los sistemas multirobot se han enfocado en los siguientes puntos: coordinación de los robots, estimaciones de las posiciones relativas y creación de mapas, los cuales se describen a continuación.

### 3.2.1. Coordinación de los robots

El empleo de sistemas multirobot para la construcción de mapas nos proporciona la ventaja de construir el mapa en menor tiempo. En el mejor de los casos, en teoría, el tiempo consumido para la construcción del mapa es de  $1/2 N$  del tiempo empleado por un único robot. En la práctica sin embargo, el tiempo es sublineal.

Una técnica empleada para la coordinación de robots enfocada a la exploración de ambientes, es conocida como técnica basada en fronteras (Thrun (2001); Fox et al. (2005); Howard et al. (2004)). Dicha técnica consiste en asignar a los robots áreas llamadas fronteras (ver Figura 3.1), las cuales representan límites entre las áreas exploradas y las no exploradas. El método basado en fronteras propuesto por Yamauchi (1997) consiste en construir un

mapa del ambiente de manera incremental y después analizarlo para encontrar las áreas frontera. Una vez encontrada un área frontera, un robot es asignado a dicha área para su exploración. Howard et Mataric (2002) proponen un método muy similar al anterior; sin embargo, para que un robot sea asignado a una zona del ambiente, debe haber un robot cercano el cual le sea visible al primer robot. Son pocas las situaciones en las cuales se puede implementar esta técnica debido a que, para construir un único mapa global consistente, se deben conocer las posiciones relativas de cada uno de los robots y además, los robots deben comenzar sus operaciones en zonas cercanas unos con otros.

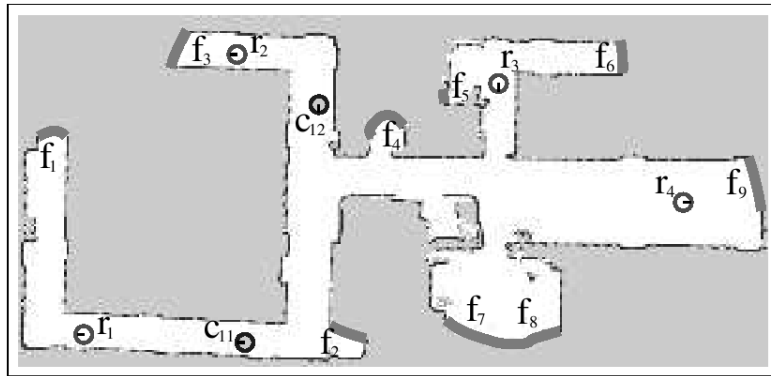


Figura 3.1: Coordinación de robots  $r_1, r_2, r_3, r_4, c_{11}$  y  $c_{12}$  basada en fronteras  $f_1-f_9$  (Fox et al. (2005)).

Stroupe et Balch (2002) proponen un método para la coordinación de un sistema multirobot, con el propósito de lograr una integración de las observaciones colaborativas y localización. Las tareas que los miembros del sistema pueden realizar son agrupadas en “misiones”. Para que un robot seleccione una tarea a ejecutar en una misión, debe evaluar la prioridad de todas las tareas existentes para dicha misión, además de las necesidades, capacidades y las tareas que los demás miembros del sistema han seleccionado. Las prioridades están definidas en función de todas las tareas, así como de la posición del robot y de la capacidad del sistema para completar con éxito dicha tarea. Por ejemplo, la tarea de realizar un mapa tiene una prioridad alta cuando la incertidumbre en la estimación de la posición del robot es baja. Lo anterior debe a que si la incertidumbre en la posición de un robot es alta, se empleará más tiempo de procesamiento en crear un mapa del ambiente y los

errores pueden ser mayores.

Parker et al. (2002) proponen que inicialmente cada robot planifique de manera independiente una ruta a seguir. Las rutas son compartidas entre todos los miembros del sistema. De esta manera se tiene una visión global de las rutas que cada uno de los robots recorrerá para completar una tarea. Cuando las rutas de dos o más robots coinciden en un punto en el mismo instante de tiempo, se agregan tiempos de espera en algunas rutas con la finalidad de evitar colisiones entre los robots. Si después de esto aún existen colisiones, los lugares donde los robots colisionan son marcados como obstáculos y se planea una nueva ruta. La planificación de trayectorias únicamente fue probada en ambientes de simulación de robots en 3D.

Otras investigaciones se han orientado a coordinar robots heterogéneos. La idea de tener un sistema multirobot heterogéneo es tener robots con distintas capacidades y equipamiento, por lo que se pueden tener robots especializados en ciertas tareas. Howard et al. (2004) emplean dos tipos de robots, los primeros con capacidades de sensado limitadas, i.e únicamente equipados con un micrófono y una cámara, mientras que los robots del segundo tipo disponen de un mayor equipamiento, i.e. robots con láser y cámara. Los robots con mayor equipamiento son asignados como líderes dentro del sistema multirobot. De esta manera un robot líder es encargado de guiar a otros robots a través de obstáculos hasta su posición correspondiente. El método es como sigue: los robots se forman en cadena detrás de un robot líder. Después, los robots formados siguen al robot líder el cual se dirige a cada una de las zonas del ambiente en las que se deben colocar los robots. En cada una de las zonas, los robots seguidores toman la posición que les fue previamente asignada (ver Figura 3.2).

### **3.2.2. Estimación de la posición**

La localización es el proceso de estimar la posición de un robot móvil dado un mapa del ambiente. Dicha localización se realiza empleando la información proveniente de las observaciones del ambiente que el robot ha realizado junto con la información de su odometría. Las investigaciones para lograr la localización de un robot se centran generalmente en dos enfoques distintos, dependiendo de si se conoce la posición inicial del robot (localización local) o no (localización global). Para estimar la posición relativa de un sistema multirobot, Howard et al. (2002) desarrollaron un sistema multirobot en el que los miembros del sistema tienen la habilidad de detectarse entre sí. En este



Figura 3.2: Coordinación de los robots mediante un robot líder (Howard et al. (2004)).

enfoque, todos los miembros del sistema exploran de manera independiente el ambiente hasta que detectan a otros robots. Una vez que dos robots se detectan, intercambian información y calculan sus posiciones relativas. Esta técnica es conocida como *localización cooperativa* y es empleada en otros trabajos como en el desarrollado por Stroupe et Balch (2002).

Para la detección de los robots se pueden emplear diversos sensores como sonares, láseres y cámaras. La tarea de estimar la posición de un robot respecto a la posición de otro robot, se ha resuelto determinando la posición relativa de un robot en un mapa construido por otro robot y empleado el método de Monte Carlo para resolver el problema de localización global. No obstante, la principal restricción del método de Monte Carlo es asumir que el robot que se desea localizar debe encontrarse dentro del ambiente correspondiente al mapa creado por otro robot. El problema ahí radica en que los robots pueden comenzar su operación en distintas zonas desconocidas del ambiente, las cuales no se traslapan y por lo tanto, ningún robot podrá estar dentro de la zona correspondiente al mapa de otro robot.

Fox et al. (2005) demostraron que si un robot se encuentra dentro de la zona del ambiente correspondiente al mapa construido por otro robot, la localización de Monte Carlo puede ser aplicada. Lo primero que se realiza en el método que los autores propusieron, es asignar un robot como líder que será el encargado de crear un mapa parcial inicial. Un segundo robot es colocado en una posición desconocida en la zona del ambiente de la cual el robot líder creó el mapa. Por último, el robot líder estima la posición del

segundo robot mediante un filtro de partícula.

Rekleitis et al. (2002) centran su trabajo en las tareas de construcción de mapas y la estimación de la posición empleando un sistema multirobot utilizando localización cooperativa. Se realiza un análisis del efecto que tiene el tamaño del sistema multirobot en la precisión de la localización; sin embargo, los experimentos no se realizan con robots físicos sino únicamente en experimentos en simulación. Los sensores simulados pueden proporcionar distintos tipos de información como la distancia entre los robots y su posición relativa. Los experimentos en simulación fueron divididos en tres tipos de acuerdo a la información empleada para estimar la localización de los robots. Dichos experimentos fueron los siguientes:

- *Empleando únicamente la distancia entre los robots, medida a partir de la información de un sonar.* Para estimar la posición de un robot en movimiento, al menos son requeridos dos robots que permanezcan estáticos, por lo que el tamaño mínimo de robots en el sistema para este experimento en particular fue de tres robots.
- *Empleando únicamente la información de Azimuth (ángulo).* Se realizaron pruebas utilizando únicamente el ángulo al cual un robot es observado, empleando sistemas multirobot de distintos tamaños. En estos experimentos no existió mejora en el cálculo de la posición del robot conforme el número de robots aumentó. Esto se debió al error existente en la estimación de la orientación de los robots que se encuentran estacionarios. Para solucionar este problema se emplearon filtros partícula para estimar la posición del robot observado.
- *Empleando únicamente posición.* El tercer grupo de pruebas consistió en calcular la posición relativa de un robot. i.e. la posición de un robot estimada en el marco de referencia de otro robot. En estos experimentos las distancias y ángulos entre los robots son combinados para estimar la posición de otro robot.
- *Empleando una combinación de los tres parámetros anteriores.* Algunos sensores de seguimiento proporcionan información acerca de la posición, orientación y distancia a la que se encuentra un robot. Debido a esto, fueron simulados este tipo de sensores agregando ruido gaussiano.

Después de realizar diversos conjuntos de pruebas, los autores concluyeron que la exactitud en la estimación de la posición relativa de los robots aumentaba conforme el número de robots también aumentaba.

Las técnicas para la estimación de la posición varían dependiendo del tipo de ambiente sobre el que se trabaja. En el caso de ambientes exteriores se puede emplear un GPS para obtener una estimación de la posición. Sin embargo, la información proporcionada por dichos sensores puede ser incompleta o inexacta. Parker et al. (2002) estiman la posición de los robots de dos maneras, empleando un barrido láser y aplicando en enfoque de localización cooperativa basada en visión. Debido a que en ambientes exteriores los errores introducidos a consecuencia de las grandes distancias recorridas son significativos e impredecibles, se emplea un Filtro Kalman Extendido (EKF) para la localización de un sistema multirobot heterogéneo.

Si bien, para el caso de ambientes exteriores existen sensores que pueden proporcionar la posición de los robots con un alto grado de certeza, en el caso de ambientes interiores no existen sensores con dichas capacidades. Debido a esto los robots deben ser capaces de manejar la incertidumbre en sus sensores para la estimación de sus posiciones relativas. Howard et Mataric (2002) emplean una técnica basada en el cálculo de la **máxima verosimilitud** (*a maximum likelihood approach*). Para resolver el problema de la localización en un sistema multirobot se asume que cada uno de los miembros del sistema está equipado con dos tipos de mecanismos llamados “detector de robots” y “detector de movimientos”. El “detector de robots” permite a un robot estimar la posición relativa de los robots cercanos a su ubicación, además de determinar su identidad. Por otro lado, el “detector de movimientos” es un sensor propioceptivo que mide los cambios en la posición del robot. Adicionalmente, también se asume que la tarea de determinar la identidad de los robots se realiza siempre de manera correcta.

El método de localización empleado por Howard et Mataric (2002) es como sigue. Primero se construye un conjunto de estimaciones  $H$  donde cada elemento  $h \in H$  es una estimación de la posición de un robot en particular en un instante de tiempo  $t$ . La estimación de la posición es definida con respecto a un sistema global de coordenadas arbitrario. Después, se considera un conjunto de observaciones  $O$  donde cada elemento  $o \in O$  representa una medición de la posición relativa de un robot en particular, empleando el “detector de movimientos” o el “detector de robots”. En caso de emplearse el detector de movimientos, cada observación representa la medida del cambio en la posición del robot. Por otra parte, si se emplea el detector de robots

cada observación representa la medida del cambio en la posición de un robot respecto a otro robot. Finalmente, se emplea una optimización numérica para determinar el conjunto de estimaciones  $H$  que más se parece al conjunto de observaciones  $O$ . El algoritmo propuesto puede ser implementado de forma centralizada o distribuida. En el enfoque centralizado las observaciones hechas por los robots son enviadas a una estación central donde un algoritmo de optimización eficiente es aplicado a los datos para obtener la mejor estimación de la posición. Después, las estimaciones de las posiciones son enviadas a cada uno de los robots de manera individual. Por otra parte, en el enfoque distribuido cada robot es encargado de generar una optimización de un conjunto de observaciones y periódicamente intercambia las estimaciones de su posición con las estimaciones de las posiciones de otros robots.

### 3.2.3. Fusión de mapas

La creación del mapa que representa un ambiente se puede realizar de manera más rápida empleando un sistema multirobot en donde cada miembro es encargado de explorar cierta zona del ambiente y crear el mapa correspondiente. Sin embargo, la integración de los datos obtenidos por cada uno de los robots de manera separada es una tarea compleja, cuanto más parecidas sean las zonas del ambiente. Si las zonas del ambiente no son lo suficientemente diferentes entre sí, la asociación de los datos correspondientes a dichas zonas puede dar lugar a un mapa global no coherente como el que se muestra en la figura 3.3



Figura 3.3: Mapa global fusionado con problemas en la integración de mapas locales (Birk et Carpin (2006)). Los problemas se aprecian en la no coincidencia del pasillo central.

El objetivo de la fusión de mapas es identificar zonas que se traslapen y de esta manera unir los mapas locales correspondientes. Los pocos trabajos reportados se centran en métodos basados en características (Huang et Beevers

(2005)). Dichos métodos consisten en identificar marcas fijas en el ambiente (*landmarks*), de esta manera resulta más sencilla la tarea de fusión de mapas locales debido a que se cuenta con puntos de referencia para la fusión de los mismos. Los mapas que por lo general son usados en este tipo de trabajos son mapas topológicos. El trabajo presentado por Birk consiste en unir mapas locales basados en rejillas de ocupación. Las rejillas de ocupación son empleadas comúnmente para representar ambientes no estructurados en los cuales resulta difícil encontrar características. Las rejillas de ocupación en este caso pueden ser tratadas como imágenes en las que la probabilidad de ocupación es representada por un color. En dicho caso, la fusión de los mapas locales puede realizarse aplicando operaciones geométricas a las imágenes que los representan hasta encontrar la máxima similaridad entre las imágenes. Para encontrar la máxima similaridad entre las imágenes suelen emplearse algún tipo de correlación o la distancia Euclidiana entre los píxeles del mismo color.

#### **3.2.4. Estrategias para la generación de mapas**

En general, para que un robot pueda crear un mapa de su ambiente de manera eficiente, se requiere hacer uso de estrategias adecuadas para la exploración de dicho ambiente. En particular, es útil conocer qué área es más convenientes recorrer y cómo se deben distribuir los robots en el ambiente con la finalidad de abarcar la mayor extensión posible del ambiente previamente conocido.

La mayor parte de las investigaciones que conciernen la creación de mapas se centran en un sólo robot. Los modelos y métodos propuestos para resolver el problema de creación de mapas deben abordar ciertas limitaciones y restricciones impuestas por la naturaleza de los ambientes del robot y los sensores empleados. Los sensores presentan problemas en cuanto al rango de percepción, el cual limita la tarea de sensado a ciertas áreas alrededor del robot. En cuanto al ambiente del robot, éste suele ser dinámico en diversas aplicaciones reales, por lo que implica que dicho ambiente puede resultar difícil de modelar al menos con las estructuras y heurísticas de representación hasta ahora propuestas.



# Capítulo 4

## SMR para la exploración colectiva de un ambiente

En este capítulo se presentan las especificaciones, diseño, construcción y comportamiento de un sistema multirobot con la capacidad de crear y fusionar mapas locales en un único mapa global. El sistema multirobot consta de tres miembros homogéneos y autónomos, con capacidades de navegación en un ambiente semi-estructurado, comunicación inalámbrica con otros miembros del sistema y con un servidor externo, y construcción de mapas métricos locales. Se presentan a continuación los requerimientos del sistema multirobot.

### 4.1. Requerimientos de sistema

En esta sección se presentan los requerimientos del sistema multirobot, así como una taxonomía del tipo de ambientes que se emplearán para evaluar su funcionamiento. Finalmente se presenta una descripción de la organización de los robots del sistema.

#### 4.1.1. Especificación de los tipos de ambientes a explorar por el sistema multirobot

Las características de los ambientes en los cuales el sistema multirobot efectuará la tarea de “mapeo” son las siguientes:

1. Ambiente con un único nivel.

2. Un pasillo principal o distribuidor.
3.  $N$  áreas disjuntas, i.e. no interconectadas entre ellas, cada una de las cuales con un único acceso al pasillo principal.
4. Una sola entrada al ambiente, conectada directamente al distribuidor.

Con base en las especificaciones anteriores podemos agrupar los ambientes de pruebas en tres grupos:

- **Tipo 1.** Ambientes con pasillo simple.
- **Tipo 2.** Ambientes un distribuidor y al menos una bifurcación.
- **Tipo 3.** Ambientes con un distribuidor con ciclos

A continuación se describen éstos tres tipos de ambientes y se ilustran con ejemplos.

**Tipo 1. Ambientes con distribuidor simple.**

Están formados por un distribuidor principal y  $n$  áreas disjuntas con  $n \geq 0$ . Ejemplos de estos ambientes se ilustran en la figura 4.1.

**Tipo 2. Ambientes con un distribuidor con al menos una bifurcación.**

Ambientes formados por  $n$  áreas disjuntas, con  $n \geq 0$ , y un distribuidor que puede subdividirse en dos distribuidores distintos que no se vuelven a conectar, los distribuidores formados por las divisiones pueden a su vez subdividirse nuevamente. En la Figura 4.2. se pueden ver ejemplos de éste tipo de ambientes

**Tipo 3. Ambientes con un distribuidor con ciclos.** Tipo de ambiente similar al tipo 2 con bifurcaciones; sin embargo, en este caso las bifurcaciones se vuelven a conectar formando un ciclo. Ejemplos de este tipo de ambientes se muestran en ella Figura 4.3.

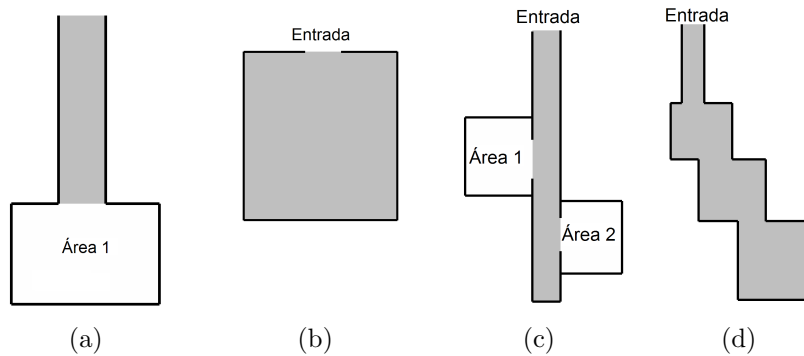


Figura 4.1: Ejemplos de ambientes con distribuidor simple.

