## INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY CAMPUS ESTADO DE MÉXICO



# ESTRATEGIAS DE MOVIMIENTO PARA LA LOCALIZACIÓN Y CONSTRUCCIÓN DE MAPAS CON MÚLTIPLES ROBOTS MÓVILES EN INTERIORES

TESIS QUE PRESENTA

**LOURDES MUÑOZ GÓMEZ** 

DOCTORADO EN CIENCIAS COMPUTACIONALES
DCC 03

DICIEMBRE, 2007

## INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY CAMPUS ESTADO DE MÉXICO



### ESTRATEGIAS DE MOVIMIENTO PARA LA LOCALIZACIÓN Y CONSTRUCCIÓN DE MAPAS CON MÚLTIPLES ROBOTS MÓVILES EN INTERIORES

#### TESIS QUE PARA OPTAR EL GRADO DE DOCTOR EN CIENCIAS COMPUTACIONALES PRESENTA

#### LOURDES MUÑOZ GÓMEZ

Asesores: Dr. RICARDO SWAIN OROPEZA

Dr. RAFAEL MURRIETA CID

Comité de Tesis: Dr. ISAAC RUDOMÍN GOLDBERG

Dr. JORGE RAMÍREZ URESTI

Dr. EDUARDO GÓMEZ RAMÍREZ

Dr. ANTONIO MARÍN HERNÁNDEZ

Atizapán de Zaragoza, Estado de México, Diciembre de 2007

A Moisés, Francisco, Elia, Elba, Hilda, a Marissa, a Carlos y a todos mis familiares presentes y ausentes, a todos mis amigos y a todas aquellas personas que muchas veces sin darse cuenta, me dieron un mirada, un gesto, una sonrisa, una frase, una carta que me hizo continuar en los momentos buenos y en los momento difíciles. **LMG** "La imaginación es más importante que el conocimiento." Albert Einstein "... caminante no hay camino, se hace camino al andar ..." Antonio Machado

## ÍNDICE

Capítulo 1. Robótica móvil y construcción de mapas	1
1.1. Robots móviles de interior y de exterior	3
1.2. Navegación en ambientes de interior	5
1.3. Construcción de mapas	8
1.3.1. Construcción de mapas con un sólo robot	9
1.3.2. Construcción de mapas con múltiples robots	12
1.4. Exploración para la construcción de mapas	16
1.4.1. Exploración sistemática	16
1.4.2. Estrategias de movimiento óptimas bajo algún criterio	18
1.5. Planteamiento del problema	21
Capítulo 2. Localización y construcción de representaciones en ambientes de interior	25
2.1. Definición del problema de SLAM	25
2.1.1. El problema de localización y el problema de construcción	26
2.1.2. El problema de localización y construcción simultánea	29
2.2. Representación del ambiente	33
2.2.1. Tipos de representaciones	33
2.2.2. Extracción de segmentos	38
2.2.3. Fusión de la información	41
Capítulo 3. Exploración para la construcción de mapas	45
3.1. Exploración basada en fronteras	45
3.1.1 Procesos de Decisión de Markov	47
3.1.2 Exploración basada en fronteras utilizando MDPs	49
3.2 Exploración con múltiples robots	55
Capítulo 4. Exploración bajo múltiples criterios.	57
4.1. Exploración basada en la generación de muestras	57

4.2. Definición de la función de utilidad a evaluar	60
4.3. Exploración con un robot	69
4.3.1. Considerando un paso adelante en el tiempo	69
4.3.2. Considerando varios pasos adelante en el tiempo	69
4.4. Exploración con múltiples robots	75
4.4.1. Considerando un paso adelante en el tiempo	75
4.4.2. Considerando varios pasos adelante en el tiempo	76
4.5. Exploración con múltiples robots con diferentes características	80
4.5.1. Definición del problema	81
4.5.2. Sistema dinámico con estado de información imperfecta y Programación	
Dinámica.	83
4.5.3. Planteamiento utilizando programación dinámica	86
Capítulo 5. Implementación y resultados	92
5.1. Robots utilizados	92
5.2. Construcción de mapas geométricos usando MDPs	93
5.2.1. Descripción del sistema implementado	93
5.2.2. Experimentos simulados y reales	94
5.3. Exploración utilizando la función de utilidad	99
5.3.1. Simulaciones con un sólo robot	99
5.3.2. Simulaciones con múltiples robots	102
5.3.3. Comparación entre un paso y varios pasos en el tiempo	105
5.3.4. Experimentos con robots reales	107
5.4. Exploración con robots con distintas características	109
Capítulo 6. Conclusiones y trabajo futuro	113
Bibliografía	120
Apéndice A.	126

#### LISTA DE FIGURAS Y TABLAS

- Figura 1.1. Robot móvil de interior y robot móvil de exterior.
- Figura 1.2. Pasos de construcción de un mapa.
- Figura 1.3. Mapa representado por medio de segmentos [Zhang00].
- Figura 1.4. Mapa parcial generado por dos robots móviles [Yamaucho98b].
- Figura 1.5. Trayectorias de robots durante la exploración [Konoliege04].
- Figura 1.6. Generación de muestra para la exploración en [Newman03].
- Figura 2.1. Localización de un robot móvil.
- Figura 2.2. Construcción de un mapa con posiciones conocidas.
- Figura 2.3. Localización y Construcción Simultánea.
- Figura 2.4. Modelo gráfico de SLAM incremental.
- Figura 2.5. Plano de un ambiente de interior.
- Figura 2.6. Ejemplo de una malla de ocupación.
- Figura 2.7. Ejemplo de un mapa geométrico utilizando segmentos de línea.
- Figura 2.8. Ejemplo de un mapa topológico.
- Figura 2.9. Dos conjuntos de puntos formados con los datos de un LRF.
- Figura 2.10. Ajuste de líneas utilizando mínimos cuadrados.
- Figura 2.11. Ejemplo donde se aplica el algoritmo de divide y vencerás.
- Figura 2.12. Fusión de información durante el proceso incremental de SLAM.
- Figura 3.1. Exploración basada en fronteras.
- Figura 3.2. Ciclo que ejecuta un robot mientras explora y construye un mapa.
- Figura 3.3. Mapa parcial del ambiente y fronteras indicadas con líneas punteadas.
- Figura 3.4. Celdas libres y celdas ocupadas.
- Figura 3.5. Probabilidades de transición de una celda a otra.
- Figura 3.6. Ejemplo de trayectoria del robot utilizando un MDP.
- Figura 3.7. Configuración inicial de dos robots antes de iniciar la exploración.
- Figura 4.1. Exploración basada en la generación de muestras cercanas a las fronteras.
- Figura 4.2. Longitud de la frontera y distancia de la muestra a la frontera.
- Figura 4.3. Una posible trayectoria para el robot durante la exploración.