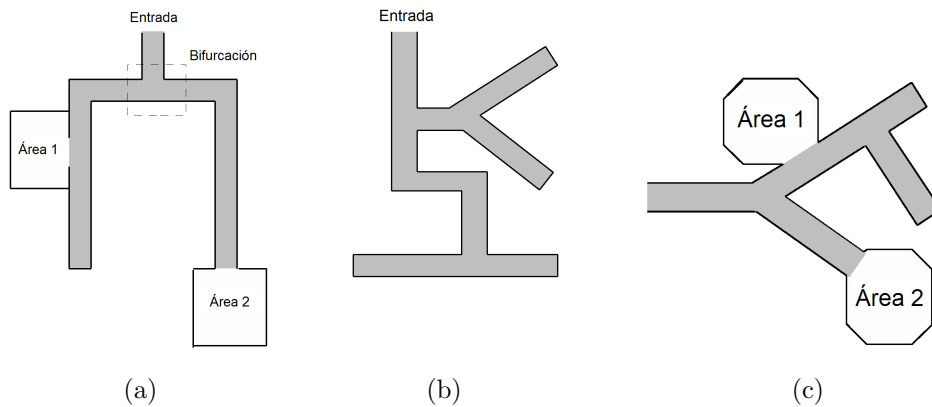


Figura 4.2: Ejemplos de ambientes con distribuidor con al menos una bifurcación.

En esta tesis se abordarán los ambientes del tipo 2 hasta con tres nodos terminales.

## 4.2. Diseño y equipamiento de los robots

Con base en los requerimientos establecidos en la definición del problema (c.f. sección 1.2), se diseñaron y equiparon tres robots móviles funcionales, construidos a partir de una plataforma comercial basada en control diferencial, los cuales se muestran en la Figura 4.4. Cada uno de los robots fue equipado con los siguientes sensores:

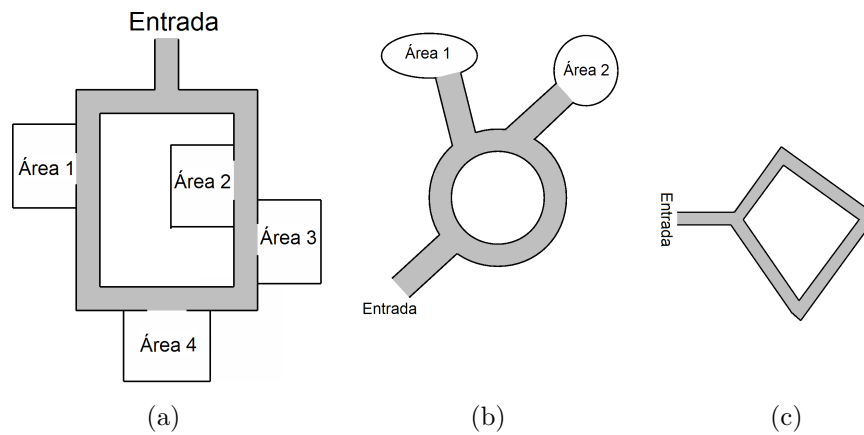


Figura 4.3: Ambientes con distribuidor con ciclo.

- Un dispositivo de sensado consistente de un sonar montado sobre un servomotor para realizar un “barrido” de la zona del ambiente de la cual se desea realizar o actualizar un mapa.
- Un sensor de distancia infrarrojo para calcular la distancia a las paredes.
- Dos odómetros montados en las ruedas del robot para calcular su desplazamiento.

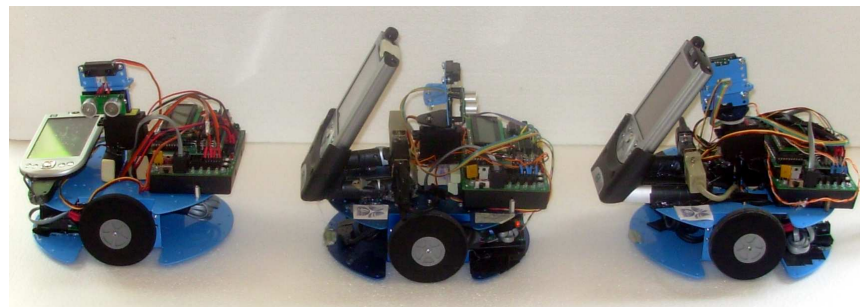


Figura 4.4: Los tres robots miembros del Sistema Multirobot.

Se caracterizó cada uno de estos sensores para determinar, entre otros, su rango de acción y sensibilidad, en distintas condiciones experimentales. Así mismo, para el control del robot y la construcción del mapa, se equipó al robot con los siguientes módulos de control:

- Módulo de Bajo Nivel.
- Módulo de Alto Nivel.

A continuación se describe cada uno de estos módulos.

**Módulo de Bajo Nivel.** Se encuentra formado por una tarjeta electrónica comercial Handyboard<sup>®</sup>, la cual se encarga de controlar tanto los actuadores del robot, como de recabar información de los sensores como el sonar y los odómetros. Las tareas que el módulo de bajo nivel ejecuta responden a las solicitudes hechas por el módulo de alto nivel bajo un esquema de servidor-esclavo en donde el módulo de bajo nivel es el esclavo, y el servidor el módulo de alto nivel. Las tareas ejecutadas por este módulo son:

- Lectura de los sensores de distancia, tanto del sonar como del sensor infrarrojo y envío de dicha información al módulo de alto nivel.
- Ejecución de las habilidades del robot tales como avanzar, girar a la derecha, girar a la izquierda, detectar pared y seguir pared. Dichas habilidades se detallarán en la siguiente sección.

**Módulo de Alto Nivel.** Este módulo está conformado por una Pocket PC <sup>®</sup> la cual se conecta por medio de un puerto serie al módulo de bajo nivel. El módulo de alto nivel se encarga de las siguientes tareas:

- Establecimiento de la comunicación con un servidor y con otros miembros del sistema multirobot.
- Envío de comandos al módulo de bajo nivel para el control del robot.
- Adquisición de información de los sensores del robot recibidos por parte del módulo de bajo nivel para la toma de decisiones.
- Almacenamiento y envío de información de los sensores al robot servidor.

En la Figura 4.5 se pueden apreciar todos los componentes del prototipo de robot. En la Figura 4.6 se presentan los módulos de alto y bajo nivel así como el flujo de comunicación entre estos módulos y los distintos dispositivos del robot. Como se puede apreciar en dicha figura, el módulo de bajo nivel es el único elemento de control que tiene comunicación directa con los sensores y actuadores del robot.

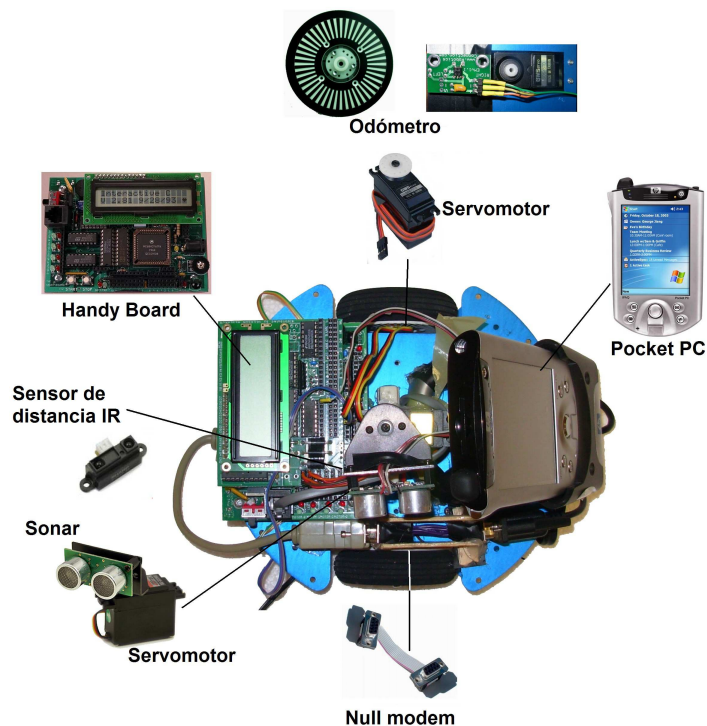


Figura 4.5: Equipamiento de los prototipos de robot.

### 4.3. Especificación del Sistema

#### 4.3.1. Especificación general

En esta tesis definimos a un sistema multirobot **SMR** como una tupla compuesta por un grupo de robots **GR**, un servidor **S** y un ambiente **A** en el cual los miembros del grupo de robots se encuentran inmersos. Esto es:

$$SMR = \langle GR, A, S \rangle$$

donde:

$$S = \{ Cs \}$$

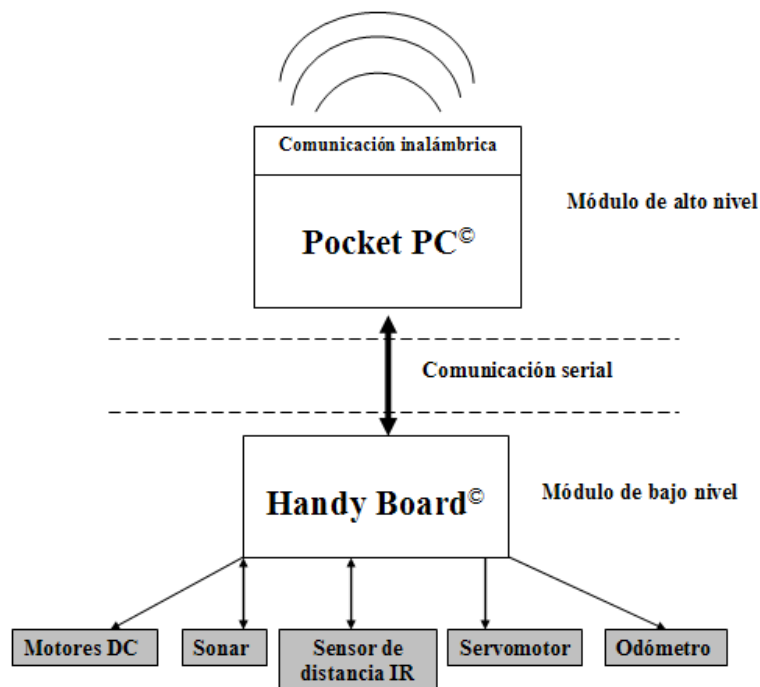


Figura 4.6: Organización interna de los prototipos de robot.

$$\mathbf{GR} = \{R_1, R_2, \dots, R_n\} \text{ y } R_i = \{Ca, Co\}$$

y para los cuales:

$\mathbf{Cs}$  denota las capacidades del servidor.  $R_i$  denota a cada uno de robots miembros el sistema, con  $1 \leq i \leq n$ .

$\mathbf{Ca}$  denota el conjunto de habilidades individuales para todo robot del sistema.

$\mathbf{Co}$  denota el conjunto de comportamientos que un robot miembro del sistema puede ejecutar de manera individual o colectiva., aplicando elementos de  $\mathbf{Ca}$  y otros  $\mathbf{Co}$ .

**A** es el ambiente del robot definido por dos tipos de mapas, un mapa topológico y un mapa métrico. Esto es:

$$\mathbf{A} = \langle \mathbf{MT}, \mathbf{MM} \rangle$$

donde:

**MM** = Mapa métrico basado en rejillas de ocupación. Donde la rejilla está representada por una matriz  $M[m][n]$  de dimensiones  $n$  filas y  $m$  columnas.

**MT** = Mapa topológico definido por un par de conjuntos finitos  $\langle V, C \rangle$ , donde  $V$  es el conjunto de vértices y  $C \subset V \times V$  es un conjunto de pares de vértices llamados aristas o caminos.

### 4.3.2. Especificación del SMR

Con base en la especificación general anterior, definimos al sistema multirobot de esta tesis de la siguiente manera.

$$\mathbf{SMR} = \langle \mathbf{GR}, \mathbf{A} \rangle$$

donde:

$$\mathbf{GR} = \{R_1, R_2, R_3\} \text{ y } R_i = \{Ca, Co\}$$

**Ca** = { avanzar,  
 girar, detenerse,  
 medir\_distancia\_odómetro,  
 medir\_distancia\_sonar,  
 enviar\_datos\_servidor,  
 esperar\_datos\_servidor,  
 enviar\_datos\_robot,  
 recibir\_datos\_robot  
 }

$\mathbf{Co} = \{$  detectar\_pared,  
 corregir\_posición,  
 buscar\_pared,  
 recibir\_orden\_de\_acceso,  
 enviar\_orden\_de\_acceso,  
 identificar\_nodo,  
 estimar\_posición,  
 seguir\_pared,  
 navegar,  
 actualizar\_mapa\_local,  
 acceder\_al\_ambiente,  
 planificar\_trayectoria  $\}$ .

$\mathbf{A}$  es el ambiente del robot definido por dos tipos de mapas, un mapa topológico y un mapa métrico en este caso basado en rejillas de ocupación, esto es:

$$\mathbf{A} = \langle \mathbf{MT}, \mathbf{MM} \rangle$$

donde:

$\mathbf{MM}$  = Mapa métrico basado en rejillas de ocupación. Donde la rejilla está representada por una matriz  $M[i][j]$  de dimensiones  $i$  filas y  $j$  columnas. Se utilizan mapas métricos de distintas dimensiones. Las especificaciones precisas de los mismos se proporcionan en el capítulo 5 de experimentos.

$\mathbf{MT}$  = Mapa topológico definido por un par de conjuntos finitos  $\langle \mathbf{V}, \mathbf{C} \rangle$ , donde  $V$  es el conjunto de vértices y  $C \subset V \times V$  es un conjunto de pares de vértices llamados aristas o caminos. Se utilizarán mapas topológicos de distintas dimensiones y topologías. Las especificaciones precisas se dan en el capítulo 5 de experimentos.

## 4.4. Capacidades del sistema (Ca)

Las capacidades específicas (Ca) del robot, pueden ser de tres tipos: capacidades de acción, capacidades de percepción y capacidades de comunicación. Las capacidades de acción son aquellas que tienen que ver directamente con los actuadores del robot, tales como servomotores y motores DC. Las capacidades de percepción son aquellas que involucran a los sensores como el sonar, el sensor de distancia infrarrojo (del dispositivo de sensado c.f 4.2) y los odómetros. Y por último, las capacidades de comunicación permitan la transmisión de información entre los robots mediante algún medio físico.

A continuación se describen las capacidades del sistema, para la presente tesis se manejan de manera indistinta las capacidades de acción, de percepción y de comunicación.

1. **avanzar**: Esta capacidad consiste en enviar datos a los motores para accionar las llantas correspondientes en el sentido adecuado para que el robot avance una distancia  $d$  determinada.
2. **girar**. Esta capacidad consiste en enviar datos a los motores para accionar las llantas correspondientes en el sentido adecuado para que el robot gire un número de grados específicos, 45 para estos experimentos. La dirección de giro del robot es a la derecha o izquierda, sobre su eje central.
3. **detenerse**. Ordena a los motores detenerse, desactivándolos.
4. **medir\_distancia\_odómetro**. Toma lecturas de los odómetros las cuales dan evidencia de la distancia recorrida por el robot.
5. **esperar**. El robot espera un tiempo  $t$  establecido antes de ejecutar la siguiente acción o desplazamiento.
6. **medir\_distancia\_sonar**. Mide la distancia al objeto más cercano en la parte frontal, izquierda y derecha del robot empleando un sonar integrado al dispositivo de sensado.
7. **enviar\_datos\_servidor**. Consiste en enviar datos sobre la posición del robot y las distancias medidas con el sonar en la parte frontal, derecha e izquierda del robot.

8. **esperar\_datos\_servidor**. Consiste en esperar y recibir información del servidor tal como el mapa métrico, el mapa topológico, el orden de acceso al ambiente y punto al cual el robot se debe dirigir.
9. **enviar\_datos\_robot**. Consiste en enviar información a cualquiera de los robots del sistema multirobot.
10. **recibir\_datos\_robot**. Consiste en recibir información de cualquiera de los robots del sistema multirobot.

## 4.5. Comportamientos (Co)

A continuación se describen los comportamientos del sistema multirobot para la exploración colectiva de un ambiente interior. Dichos comportamientos están divididos en dos tipos, los comportamientos básicos (CoB) y los comportamientos complejos (CoC). Los CoB son las conductas que el robot puede ejecutar a partir de la combinación de diferentes capacidades. Por otra parte, los comportamientos compuestos (CoC) son las conductas que cada uno de los robots puede realizar empleando CoB e incluso otros CoC.

Los comportamientos básicos CoB son los siguientes:

- **corregir\_posición**. Cuando el robot se encuentra navegando dentro del ambiente es necesario que el robot se desplace en línea recta para crear el mapa local de manera coherente. La mayor parte del tiempo el robot se encuentra navegando entre dos paredes, de esta manera, cuando percibe que se está acercando a una pared y por consiguiente alejándose de la pared del lado opuesto, incrementa la velocidad de la rueda que se encuentra del lado al cual el robot se está acercando. De esta manera corrige su posición para navegar lo más preciso posible en línea recta.
- **esperar\_información\_inicial**. Al iniciar la ejecución del sistema, cada robot recibe la información necesaria por parte del servidor para realizar la actualización del mapa local que le sea asignada. Dicha información consiste en datos con información sobre el orden de acceso al ambiente y los mapas tanto topológico como métrico. Este comportamiento es ejecutado por el robot de manera automática al iniciar su ejecución.

- **esperar\_orden\_de\_acceso.** Después de que un robot recibe la información inicial, compara el orden de acceso con su identificador interno. En caso de que su identificador coincida con el primer número en el orden de acceso, el robot comenzará la ejecución de su tarea, en caso contrario ejecutará el comportamiento de **esperar\_orden\_acceso** . Durante la ejecución de este comportamiento, el robot permanece inmóvil a la espera de un mensaje que le indique que debe acceder al ambiente. Dicho mensaje puede provenir del servidor o de otro robot miembro del sistema.
- **enviar\_orden\_acceso.** Cuando un robot alcanza e identifica el primer nodo a acceder en el ambiente representado en el mapa topológico, debe enviar una notificación al siguiente robot en turno para que dicho robot pueda acceder al ambiente.
- **actualizar\_mapa\_local.** Para actualizar su mapa local, el robot realiza un sensado utilizando su dispositivo de sensado en tres posiciones distintas en el ambiente con respecto a su orientación: derecha, izquierda y frente. Con dicha información, así como la de su posición, el robot actualiza el mapa local almacenado en su memoria interna y envía dicha información al servidor para que el mapa local sea actualizado en línea y sea visible por parte de un usuario ubicado fuera del ambiente.
- **planificar\_trayectoria.** Con base en la información del mapa topológico y en el punto al cual el robot debe llegar para actualizar el mapa local, el robot ejecuta el algoritmo de Dijkstra (Weiss (1992)) para planificación de trayectorias y calcula la ruta más corta en cuanto a distancia para acceder a dicho punto.
- **detectar\_pared.** Un robot puede detectar paredes en tres distintas posiciones con respecto a su propia orientación, al frente, a la derecha o a la izquierda. Para detectar una pared, el robot toma cinco lecturas del sonar en intervalos de 100 *ms* y obtiene un promedio de estas distancias. Si el promedio de las lecturas del sonar es menor que un umbral previamente definido, del orden de la medida del ancho de un pasillo, se considera que se ha detectado una pared.

Los comportamientos compuestos CoC son los siguientes:

- **buscar\_pared.** El robot busca una pared que se encuentre del lado izquierdo o derecho de su estructura. Para realizar esto, el robot dirige su dispositivo de sensado a una de dos posiciones dependiendo de la pared que se desea buscar. Una vez posicionado el dispositivo de sensado, avanza de manera constante en línea recta hacia al frente ejecutando el comportamiento **detectar\_pared** en intervalos de  $200ms$ . Una vez detectada la pared, el robot mueve su dispositivo de sensado al frente y se detiene.
- **acceder\_ambiente.** Los robot miembros del sistema multirobot se encuentran fuera del ambiente formados uno al costado del otro a una distancia mínima aproximada de  $20\text{ cm}$ , determinada experimentalmente para evitar colisiones entre los robots. Para acceder al ambiente, un robot debe avanzar en línea recta  $40\text{ cm}$ , girar a su derecha  $90^\circ$  mover su dispositivo de sensado a su derecha y ejecutar el comportamiento **buscar\_pared**. Una vez detectada la pared derecha, el robot considera que ha entrado al ambiente.
- **identificar\_nodo.** Para identificar un nodo en el ambiente el robot mueve su dispositivo de sensado a tres posiciones, al frente, derecha e izquierda de su estructura y en cada una de ellas ejecuta el comportamiento básico **detectar\_pared**. Con esto, el robot tendrá información que le indica si existen paredes al frente, a la izquierda y a la derecha de su estructura. Con ello, además de obtener información de su posición, el robot puede determinar en qué nodo del mapa topológico se encuentra.
- **estimar\_posición.** El robot estima su posición con respecto a la distancia a las paredes y a la información de su posición previa.
- **seguir\_pared.** El robot se desplaza de forma paralela a una pared en intervalos de  $5\text{ cm}$ , dicha medida se determinó experimentalmente para que el robot abarcara la mayor área posible con su dispositivo de sensado. En cada intervalo el robot ejecuta el comportamiento básico **detectar\_pared**, por lo que avanzará mientras detecte la pared a seguir. Cuando no es detectada ninguna pared, el robot se detiene.
- **navegar.** La navegación de robot se realiza ejecutando los comportamientos de **detectar\_pared** e **identificar\_nodo** así como el la capacidad de **girar** cuando sea necesario.

## 4.6. Organización del sistema

En la Figura 4.7 se presenta la organización general del sistema multirobot. A continuación se detalla dicha organización.

Inicialmente se tiene un ambiente interior sin modificaciones el cual es considerado como el ambiente interior original. En general, el ambiente interior original puede ser de cualquiera de los tipos descritos en la sección 4.1.1. Sin embargo, en esta tesis se abordarán los ambientes del tipo 2 hasta con tres nodos terminales. De esta manera, en una fase previa a los experimentos realizados al sistema multirobot, un robot se encarga de entrar al ambiente y explorarlo mediante una trayectoria previamente programada en el robot (ver Figura 4.7 a)). Durante la exploración el robot genera un mapa métrico y lo almacena en su memoria interna de tal forma que al salir del ambiente se tiene una representación del ambiente original (ver Figura 4.7 b)). Dicha representación en forma de mapa métrico es entregada a un servidor externo ((ver Figura 4.7 c))).

El servidor externo genera de manera automática un mapa topológico (ver Figura 4.7 d)) empleando el algoritmo **AAMT** a partir del mapa métrico que le fue entregado, el algoritmo **AAMT** se describe a detalle en la sección 4.9. El mapa topológico es empleado por el servidor para la división en zonas disjuntas del ambiente aplicando el esquema descrito en la sección 4.12.1. Una vez realizada la división se asignan zonas a cada miembro del sistema multirobot (ver Figura 4.7 e)) tal como se describe en la sección 4.12.1. Los tres robots establecen comunicación con cada uno de los miembros del sistema y a su vez con el servidor externo y acceden al ambiente por turnos (ver Figura 4.7 f)) como se describe en la sección 4.12.2.

Los robots exploran de manera independiente y autónoma el ambiente, toman lecturas de sus sensores las cuales son enviadas al servidor externo empleando una comunicación inalámbrica (ver Figura 4.7 g)). El servidor externo es el encargado de la construcción de los mapas locales y de su fusión en un único mapa global coherente (ver Figura 4.7 h)), el esquema de fusión de los mapas locales en un mapa global se presenta en la sección 4.12.1.

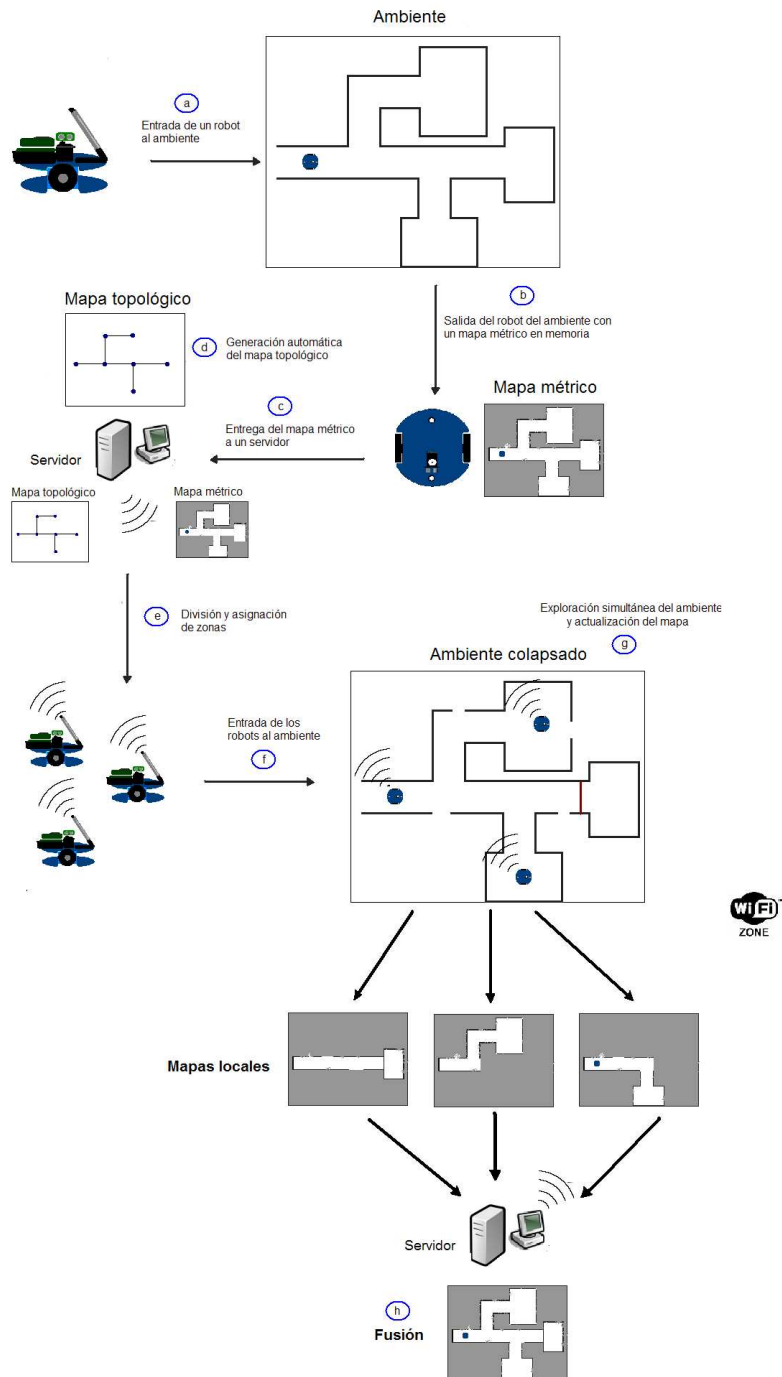


Figura 4.7: Organización del sistema multirobot  
49

## 4.7. Generación automática del mapa original del ambiente

Los modelos del ambiente empleados en la presente tesis son un mapa topológico y un mapa métrico basado en rejillas de ocupación. Originalmente el sistema cuenta con un mapa métrico cuyas probabilidades *a priori* son inicializadas en 0.5, lo cual indica que no existe conocimiento previo acerca del ambiente que se desea explorar. A continuación se describe cómo fueron adecuados estos dos tipos de modelos al sistema multirobot y la forma en la que se generan cada uno de ellos.

La percepción de cada miembro del sistema multirobot está limitada a un único sonar montado sobre un servomotor que le permite tomar lecturas en diferentes orientaciones del robot y que forma parte del dispositivo de sensado. El servomotor sobre el cual se encuentra montado el sonar puede girar  $180^\circ$  relativos a la posición del robot en el ambiente. Sin embargo, el tomar lecturas del sonar en cada una de las posiciones posibles es un proceso costoso y lento, por lo que se realizó una discretización de las posiciones posibles del servomotor para tomar lecturas. Dicha discretización consistió en tomar lecturas únicamente en  $0^\circ$ ,  $90^\circ$  y  $180^\circ$  que corresponden a la derecha, frente e izquierda del robot (ver Figura 4.8 a)) cada vez que el robot avanza en intervalos de 5 *cm.* (ver Figura 4.8 b)). Cabe mencionar que dicha discretización es suficiente para los propósitos de la presente tesis debido a que permite al robot, una vez en movimiento, cubrir un área considerable del ambiente a su alrededor y reducir el tiempo de sensado. Además, siempre que se elige un modelo para representar un ambiente se debe considerar que dicha representación contiene sólo las características útiles para que un robot en particular ejecute las tareas que le sean asignadas, y que por lo tanto existirán elementos en el ambiente que no serán modelados.

## 4.8. Mapa basado en rejillas de ocupación

El modelo para el ambiente basado en rejillas de ocupación que emplea el sistema multirobot está basado en el modelo propuesto por Murphy (2000) descrito en el capítulo 2 de este trabajo. El algoritmo para la creación y actualización del mapa basado en rejillas de ocupación se detalla en el algoritmo 1. Dicho algoritmo recibe como parámetros la posición ( $posX, posY$ ) del robot y una matriz que representa la rejilla de ocupación llamada *reji-*

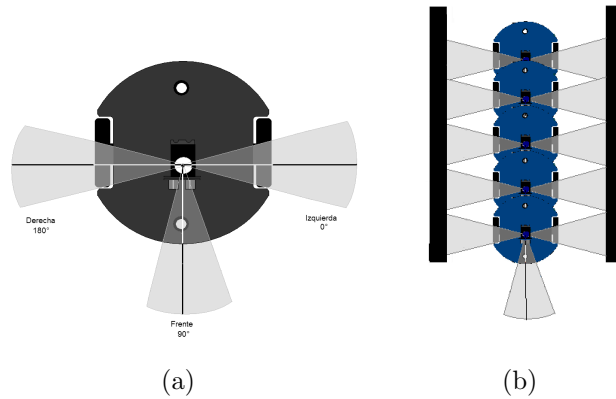


Figura 4.8: Discretización de las posiciones en las que el robot puede sentir el ambiente. a) Tres posiciones del dispositivo de sensado, b) recorrido del un ambiente por el robot.

lla. La rejilla contiene las probabilidades *a posteriori*  $P(\text{ocupado}|s)$  para cada una de las celdas, i.e. la probabilidad de que el área correspondiente a una zona del ambiente asociada a una celda de la rejilla esté ocupada dada una lectura del sensor. La rejilla es inicializada con probabilidades *a priori* de 0,5 y conforme el robot toma nuevas lecturas del sonar, la rejilla se actualiza empleando el algoritmo 1.

El algoritmo 1 **actualiza-rejilla** actualiza el mapa basado en rejillas de ocupación. Las probabilidades *a priori* de que las celdas de la rejilla se encuentren o no ocupadas se calculan en la línea 2 a). Las probabilidades de ocupación dadas la posición actual del robot y la distancia medida por el sonar dependen del modelo del sonar y son calculadas en la línea 2 b). En la línea 2 c) se actualizan las probabilidades *a posteriori* aplicando la regla de Bayes y almacenando solamente la probabilidad *a posteriori* de que una celda esté ocupada. El proceso de actualización de las celdas de la rejilla de ocupación se realiza únicamente para las celdas que se encuentren dentro del campo perceptual del sonar. Dicha verificación es realizada en la línea 2.

El algoritmo 2 **modeloSonar** aplica el modelo del sonar para la interpretación de los datos procedentes de dicho sensor. Recibe como parámetros de entrada la posición  $(posX, posY)$  que representa el origen del sonido en el modelo del sonar, además de la posición de la celda a considerar  $posI, posJ$ , y calcula la probabilidad de ocupación de dicha celda dada la lectura del sonar.

Cada celda de la rejilla de ocupación se representa  $5\text{cm} \times 5\text{cm}$  del ambiente. La variable *maxOccupied* representa la máxima probabilidad de ocupación que puede tener una celda. Para el caso de la presente tesis se manejó el valor de esta última a 0.98, lo que indica que ninguna celda puede tener una probabilidad de estar ocupada que sea mayor a 0.98, y una probabilidad de estar vacía menor a 0.2. Lo anterior es representado en la línea 6 del algoritmo 2 al multiplicar la probabilidad  $P$  por *maxOccupied*. La variable  $R$  indica la distancia mayor que puede tener una lectura del sonar, por lo que cualquier distancia mayor a  $R$  será considerada únicamente como  $R$ , según se ajusta en la línea 4 de este algoritmo. El valor de esta variable depende, en general, de las condiciones del ambiente. Para los ambientes empleados en esta tesis el valor de  $R$  fue inicializado con 10, lo que significa que cualquier obstáculo que se encuentre a una distancia mayor de 10 celdas, i.e.  $50\text{ cm.}$ , no serán tomados en cuenta, mientras que las celdas comprendidas dentro del campo perceptual del sonar tendrán una alta probabilidad de estar vacías, cuando la lectura del sonar así lo indica.

La región en la que se encuentra una celda determinada es calculada con el algoritmo 3 *determinar-región*, el cual tiene como parámetros de entrada la distancia del origen del modelo, i.e. la posición  $(posX, posY)$  del robot, a una celda dada, y la distancia medida por el sensor. El algoritmo 3 regresa un valor que representa una de las tres regiones consideradas dentro del modelo del sensor.

El algoritmo 4 *campoPerceptual* recibe como parámetros de entrada la posición del robot, así como su orientación y la posición de la celda a actualizar. Con base en esa información se determina si la celda a actualizar se encuentra dentro del campo perceptual del robot mediante la función *atan2* empleado en la línea 1. De esta manera, si el valor absoluto de la distancia angular entre la línea formada por los puntos de la posición de la celda y la posición del robot con respecto del ángulo de la orientación del robot es menor a  $15^\circ$ , indicará que la celda se encuentra dentro del campo perceptual del robot.

---

**Algoritmo 1. Algoritmo *actualiza-rejilla***

**Variables globales:**

Rejilla de ocupación  $rejilla[n][m]$

**Datos de entrada:**

Posición del robot ( $posX, posY$ ), Orientación del robot  $phi$

1. Tomar lectura del sonar con la primitiva correspondiente del conjunto **Ca** y asignarla a una medida de distancia  $da$ .

$da \leftarrow \text{medir\_distancia\_sonar}$

2. Para cada elemento  $(i, j)$  de la rejilla, actualizar la probabilidad de ocupación de la siguiente forma: Si el elemento  $(i, j)$  de la rejilla se encuentra en el campo perceptual del modelo del sonar, esto es

Si ( $\text{campoPerceptual}(posX, posX, i, j, phi) = 1$ ) hacer

- a) Obtener las probabilidades a priori de ocupación  $probOcupadoapriori$ , y de no ocupación  $probVacioapriori$ , a partir de la información almacenada en la rejilla de ocupación.

$probOcupadoapriori \leftarrow rejilla[i][j]$

$probVacioapriori \leftarrow 1,0 - probOcupadoapriori$

- b) Obtener la probabilidad de ocupación dado el modelo del sonar.

$probOcupado \leftarrow \text{modeloSonar}(i, j, x, y, da)$

$probVacio \leftarrow 1,0 - probOcupado$

- c) Actualizar las probabilidades a posteori aplicando la regla de Bayes.

$rejilla[i][j] \leftarrow \frac{(probOcupado \times probOcupadoapriori)}{((probOcupado \times probOcupadoapriori) + (probVacio \times probVacioapriori))}$

En otro caso, no actualizar el elemento  $(i, j)$  de la rejilla de ocupación.

---

---

Algoritmo 2. Algoritmo *modeloSonar*

**Datos de entrada:**

Posición del robot  $(posX, posY)$ , Orientación del robot  $phi$ , Distancia medida por el sonar  $da$

**Datos de salida:**

Probabilidad de ocupación de la celda  $(posI, posJ)$

**Datos globales:**

Distancia máxima permitida para el sonar  $R$

1. Obtener la distancia Euclidiana  $r$  de la posición en donde se encuentra el robot  $(posX, posY)$  a la posición  $(posI, posJ)$  de la celda a actualizar.

$$r \leftarrow \text{sqrt}((posI - posX)^2 + (posI - posY)^2)$$

2. Calcular la distancia angular  $\alpha$  entre el centro del cono del sonar y la celda en la posición  $(posI, posJ)$ .

$$\alpha \leftarrow \text{atan2}(posI, posJ, posX, posY, phi)$$

3. Calcular la probabilidad de ocupación de la celda  $(posI, posJ)$ , donde  $\beta$  es el ángulo del campo de vision del sonar.

$$P \leftarrow (((R - r)/R) + ((\beta - \alpha)/\pi))/2;$$

4. Si la distancia medida por el sonar  $da$  es mayor que la distancia máxima permitida  $R$ , entonces emplear la distancia máxima permitida como la medida del sonar:  $da \leftarrow R$

5. Obtener la zona  $reg$  en la que se encuentra la celda  $(posI, posJ)$ :  $reg \leftarrow \text{determinar\_region}(r, da)$

6. Si la celda  $(posI, posJ)$  se encuentra en la **Región I** regresar  $P \times \text{maxOccupied}$  y terminar algoritmo.

7. Si la celda  $(posI, posJ)$  se encuentra en la **Región II** regresar  $(1.0 - P)$  que representa la probabilidad de que la celda no esté ocupada y terminar algoritmo

8. En cualquier otro caso regresar 0, dado que la celda se encuentra en la **Región III**.

---

Algoritmo 3. algoritmo **determinar-region**

**Datos de entrada:**

Distancia a la celda  $r$  , Distancia medida por el sonar  $da$

**Datos de salida:**

identificador de la región en la cual se encuentra la celda.

**Datos globales:**

$tolerance$ , denota una medida de tolerancia que representa el grosor de los muros en el ambiente.

1. Si  $((r \geq (s - tolerance)) \wedge (r \leq (s + tolerance)))$  regresar indicador de región 1 y terminar algoritmo
2. Si  $(r < s - tolerance)$  regresar indicador de región 2 y terminar algoritmo
3. En otro caso, regresar indicador de región 3

---

---

Algoritmo 4 Algoritmo *campoPerceptual*

**Datos de entrada:**

Posición del robot  $(posX, posY)$ , orientación del robot  $phi$ , posición de la celda a actualizar  $(posI, posJ)$

**Datos de salida:**

*TRUE* Si la celda  $(posX, posY)$  se encuentra dentro del campo perceptual del sonar.

*FALSE* En caso contrario.

1. Si  $(atan2(posJ - posY, posI - posX) < 15)$  regresar *TRUE* y terminar
  2. En caso contrario regresar *FALSE* y terminar
-

## 4.9. Mapa topológico

Además del mapa métrico presentado en la sección anterior, el sistema multirobot cuenta con un segundo modelo del ambiente consistente en un mapa topológico. Este segundo modelo es creado a partir del mapa basado en rejilla de ocupación creado en una primera fase por parte de un miembro del sistema multirobot. Para la presente tesis definimos un mapa topológico  $\mathbf{MT}$  como una tupla de nodos y arcos de la siguiente manera:

$$\mathbf{MT} = \langle \text{nodos, caminos} \rangle$$

donde cada nodo y arco tiene asociado un tipo de estructura de datos de la siguiente forma:

```
Tipo Nodo {
    int nodoPosX;
    int nodoPosY;
    Arco listaArcos[tamMaxNodos];
}
```

```
Tipo Camino {
    int nodoOrigen;
    int nodoDestino;
    orientacion Or;
    distancia dist;
}
```

La orientación  $Or$  en los arcos existentes de la estructura *Camino* entre nodos representa la orientación apropiada hacia la cual el robot debe orientarse para llegar desde el primer nodo *nodoOrigen* con coordenadas:

$(\text{nodoOrigen.nodoPosX}, \text{nodoOrigen.nodoPosY})$

hasta el segundo nodo denominado *nodoDestino* con coordenadas:

$(nodoDestino.nodoPosX, nodoDestino.nodoPosY)$

El nodoDestino se encuentra a una distancia  $dist$  también definida como parte del camino. Para el caso del mapa topológico empleado se asume que el robot, una vez orientado hacia la posición apropiada, únicamente debe ejecutar un movimiento en línea recta para llegar de un nodo a otro.