- Figura 4.4. Tres diferentes rutas.
- Figura 4.5. Gráfica que muestra la evaluación de utilidad de las rutas de la figura 4.4.
- Figura 4.6. Función *fmin* que penaliza las posiciones muy cercanas a las paredes.
- Figura 4.7. Robot enfrente de tres fronteras por explorar.
- Figura 4.8. Árbol de búsqueda para visitar las tres fronteras de la figura 4.7.
- Figura 4.9. Árbol de búsqueda con los estados del robot.
- Figura 4.10. Árbol de búsqueda reducido.
- Figura 4.11. Dos robots con tres fronteras por explorar.
- Figura 4.12. Árbol de búsqueda para ejemplo de la figura 4.11.
- Figura 4.13. Secuencia de decisiones para asignar los robots a sus respectivas fronteras.
- Figura 5.1. Vistas del ambiente de interior.
- Figura 5.2. Configuración inicial de los robots antes de comenzar la construcción del mapa.
- Figura 5.3. Secuencia de imágenes durante la construcción de un mapa.
- Figura 5.4. Los robots durante la exploración.
- Figura 5.5. Secuencia de imágenes del proceso de construcción de un mapa utilizando el simulador.
- Figura 5.6. Mapa final obtenido por un sólo robot: con sensor de 360°, con sensor de 180°.
- Figura 5.7. Exploración con un solo robot considerando un paso adelante en el tiempo.
- Figura 5.8. Exploración con un solo robot considerando varios pasos adelante en el tiempo.
- Figura 5.9. Mapa obtenido por 4 robots.
- Figura 5.10. Exploración con dos robots considerando un paso adelante en el tiempo.
- Figura 5.11. Exploración con dos robots considerando varios pasos adelante en el tiempo.
- Tabla 5.1. Comparación: Un paso adelante vs. Varios pasos adelante.
- Figura 5.12. Comparación entre uno y varios pasos. Mapa y trayectoria para un sólo robot.
- Figura 5.13. Ambiente de pruebas para experimentos con robots reales.
- Figura 5.14. Mapas parciales obtenidos conforme avanza la exploración.
- Figura 5.15. Mapas parciales representados por los puntos del láser y alineamiento de mapas.

## CAPÍTULO 1. ROBÓTICA MÓVIL Y CONSTRUCCIÓN DE MAPAS

La robótica móvil, se encarga de estudiar distintos tipos de vehículos autónomos, estos vehículos pueden ser acuáticos, terrestres, aéreos, con extremidades articuladas o con ruedas [Dudek00]. Generalmente se requiere que los robots móviles trabajen en ambientes donde comúnmente laboran personas, por lo que es necesario que los robots móviles puedan desplazarse de un sitio a otro, es decir es necesario que los robots naveguen por el ambiente. Para llevar a cabo esta navegación, es necesario que los robots cuenten con un mapa que les permita identificar donde se encuentran y así llevar a cabo las tareas que se les asignen. Y para que el robot cuente con un mapa, es necesario que se explore el ambiente para conocerlo y generar un modelo del mismo.

Muchos trabajos se han centrado en la generación de mapas lo más preciso posible, considerando que un robot no pueden obtener representaciones perfectas debido a errores al utilizar sensores para descubrir el ambiente o a la falta de exactitud en los movimientos del robot debido a derrapamientos por ejemplo. Sin embargo la meta principal de este trabajo de tesis es el desarrollo de distintas estrategias de movimiento que se utilicen durante la exploración que realizan uno o varios robots para obtener un mapa del ambiente. Se proponen principalmente tres estrategias de movimiento, una sistemática, una óptima y una para múltiples robots con distintas capacidades. La estrategia de movimiento sistemática hace que uno o varios robots siempre

acudan al lugar desconocido más cercano sin importar si es o no conveniente. La estrategia óptima considera información a cerca del medio ambiente y a cerca del robot mismo para decidir cuál es el lugar más conveniente para que el robot o los robots continúen la exploración. Y finalmente una estrategia de movimiento calculada teniendo en cuenta que puede haber robots con mejores sensores que otros y robots con movimientos más precisos que otros.

Esta investigación se realizó en el marco del proyecto "Sensor Based Robotics" 360001-A financiado por NSF-CONACyT y de la Cátedra de Investigación "Robots Colaborativos Multisensoriales" del Tecnológico de Monterrey Campus Estado de México. Durante el desarrollo de este trabajo se realizaron las siguientes actividades:

- Asistencia a la Escuela de Verano en Imágenes y Robótica (Summer School on Images and Robotics, SSIR) en la Universidad de las Américas (UDLA), Puebla, México en el verano del 2003.
- Asistencia al curso de KINEO WORKS, en el Tecnológico de Monterrey, Campus Estado de México (ITESM-CEM), en septiembre del 2003.
- Asistencia a la Escuela de Verano de SLAM (Summer School on Simultaneous Localization and Mapping) en el Laboratoire d'Analyse et d'Architectura des Systèmes du CNRS (LAAS-CNRS) Toulouse, Francia en el verano del 2004.
- Estacia de investigación con el Dr. Seth Hutchinson en el Beckman Institute en University of Illinois at Urbana-Champaign (UIUC) Urbana, USA de septiembre a diciembre del 2004.
- Asistencia al curso "Principles of Robot Motion", impartido por el Dr. Seth Hutchinson en el ITESM-CEM, durante el verano del 2005.
- Publicación de varios artículos en congresos nacionales e internacionales, journals y capítulos de libro, así como en el congreso de Investigación y Extensión del ITEM.
- Participación en el seminario semanal del Centro de Investigación en Matemáticas
   (CIMAT) con la plática "Estrategias de Movimiento para Construcción y Localización

Simultánea con Múltiples Robots" en abril del 2006, en Guanajuato, México.

#### 1.1 ROBOTS MÓVILES DE INTERIOR Y DE EXTERIOR

Un robot móvil es una combinación de varios componentes que se pueden agrupar en uno de los cuatro subsistemas siguientes [Dudek00]:

- a) Localización: Esta formado por el hardware y software necesarios para proveer de movimiento al robot. En la parte de hardware se encuentran por ejemplo los motores, engranajes, ruedas, patas, hélices, etc. Y en la parte de software las librerías necesarias para utilizar dicho hardware. De acuerdo a este subsistema y el medio en el cuál se desplazarán, los robots móviles se pueden clasificar en aéreos, terrestres, acuáticos y del espacio. En este trabajo todos los robots utilizados son terrestres y con ruedas; de manera que de aquí en adelante al referirse a los robots, se asume que se está hablando de robots móviles terrestres con ruedas.
- b) Sensado: Este subsistema permiten a los robots tener información a cerca de ellos mismos o bien del medio ambiente en donde se encuentran. Los sensores que permiten conocer información sobre el robot mismo, son los llamados sensores internos: ejemplos de éstos con los giroscopios, acelerómetros y en general sensores que permiten calcular la odometría (es decir sensores que permitan calcular la posición del robot integrando información incremental a cerca del movimiento). Los sensores que permiten percibir el medio ambiente, son los sensores externos; y entre estos se encuentran por ejemplo sensores táctiles, infrarrojos, de rango, de posicionamiento (GPS) y cámaras. Los sensores de rango más comunes son el *sonar* y el *sensor láser de rango* (laser range finder, LRF). Estos últimos, funcionan lanzando una onda que al rebotar con los obstáculos regresa y es detectada para estimar la distancia a la que se encuentra el objeto.
- c) Razonamiento: Este subsistema es el que permite que el robot sea autónomo, es decir es el

- que permite que de acuerdo a la información sensada, el robot tome una decisión de la acción o acciones que debe realizar.
- d) Comunicación: Estes subsistema está formado por los dispositivos y programas que permiten que el robot se comunique ya sea con otros robots, con computadoras o con operadores.

Otro aspecto a considerar para los robots móviles, es el tipo de ambiente en el cuál se encontrarán. Comúnmente los ambientes se dividen en *ambientes de interior* y *ambientes de exterior*. Se consideran como ambientes de interior generalmente a lugares que se encuentran dentro de edificios, por lo que este tipo de ambientes se caracterizan por ser cerrados, con pisos planos o con poca pendiente, así como con la gran mayoría de estructuras que los forman son del tipo geométrico y con ángulos generalmente rectos (por ejemplo las paredes, los marcos de las puertas, esquinas, cruces de pasillos, etc.). Por otro lado, se consideran como ambientes de exterior a espacios abiertos, que pueden incluir lugares con vegetación, calles y caminos por ejemplo. Los ambientes de exterior de manera general se considera que pueden tener pendientes pronunciadas, piso rugoso y con desniveles; además de que el ambiente puede tener elementos poco estructurados (por ejemplo rocas, arbustos, edificaciones, etc.). También se asume que en los ambientes de interior es más fácil tener controlado algunos aspectos del ambiente como pueden ser iluminación y temperatura, cosa que no sucede en los ambientes de exterior.

De acuerdo a las características del ambiente, se han diseñado robots móviles para ambientes de interior y robots móviles para ambientes de exterior. La principal diferencia entre estos dos tipos de robots se encuentran en sus respectivos subsistemas de locomoción y de sensado. Los robots móviles de interior cuentan con un sistema de locomoción más sencillo como pueden ser ruedas poco anchas, así como sensores que pueden trabajar en condiciones controladas de luz. En cambio los robots móviles de exterior tienen un sistema de locomoción que se pueda adaptar a terrenos desnivelados, por ejemplo los robots de exterior generalmente cuentan con

llantas más anchas, con un sistema de tracción o bien orugas; así como cuentan con sensores ajustables para trabajar con diversas condiciones de luz, u otro tipo de sensores (por ejemplo, es común que los robots de exterior cuenten con un sistema GPS, mientras que para robots de interior esto no es útil). En la figura 1.1, se muestran respectivamente un robot de interior modelo Pioneer 3-DX y un robot de exterior modelo Pioneer 3-AT.

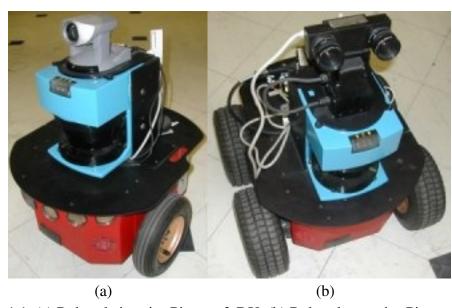


Figura 1.1. (a) Robot de interior Pioneer 3-DX. (b) Robot de exterior Pioneer 3-AT.

Desde el punto de vista de los cuatro subsistemas de un robot móvil, el trabajo desarrollado en esta tesis pertenece al subsistema de razonamiento: el robot debe decidir por donde explorar, es decir hacia donde debe moverse para descubrir el ambiente. Además se consideran solamente que el robot se encontrará en ambientes de interior.

#### 1.2 NAVEGACIÓN EN AMBIENTES DE INTERIOR

Cuando se trabaja con robots móviles, el primer problema a ser resuelto es el problema de la navegación autónoma. La navegación es un proceso incremental que consta de cuatro pasos

#### definidos en [Chatila94]:

- a) Percepción y modelado del ambiente: El robot necesita conocer lo que le rodea y contar con una representación del lugar donde se encuentra.
- b) Localización: El robot necesita saber en dónde se encuentra generalmente utilizando su propia representación del ambiente.
- c) Decisión y planificación de movimiento: El robot debe definir a dónde (meta) y por dónde ir (trayectoria), de acuerdo a una tarea que se le haya asignado.
- d) Ejecución de movimiento: La secuencia de comandos que controlen movimientos para seguir una trayectoria o dirección.

La complejidad de la tarea de navegación se puede ver afectada por el tipo de ambiente en cuál el robot se encuentra. En los ambientes de interior se asume generalmente que el robot se desplazará en una superficie plana o con muy poca pendiente y dentro de espacios considerados como estructurados (donde predominan primitivas geométricas, ángulos rectos y superficies planas). Por lo tanto es común tomar en cuenta éstas características del ambiente para el desarrollo de algoritmos de navegación adecuados en ambientes de interior; por ejemplo definir la navegación de un robot basada en la ubicación de las puertas de las oficinas. Ejemplos de estos ambientes son las oficinas, laboratorios, fábricas, casas, etc. En cambio, en los ambientes de exterior no existe una estructura definida como por ejemplo en los ambientes naturales donde existen árboles, arbustos, rocas, etc. Por lo tanto los algoritmos de navegación en este tipo de ambientes están basados generalmente en objetos que ahí se encuentren, como pueden ser árboles, rocas prominentes, la orilla de senderos, etc. Un ejemplo de navegación en ambientes de exterior se puede encontrar en [Haddad98].

Para que un robot móvil pueda navegar en un ambiente completando los cuatro pasos anteriormente descritos, es necesario que cuente con una representación del ambiente que le permita localizarse de acuerdo a lo percibido y tomar decisiones de hacia dónde dirigirse o cuál

será el siguiente objetivo. El proceso de obtención de una representación o mapa del ambiente se le conoce cómo *construcción* (mapping); en este proceso el robot de manera autónoma va generando un modelo de un ambiente generalmente desconocido (aunque puede ser parcialmente conocido).

Durante el proceso de construcción el robot debe llevar a cabo las siguientes etapas o pasos de manera cíclica o repetida hasta que complete el mapa (ver figura 1.2):

- a) Percibir el ambiente y generar una representación o modelo de lo percibido, y añadir esta nueva información al mapa total que se tenga hasta determinado momento.
- b) Reubicar al robot teniendo como referencia el último mapa actualizado.
- c) Decidir a dónde desplazarse teniendo en cuenta lo que conoce.
- d) Llegar a la siguiente posición para tomar la siguiente lectura.

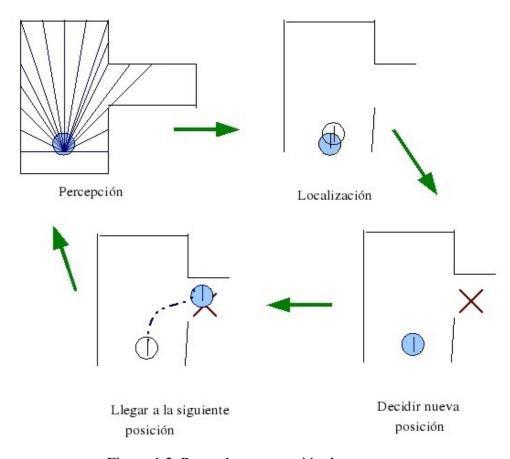


Figura 1.2. Pasos de construcción de un mapa.

Tal como se ilustran los pasos para completar un mapa en la figura 1.2, el robot debe moverse para descubrir nuevas partes del ambiente y a esto se le conoce como *exploración*. Dicho de otra manera, el robot debe explorar el ambiente para poder construir una representación del mismo. Generalmente se buscan estrategias de exploración que permitan obtener la mayor cantidad de información con el mínimo de recursos utilizados por el robot (menor cantidad de movimientos, menor cantidad de procesamiento, menor cantidad de energía utilizada, etc.). Sin embargo, también existen casos en donde la exploración está dirigida por otros criterios como puede ser una mayor cobertura del espacio. Una forma de obtener de manera más eficiente un mapa puede considerar la exploración de varios robots móviles en ambientes desconocidos, cada uno de los cuales genera representaciones parciales del ambiente que se fusionan dando como resultado un mapa completo de un ambiente determinado.

El propósito de esta investigación es la de obtener una representación de un ambiente de interior utilizando varios robots; para lo cuál se consideran diferentes estrategias de movimiento. Dicho de otra manera, esta tesis está enfoca en definir la forma en que varios robots se deben mover de acuerdo a las características sensoriales de cada robot, para obtener una sola representación del ambiente en el que se encuentran.