El algoritmo 5 **AAMT** para la abstracción de un mapa topológico recibe como dato de entrada un mapa métrico basado en rejillas de ocupación **MM** y devuelve el mapa topológico **MT** asociado a dicho mapa métrico. La idea general del algoritmo **AAMT** es realizar un recorrido virtual iniciando en una posición y con una orientación conocidas a través del mapa métrico buscando características tales como bifurcaciones del pasillo principal y esquinas en las paredes. Este recorrido se realiza celda por celda. Se asume que existe una pared a partir de las probabilidades de ocupación del mapa métrico. Las bifurcaciones y esquinas en el mapa métrico representan nodos del mapa topológico, en cambio los pasillos representan caminos entre los nodos. Una vez dentro de una habitación el algoritmo realiza el recorrido de forma paralela a las paredes y cuando que se han tocado tres esquinas se asume que se cubrió toda la habitación. En el cuadro 4.1 se describen los estados aludidos por el algoritmo 5.

Izquierda	Frente	Derecha	Estado	Id.
1	0	1	Pasillo	$ps$
0	1	0	Bifurcación en ambos lados	$bAL$
1	0	0	Bifurcación a la derecha	$bD$
0	0	1	Bifurcación a la izquierda	$bI$
0	1	1	Esquina izquierda	$eI$
1	1	0	Esquina derecha	$bD$

Cuadro 4.1: Estados posibles de las celdas analizadas durante el recorrido virtual del mapa métrico  $MM$  realizado por el algoritmo **AAMT**.

---

## Algoritmo 5. Algoritmo *AAMT*

### Datos de entrada:

Mapa basado en rejillas de ocupación de dimensiones  $m$  filas  $\times$   $n$  columnas  
 $MM[m][n]$

### Datos locales:

Estado actual de la celda  $edoCelda$ , estado anterior al estado actual de la celda  $edoCeldaAnt$ , pila para almacenar estados de las celdas  $P$ , bandera que indica si se encontró un nodo terminal  $BnodoTerminal$ , mapa topológico  $MT$ .

1. Establecer el estado de la celda como pasillo  $edCelda \leftarrow ps$  e insertarlo en la pila  $P$ .
2. Igualar el estado anterior al estado actual  $edCeldaAnt \leftarrow edCelda$ , y hacer  $BnodoTerminal \leftarrow FALSE$ .
3. Mientras existan elementos en la pila  $P$  hacer
  - 3.1 Mientras ( $BnodoTerminal == FALSE$ ) hacer
    - 3.1.1 Avanzar sobre la rejilla hasta encontrar una esquina o una bifurcación y actualizar el estado actual.
    - 3.1.2 **Si** se encontró una bifurcación  $bD$ ,  $bI$ , o  $bAL$  **hacer**
      - 3.1.2.1 Marcar el camino derecho para  $bD$  y  $bAL$  e izquierda para  $bI$  como analizado e insertar el estado actual en el  $MT$
      - 3.1.2.2 cambiar la orientación a la derecha para  $bD$  y  $bAL$  e izquierda para  $bI$ .
      - 3.1.2.3 Si se han visitado tres esquinas de una habitación, hacer  $BnodoTerminal \leftarrow TRUE$ .
    - 3.1.3 **Si** se encontró una esquina, cambiar la orientación a la derecha si se está en el estado  $eD$ , e izquierda si está en el estado  $eI$ .

---

**Algoritmo 5. Algoritmo AAMT (Continuación)**

3.1.4 Verificar si el giro corresponde a una esquina dentro de una habitación.

3.1.5 Insertar en MT *edoCelda*

3.1.6 Crear camino de *edCeldaAnt* a *edCelda* e insertarlo en el mapa topológico MT.

3.1.7  $edCeldaAnt \leftarrow edCelda$ .

3.2  $edoCelda \leftarrow$  sacar estado de  $P$

4 Resgresar el mapa topológico MT.

---

## 4.10. Corrección del error odométrico

En el enfoque propuesto, son empleadas las características de los ambientes descritos anteriormente para corregir el error odométrico. La corrección del error odométrico se realiza a partir de la distancia a las paredes, i.e. si el robot debe seguir una pared derecha o izquierda, debe mantenerse siempre a la misma distancia empleando el sonar y el control diferencial con los que cuenta. Antes de iniciar el seguimiento de una pared, se debe almacenar una distancia inicial estimada entre el robot y la pared a seguir. Una vez iniciado el seguimiento de la pared, los movimientos del robot son divididos en pasos. En cada paso el robot debe ejecutar las siguientes acciones:

- **Medir la distancia a la pared en seguimiento.** Se aplica para ello la primitiva del conjunto `Ca medir_distacia_sonar`.
- **Calcular velocidades para los motores de la llantas.** Las velocidades para los motores que controlan las llantas del robot son calculadas con base en la distancia hacia un objeto de referencia, en este caso una pared. Esta distancia es medida en el paso actual y comparada con la distancia inicial al objeto de referencia. Si la distancia medida en el paso actual es mayor que la distancia inicial, se debe aumentar la velocidad de la llanta que se encuentra del lado opuesto a la pared en

seguimiento. En caso contrario, se aumenta la velocidad de la llanta del mismo lado de la pared.

- **Avanzar.** Una vez calculadas las velocidades para los motores, el robot debe ejecutar la habilidad `avanzar` del conjunto **Ca**.

Antes de la ejecución de cada paso, las velocidades de los motores son igualadas a una velocidad establecida previamente, para evitar que las velocidades se incrementen de manera indefinida.

Un ejemplo de la corrección odométrica empleada se puede apreciar en la Figura 4.9. En este caso, antes de iniciar el seguimiento de la pared izquierda, el robot toma una distancia llamada  $d_{inicial}$  a la pared izquierda, después ejecuta el primer paso en el tiempo  $t = 1$  y nota que la distancia a la pared derecha es menor que la distancia inicial, i.e.  $d_{t=1} < d_{inicial}$  por lo que aumenta la velocidad de la llanta izquierda para alejarse. El robot ejecuta dos pasos sin cambios en las velocidades. En el tiempo  $t = 4$  el robot se encuentra a una distancia mayor a la pared izquierda que la distancia inicial, i.e.  $d_{t=4} > d_{inicial}$  por lo que ahora aumenta la velocidad de la llanta derecha para acercarse.

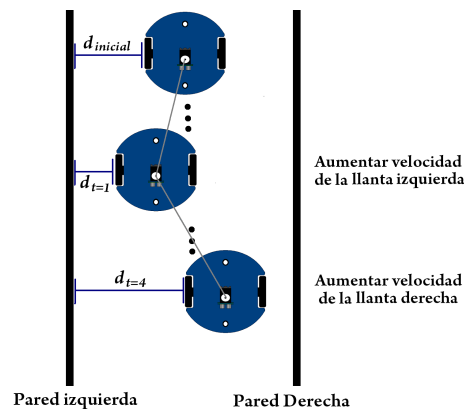


Figura 4.9: Corrección de la odometría del robot con base en información del ambiente.

## 4.11. Estimación de la pose del robot

La estimación de la pose del robot se calcula con respecto al mapa basado en rejillas de ocupación, cuya discretización consistió en representar una zona del ambiente de dimensiones de  $5 \times 5 \text{ cm}$  en una celda. Cuando se dice que el robot se encuentra en la pose  $(posX, posY)$  nos referimos a que el centro del robot se encuentra en dicha pose, aunque en realidad el robot abarque varias celdas alrededor de ella. La pose inicial de los robots es conocida, por lo que la estimación de la nueva pose, después de un movimiento del robot, es calculada con base en la información de sus odómetros. Sin embargo, lo anterior puede provocar errores en la estimación dada la imprecisión de los odómetros. Para corregirlos, se emplean dos enfoques adicionales para realizar la estimación. El primero emplea la distancia a las paredes, de esta manera se conoce si el robot ha tenido un desplazamiento a su derecha o a su izquierda (ver Figura 4.10). El segundo emplea la información del mapa topológico, i.e. en cuanto el robot reconoce un punto del mapa topológico, corrige la estimación de su pose con respecto a la información conocida (ver Figura 4.11).

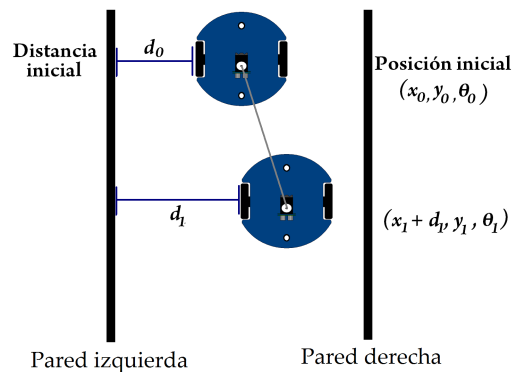


Figura 4.10: Estimación de la pose del robot con base en las características del ambiente, i.e. distancia a las paredes del ambiente.

## 4.12. División y fusión del mapa

La división del ambiente en zonas y la fusión de los mapas locales basados en rejillas de ocupación se realiza a partir de la información proporcionada

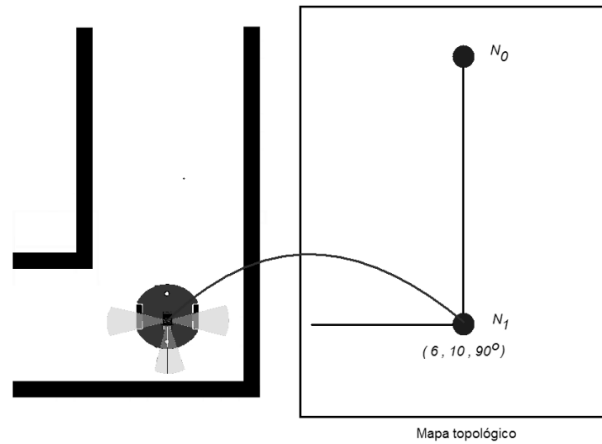


Figura 4.11: Estimación de la pose del robot empleando un mapa topológico.

por el mapa topológico. A continuación se describe el esquema tanto para la división como para la fusión de los mapas locales.

#### 4.12.1. División, repartición y asignación del espacio

La división del ambiente en zonas tiene como propósito asignar zonas a los robots miembros del sistema multirobot para la exploración de esas zonas y la actualización del mapa global. Esta tarea es realizada por el servidor del sistema multirobot empleando el mapa topológico creado por el mismo servidor fuera de línea a partir del mapa basado en rejillas de ocupación.

En la Figura 4.12 a) se muestra un ambiente parcial de dimensiones  $1,6 m. \times 2,6 m.$  para el experimento de construcción de un mapa topológico empleando el algoritmo de Abstracción de un Mapa Topológico AAMT (c.f. 4.7). El mapa topológico para este ambiente es el siguiente:

$$MT = \langle V, C \rangle$$

donde:

$$V = \{P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8, P_9, P_{10}, P_{11}, P_{12}\}$$

y

$$\mathbf{C} = \{(P_1, P_2), (P_2, P_3), (P_3, P_4), (P_4, P_5), (P_5, P_6), (P_6, P_7), (P_2, P_8), (P_8, P_9), (P_9, P_{10}), (P_{10}, P_{11}), (P_{11}, P_{12})\}$$

La representación gráfica del mapa topológico se muestra en la Figura 4.12 b). Los nodos terminales son aquellos que aparecen únicamente como segundo componente de los pares ordenados de  $\mathbf{C}$ , para éste mapa topológico los nodos terminales son  $P_7$  y  $P_{12}$ . Como el número máximo de robots de nuestro sistema multirobot es igual a 3, y considerando que se asigna un nodo terminal por robot, el número máximo de nodos terminales que pueden ser asignados a nuestro sistema multirobot es igual a 3

Las zonas son asignadas a los robots de acuerdo a prioridades. Estas prioridades se determinan tomando en cuenta lo siguiente:

- Cada robot tiene asociado un identificador  $id_i$  con  $1 \leq i \leq 3$ .
- Se calculan las distancias entre el nodo inicial y los nodos terminales empleando el algoritmo de Dijkstra (Weiss (1992)) y el mapa topológico.
- El nodo terminal con la mayor distancia es asignado al robot con el identificador  $id_1$ , el segundo nodo con mayor distancia se asigna al robot con identificador  $id_2$  y el tercer nodo con mayor distancia se asigna al robot con identificador  $id_3$ .
- En caso de existir menos zonas que robots, se emplean menos robots para la exploración del ambiente.

Por otro lado, existen áreas comunes, i.e. áreas recorridas por más de un robot para acceder a las zonas asignadas. Estas áreas comunes corresponden a uno o varios segmentos del pasillo principal. Para el mapa topológico de la Figura 4.12 b), el área común a los robots corresponde al área representada en el mapa topológico por el camino  $(P_1, P_2)$  debido a que los dos robots tienen que recorrer este camino para acceder a sus zonas asignadas, nodos terminales  $P_3$  y  $P_5$ .

#### 4.12.2. Acceso a zonas

Para el ingreso de los robots al ambiente es necesario que exista comunicación entre todos los robots del sistema y entre el servidor. De esta manera,

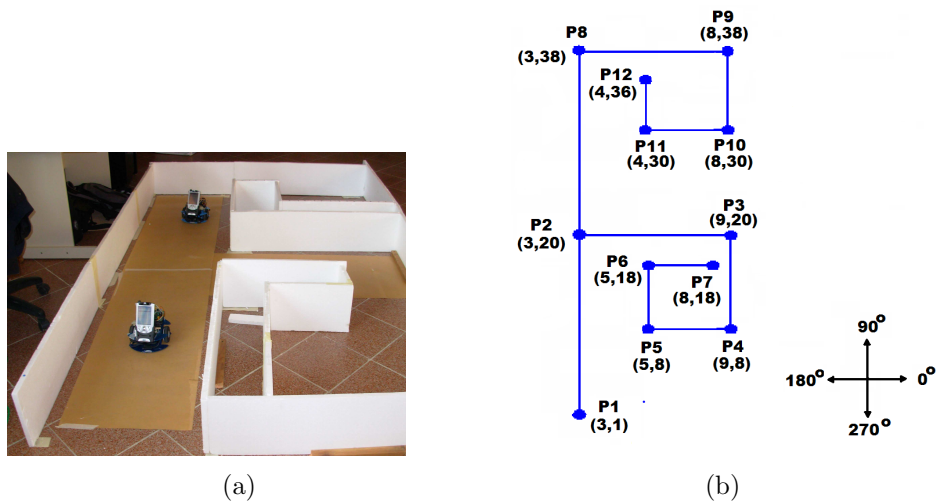


Figura 4.12: Generación de un mapa topológico. a) Ambiente para experimentos parciales, b) mapa topológico con dos nodos terminales  $P_3$  y  $P_5$ .

el robot que tiene el identificador más bajo es el primero en acceder al ambiente. Los robots reciben tres piezas de información por parte del servidor: el mapa topológico, el mapa métrico y el nodo al cual dirigirse. Los robots comienzan su operación fuera del ambiente formados uno junto al otro como se muestra en la Figura 4.13 a).

Para ingresar al ambiente, un robot debe primeramente buscar la entrada. Para buscar la entrada al ambiente, el robot debe avanzar al frente para salir de su formación inicial, después girar a la derecha y posicionar su módulo de sensado a la derecha con respecto a su orientación y avanzar hasta localizar una pared a su lado derecho empleando la primitiva `buscar_pared` del conjunto **Ca**. Una vez encontrada la pared, esta es considerada como la entrada al ambiente (ver Figura 4.13 b)). Después de entrar al ambiente, el robot navega a través del ambiente empleando el mapa topológico hasta alcanzar el primer nodo del mapa topológico, en este momento el robot que explora notifica al siguiente robot en turno que puede ingresar al ambiente, y así sucesivamente hasta que todos los robots hayan ingresado.

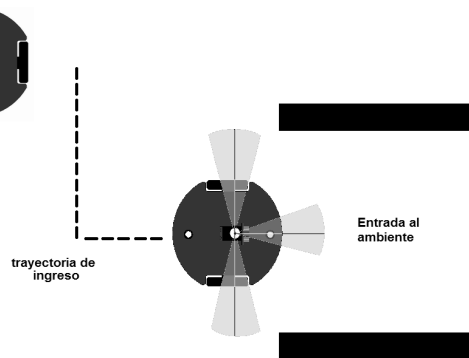


Figura 4.13: Entrada al ambiente de un miembro del sistema multirobot.

### 4.12.3. Exploración colectiva, actualización y corrección de los mapas locales de los robots

La actualización del mapa global basado en rejilla de ocupación se puede realizar de dos maneras, dependiendo de si los robots tienen o no comunicación con el servidor, según se describe a continuación:

1. **Comunicación continua.** Si los robots se pueden comunicar con el servidor, entonces envían datos como la estimación de su posición sin corrección odométrica, así como la información cruda de los sensores. En este caso el robot no realiza la estimación de la posición y tampoco crea su mapa local, sino que dichas tareas corresponden al servidor que lo logra fusionando la información que es enviada por cada uno de los robots en el mapa global.
2. **Comunicación intermitente.** Si alguno de los robots pierde la comunicación con el servidor, entonces almacena la información tanto de su posición en cada paso, como de la información obtenida del sonar. Esta información será enviada al servidor una vez que se reestablezca la comunicación

Este esquema permite la actualización del mapa en tiempo real y su visualización de forma gráfica en el servidor, además de contender con los problemas de comunicación que se pueden presentar durante el proceso de

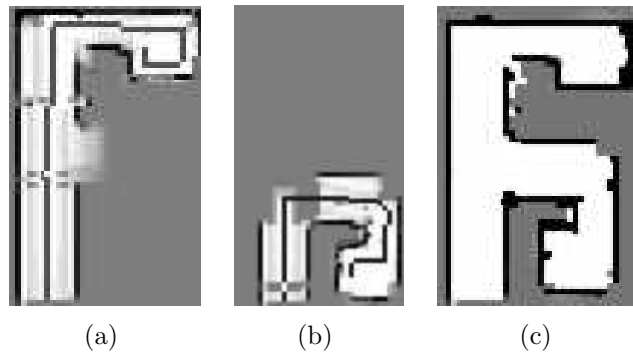


Figura 4.14: Mapas locales creados por dos robots a) y b) y su fusión en un sólo mapa global

actualización del mapa. Una segunda ventaja de la combinación de estos dos esquemas es la descarga a los robots del trabajo de la creación del mapa y navegación al mismo tiempo.

En cualquiera de los dos escenarios planteados anteriormente, para lograr una fusión de los mapas locales de manera coherente es necesario que los robots se encuentren localizados de manera correcta. Una vez localizados los robots, la fusión de los nuevos datos del sonar se realiza empleando el teorema de Bayes, de aquí la importancia de crear un mapa topológico a partir del mapa basado en rejillas de ocupación y de la localización de los robots empleando dicho mapa.

En la Figura 4.14 se muestran los mapas locales creados del ambiente de la Figura 4.12 a) con base en el mapa topológico generado automáticamente por el algoritmo de abstracción de un mapa topológico *AAMT*, la representación gráfica de dicho mapa se muestra en al Figura 4.12 b).