#### 1.3 CONSTRUCCIÓN DE MAPAS

El problema de construcción de un mapa por un vehículo autónomo, consiste en obtener un modelo o representación de un ambiente físico, utilizando generalmente los sensores de un robot móvil. Por lo tanto, un robot autónomo debe tener la capacidad de explorar ambientes desconocidos, construir representaciones del ambiente y utilizar dichas representaciones para

poder navegar. Estas capacidades, definen el problema de *Localización y Construcción Simultánea* (Simultaneous Localization and Map building, SLAM) [Castellanos99]. Este problema se encuentra estrechamente relacionado con el problema de construcción, y en algunos trabajos se les nombra casi de manera indistinta. En el capítulo 2, se definirán con más detalle ambos problemas.

El problema de SLAM fue definido a partir del trabajo inicial de Smith y Cheeseman [Cheeseman86], en donde se propone un método para estimar la posición y el error entre dos marcos de referencia considerando la incertidumbre debida a los sensores y a los movimientos de un robot. La solución al problema supone que un robot sea capaz de iniciar en un ambiente y localización desconocidas y de manera incremental ir construyendo un mapa (representación) del ambiente al mismo tiempo que se utiliza dicho mapa para conocer su posición. En cambio, en el problema de construcción, se supone como conocida la localización del robot mientras se encuentra sensando el ambiente para construir la representación del mismo. Sin embargo, en ambos casos el resultado de solucionar dichos problemas es la obtención de un mapa del ambiente.

A continuación se describen brevemente algunos trabajos previos relacionados con la obtención de mapas, tanto para el caso en que dichos mapas sean obtenidos por un sólo robot o por múltiples robots.

#### 1.3.1 CONSTRUCCIÓN DE MAPAS CON UN SÓLO ROBOT

A lo largo de las últimas dos décadas, se han propuesto distintas soluciones para la construcción de mapas. Estas soluciones generan mapas con diferentes tipos de representación del ambiente y además han utilizado gran variedad de sensores para el paso de la percepción. Los diferentes tipos de representación de un ambiente se detallan en el capítulo 2, sin embargo a continuación se

mencionan algunas de las posibles representaciones para hacer referencias a los trabajos previos.

Entre los primeros trabajos para la obtención de un mapa, se encuentra el desarrollado en [Elfes87], en el cuál se representaba el mapa del ambiente por medio de una *malla de ocupación*. La idea de este tipo de mapa es dividir el espacio en celdas regulares, indicando si cada celda se encuentra ocupada o vacía con cierto valor de probabilidad asociado. En este caso, se utilizaron sensores de rango, para obtener información del medio ambiente. Este tipo de representación para mapas también ha sido utilizado más recientemente en varios trabajos, por ejemplo el desarrollado en [Yamauchi98a].

Algunos otros trabajos, han optado por una representación del mapa semejante a la que se presenta en los planos en arquitectura. Este tipo de presentación es del tipo geométrica, en donde se utilizan generalmente segmentos de línea para definir las paredes y los contornos de obstáculos, aunque también en algunos casos se utilizan polígonos para definir el mapa [Marín06]. De los primeros trabajos que utilizaron este tipo de representación, son los presentados en [Chatila85] y [Crowles89]; en ambos casos se utilizaron sensores de rango para la obtención de un mapa y consideraron ajustes a los segmentos del mapa debido a los posibles errores durante el desplazamiento del robot. Otros ejemplos de trabajos que utilizan esta representación son los desarrollados en [Zhang00], [Gonzalez94] y [Pfister03]. En estos trabajos, se utiliza un LRF para tomar lecturas en un plano paralelo al piso y con esta información se construyen los segmentos. Un ejemplo de un mapa de segmentos se muestra en la figura 1.3, correspondiente al trabajo desarrollado en [Zhang00]. En la figura se encuentra el robot, representado por un círculo, en la esquina superior derecha del mapa y también se muestra lo que está observando el robot en esa posición. En esta figura se representa con líneas punteadas lo que más adelante se definirá como fronteras, es decir esas líneas punteadas representan el límite entre lo que se conoce del ambiente y lo que se desconoce.

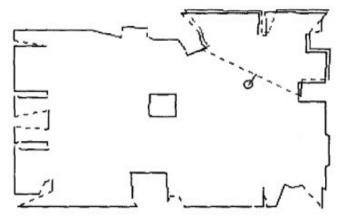


Figura 1.3. Mapa representado por medio de segmentos [Zhang00].

Otro tipo de mapas, son los llamados *mapas topológicos*; en donde el ambiente es representado por medio de un grafo, en el cuál los nodos indican lugares como cuartos, puertas o puntos de reunión; mientras que las aristas indican conexiones entre esos lugares. Ejemplos de mapas topológicos se muestran en [Korten94] y [Huang04]. El primero [Korten94] utiliza como sensores varios sonares y un sistema de visión para encontrar intersecciones y puertas que definirán los nodos del mapa, en cambio en el segundo [Huang04] se utilizan sensores infrarrojos y el robot explora el ambiente siguiendo el contorno de la pared para encontrar esquinas que definirán los nodos del grafo.

Finalmente se mencionan un ejemplo, en el cuál el mapa está formado por un conjunto de *marcas* (landmarks) u objetos prominentes dentro del ambiente, este tipo de representación es muy común en los trabajos en donde se resuelve el problema de SLAM utilizando el filtro extendido de Kalman [Choset05], como se verá en el capítulo 2. En el trabajo desarrollado en [Dissanayake01], el ambiente contaba con un conjunto de postes que funcionaban como landmarks. Las posiciones de los postes en el ambiente se iban añadiendo de manera incremental hasta obtener la representación completa del total de landmarks. Al final, en estos casos, el mapa consiste de una lista con las posiciones de todas los landmarks que se encuentran en el ambiente.

#### 1.3.2 CONSTRUCCIÓN DE MAPAS CON MÚLTIPLES ROBOTS

La construcción de mapas con múltiples robots es un problema que se ha abordado en los últimos años, probablemente debido a que existen mayores posibilidades de contar con más de un robot, generalmente robots pequeños de bajo costo. Sin embargo existen algunos trabajos de múltiples robots a inicios de la década de los noventa que fueron implementados solamente a nivel de simulación, como el desarrollado en [Singh93], que propone la creación de un mapa utilizando múltiples robots con sensores de rango y cada uno de distinto tamaño. Al asumir la existencia de robots de diferentes tamaños, a cada robot se le asignaba un área por descubrir dependiendo si podía pasar o no por ciertas partes del mapa debido a su tamaño.

Al igual que los mapas generados con un robot, en el caso de múltiples robots distintos autores han propuesto soluciones con distintas representaciones, siendo las más comunes la malla de ocupación y los mapas formados por un conjunto de landmarks. A continuación se mencionan trabajos realizados de construcción utilizando múltiples robots, que se encuentran ordenados de acuerdo al tipo de representación que utilizan. De cada trabajo se describe el tipo de representación utilizada, la estrategia de exploración (si es que existe alguna) y la forma en que se obtiene el mapa.

La forma más sencilla de representar el ambiente simplemente consiste en ir almacenando la información obtenida como una nube de puntos que define los contornos de objetos en el ambiente, esta nube de puntos pueden ser los datos obtenidos directamente por un sensor de rango. Un ejemplo de este tipo de mapas se puede encontrar en [Thrun00], donde se propone la cooperación de dos robots. Cuando un robot cuenta con una representación parcial del ambiente, un segundo robot puede cooperar si se puede localizar en el mapa del primero, no se propone una estrategia de exploración y todos los robots trabajan con un láser.

Algunos ejemplos de trabajos donde se utilizó como representación un conjunto de landmarks, son los trabajos mostrados en [Stroupe03], [Williams03] y [Fenwick04]. En [Stroupe03] la estrategia de exploración consiste en que cada robot selecciona la posición a donde se va a desplazar teniendo como criterio la reducción de la incertidumbre en la posición de un landmark en particular. La medida de la incertidumbre tomo en cuenta la posición actual del robot en cuestión, la del resto de los robots y la capacidad de los sensores simulados con que cuenta. El trabajo propuesto en [Williams03], se centra en el proceso de construcción del mapa. En este caso cada robot va construyendo un mapa local con un marco de referencia propio, al final los mapas locales son integrados en un solo global teniendo en cuenta las transformaciones correspondientes entre cada uno de los marcos de referencia locales. En cambio en [Fenwick04], se requiere en todo momento conocer la posición de todos los robots en un mismo marco de referencia. Esto debido a que se tiene en un solo vector de estado las posiciones de todos los robots y de todos los landmarks. Cabe aclarar que en estos últimos dos trabajos no se definen estrategias de exploración.

Finalmente, como se mencionó al inicio de esta sección, la representación más común en el caso de múltiples robots es la malla de ocupación. Esto debido a que la fusión de la información proveniente ya sea de varios sensores o de varios robots se puede realizar con mayor facilidad siguiendo el esquema definido en [Elfes87]. En [Singh93], los robots simulados van construyendo una malla de ocupación que se envía a todos los robots; cada robot decide hacia donde explorar siguiendo los contornos de la parte conocida del mapa hasta encontrar nueva información.

El primer trabajo de contrucción de mapas con múltiples robots implementado con robots reales, fue desarrollado por [Yamauchi98b]. En este trabajo cada robot cuenta con sensores de rango y la exploración la realiza cada robot de manera independiente acercándose a las fronteras (en este caso una frontera está definida por celdas conocidas que se encuentra al lado de celdas

desconcidas). Cada robot envía a los otros su mapa local, de manera que cada uno posee una representación global del ambiente. En la figura 1.4 se muestra un mapa parcial que hasta determinado momento han generado un par de robots que se encuentran explorando el ambiente. Los robots se encuentra representados por los círculos pequenos y la siguiente posición a la cuál se desplazará cada robot para continuar la exploración, se encuentra marcada con una X.

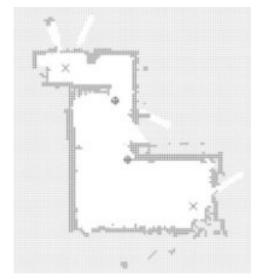


Figura 1.4. Mapa parcial generado por dos robots móviles [Yamauchi98b].

El proyecto Millibots [Navarro02], fue desarrollado en Carnegie Mellon University (CMU); la idea de este proyecto es la de contar con varios robots de tamaño pequeño con distintos tipos de sensores (algunos robots con sonares, otros con cámaras, algunos con sensores infrarrojos) que se encargan de explorar un ambiente desconocido y generar una representación del mismo. En este trabajo, se genera un mapa que consiste de varias mallas de ocupación, cada malla obtenida por un tipo de sensor diferente (cámaras, sonares, etc.). Cada robot se mueve de manera independiente siguiendo una trayectoria predefinida y la información que se va recolectando es enviada a un robot líder que concentra todo en un sólo mapa.

El proyecto Centibots [Konolige04] desarrollado en el Stanford Research Institute (SRI), consiste en trabajar con un número muy grande de robots (aproximadamente 60 robots) que

puedan percibir, explorar, planificar y colaborar en un ambiente desconocido. Los robots trabajan en tres etapas sucesivas: a) la etapa de construcción de mapa donde trabajan aproximadamente entre 2 y 5 robots utilizando cada uno un láser; b) la etapa de búsqueda en donde un gran número de robots equipados con cámaras buscan un objeto contando ya con el mapa del ambiente y c) la etapa de detección de intrusos, donde aproximadamente unos 60 robots son distribuidos por todo el ambiente para funcionar como guardias.

Cuando se está generando el mapa, cada robot de manera independiente va obteniendo su propia malla de ocupación, acercándose a la frontera donde estima que obtendrá una mayor cantidad de información. Cada uno de los robots puede iniciar la exploración sin tener conocimiento de la existencia de otros; sin embargo, si dos robots se encuentra lo suficientemente cercanos, pueden crear una hipótesis de su posición relativa que es comprobada cuando ambos robots se acercan a un sitio determinado; si la hipótesis es verificada entonces los dos pertenecen al mismo grupo y realizan una exploración cooperativa (ver figura 1.5).

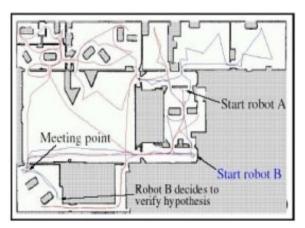


Figura 1.5. Trayectorias de robots durante la exploración y punto de encuentro [Konoliege04].

En la exploración cooperativa, existe un robot líder que se encarga de dirigir a los robots de su grupo indicando a cada uno la frontera que se debe explorar. Un avance significativo en este proyecto es que la aproximación propuesta para unir los mapas parciales de cada robot o grupo de robots considera que se desconocen sus posiciones relativas, en este método se puede saber si un

robot se encuentra dentro del mapa de otro robot o no, y con esto determinar si los mapas de dos robots se traslapan o no, y finalmente decidir como fusionarlos.

#### 1.4 EXPLORACIÓN PARA LA CONSTRUCCIÓN DE MAPAS

La mayoría de los trabajos de SLAM y de construcción de mapas, se han enfocado principalmente en aspectos relacionados con la generación precisa de un modelo del ambiente, donde se busca fusionar de alguna manera los nuevos datos obtenidos y estimar la posición del robot. En la mayoría de dichos trabajos no existe una estrategia de movimiento para efectuar la exploración, típicamente la trayectoria que sigue el robot móvil se encuentra predefinida o es generada por un operador mediante teleoperación mientras el robot va sensando el ambiente.

Existen pocos resultados en donde se defina una estrategia de exploración para la construcción de mapas. De manera general se pueden encontrar dos tipos de estrategias de exploración: aquellas en donde no se considera la información percibida para efectuar la exploración y aquellas en donde sí se toma en cuenta. La información percibida que se pueden utilizar es variada, y puede incluir la distancia recorrida, el área del ambiente que se espera descubrir, la energía utilizada, entre otra cosas. A continuación se describen brevemente trabajos previos de estrategias de exploración y se encuentran divididos en dos secciones de acuerdo a si dichos trabajos consideran o no información.

#### 1.4.1 EXPLORACIÓN SISTEMÁTICA

En esta tesis se consideran como estrategias de movimiento sistemáticas a aquellas en las cuáles se definen movimientos repetitivos o que cumplen con un patrón sin tomar en cuenta la información que se tenga hasta cierto momento durante la exploración. Una estrategia sencilla en el momento de la exploración, es hacer que el robot se mueva siguiendo el contorno de la pared mientras se encuentra sensando el ambiente [Bemporad97]; de esta manera se puede obtener una representación que incluya los límites del ambiente, sin embargo no incluirá a aquellos obstáculos que se encuentren en el medio.