# Capítulo 5

## Experimentos y evaluación

En esta sección se presentan los experimentos realizados con el sistema multirobot para evaluar el desempeño de los miembros del sistema de manera individual y colectiva. Se presentan los resultados obtenidos de ocho experimentos, cuatro de ellos individuales y cuatro colectivos.

Los experimentos individuales se realizaron con la finalidad de evaluar las capacidades de cada uno de los miembros del sistema de manera independiente, debido a que el buen funcionamiento de cada miembro es importante para el desempeño de todo el sistema en general. Dichos experimentos son:

1. Entrada al ambiente.
2. Navegación y localización en el ambiente dado el mapa topológico.
3. Seguimiento de pared y estimación de la posición.
4. Generación de un mapa local.

Los experimentos colectivos que se presentan se realizaron con el objetivo de evaluar la capacidad principal del sistema multirobot consistente en la generación de mapas locales para ser fusionados en un mapa global. Para esto se emplearon tres robots homogéneos cuya construcción fue descrita a detalle en la sección 4.2. Además de los robots, también se empleó como servidor una computadora externa al ambiente, la cual realizó la fusión y visualización de los mapas locales y del mapa global fusionado. Los experimentos colectivos son:

1. Actualización del mapa global de un ambiente sin modificaciones

2. Actualización del mapa de un ambiente con modificaciones.
3. Coordinación y comunicación entre los miembros del sistema multirobot.
4. Actualización de un mapa del ambiente sin modificaciones y sin comunicación con el servidor.

En una fase previa a los experimentos colectivos, un robot fue colocado en un ambiente sin modificaciones, considerado como el ambiente original, a partir del cual se generó un mapa métrico mediante una trayectoria previamente definida y programada en el robot. A partir de este mapa métrico se generó un mapa topológico aplicando el algoritmo *AAMT* descrito en la sección 4.9. El mapa topológico generado en esta fase previa fue empleado como base para los experimentos descritos en esta sección.

Para evaluar la capacidad del sistema multirobot para actualizar el mapa global, se empleó, en el caso de los ambientes sin modificaciones, la correlación entre el mapa generado por el robot con una trayectoria previamente programada y el mapa generado por el sistema multirobot. En el caso del ambiente con modificaciones, además del algoritmo de correlación, fue empleada una métrica cualitativa consistente en verificar de forma visual si los cambios en el ambiente fueron reflejados en el mapa global fusionado por el servidor a partir de la información enviada por los tres robots.

## 5.1. Experimentos individuales

Los experimentos individuales fueron cuatro: entrada al ambiente; navegación y localización en el ambiente dado un mapa topológico; seguimiento de pared y estimación de la posición; y finalmente, generación de un mapa local. Dichos experimentos se denominan individuales debido a que fueron ejecutados por un solo robot en el ambiente y se diseñaron para evaluar las habilidades y capacidades de cada miembro del sistema por separado. A continuación se detallan cada uno de los experimentos, así como los resultados obtenidos.

### 5.1.1. Experimento no. 1: entrada al ambiente

Las especificaciones del sistema para este experimento son:

$$\text{SMR} = \langle \text{GR}, \text{A} \rangle$$

donde:

$$\text{GR} = \{ \text{R}_1 \} \text{ y } \text{R}_1 = \{ \text{Ca}, \text{Co} \}$$

$\text{Ca} = \{ \text{avanzar, girar, detenerse, medir\_distancia\_odómetro, medir\_distancia\_sonar} \}$

$\text{Co} = \{ \text{detectar\_pared, buscar\_pared, acceder\_al\_ambiente} \}$ .

$\text{A}$  es el ambiente del robot definido por un mapa topológico esto es:

$$\text{A} = \langle \text{MT} \rangle$$

donde:

$\text{MT}$  = Mapa topológico definido por un par de conjuntos finitos de vértices y caminos  $\langle \text{V}, \text{C} \rangle$ , donde  $V = \{n_0, n_1\}$  y  $C = \{(n_0, n_1)\}$

**Descripción:** Esta prueba consiste en probar la habilidad de un robot para entrar al ambiente partiendo de una posición externa, y encontrar la entrada al ambiente. Se posiciona un robot fuera del ambiente de pruebas, del lado izquierdo a la entrada y con un ángulo inicial de  $270^\circ$  y una distancia de  $50 \text{ cm}$ . de la entrada. Una vez recibida la orden de acceso al ambiente por parte del servidor, el robot avanza aproximadamente  $15 \text{ cm}$ . hacia el frente, gira a la derecha  $90^\circ$ , posiciona su dispositivo de sensado a la derecha y avanza hasta localizar una pared a su lado derecho, la cual indica que se ha encontrado la entrada al ambiente. Una vez encontrada la entrada, el robot se detiene. El ángulo de  $270^\circ$  es la orientación inicial del robot, por lo que éste debe tener siempre esta orientación previamente a la ejecución del experimento. Sin embargo, la distancia de  $50 \text{ cm}$ . puede variar, para que los tres robots miembros del sistema puedan ser colocados a distintas distancias de la entrada principal

**Resultados esperado:** El robot encuentra la entrada al ambiente y se detiene.

**Resultado obtenido:** El robot logró desplazarse, girar a la derecha correctamente y después avanzó hasta encontrar la entrada al ambiente, por lo que la prueba fue considerada exitosa. En la Figura 5.1 se muestran distintas fases del experimento.

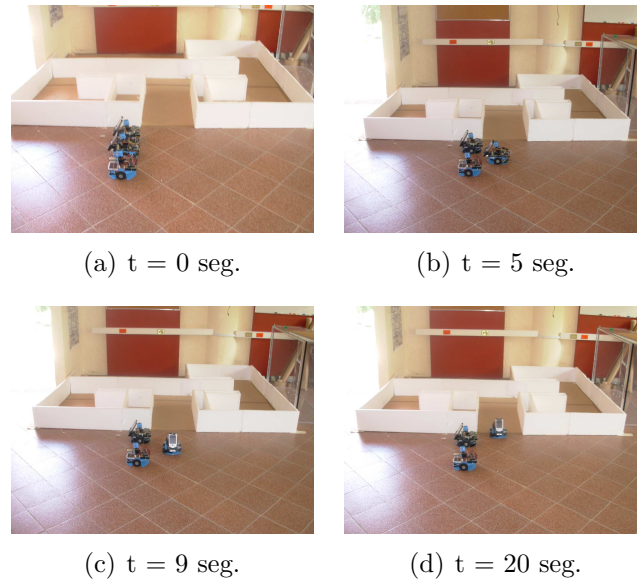


Figura 5.1: Entrada al ambiente, (a) estado inicial del robot al comenzar la prueba, (b) el robot avanza, (c) el robot gira  $90^\circ$  y avanza sensando para detectar la pared del lado derecho, y (d) el robot detecta la pared derecha y se detiene

### 5.1.2. Experimento no. 2 navegación y localización en el ambiente dado un mapa topológico

Las especificaciones del sistema para este experimento son:

$$\mathbf{SMR} = \langle \mathbf{GR}, \mathbf{A} \rangle$$

donde:

$$\mathbf{GR} = \{R_1\} \text{ y } \mathbf{R}_1 = \{Ca, Co\}$$

$\mathbf{Ca} = \{ \text{avanzar, girar, detenerse, medir\_distancia\_odómetro, medir\_distancia\_sonar} \}$

$\mathbf{Co} = \{ \text{detectar\_pared, corregir\_posición, identificar\_nodo, estimar\_posición, seguir\_pared, navegar, planificar\_trayectoria} \}$ .

$\mathbf{A}$  es el ambiente del robot definido por un mapa topológico esto es:

$$\mathbf{A} = \langle \mathbf{MT} \rangle$$

donde:

$\mathbf{MT}$  = Mapa topológico definido por un par de conjuntos finitos de vértices y caminos  $\langle \mathbf{V}, \mathbf{C} \rangle$

y para el cual:

$$\mathbf{V} = \{n1, n2, n3, n4, n5, n6, n7, n8, n9, n10, n11, n12, n13, n14, n15, n16, n17\}$$

y

$$\mathbf{C} = \{(n1, n2), (n2, n3), (n2, n12), (n3, n4), (n3, n8), (n4, n5), (n5, n6), (n6, n7), (n7, n4), (n8, n9), (n9, n10), (n10, n11), (n11, n8), (n12, n13), (n13, n14), (n14, n15), (n15, n16), (n16, n17), (n17, n14)\}$$

$\mathbf{MT}$  es ilustrado gráficamente en la Figura 5.6.

**Descripción:** Esta prueba consiste en probar la capacidad de un robot para localizarse en un ambiente dado un mapa topológico del mismo. En este experimento el robot tiene como información *a priori* el mapa del ambiente y su posición inicial, además de un nodo del mapa topológico que le indicaba a dónde dirigirse partiendo de la posición inicial dada previamente. Una vez iniciada la prueba, el robot navega hasta encontrar el nodo del mapa topológico que le ha sido asignado.

**Resultados esperado:** El robot navega desde la zona en el ambiente que representa el nodo inicial en el mapa topológico hasta una segunda zona que representa el nodo final, gira a la derecha para indicar que alcanzó el nodo y se detiene.

**Resultado obtenido:** El robot logró navegar desde la entrada del ambiente la cual representaba el nodo inicial, hasta encontrar una bifurcación en el pasillo principal la cual estaba representada por un nodo en el mapa topológico. Una vez alcanzado dicho nodo, el robot se detuvo por lo que la prueba fue considerada como exitosa. En la Figura 5.2 se pueden apreciar diferentes fases del experimento.

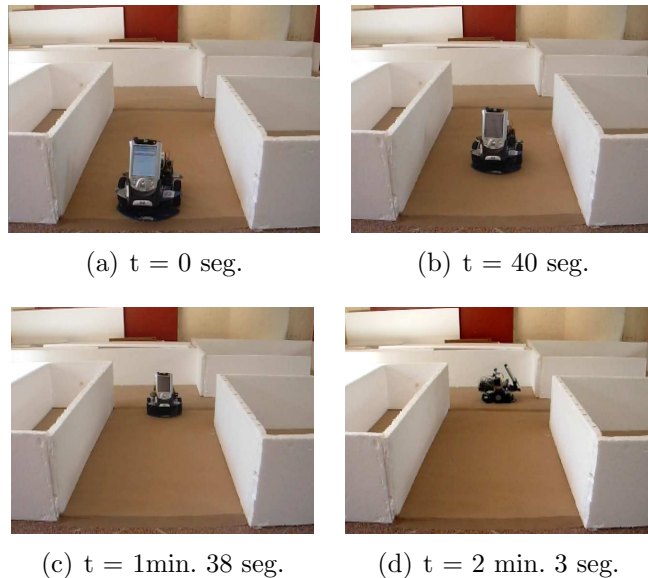


Figura 5.2: Navegación y localización en un ambiente empleando un mapa topológico, (a) estado inicial del robot al comenzar la prueba, (b) el robot avanza corrigiendo su posición en caso de ser necesario, y detectando paredes a la derecha, izquierda y frente, (c) el robot detecta que la pared se terminó y estima su posición, y (d) el robot gira a la derecha y se detiene.

### 5.1.3. Experimento 3: seguimiento de pared y estimación de la posición

Las especificaciones del sistema para este experimento son:

$$\text{SMR} = \langle \text{GR} \rangle$$

donde:

$$\text{GR}, \mathbf{A} = \{ \mathbf{R}_1 \} \text{ y } \mathbf{R}_1 = \{ \text{Ca}, \text{Co} \}$$

$$\text{Ca} = \{ \text{avanzar}, \text{detenerse}, \text{medir\_distancia\_odómetro}, \\ \text{medir\_distancia\_sonar} \}$$

$$\text{Co} = \{ \text{detectar\_pared}, \text{corregir\_posición}, \text{estimar\_posición}, \\ \text{seguir\_pared} \}.$$

$\mathbf{A}$  es el ambiente del robot definido por un mapa topológico esto es:

$$\mathbf{A} = \langle \text{MT} \rangle$$

donde:

$\text{MT}$  = Mapa topológico definido por un par de conjuntos finitos de vértices y caminos  $\langle \mathbf{V}, \mathbf{C} \rangle$

y para el cual:

$$\mathbf{V} = \{ n1, n2, n3, n4, n5, n6, n7, n8, n9, n10, n11, n12, n13, n14, n15, n16, n17 \}$$

y

$$\mathbf{C} = \{ (n1, n2), (n2, n3), (n2, n12), (n3, n4), (n3, n8), (n4, n5), (n5, n6), (n6, n7), \\ (n7, n4), (n8, n9), (n9, n10), (n10, n11), (n11, n8), (n12, n13), (n13, n14), \\ (n14, n15), (n15, n16), (n16, n17), (n17, n14) \}$$

MT es ilustrado gráficamente en la Figura 5.6.

**Descripción.** La prueba consiste en colocar al robot al inicio del pasillo principal para que navegue a través del pasillo siguiendo la pared izquierda del mismo hasta que dicha pared se termine. El largo del pasillo es de  $100\text{cm}$ . Este proceso se debe realizar por pasos, en cada paso el robot se mueve  $5\text{cm}$  y toma tres lecturas de sus sensores de distancia. Con base en dicha información, el robot debe determinar si se ha alejado o acercado a una de las paredes del ambiente, en cuyo caso debe estimar su posición tomando en cuenta la posición anterior y la distancia medida a una de las paredes. Posteriormente, debe aumentar la velocidad de uno de los lados para intentar corregir su posición en el siguiente paso. Una vez alcanzado el final del pasillo el robot debe detenerse.

**Resultado esperado:** El robot encuentra el final de la pared derecha y se detiene.

**Resultado obtenido.** El robot logró seguir la pared derecha del pasillo principal y alcanzó el final de la misma y se detuvo, por lo que la prueba fue considerada como exitosa. Finalmente en la Figura 5.3 se ilustran distintas etapas del experimento.

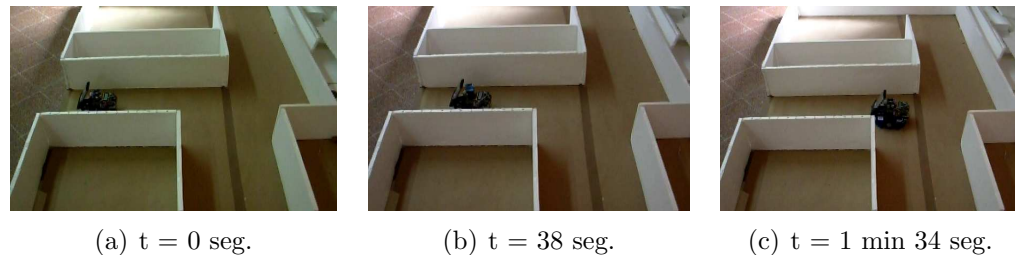


Figura 5.3: Seguimiento de pared y estimación de la pose, (a) estado inicial del robot al comenzar la prueba, (b) el robot sigue una trayectoria paralela a la pared izquierda (d) el robot detecta que la pared izquierda terminó y se detiene.

#### 5.1.4. Experimento 4: generación de un mapa local

Las especificaciones del sistema para este experimento son:

$$\mathbf{SMR} = \langle \mathbf{GR}, \mathbf{A} \rangle$$

donde:

$$\mathbf{GR} = \{ \mathbf{R}_1 \} \text{ y } \mathbf{R}_1 = \{ \mathbf{Ca}, \mathbf{Co} \}$$

$$\mathbf{Ca} = \{ \text{avanzar, detenerse, medir\_distancia\_odómetro,} \\ \text{medir\_distancia\_sonar} \}$$

$$\mathbf{Co} = \{ \text{detectar\_pared, corregir\_posición, estimar\_posición,} \\ \text{seguir\_pared, navegar, actualizar\_mapa\_local} \}$$

$\mathbf{A}$  es el ambiente del robot representado por dos tipos de mapas complementarios, un mapa topológico y un mapa métrico basado en rejillas de ocupación, esto es:

$$\mathbf{A} = \langle \mathbf{MM}, \mathbf{MT} \rangle$$

donde:

$\mathbf{MM}$  = Mapa métrico basado en rejillas de ocupación. Donde la rejilla está representada por una matriz  $M[100][100]$  es decir de dimensiones 100 filas y 100 columnas, donde cada celda representa una porción de  $5 \text{ cm}^2$  del ambiente

$\mathbf{MT}$  = Mapa topológico definido por un par de conjuntos finitos de vértices y caminos  $\langle \mathbf{V}, \mathbf{C} \rangle$

y para el cual:

$$\mathbf{V} = \{ n1, n2, n3, n4, n5, n6, n7, n8, n9, n10, n11, n12, n13, n14, n15, n16, n17 \}$$

y

$$C = \{(n1, n2), (n2, n3), (n2, n12), (n3, n4), (n3, n8), (n4, n5), (n5, n6), (n6, n7), (n7, n4), (n8, n9), (n9, n10), (n10, n11), (n11, n8), (n12, n13), (n13, n14), (n14, n15), (n15, n16), (n16, n17), (n17, n14)\}$$

MT es ilustrado gráficamente en la Figura 5.6.

**Descripción.** La prueba consiste en colocar al robot al inicio del pasillo principal para que navegue a través del pasillo siguiendo la pared derecha del mismo hasta que dicha pared se termine, de manera similar al experimento anterior. Sin embargo, en este caso el robot debe además, construir un mapa local de la zona del ambiente recorrido.

**Resultado esperado:** El robot encuentra el final de la pared derecha y se detiene. Durante este proceso el robot construye un mapa local.

**Resultado obtenido.** El robot logró navegar desde el inicio del pasillo principal hasta la zona del ambiente que representa el nodo final que le fue asignado previamente, además de generar durante todo el proceso un mapa local.

En la Figura 5.4 se presenta un mapa local generado realizando estimación de posición, la línea del centro del pasillo, representado este último por el área blanca, indica la posición del robot durante el experimento.