En el trabajo propuesto por [Sim03] se definieron varias patrones a seguir durante la exploración, tales como que el robot avance en zigzag, avance en círculos concéntricos o bien siguiendo un patrón en forma de triángulo. La idea de esta trabajo fue la de comparar esos diferentes patrones de movimiento para saber con qué tipo de exploración se obtenía un mapa más preciso y se cubría mejor el espacio libre. Otro trabajo donde el objetivo de la exploración era simplemente cubrir todo el espacio libre, es el presentado en [Zunino01], este debido a que el robot que iba construyendo el mapa era una aspiradora.

En algunos casos también podría considerarse también como una estrategia sistemática, acudir siempre a la frontera más cercana o a la más grande. Acudir a una frontera durante la exploración, significa que el robot debe acercarse a la línea imaginaria que limita lo que hasta determinado momento se conoce del ambiente, con la parte desconocida del mismo. Por ejemplo, en el trabajo de Yamauchi [Yamauchi98b], el robot siempre acude a la frontera más cercana.

En una primera propuesta que se realizó durante este trabajo de tesis [Muñoz04], se definió una estrategia de exploración en la cuál los robots móviles siempre acudían a la frontera más grande y más cercana al robot. En esta forma de exploración se asumía que al acudir a la frontera más grande, el robot descubriría una mayor área del ambiente. Esto se describe con más detalle el capítulo 3 de esta tesis.

#### 1.4.2 ESTRATEGIAS DE MOVIMIENTO ÓPTIMAS BAJO ALGÚN CRITERIO

Las estrategias de movimiento que utilizan la información percibida por el robot, generalmente buscan optimizar la exploración bajo algún criterio en particular; por ejemplo realizar la menor cantidad de movimientos, descubrir una mayor área del ambiente desconocido, obtener una representación más precisa, en otros criterios. Para este propósito se define una función de utilidad o una función de costo que se evalúa para determinar los mejores movimientos que debe efectuar el robot para llevar a cabo la exploración.

Este tipo de estrategias, pueden ser vistas como estrategias donde se busca la *mejor vista siguiente posible* (next best view, NBV), de acuerdo con la información con la que el robot cuenta hasta el momento en que debe decidir el siguiente movimiento para llegar al lugar donde volverá a sensar. Ya sea que se considere información acerca de la distancia, de la incertidumbre, del espacio nuevo a descubrir, etc, generalmente lo que se busca es una función objetivo que represente el criterio o criterios que se desean optimizar. Esta idea se ha aplicado no solamente en el caso de la exploración como se describe en esta sección, sino también se ha empleado para otro tipo de tareas como es la navegación y localización. Por ejemplo en el trabajo propuesto en [Marín05], se define una función de utilidad para seleccionar la mejor combinación de parámetros de la cámara del robot (pan, tilt, zoom) y el subconjunto de landmarks que verá el robot para que éste se localice correctamente a lo largo de una trayectoria.

En el trabajo presentado en [Sim05], durante la exploración de un ambiente desconocido, se quiere que se maximice la cantidad de información que se obtendrá en cada sensado del robot. En este trabajo, la información es representada por medio de la entropía de la distribución de la probabilidad (la función de probabilidad de la posición del robot y de la posición de los elementos del mapa en algunos casos). Debido a que la entropía puede ser calculada utilizando la inversa de la matriz de covarianza de la distribución, el maximizar la información se traduce en minimizar la

matriz de covarianza. Con la función objetivo definida en términos de la matriz de covarianza, se calcula la ruta que debe seguir el robot dividiendo el espacio en una malla regular. Otro ejemplo en el cuál se utiliza como criterio para la exploración la ganancia de información, es el trabajo presentado en [Bourgault02], donde se utiliza como función objetivo la inversa de la matriz de covarianza. En [Feder99], se utiliza la información de Fisher como función a optimizar (este valor de información se calcula como el gradiente del logaritmo de la probabilidad total). En este caso se encuentra la acción para el robot que maximice la ganancia de información en la siguiente lectura del ambiente. El conjunto de posibles acciones del robot son distancias y ángulos predefinidos.

En cuanto a trabajos de exploración con múltiples robots, se menciona el trabajo realizado en [Simmons00]. En este caso, se representa el ambiente con una malla de ocupación que indica si cada celda de la malla esta vacía, ocupada o es desconocida. La exploración está basada en fronteras y para cada robot se determina qué frontera debe de visitar considerando una función de utilidad que evalúa la ganancia de información menos el costo de exploración. La ganancia de información se determina para cada posible frontera, para cada posible robot con el número de celdas desconocidas que se encuentran en el rango del sensor del robot; y el costo de exploración se determina con la distancia que hay entre la posición actual del robot y la frontera a la cuál pretende ir. Existe un procesamiento central que calcula la utilidad máxima para el conjunto de los robots y así se define la frontera que explorará cada robot en el siguiente movimiento.

En varios trabajos en donde se han definido estrategias de exploración basadas en algún criterio representado por medio de una función, se ha propuesto el uso de técnicas aleatorias para la generación de posibles lugares a los que debe ir un robot como en [Gonzalez02], [Makarenko02] y [Newman03]. En estos trabajos se generan muestras aleatorias de posibles configuraciones para el robot; dichas muestras son evaluadas de acuerdo con la función de utilidad o función de costo y de acuerdo al valor obtenido, se identifica la siguiente posición a la

cuál el robot acudirá para continuar con la exploración.

En [Gonzalez02] se propone una función que refleja intuitivamente cómo un robot debe explorar el espacio. La función de evaluación asigna un valor mayor a aquellas posiciones que mejor definen el compromiso entre la eliminación del espacio inexplorado y la distancia a viajar. En el trabajo presentado por Makarenko en [Makarenko02], se propone una función de utilidad que es la suma ponderada de varias funciones más pequeñas que representan varios criterios: ganancia de información (calculada de acuerdo al valor de entropía de las celdas de la malla de ocupación), la distancia que el robot debe viajar y la facilidad de localización del robot (basado en la submatriz de covarianza de la posición del robot).

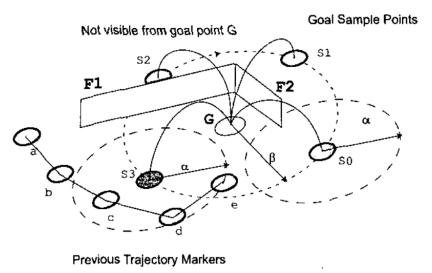


Figura 1.6. Generación de muestra para la exploración en [Newman03].

En [Newman03] se describe un método que considera las características geométricas del ambiente, así como la incertidumbre asociada a dichas características. Este método genera *metas*, denotadas en la figura 1.6 como G (posibles localizaciones siguientes del robot), asociadas a una característica del ambiente (por ejemplo un segmento) y alrededor de cada meta se generan *muestras* a un radio predefinido  $\beta$  denotadas como Sn en la figura. Y alrededor de cada muestra se "trazan" círculos de un radio  $\alpha$  (ver figura 1.6). El tamaño final del conjunto de muestras para

cada meta debe satisfacer dos condiciones: (a) cada muestra debe tener un línea directa de visión a la meta, (b) cada muestras no debe de tener una línea directa de visión a cualquier círculo alfa de otra muestra. Dependiendo del tamaño del conjunto muestras para cada meta, se selecciona aquella meta con mayor calificación. De esta manera el robot se dirige a la meta mejor evaluada para continuar la exploración.

En este trabajo de tesis de definió una segunda estrategia que permite a uno o varios robots explorar de manera eficiente el ambiente, para lo cuál se desarrolló una función de utilidad [Tovar06] que mide la calidad de posibles posiciones del robot generadas aleatoriamente. Dicha función de utilidad toma en cuenta la facilidad de reconocer landmarks en el ambiente, la menor cantidad de movimientos del robot, la cercanía a obstáculos o paredes, la facilidad de incorporar nueva información al mapa y la cantidad de información que se espera conocer en la posición evaluada. Una característica importante considerada en esta función, es que no sólo se toma en cuenta la cantidad de movimientos que debe realizar el robot sino el tipo de movimientos que efectuará (una o más translaciones o rotaciones) y cómo este tipo de movimientos afectan la incertidumbre en la posición del robot, es decir se consideran casos como el siguiente: no es lo mismo que el robot se mueva en línea recta una distancia dada en un sólo movimiento, a qué avance la misma distancia avanzando y deteniéndose varias veces. Finalmente también se definió una tercera estrategia que considera la utilización de robots con diferentes capacidades sensoriales y motrices y en este casos se utiliza una función más sencilla en donde se toma en cuenta el área que el robot espera conocer y la distancia que debe recorrer.

#### 1.5 PLANTEAMIENTO DEL PROBLEMA

A partir de la revisión de trabajos relacionados con la construcción de mapas y la exploración, se puede concluir que la mayoría se enfocan en la obtención de representaciones del ambiente precisas dejando de lado la definición de estrategias de movimiento durante la etapa de exploración.

La meta principal en este trabajo de tesis es el definir varias estrategias de movimiento para la exploración de uno o varios robots móviles que están encargados de construir una representación del ambiente en el que se encuentran. A pesar de que la obtención del mapa no es el objetivo principal de esta tesis, sino las estrategias de exploración; es necesario seleccionar no sólo una manera de representar el ambiente, sino también una manera de incorporar nueva información a esa representación. De manera que contando con una representación del ambiente y los algoritmos necesarios para añadir nuevas partes a esa representación, entonces se pueden estudiar de manera más detallada las estrategias de movimiento para los robots. Para poder generar una representación del ambiente o mapa y para poder definir las distintas estrategias de movimiento, se deben resolver los siguientes problemas específicos:

- Se quiere obtener un mapa del ambiente en el cuál se pueda representar de manera compacta la información obtenida por los robots. La información es obtenida por medio de un sensor de rango del robot (tanto en experimentos simulados como con los robots reales) y de una cámara (sólo en algunos experimentos simulados). Esta información debe ser procesada de manera que los datos obtenidos directamente del sensor (en este caso puntos) puedan ser utilizados para obtener la representación del ambiente seleccionada (en este caso segmentos de línea).
- Se necesita contar con un algoritmo que permita fusionar la información obtenida, ya sea fusionar la información entre las sucesivas lecturas que realiza un solo robot, o bien fusionar la información de varias lecturas realizadas por diferentes robots. De manera que se obtenga cada vez un solo mapa al cuál se le añadirá más información en las etapas posteriores de la exploración.
- Se define una primera estrategia de movimiento en la cuál los robots móviles siempre deciden acudir a la frontera más grande y más cercana durante la exploración.

- Se define una estrategia de movimiento óptima para un conjunto discreto de configuraciones candidatas de sensado, para la cuál se propone una función de utilidad que debe ser evaluada para saber a dónde se debe mover el robot durante la exploración en el siguiente paso en el tiempo. La función de utilidad toma en cuenta varios criterios, tales como el número de veces que se debe mover el robot, la cercanía a objetos y paredes, la cantidad de información nueva que espera descubrir, entre otros. Esta estrategia de movimiento está situada en el contexto de lo que se conoce como planificación de la percepción, es decir el sensado es incorporado como una restricción que debe satisfacer un plan de movimientos para el robot. En este caso, escoger una estrategia de exploración es equivalente a seleccionar configuraciones donde se realizará el sensado y la forma en que alcanzarán dichas configuraciones (trayectorias entre esas configuraciones).
- Se presentan extensiones para la estrategia de movimiento óptima, en las cuáles se consideran los siguientes casos: la exploración con múltiples robots y la obtención de una ruta de varios movimientos (más de un paso adelante en el tiempo) para cada uno de los robots. En este último caso, se considera que con una sola ejecución del algoritmo de exploración se obtienen varias posiciones a las cuáles los robots se deben desplazar antes de ejecutar nuevamente el algoritmo de exploración.
- Se define una estrategia de movimiento para que varios robots con distintas características sensoriales y motrices pueden llevar a cabo la exploración de un ambiente desconocido.

Para el desarrollo de esta tesis se emplearon los robots móviles Pioneer 3AT y Pioneer 3DX, que se muestran en la figura 1.1, ambos cuentan con un láser que se utilizó para la obtención de datos. Las pruebas se realizaron en ambientes de interior, en laboratorios cerrados donde se armaron laberintos de prueba. La descripción detallada de la representación seleccionada y los algoritmos para la fusión de información, se encuentran en el capítulo 2. La estrategia de exploración basada en fronteras se describe en el capítulo 3. Y en el capítulo 4 se muestran los casos de la exploración óptima con una función de utilidad, así como el caso de la exploración

con robots con distintas características. En el capítulo 5 se describen las implementaciones y experimentos realizados, y finalmente en el capítulo 6 se muestran las conclusiones y trabajo futuro.

# CAPÍTULO 2. LOCALIZACIÓN Y CONSTRUCCIÓN DE REPRESENTACIONES EN AMBIENTES DE INTERIOR

Cuando se trata el problema de la navegación autónoma de un robot móvil, los dos primeros pasos son primero la percepción y modelado del ambiente y segundo la localización del robot. En la primera parte de este capítulo se definen los problemas de localización, construcción de modelos del ambiente; así como la localización y construcción simultánea. En la segunda parte del capítulo se describen los tipos de representaciones o tipos de modelos que existen y se detalla a cerca del modelo que se utiliza en este trabajo de tesis.

#### 2.1 DEFINICIÓN DEL PROBLEMA DE SLAM

El problema de localización responde a la pregunta ¿dónde está el robot?, y considera que el robot cuenta con un mapa y con sensores que le permiten percibir el ambiente; de tal manera que se puede comparar lo que el robot acaba de percibir con el mapa para determinar en donde se encuentra. Por otro lado el problema de construcción del mapa supone que el robot conoce con exactitud su posición y conforme va percibiendo el ambiente, va guardando cierta información de lo que percibe y la posición en que lo percibió para crear un modelo o representación del ambiente. Sin embargo en el caso general, cuando el robot necesita construir una representación

del ambiente, inicialmente se encuentra en una posición desconocida, en un ambiente desconocido y de manera incremental va construyendo el mapa al mismo tiempo que utiliza dicho mapa para conocer su posición.

#### 2.1.1 EL PROBLEMA DE LOCALIZACIÓN Y EL PROBLEMA DE CONSTRUCCIÓN

El *problema de localización*, puede formularse como un problema en que se debe encontrar la correspondencia entre lo que el robot está sensando y un modelo que tiene del ambiente. Dicho de otra manera, para poder localizarse el robot debe tomar una lectura del ambiente con sus sensores, con esa información debe construir un modelo que llamaremos parcial y tratar de ajustar ese modelo parcial al mapa (modelo global) que posee del ambiente en donde se encuentra. Para solucionar el problema, se plantea un conjunto de hipótesis a cerca de la posición del robot [Castellanos99], donde cada hipótesis representa una posible posición estimada del robot con respecto al marco de referencia del ambiente; estas hipótesis deberán ser verificadas. Generalmente se utiliza un marco de referencia global, aunque se han desarrollado trabajos donde también se utilizan marcos de referencia locales [Alencastre06b]. La localización estimada del robot se puede saber si se conoce la transformación relativa que existe entre el marco de referencia del robot y el marco de referencia del ambiente; o bien se pueden tener numerosas hipótesis a cerca de la localización del robot y calcular un valor de probabilidad para cada una de las hipótesis y la localización estimada del robot será aquella que tenga la mayor probabilidad.