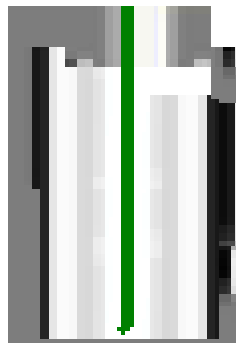


Figura 5.4: Mapa local generado por un robot en el experimento individual de generación automática del mapa local

## 5.2. Experimentos colectivos de coordinación, comunicación, generación y fusión de mapas

Los experimentos colectivos fueron realizados empleando tres robots homogéneos y un servidor remoto. Dichos experimentos fueron realizados con la finalidad de evaluar la capacidad del sistema multirobot para generar mapas métricos tanto del ambiente original como del ambiente una vez que ha sufrido modificaciones, así como un experimento para evaluar la coordinación de los robots y uno más para evaluar la comunicación. Las modificaciones que se realizaron al ambiente original fueron de dos tipos y consistieron en abrir paredes y cerrar pasillos. El objetivo de realizar estas modificaciones fue evaluar la capacidad de sistema para reflejar los cambios en el mapa global generado de un ambiente interior. A continuación se detallan estos experimentos realizados. Las especificaciones del sistema multirobot para los experimentos colectivos son:

$$\text{SMR} = \langle \text{GR}, \text{A}, \text{S} \rangle$$

donde:

$$\text{GR} = \{R_1, R_2, R_3\} \text{ y } R_i = \{Ca, Co\}$$

$$\text{S} = \{Cs\}$$

$$\text{Cs} = \{ \text{recibir\_datos\_robot}, \text{enviar\_datos\_robot}, \text{estimar\_posición}, \text{actualizar\_mapa\_local}, \text{fusionar\_mapa\_local} \}$$

$$\text{Ca} = \{ \text{avanzar}, \text{girar}, \text{detenerse}, \text{detectar\_pared}, \text{medir\_distancia\_sonar}, \text{enviar\_datos\_servidor}, \text{esperar\_datos\_servidor}, \text{enviar\_datos\_robot}, \text{recibir\_datos\_robot} \}$$

$$\text{Co} = \{ \text{detectar\_pared}, \text{corregir\_posición}, \text{buscar\_pared}, \text{recibir\_orden\_de\_acceso}, \text{enviar\_orden\_de\_acceso}, \text{identificar\_nodo}, \text{estimar\_posición}, \text{seguir\_pared}, \text{navegar}, \text{actualizar\_mapa\_local}, \text{acceder\_al\_ambiente}, \text{planificar\_trayectoria} \}.$$

**A** es el ambiente del robot definido por dos tipos de mapas, un mapa topológico y un mapa métrico en este caso basado en rejillas de ocupación, esto es:

$$\mathbf{A} = \langle \text{MT}, \text{MM} \rangle$$

donde:

MM = Mapa métrico basado en rejillas de ocupación. Donde la rejilla está representada por una matriz  $M[100][100]$  es decir de dimensiones 100 filas y 100 columnas, donde cada celda representa una porción de  $5 \text{ cm}^2$  del ambiente

MT = Mapa topológico definido por un par de conjuntos finitos de vértices y caminos  $\langle \mathbf{V}, \mathbf{C} \rangle$

y para el cual:

$$\mathbf{V} = \{n1, n2, n3, n4, n5, n6, n7, n8, n9, n10, n11, n12, n13, n14, n15, n16, n17\}$$

y

$$\mathbf{C} = \{(n1, n2), (n2, n3), (n2, n12), (n3, n4), (n3, n8), (n4, n5), (n5, n6), (n6, n7), (n7, n4), (n8, n9), (n9, n10), (n10, n11), (n11, n8), (n12, n13), (n13, n14), (n14, n15), (n15, n16), (n16, n17), (n17, n14)\}$$

MT es ilustrado gráficamente en la Figura 5.6.

Para todos los experimentos colectivos presentados, fue empleado un servidor consistente en una computadora con procesador AMD Sempro Mobile © con 712 MB de memoria RAM, la cual fue colocada fuera del ambiente a una distancia aproximada de 4 metros. Para la comunicación entre los robots y el servidor fue empleada una red inalámbrica *WiFi*.

En todos los casos, una vez que fueron colocados los robots fuera de ambiente y establecidas las comunicaciones entre ellos y el servidor encargado de recibir la información, se inició la ejecución de la prueba. En una fase

previa a los experimentos, un robot con una trayectoria predefinida generó el métrico MM el cuál fue empleado por el servidor para generar de manera automática el mapa topológico MT del ambiente inicial. El mapa métrico original se muestra en la Figura 5.5, mientras que el mapa topológico original se presenta en la figura 5.6.

Los mapas MM y MT generados en una fase previa a los experimentos colectivos, fueron empleados para medir la capacidad del sistema multirobot para generar y actualizar mapa locales así como fusionarlos en un mapa global. Para ello, se empleó una medida cuantitativa basada en la fórmula de correlación cruzada, la cual mide, para la presente tesis, la similaridad entre dos mapas métricos. La fórmula empleada es la siguiente:.

$$NCC = \frac{\sum_{i=1}^m \sum_{j=1}^n M(i, j)M'(i, j)}{\sum_{i=1}^n \sum_{j=1}^m |M(i, j)|^2} \quad (5.1)$$

donde  $M$  es la matriz que contiene las probabilidades de ocupación asociadas al mapa del ambiente original,  $M'$  es la matriz que contiene las probabilidades de ocupación del mapa del ambiente actualizado. Los dos mapas deben tener dimensiones  $m \times n$ . En todos los experimentos colectivos la correlación de dos mapas se calculó aplicando la fórmula 5.1 implementada en MATLAB 7.0©.

Por otra parte, una medida cuantitativa en el caso de la actualización de mapas de un ambiente con modificaciones, consiste en la verificación visual de los cambios en el mapa.

A continuación se describen y presentan los resultados obtenidos en los cuatro experimentos colectivos.

### 5.2.1. Experimento 1: Actualización de un mapa del ambiente sin modificaciones

El objetivo de esta prueba fue evaluar los mecanismos para la división y asignación automática de las regiones a los robots, el acceso de los robots a las regiones que les fueron asignadas, así como la exploración de las regiones y la generación de los mapas locales que las representan.

**Descripción de la prueba:** Para esta prueba se emplea un ambiente interior de dimensiones  $3 m. \times 2,38 m.$ . La prueba consiste en colocar a

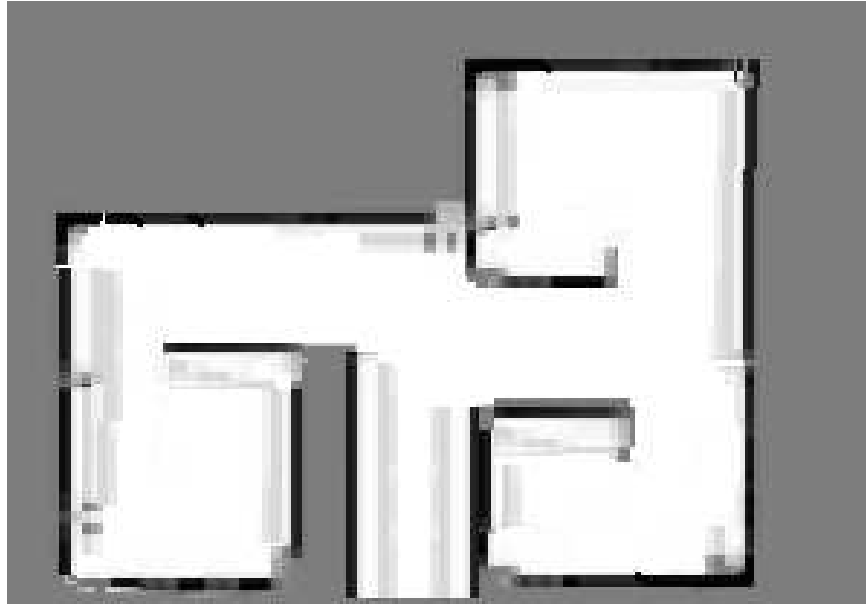


Figura 5.5: Ilustración gráfica del mapa métrico MM para el ambiente de pruebas generado por un robot con trayectoria predefinida.

los tres robots  $R_1$ ,  $R_2$  y  $R_3$  fuera del ambiente, del lado izquierdo de la entrada principal con un ángulo absoluto de  $90^\circ$  y colocados uno junto al otro, empezando con el robot  $R_1$  y terminando con el robot  $R_3$  a una distancia de  $20\text{ cm}$ . uno de otro. Cabe mencionar que el orden en que son colocados los robots fuera del ambiente no necesariamente corresponde al orden para acceder al ambiente, este último debe ser determinado por el servidor con base en la distancia que cada robot debe recorrer para alcanzar la zona asignada. Así mismo, el primer robot debe colocarse a una distancia aproximada de  $20\text{ cm}$ . con respecto a la entrada principal.

Los robots reciben tres piezas de información de parte del servidor: el mapa topológico, el mapa métrico y el nodo al cual dirigirse. El nodo al cual deben dirigirse los robots es asignado por el servidor con base en la distancia que se debe recorrer para acceder hasta la zona asignada (c.f. 4.12). Una vez que un robot ha recibido la información necesaria, debe verificar su identificador para comprobar que es el robot con el identificador  $id_1$ , en caso de coincidir debe ingresar al ambiente y navegar a través de él hasta llegar a la zona que le ha sido asignada identificada por un nodo terminal. En caso contrario, debe esperar a recibir la notificación de acceso exitoso por parte

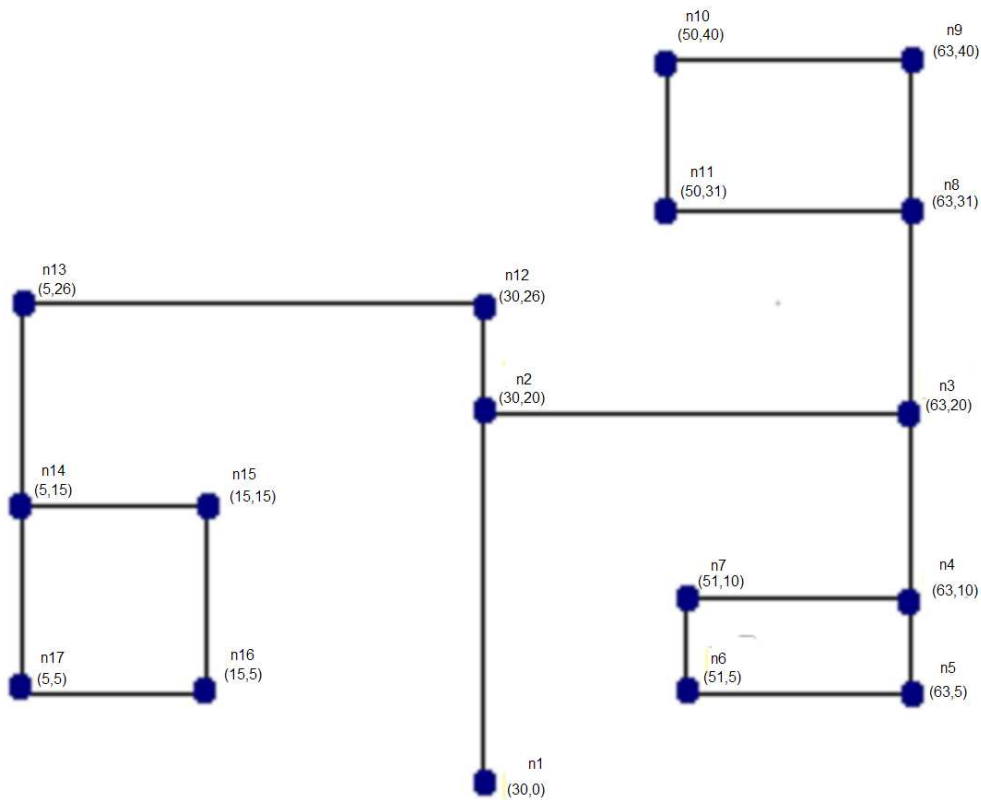


Figura 5.6: Representación gráfica del mapa topológico MT para el ambiente de pruebas. Este mapa fue generado automáticamente por el servidor.

de otro robot para poder acceder al ambiente.

Cuando un robot que esté navegando en el ambiente alcance el primer nodo del mapa topológico siguiente al nodo inicial, debe enviar una notificación de acceso exitoso al siguiente robot en turno y así sucesivamente. En todo momento de la navegación los robots deben estimar su posición y tomar lecturas de los sensores de distancia en intervalos de 5 *cm.*, dicha información debe ser enviada al servidor para que éste generara los mapas locales y los fusione en un único mapa global, además de mostrar en cada etapa los mapas generados. La prueba concluye en cuanto el último robot terminara su recorrido.

**Resultado esperado:** El servidor logra la división del mapa topológico así como la asignación de las zonas a los robots. Los robots logran entrar al ambiente, realizan el recorrido y envían la información necesaria al servidor para la generación de los mapas locales. Por último, el servidor logra generar los mapas locales y fusionarlos con el mapa métrico previamente generado, además de mostrar en pantalla la actualización y fusión de mapa global.

**Resultados obtenidos:** Los tres robots  $R_1$ ,  $R_2$  y  $R_3$  junto con el servidor  $S$  lograron cumplir con todas las tareas antes mencionadas y generar tres mapas locales los cuales fueron fusionados con el mapa global previamente generado para su actualización. El mapa topológico del ambiente se muestra en la figura 5.6. En la Figura 5.7 se muestra una secuencia de imágenes tomadas durante el experimento. En la Figura 5.8 se muestran los mapas locales generados por cada uno de los robots, las trayectorias que siguieron los robots y el mapa global fusionado, por lo que la prueba fue considerada como exitosa. La correlación entre el mapa generado por los tres robots (5.8) y el mapa global generado por un robot con la trayectoria previamente programada (5.5) fue de 93,1546. Dicho valor indica que el mapa generado por el sistema multirobot es muy similar al mapa original.

### 5.2.2. Experimento 2: Actualización de un mapa del ambiente con modificaciones

El objetivo de esta prueba fue evaluar la capacidad del sistema multirobot para contender con un ambiente cuyas condiciones iniciales fueron modificadas, i.e. algunas paredes fueron cortadas y un pasillo fue cerrado. Además de esto, se evaluó la capacidad del sistema para reflejar dichos cambios en el mapa métrico original. La dificultad de explorar un ambiente con modificaciones radica en que este ambiente a pesar de basarse en un mapa conocido, es un ambiente alterado, que para fines prácticos, es un ambiente nuevo para los robots. Ahora bien, cada uno de los robots se basa en características del ambiente tales como pasillo y bifurcaciones para localizarse con ayuda del mapa topológico, si dichas condiciones cambian, el robot puede creer estar en un punto cuando en realidad se encuentra en un punto distinto.

**Descripción de la prueba:** Para esta prueba se debe emplear el mismo

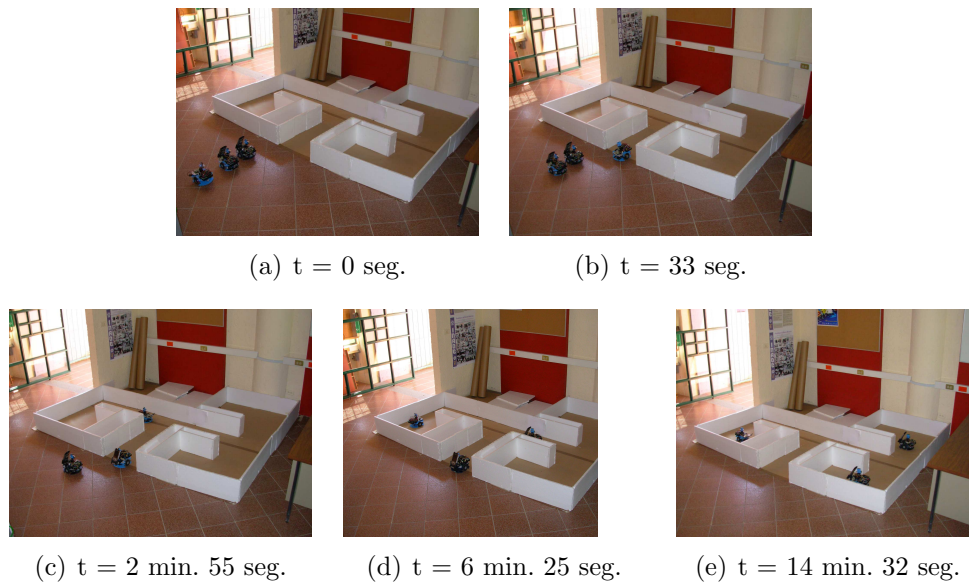


Figura 5.7: Experimento colectivo 1, a) estado inicial de los robots fuera de ambiente, b) el primer robot entra al ambiente, c) el segundo robot entra al ambiente, d) el tercer robot entra al ambiente y e) posición final de los robots.

ambiente de dimensiones  $3\text{ m} \times 2,38\text{ m}$  de la prueba anterior de los experimentos colectivos con las modificaciones que se pueden apreciar en la figura 5.9. Dichos cambios consisten en realizar tres aberturas de  $20\text{ cm}$ . a dos paredes distintas del ambiente así como bloquear la entrada a una de las regiones. Las tareas que el sistema multirobot debe ejecutar son las mismas que las descritas para la prueba anterior de los experimentos colectivos.

**Resultado esperado:** El servidor logra la división del mapa topológico así como la asignación de las regiones a los robots. Los robots logran entrar al ambiente, realizar el recorrido conteniendo con los cambios en el ambiente y envían la información necesaria al servidor para la generación de los mapas locales. Por último, el servidor logra generar los mapas locales y fusionarlos con el mapa métrico previamente generado, así como reflejar en el mapa métrico los cambios del ambiente.

**Resultados obtenidos:** Los tres robots  $R_1$ ,  $R_2$  y  $R_3$  junto con el ser-

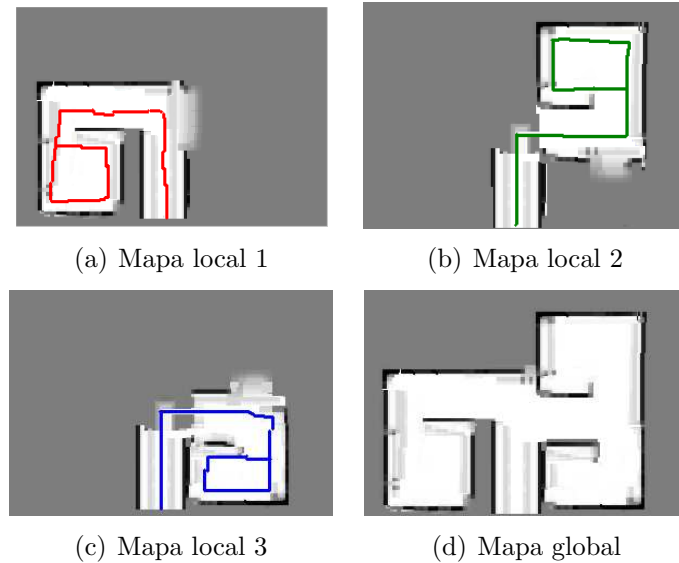


Figura 5.8: Mapas métricos generados por el sistema multirobot en el experimento colectivo 1. a) Mapa local generado por el robot  $R_1$ , b) mapa local generado por el robot  $R_2$ , c) mapa local generado por el robot  $R_3$ , d) mapa global fusionado por el servidor  $S$ .

vidor  $S$  lograron cumplir con todas las tareas antes mencionadas y generar tres mapas locales, los cuales fueron fusionados con el mapa global previamente generado para su actualización. En el mapa métrico se reflejaron los cambios realizados al ambiente, por lo que la prueba fue considerada como exitosa (ver Figura 5.11). La correlación entre el mapa generado por los tres robots (5.11 d)) y el mapa global generado por un robot con la trayectoria previamente programada (5.5) fue de 90,2312, esta correlación es menor que la correlación obtenida del ambiente sin modificaciones. Lo anterior indica que el mapa presenta cambios, en este caso las modificaciones realizadas intencionalmente al ambiente.

Es importante señalar que las probabilidades *a priori* proporcionadas por el servidor a los robots, son probabilidades de ocupación y de no ocupación muy cercanas a 0 y a 1 respectivamente, por lo que cuando un robot intentaba actualizar las probabilidades en las zonas con modificaciones el cambio no se reflejaba en el mapa métrico. Para resolver este problema, fue necesario



(a) Pasillo original.



(b) Pasillo con obstrucción en la entrada del extremo superior izquierda.



(c) Pasillo original.



(d) Pasillo con abertura en el extremo superior central.

Figura 5.9: Dos de los cuatro cambios realizados al ambiente original de pruebas.

reducir las probabilidades *a priori* antes de ser entregadas a los robots. Con esto se dio mayor importancia a las nuevas lecturas de los sensores y los cambios se reflejaron correctamente en el mapa global fusionado. Por otro lado, como en este experimento se cerró el acceso a una zona, dicha zona no fue actualizada, lo anterior se muestra en la Figura 5.11) a) con líneas entre cortadas sobre la zona no actualizada.

### 5.2.3. Experimento 3: Comunicación y coordinación entre los miembros del sistema multirobot

El objetivo de esta prueba fue evaluar la comunicación entre los miembros del sistema multirobot, así como la capacidad de sus miembros para coordinarse entre ellos. Para esto se diseñó una prueba que tomara en cuenta estas dos características de los robots.

**Descripción de la prueba:** Los robots son colocados fuera del ambiente

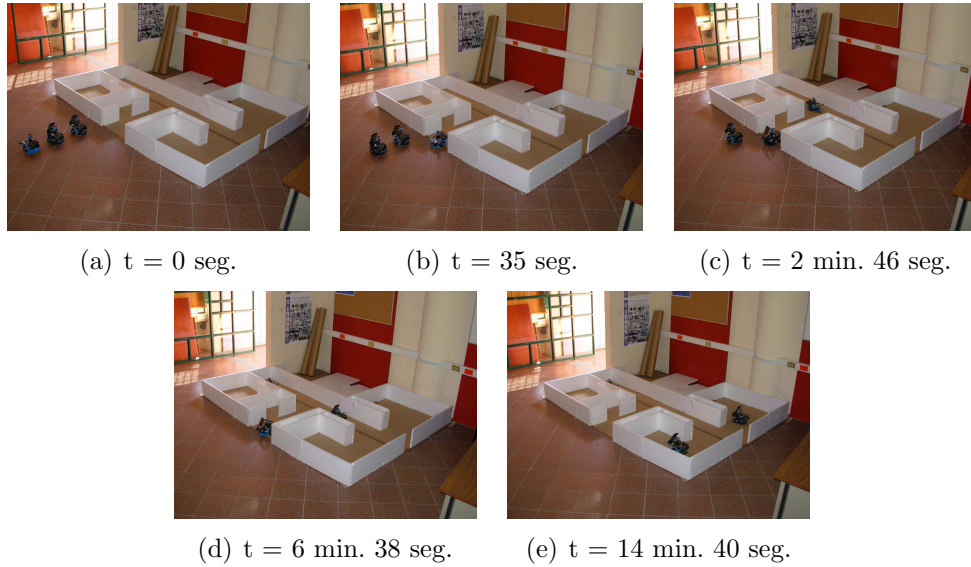


Figura 5.10: Experimento colectivo 2, a) estado inicial de los robots fuera del ambiente, b) el primer robot entra al ambiente, c) el segundo robot entra al ambiente, d) el tercer robot entra al ambiente y e) estado final de los robots, cada uno al interior de la zona que le fue asignada.

como en todos los experimentos colectivos. El servidor  $S$  debe enviar las piezas de información correspondientes a cada uno de los robots, consistentes en el orden de acceso y, para este experimento el mapa topológico. Una vez que los robots hayan recibido la información se inicia la ejecución de la prueba. Cada uno de los robots debe acceder a la zona que le fue asignada sin crear mapas locales, únicamente navegando en el ambiente. Cuando cada uno de los robots alcance el punto final debe comunicar a los demás miembros del sistema que ha accedido a la zona asignada por el servidor. Una vez que todos los robots hayan alcanzado la zona asignada y comunicado a todos los miembros del sistema su estado, cada uno de ellos debe salir del ambiente bajo los siguientes criterios:

- El primer robot en acceder al ambiente es el primero en salir, sin ninguna restricción, i.e. no es necesario esperar ningún tipo de información de otro robot. Cuando el primer robot se encuentre en la salida, debe comunicar su estado al siguiente robot en turno, con base en el orden

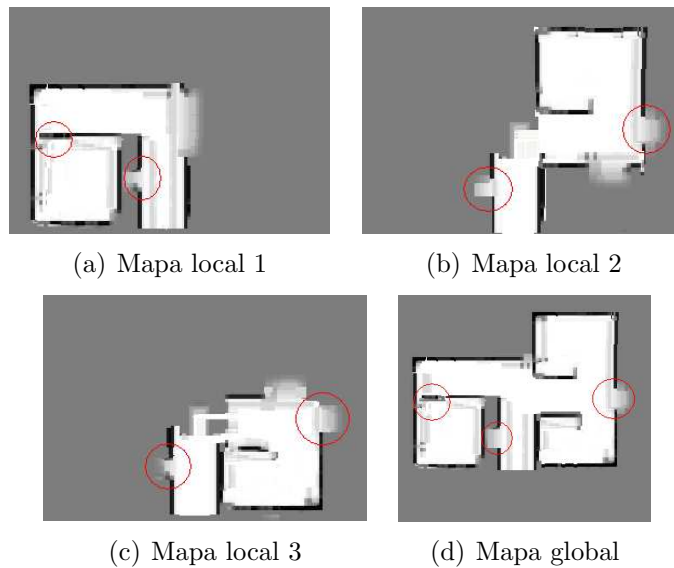


Figura 5.11: Mapas métricos generados por el sistema multirobot en el experimento colectivo 2. Los cambios en el ambiente original se pueden apreciar remarcados por círculos en las figuras. a) Mapa local generado por el robot  $R_1$ , b) mapa local generado por el robot  $R_2$ , c) mapa local generado por el robot 3  $R_3$ , d) mapa global fusionado por el servidor  $S$ .

de entrada, para que éste pueda salir.

- Todos los robots que no sean el primer robot que entró al ambiente, deben esperar la notificación de otro robot para salir del ambiente.
- Si un robot que no es el primer robot que entró al ambiente, no recibe la notificación de que los otros robots han salido del ambiente después de un tiempo determinado por:

$$\text{orden\_de\_acceso} \times ct$$

donde  $ct$  es una constante de tiempo, para este experimento del orden de 10 *minutos*, a partir del momento en que el robot terminó su recorrido por la zona asignada, dicho robot debe salir del ambiente.

El recorrido que cada robot debe realizar para salir del ambiente, es el mismo que el empleado para acceder a la zona asignada pero en sentido inverso.

**Resultados obtenidos:** Los tres robots  $R_1$ ,  $R_2$  y  $R_3$  lograron entrar al ambiente en el siguiente orden  $R_1$ ,  $R_3$ ,  $R_2$ . El primer robot en entrar al ambiente fue el primero en alcanzar el punto que le fue asignado y así sucesivamente en el orden que entraron al ambiente. Cuando el primer robot alcanzó su zona, esperó hasta que recibió la confirmación de que los otros dos robots habían alcanzado sus respectivas zonas. En ese momento, el primer robot comenzó a navegar para salir del ambiente siguiendo la ruta calculada inicialmente con el mapa topológico pero en sentido inverso. Una vez que el primer robot logró salir del ambiente, éste comunicó al segundo robot su estado fuera del ambiente y el segundo robot comenzó su recorrido para salir del ambiente. El tercer robot de manera similar comenzó su salida una vez recibida la notificación del segundo robot. Cada vez que un robot salía del ambiente, éste era colocado a un lado de la entrada de forma manual, de tal forma que no interfiriera con la salida de los demás robots.

En la Figura 5.12 se pueden apreciar distintas fases del experimento, desde la entrada de los robots hasta su salida.

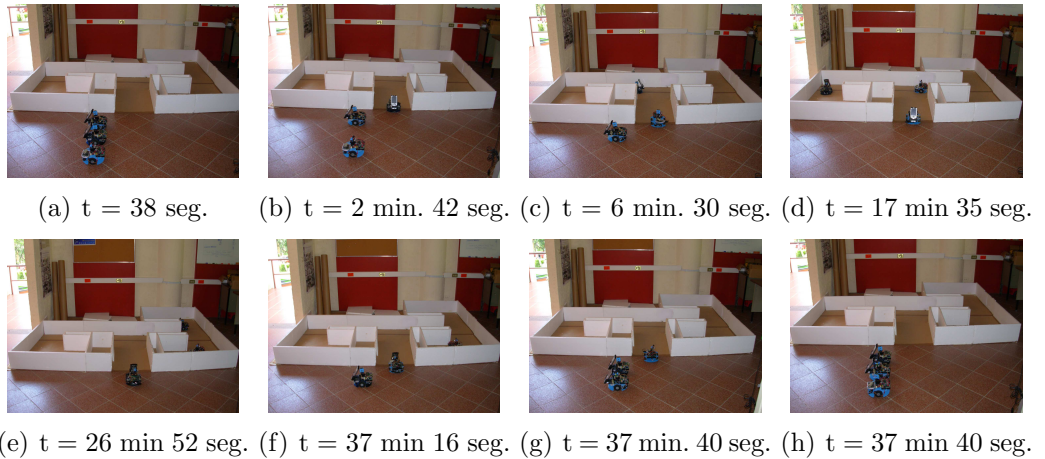


Figura 5.12: Experimento colectivo 3, a) estado inicial de los robots fuera del ambiente, b)  $R_1$  entra al ambiente, c)  $R_2$  entra al ambiente, d)  $R_3$  entra al ambiente, e) salida del robot  $R_1$  f) salida del robot  $R_2$  g) salida del robot  $R_3$  y h) estado final de los robots.

#### 5.2.4. Experimento 4: Actualización de un mapa del ambiente sin modificaciones y sin comunicación con el servidor

El envío de datos al robot para la actualización del mapa global basado en rejilla de ocupación se puede realizar de dos maneras, mediante comunicación continua y por medio de comunicación intermitente (c.f 4.12.3). Si la comunicación es continua, cada robot miembro del sistema envía la información al servidor en cada momento y no es necesario esperar a que todos los robots salgan del ambiente para tener el mapa global actualizado. En el caso de la comunicación intermitente, si la información se pierde sólo en intervalos pequeños del orden de 30 *segs.*, el mapa también es actualizado en línea. Sin embargo, cuando la información se pierde completamente durante todo el proceso de navegación del sistema multirobot, es necesario esperar a que todos los robots salgan del ambiente y que la comunicación se reestablezca para recuperar la información que los robots tienen almacenada en su memoria interna y de esta forma actualizar el mapa global.

El objetivo de esta prueba fue evaluar las capacidades de generación de mapas locales y la fusión del mapa global cuando la comunicación con el servidor se pierde totalmente.

**Descripción de la prueba:** El servidor  $S$  debe establecer comunicación con los tres robots  $R_1$ ,  $R_2$  y  $R_3$  y enviarles las piezas de información necesarias para la navegación del ambiente. Una vez recibida la información el servidor fue interrumpido de manera manual, de tal forma que se perdiera la comunicación entre los robots y el servidor. Los robots deben entrar al ambiente, navegar hasta sus zonas asignadas, almacenando en todo momento la información de sus sensores, y salir del ambiente bajo el esquema planteado en el experimento colectivo 3. Cuando los robots hayan salido del ambiente se reestablece la comunicación del servidor, de tal manera que los robots se comuniquen de nuevo con él y envíen la información recolectada a lo largo de la navegación. Conforme el servidor reciba la información de cada robot, debe generar los mapas locales y fusionarlos con el mapa global de forma transparente, i.e. el servidor no percibe ni considera que los robots se encuentran o no navegando en el ambiente.

**Resultados esperados:** Los tres robots entran a sus zonas asignadas y recuperan la información necesaria para actualizar el mapa global sin comuni-

cación con el servidor. Cuando los robots salen del ambiente, la comunicación con el servidor se reestablece y los tres robots transmiten toda la información almacenada en su memoria interna, el servidor  $S$  la recibe, genera los mapas locales y los fusiona en un único mapa global.

**Resultados obtenidos:** Los tres robots  $R_1$ ,  $R_2$  y  $R_3$ , una vez recibida la información del servidor, entraron al ambiente, lo recorrieron sin tener comunicación con el servidor y lograron salir del ambiente sin problemas. Una vez que los robots salieron del ambiente se reestableció la comunicación con el servidor y los robots enviaron la información de sus sensores recolectada durante el proceso de navegación. El servidor recibió la información, generó los mapas locales y los fusionó en un mapa global. En la Figura 5.13 se muestran los mapas locales generados por el servidor una vez reestablecida la comunicación, así como el mapa global fusionado. La correlación entre el mapa generado por los tres robots (Figura 5.13 d)) y el mapa global generado por un robot con la trayectoria previamente programada (Figura 5.5) fue de 94,124, dicho valor indica que el mapa generado por el sistema multirobot es muy similar al mapa original.

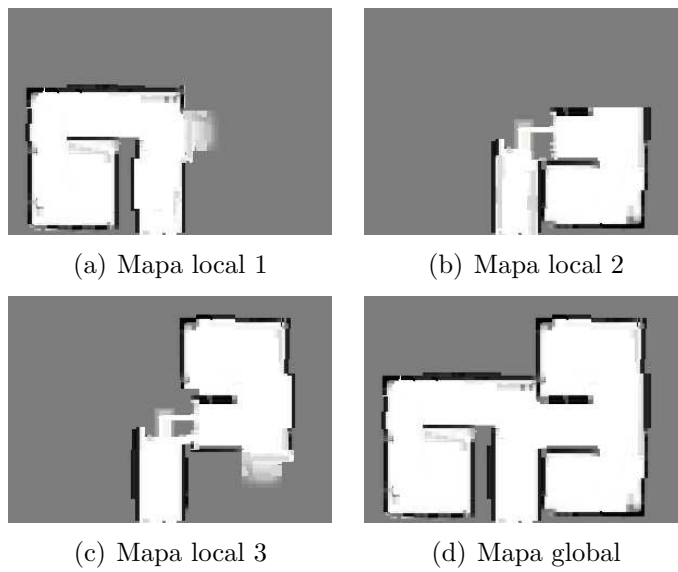


Figura 5.13: Mapas métricos generados por el sistema multirobot en el experimento colectivo 4. a) Mapa local generado por el robot  $R_1$ , b) mapa local generado por el robot  $R_1$ , c) mapa local generado por el robot  $R_1$ , d) mapa global fusionado por el servidor  $S$ .



# Capítulo 6

## Conclusiones

### 6.1. Recapitulativo

El presente trabajo de tesis consistió en el diseño y construcción de un sistema para la fusión de mapas locales en un único mapa global. Se equiparon tres robots móviles homogéneos con capacidades de acción, percepción y comunicación. La construcción de cada robot se dividió de manera general en dos módulos distintos, el módulo de bajo nivel y el módulo de alto nivel. El módulo de bajo nivel contiene los dispositivos encargados de la actuación del robot y consiste de una tarjeta controladora Handyboard©. Este módulo fue el encargado de ejecutar todas las acciones ligadas directamente a los sensores y actuadores del robot. Por otra parte, para el módulo de alto nivel se empleó una Pocket PC© y sus principales tareas fueron la generación y almacenamiento de los mapas locales, y la comunicación, de manera inalámbrica, con los demás miembros del sistema multirobot. La capacidad sensorial de cada miembro del sistema fue limitada y se basó únicamente en un sonar, un sensor de proximidad infrarrojo y un odómetro..

Para la representación, tanto de los mapas locales como del mapa global fue empleado un modelo de ambiente conocido como rejilla de ocupación. Un mapa basado en rejilla de ocupación discretiza el ambiente en celdas de igual dimensión, cada una de las cuales mantiene un estado de ocupación (1) o de no ocupación (0) de la zona del ambiente asociada a dicha celda.

Para actualizar el mapa global basado en rejillas de ocupación se propuso

dividir de manera automática el ambiente en zonas a las que cada miembro del sistema multirobot pudiera acceder. La división fue realizada empleando un segundo modelo del ambiente basado en un tipo de representación conocido como mapa topológico. Un mapa topológico también conocido como mapa relacional, es aquel que representa mediante un grafo zonas distintas del ambiente y las relaciones existentes entre ellas. La generación del mapa topológico se realizó partir de un mapa basado en rejillas de ocupación previamente generado por un robot con una trayectoria predefinida. Una vez generado el mapa métrico inicial, éste fue empleado para generar un mapa topológico de manera automática en una computadora que fungió como servidor dentro del sistema multirobot.

Se definieron los tipos de ambientes en los cuales el sistema multirobot se podía desempeñar. Las características principales fueron, un pasillo principal el cual podía tener bifurcaciones, una única entrada al ambiente y  $n$  zonas disjuntas conectadas directamente al pasillo principal. Estos ambientes comparten diversas características que se pueden encontrar en algunos edificios, principalmente de oficinas.

Para la navegación y localización de los robots en los ambientes definidos se propuso un esquema que utilizó tanto el mapa topológico como características del ambiente tales como paredes y bifurcaciones de los pasillos derivadas del mismo mapa topológico.

La fusión de los mapas locales generados por los miembros del sistema multirobot se realizó aplicando la regla de Bayes en el servidor. La información colectada por los robots durante su exploración del ambiente fue enviada al servidor. Para contener con los problemas de comunicación que pudieran surgir durante la exploración colectiva del ambiente, se propusieron dos esquemas de comunicación entre los robots y el servidor, dependiendo de si existía o no comunicación, llamados comunicación continua y comunicación intermitente.

Todos los mecanismos de coordinación y actuación de los robots antes mencionados fueron reflejados en la definición del sistema multirobot dada en la sección 4.3. De esta manera se propusieron e implementaron las habilidades y capacidades de los miembros de sistema. Las habilidades de cada robot que fueron programadas son **avanzar**, **girar**, **detenerse**,

`medir_distancia_ir`, `medir_distancia_sonar`, `enviar_datos_servidor`, `esperar_datos_servidor`, `enviar_datos_robot`, `recibir_datos_robot`. Las capacidades de cada robot fueron `detectar_pared`, `corregir_posición`, `buscar_pared`, `recibir_orden_de_acceso`, `enviar_orden_de_acceso`, `identificar_nodo`, `estimar_posición`, `seguir_pared`, `navegar`, `actualizar_mapa_local`, `acceder_al_ambiente`, `planificar_trayectoria`. Las habilidades del servidor que fueron programadas son `recibir_datos_robot`, `enviar_datos_robot`, `estimar_posición`, `actualizar_mapa_local`, `fusionar_mapa_local`.

Las pruebas realizadas fueron propuestas con el objetivo de evaluar la capacidad del sistema multirobot para generar o actualizar un mapa basado en rejillas de ocupación, así como las habilidades y capacidades de manera individual para cada robot. Las pruebas fueron divididas en dos tipos, experimentos individuales y colectivos. Los experimentos individuales se orientaron a evaluar las habilidades y capacidades de cada miembro del sistema multirobot de manera individual. Los experimentos colectivos consistieron en la generación de un mapa basado en rejillas de ocupación a partir de un mapa topológico, mientras que en un segundo grupo de experimentos colectivos el objetivo fue la actualización del primer mapa después de una serie de modificaciones realizadas al ambiente original tales como bloqueo de pasillos y aperturas de paredes. En este segundo grupo de experimentos colectivos los cambios del ambiente fueron reflejados en un 100 % .

## 6.2. Aportaciones

Las aportaciones de esta tesis están divididas en aportaciones materiales y aportaciones conceptuales. A continuación se detallan ambas.

### 6.2.1. Materiales

1. Se construyó un sistema multirobot compuesto por tres robots homogéneos, autónomos, móviles y con capacidades de navegación en un ambiente interior semi-estructurado con características que permiten su representación en forma de un grafo acíclico no dirigido, así como con capacidades de percepción y comunicación entre los miembros del sistema multirobot. El trabajo desarrollado se integrará al acervo del Laboratorio de Robótica de la Coordinación de Ciencias Computacio-

nales del INAOE. Este sistema multirobot, a diferencia de otros sistemas multirobot diseñados y construidos en el Laboratorio de Robótica de esta coordinación, tiene capacidades de comunicación inalámbrica y capacidades de almacenamiento que no existían en los prototipos anteriores. Dichas capacidades fueron logradas después de abundante experimentación y construcción de elementos para el robot que no se pueden conseguir fácilmente en el mercado comercial como el cable para la comunicación serial entre la Pocket PC© y la tarjeta controladora Handyboard©. Se espera que este nuevo sistema multirobot pueda servir de plataforma para futuros experimentos en Robótica colectiva en el Laboratorio de Robótica de la Coordinación.

### 6.2.2. Conceptuales

1. Se propuso un esquema para división de un ambiente en zonas con la finalidad de que los miembros de un sistema multirobot puedan explorarlo abarcando la mayor área del ambiente posible.
2. Se propuso un esquema para la coordinación de los robots con la finalidad de acceder al ambiente evitando colisiones al momento de entrar. Este esquema consiste en que cada robot calcule la distancia que debe recorrer hasta un punto asignado por un servidor externo y comunique dicha distancia a todos los miembros del sistema para que el robot con la mayor distancia a recorrer acceda primero al ambiente.
3. Se propusieron esquemas para la navegación y localización de un robot en un ambiente semi-estructurado basado en un mapa topológico y características del ambiente. A diferencia de otros proyectos, estos esquemas están propuestos para ser implementados en robots con capacidades de navegación y percepción limitadas, específicamente un sensor de distancia ultrasónico, un sensor de distancia IR y dos odómetros.
4. Se propusieron e implementaron dos esquemas para la transmisión de datos entre los robots y la computadora externa que ejerce el papel de servidor dentro del sistema, los esquemas de comunicación continua e intermitente. Estos esquemas permiten la actualización del mapa en tiempo real y su visualización en línea en el servidor, además de contender con los problemas de comunicación que se pueden presentar durante el proceso de actualización del mapa global.

5. Se propuso e implementó un algoritmo para la generación de un mapa topológico, Algoritmo de Abstracción de un Mapa Topológico (**AAMT**), a partir de un mapa métrico basado en rejillas de ocupación.

### 6.3. Conclusiones y perspectivas

Los mecanismos propuestos en la presente tesis para la generación y actualización de mapas basados en rejillas de ocupación a través de la percepción de los miembros de un sistema multirobot funcionaron adecuadamente en los experimentos planteados. A continuación se presentan las conclusiones de este trabajo de investigación.

El comportamiento `seguir_pared` funcionó de manera adecuada para que los robots avanzaran de forma paralela a una pared del ambiente. No obstante, dicho comportamiento está basado únicamente en la información proporcionada por los sensores de distancia, y debido a que dichos sensores pueden presentar ruido o lecturas erróneas, el comportamiento `seguir_pared` puede fallar y provocar que el robot no siga una trayectoria paralela a la pared.

Los esquemas de comunicación continua e intermitente planteados para contender con los problemas de comunicación pueden presentar dificultades si la comunicación se pierde y se recupera de forma continua, debido a que el robot tiene en tal caso que verificar constantemente qué datos deben ser enviados al servidor, y por consiguiente el proceso de actualización del mapa podría saturarse.

El sistema de sensado toma lecturas del sonar en tres posiciones distintas con respecto a la orientación del robot. Dicha discretización fue suficiente para representar el ambiente gracias a que se combinó con pequeños desplazamientos del robot, del orden de 5 *cm.*, que representan una celda en el modelo del ambiente del robot. Además, se debe tomar en cuenta que cualquier tipo de modelo del ambiente contiene únicamente la información necesaria para que el robot ejecute sus tareas.

A futuro se proponen las siguientes modificaciones al trabajo propuesto.

- **Generación del mapa topológico.** El mapa topológico empleado por el sistema multirobot es generado a partir de la información de un mapa métrico. Para fines más generales, el mapa topológico puede ser generado a partir de planos arquitectónicos tipo CAD, o en general, de cualquier representación del ambiente que permita la generación de un mapa topológico.

- **Equipamiento de los robots.** Dotar al robot de mejores dispositivos de percepción y actuación, tales como láseres, cámaras de video, un anillo de sonares etc. Con ello se equipará a los robots con dispositivos que permitan otros tipos de localización y navegación.
- **Esquema de navegación y localización.** Implementar nuevos esquemas de navegación y localización que hagan posible explorar ambientes reales, tales como edificios, oficinas, etc.
- **Tipos de ambientes a explorar.** Extender los mecanismos propuestos para la exploración colectiva de ambientes de los tipos 1 y 3 descritos en la sección 4.1.1.
- **Esquema de comunicación.** Extender el equipamiento y los esquemas de comunicación y coordinación, para permitir por ejemplo, definición de orden de acceso y de salida más flexible, descentralización en los robots de la tarea del servidor, por mencionar algunos.

# Apéndice A

## Handyboard©

La Handyboard© (figura A.1) es una tarjeta controladora comercial que mide  $10,7\text{ cm}$  de largo  $\times 8\text{ cm}$  de ancho, su suministro de energía es a base de baterías recargables de Nicad de  $9,6\text{ V}$ , emplea un microcontrolador 68HC11 de Motorola, incluye:

- 32 *K* de RAM estática
- 4 salidas de motores DC.
- 1 pantalla LCD de  $16 \times 2$  caracteres.
- 1 *knob* o botón selector y una pieza bocina programable por el usuario.
- 7 entradas analógicas y 9 entradas digitales para uso de sensores.
- 1 emisor infrarrojo de  $38\text{ KHz}$  y la base para colocar un receptor infrarrojo de  $38\text{ KHz}$ .
- Puertos de conexión para colocar una tarjeta de expansión de puertos I/O.

La Handyboard© también cuenta con una tarjeta de expansión (figura A.2) que provee las siguientes características:

- 10 entradas adicionales para sensores analógicos
- 4 entradas para activar sensores de LEGOMindstorms© (sensores reflectivos y sensores de rotación)

- 9 salidas digitales
- 6 señales de control de servomotores con suministro de poder de la batería interna de la Handyboard©
- poder externo opcional para servomotores
- espacio para conectar un sistema ultrasónico Polaroid 6500
- área de propósito general
- conectores para conectar la pantalla LCD de la Handyboard©

Las figuras A.1 y A.2 muestran una tarjeta Handyboard© y su módulo de expansión.



Figura A.1: Tarjeta de control Handyboard©.

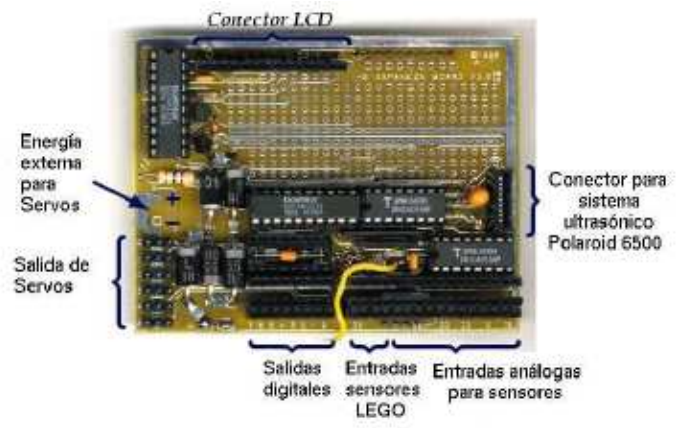


Figura A.2: Tarjeta de expansión de la Handyboard©



Figura A.3: Tarjeta controladora Handyboard© y modulo de expansión



# Apéndice B

## Pocket PCs®

### B.1 IPAQ RX1950

La IPAQ RX1950® (ver Figura B.2) es un Pocket PC® de la compañía Hewlett Packard® de dimensiones  $113,5 \times 70,6 \times 12,7 \text{ mm}$  (alto  $\times$  ancho  $\times$  grosor). Su suministro de energía es a base de una batería de ión de litio recargable de 1100 mAh. Está equipada con un procesador Samsung® SC32442 a 300 Mhz e incluye 32 MB de memoria RAM y 64 MB de memoria ROM que alberga al sistema operativo empleado, Microsoft Windows Mobile 5.0. Cuenta además con una pantalla de 3.50 pulgadas con una resolución de  $240 \times 320$  pixeles y 65536 colores.



Figura B.1: Pocket PC IPAQ RX 1950 de Hewlett Packard.

Para su comunicación, la IPAQ RX1950® está equipada con un puerto de entrada y salida que puede enviar datos y leer datos de manera serial o

USB, en cada uno de los casos, es necesario un conector especial para emplear dicho puerto. Además, cuenta con un puerto infrarrojo y Wi-Fi (802.11b) para comunicación inalámbrica.

## B.2 IPAQ HP H5550

La IPAQ HP H5550© es un Pocket PC© de la compañía Hewlett Packard© de dimensiones  $137,16mm \times 15,23mm \times 83,82mm$  (alto  $\times$  ancho  $\times$  grosor). Su suministro de energía es a base de una batería de ión de litio recargable de 1100 mAh.

La IPAQ H5550 tiene integrado un procesador con la tecnología Intel© XScale a 400 MHz, con 128 MB SDRAM y 48-MB de memoria Flash ROM, y sistema operativo Windows Mobile 2003 Premium Edition. Es posible ampliar la memoria a través de memoria SD y MMC. Cuenta además con una pantalla de 3.8 pulgadas con una resolución de  $240 \times 320$  pixeles y 65536 colores.

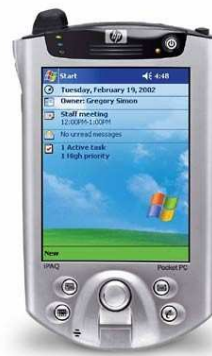


Figura B.2: Pocket PC IPAQ RX 5550 de Hewlett Packard.

Para la comunicación, la IPAQ RX1950© está equipada con un puerto de entrada y salida que puede enviar datos y leer datos de manera serial o USB, en cada uno de los casos, es necesario un conector especial para emplear dicho puerto. Además, cuenta con puerto infrarrojo IrDA, Bluetooth y Wi-Fi (802.11b) estos dos últimos para comunicación inalámbrica.

# Apéndice C

## EWE

Ewe es un sistema de programación que permite escribir aplicaciones empleando Java©. Los programas realizados con Ewe se ejecutan de la misma manera en computadoras de escritorio que en dispositivos móviles. Ewe cuenta con su propia máquina virtual. Sin embargo, los programas de Ewe puede ejecutarse en computadoras sin la máquina virtual de Ewe que cuenten con Java© 1.2 (o compatible). Las aplicaciones pueden también ejecutarse en un navegador de Internet con mínimas modificaciones.

Las características más avanzadas de Ewe son las siguientes:

- Programación del puerto infrarrojo IRDA.
- Una librería gráfica similar a la librería SWING empleada en Java © 1.2.
- Empleo de métodos nativos para acceder a APIs adecuados a diferentes dispositivos móviles.

Ewe se encuentra disponible de forma gratuita para las siguientes plataformas:

- PocketPC © (Ver Figura )
- MS SmartPhone ©
- HandHeldPC Pro ©
- Sharp Zaurus ©

- Computadoras con Linux
- Computadoras con Windows ©
- Computadoras con cualquier versión de la máquina virtual de Java © 1.2 o superior.

### **SDK de Ewe**

El SDK de Ewe incluye los siguientes componentes los cuales son independientes de la plataforma:

1. **La librería de Clases de Ewe - Ewe.jar.** Este archivo es empleado como código de las librerías de clases cuando se compilan y programas, además de emplearse como códigos de librerías, en tiempo de ejecución cuando se ejecutan programas bajo Java 1.2.
2. **El Programa Cosnstructor Jewel - Jewel.ewe y JewelData.jar.** Este programa se emplea para "empaquetar" los archivos de clases en un único archivo con extensión .ewe. También es empleado para convertir aplicaciones Ewe en aplicaciones ejecutables dependientes de la plataforma, e.g archivo EXE en Windows ©.

Además de los archivos incluidos en el SDK también se encuentran los siguientes archivos que se distribuyen de manera separada:

1. **La API de Ewe y la guía para el Desarrollo de Aplicaciones de Ewe.** La API es una lista de todas las clases y métodos de Ewe en formato HTML.
2. **Las máquinas virtuales de Ewe.** Si se desea ejecutar un programa Ewe en una computadora con Java, este se ejecutará empleando el archivo Ewe.jar. Sin embargo, si no se cuenta con la máquina virtual de Java será necesario instalar una máquina virtual de Ewe adecuada a la plataforma de trabajo.
3. **Un compilador de Java.** Este comipilador es necesario para compilar los archivos Ewe.

### **Compilar archivos Ewe**

Para compilar un archivo Ewe se debe especificar al compilador la ruta en donde se encuentra el archivo Ewe.jar además del archivo a compilar de la siguiente manera:

```
javac -classpath c:/ewe/classes/Ewe.jar MyClass.java
```

### **Ejecutar programas Ewe**

Si se desea ejecutar programas empleando la máquina virtual de Java se debe escribir lo siguiente en la línea de comandos:

- (Windows) `java -cp ewe.jar;./ Ewe mypackage.MyClass`
- (Unix ) `java -cp ewe.jar:./ Ewe mypackage.MyClass`

También puede ser empleada la máquina virtual de Ewe de la siguiente forma:

```
ewe mypackage.MyClass
```



# Apéndice D

## Modificación de los servomotores

Para la construcción de robots móviles es necesario contar con un dispositivo con la suficiente fuerza tal que permita mover las ruedas del robot hasta una posición deseada. Lo anterior se puede realizar empleando motores DC. Sin embargo, este tipo de motores en ocasiones no cumple con los requerimientos en cuanto a torque se refiere para lograr el desplazamiento del robot. Se pueden agregar una serie de engranes a los motores DC para alcanzar el torque adecuado; sin embargo, si se cuenta con servomotores, estos ya cuentan con los mecanismos adecuados que le proporcionan la fuerza adecuada al motor.

Un servomotor (ver Figura D.1 ) es un dispositivo similar a un motor de corriente continua, que tiene la capacidad de ubicarse en cualquier posición dentro de su rango de operación, y mantenerse estable en dicha posición. Está conformado por un motor, una caja reductora y un circuito de control. El problema que existe en los servomotores, y por lo que no pueden ser empleados directamente en un robot móvil para accionar las ruedas, es que tienen un rango de giro limitado a  $180^\circ$ .

Los materiales empleados para la modificación de los servomotores son:

- Pinzas de corte
- Desarmador de cruz
- Cautín



Figura D.1: Servomotor HiTec.

- Soldadura
- Cable

Los pasos para modificar un servomotor de tal forma que tenga un giro continuo de  $360^\circ$  son los siguientes:

1. Abrir la tapa posterior del servomotor quitando los cuatro tornillos con un desarmador de cruz de la medida adecuada.
2. Desmontar el servomotor y cortar con las pinzas de corte el circuito interno para el control del motor, y en su lugar soldar dos cables que sirven para controlar el motor desde la Handyboard (ver Figura D.2).
3. Cortar el tope mecánico del engranaje de salida del servomotor para que de esta forma pueda girar libremente  $360$  grados (ver Figura D.3).
4. Colocar el motor en su lugar y cerrar el servo.



Figura D.2: Servomotor modificado. Vista interna.

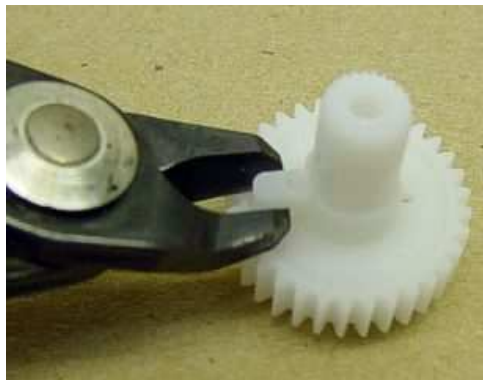


Figura D.3: Corte del tope del servomotor.



# Bibliografía

- Birk A. et Carpin S. (2006). Merging occupancy grid maps from multiple robots. En *Special issue on Multi-Robot Systems*, pp. 1384 – 1397. IEEE Press.
- Fox D., Ko J., Konolige K., Limketkai B., Schulz D., et Stewart B. (2005). Distributed multi-robot exploration and mapping. En *CRV '05: Proceedings of the 2nd Canadian Conference on Computer and Robot Vision*, pp. 1325 – 1339, Washington, DC, USA. IEEE Computer Society.
- Howard A. et Mataric M. J. (2002). Deployment and localization for mobile robot teams. En *Multi-Robot Systems: From Swarms to Intelligent Automata, Proceedings from the 2002 NRL Workshop on Multi-Robot Systems*, pp. 41 – 52. Kluwer Academic Publishers.
- Howard A., Mataric M. J., et Sukhatme G. S. (2002). An incremental deployment algorithm for mobile robot teams. *Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2849 – 2854.
- Howard A., Parker L. E., et Sukhatme G. S. (2004). The SDR experience: Experiments with a large-scale heterogeneous mobile robot team. En *9th International Symposium on Experimental Robotics 2004 (ISER-04)*, pp. 14–23, Singapore.
- Hu H. et Gu D. (2000). Landmark-based navigation of industrial mobile robots. En *Industrial Robot: An International Journal*, volume 27, pp. 458 – 467. Emerald Group Publishing Limited.
- Huang W. H. et Beevers K. R. (2005). Topological map merging. En *The International Journal of Robotics Research*, volume 24, pp. 601–613, Thousand Oaks, CA, USA. Sage Publications, Inc.

- Konolige K., Guzzoni D., et Nicewarner K. (2002). A Multi-agent system for multi-robot mapping and exploration. En *Proceedings of the 2002 NRL Workshop on Multi-Robot Systems*, pp. 11–20. Kluwer Academic Publishers.
- Moravec H. et Elfes A. E. (1985). High resolution maps from wide angle sonar. En *Proceedings of the 1985 IEEE International Conference on Robotics and Automation*, pp. 116 – 121, St. Louis, Missouri, USA.
- Murphy R. R. (2000). *Introduction to AI Robotics*. MIT Press.
- Parker L. E., Fregene K., Guo Y., et Madhavan R. (2002). Distributed heterogeneous sensing for outdoor multi-robot localization, mapping, and path planning. En *Proceedings of the 2002 NRL Workshop on Multi-Robot Systems*, pp. 3–. Kluwer Academic Publishers.
- Pereira G. A. S., Pimentel B. S., Chaimowicz L., et Campos M. F. M. (2002). Coordination of multiple mobile robots in an object carrying task using implicit communication. En *Proceedings of IEEE International Conference on Robotics and Automation (ICRA '02)*, pp. 281–286, Washington, DC.
- Rekleitis I. M., Dudek G., et Milios E. E. (2002). On the positional uncertainty of multi-robot cooperative localization. En *Proceedings of the 2002 NRL Workshop on Multi-Robot Systems*, pp. 3 – 10. Kluwer Academic Publishers.
- Romero L., Morales E., et Sucar L. E. (2006). An exploration and navigation approach for indoor mobile robots considering sensors perceptual limitations. *International Journal of Advanced Robotic Systems*, 3(3):113 – 126.
- Stroupe A. et Balch T. (2002). Mission-relevant collaborative observation and localization. En *Multi-Robot Systems: From Swarms to Intelligent Automata, Proceedings from the 2002 NRL Workshop on Multi-Robot Systems*, pp. 31 – 40. Kluwer Academic Publishers.
- Thrun S. (2001). A probabilistic online mapping algorithm for teams of mobile robots. En *International Journal of Robotics Research*, volume 20, pp. 11–19. Sage Publications, Inc.

- Thrun S., Burgard W., et Fox D. (2005). *Probabilistic Robotics*. The MIT Press.
- Weiss M. A. (1992). *Estructuras de datos y algoritmos*. Addison-Wesley Iberoamericana.
- Yamauchi B. (1997). A frontier-based approach for autonomous exploration. En *CIRA '97: Proceedings of the 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pp. 146–151, Washington, DC, USA. IEEE Computer Society.