En la figura 2.1, se muestra de manera esquemática el problema de localización. Los tríangulos sólidos representa la posición real del robot, y los pentágonos sólidos representan el mapa con el que cuenta. Cuando el robot toma una lectura parcial del mapa (las lecturas son denotadas con  $z_k$ ), y además conoce los movimientos anteriores que realizó (un solo movimiento se denota como  $u_k$  y todos los movimientos hasta el tiempo k se denotan como u[0:k]), entonces puede estimar su posición con respecto a un marco de referencia. En la figura la estimación de la

posición está representada por medio de los tríangulos sin color y se denota cada estimación como  $x_k$ . En el caso general puede haber más de una estimación de la posición del robot en un momento dado, o bien se puede obtener una distribución de probabilidad a cerca de la posición del robot, dado los movimientos que ha efectuado (u[0:k-1]) y las lecturas que ha tomado con sus sensores (z[0:k]), es decir se calcula  $p(x_k \mid u[0:k-1], z[0:k])$  [Choset05].

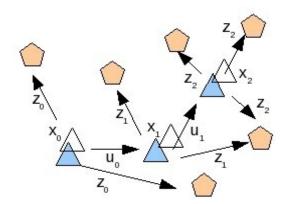


Figura 2.1. Localización de un robot móvil.

El problema de localización, tienes distintas instancias cuya solución tiene diferentes grados de dificultad [Choset05], estas instancias se mencionan a continuación:

- Seguimiento de la posición: en este caso inicialmente se conoce la posición inicial del robot y conforme se va moviendo, se va estimado su nueva posición.
- Localización global, donde se desconoce la posición inicial del robot y comúnmente el robot debe desplazarse hasta poder conocer su posición en el ambiente. Este problema se presenta por ejemplo cuando se enciende el robot sin saber en donde se encuentra.
- Relocalización, en la cuál se supone que en determinado momento el robot conocía su posición, sin embargo de manera repentina el robot es transladado a una nueva posición, por lo que lo primero que debe hacer el robot es darse cuenta que ha cambiado de lugar.

Para cada uno de los problemas anteriores, se han propuesto varias soluciones, siendo de los métodos más comunes el Filtro Extendido de Kalman (Extended Kalman Filter, EKF) y el

Filtro de Partículas [Choset05]. Ambos son métodos probabilísticos, en el caso del EKF, la distribución de probabilidad  $p(x_k \mid u[0:k-1], z[0:k])$  se asume que es gausiana y por lo que puede ser utilizado para el caso de seguimiento de la posición; mientras que en el caso del filtro de partículas, no se asume alguna distribución de probabilidad, por lo tanto se puede utilizar para resolver las otras dos instancias del problema de localización [Choset05].

El *problema de construcción*, en el caso más simple, supone que el robot conoce con exactitud su posición en todo momento mientras va sensando información del ambiente. Esta idea se muestra en la figura 2.2, en donde el robot conoce a lo largo de su trayectorio sus posiciones denotadas como  $x_k$ , y va obteniendo información del ambiente con cada lectura denotada como  $z_k$ , de manera que poco a poco va teniendo una estimación del mapa del ambiente que en la figura se denota con los pentágonos sin color (y los pentágonos con color el mapa real). En la solución probabilística a este problema en [Elfes87], se propuso obtener un mapa de celdas calculando la probabilidad de que cada celda este o no ocupada dadas las posiciones del robot y las lecturas que registraron los sensores, es decir para cada celda se calcula  $p(celda \mid x/0:k)$ , z/0:k).

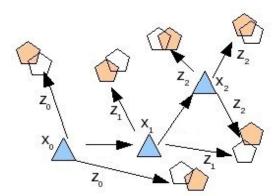


Figura 2.2. Construcción de un mapa con posiciones conocidas.

Sin embargo, para obtener el mapa de un ambiente un robot debe tener en cuenta que se tienen al menos dos tipos de ruido en los sensores: ruido en los sensores de percepción (cámaras o sensores de rango por ejemplo) y ruido en las mediciones de odometría (es decir ruido al detectar su propia posición). Debido al ruido en las mediciones de la posición del robot, el problema de construcción de un mapa crea de manera inherente un problema de localización. Por lo tanto en el caso más general, el problema de construcción de un mapa es un problema donde se debe resolver la parte de localización y la parte de construcción al mismo tiempo. Y esto es lo que se describe en la siguiente sección.

#### 2.1.2 EL PROBLEMA DE LOCALIZACIÓN Y CONSTRUCCIÓN SIMULTÁNEA

El problema de *Localización y Construcción Simultánea* (Simultaneous Localization and Map building, SLAM) supone que un vehículo autónomo comienza en una posición desconocida en un ambiente desconocido y de manera incremental construye un mapa del ambiente, mientras simultáneamente usa el mapa para calcular su posición en el ambiente. La habilidad de poner un robot en un ambiente y localización desconocidos junto con la habilidad de construir el mapa usando observaciones del ambiente y que este mapa sea utilizado simultáneamente para navegar, hacen de un robot un vehículo autónomo.

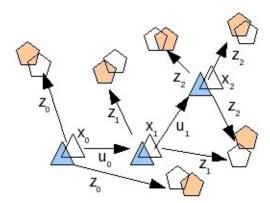


Figura 2.3. Localización y Construcción Simultánea.

En la figura 2.3, se ilustra el problema de SLAM, los tríangulos sólidos indican la posición real del robot y los tríangulos sin color indican la posición estimada del robot que se va calculando al mismo tiempo que se construye el mapa. En la figura, el mapa estimado está

representado por los pentágonos sin color.

Se han propuesta distintas formas para solucionar el problema de SLAM, principalmente se pueden dividir en tres distintos tipos de soluciones [Dissanayake01]: las que utilizan el EKF, las que utilizan alguna solución numérica diferente al EKF y las que utilizan alguna aproximación cualitativa.

### 2.1.2.1 Solución utilizando el EKF

El método más utilizado para resolver el problema de SLAM es utilizando el EKF, debido principalmente a que éste provee una solución recursiva al problema de navegación y un mecanismo para calcular la incertidumbre del robot y del mapa al mismo tiempo. En este método, se estiman de manera simultánea la posición del robot y la posición de un conjunto de caraterísticas y landmarks que definen el mapa, basándose en el modelo de movimiento del robot y en el modelo de observación de los landmarks. El modelo de movimiento del robot, en este caso no es más que una matriz que al ser multiplicada por el vector de estado (vector de posición del robot), da como resultado la nueva posición del robot en el siguiente paso en el tiempo. Y el modelo de observación, es también una matriz que permite relacionar las lecturas del robot con las posiciones de los landmarks en el ambiente. Los landmarks que definen el mapa pueden ser segmentos, postes, marcas artificiales, árboles (en ambientes de exterior), esquinas, etc.

El EKF, permite una solución incremental al problema de SLAM, como se ilustra el la figura 2.4; es decir mientras el robot se va desplazando, y poco a poco van obteniendo una mayor parte mapa del ambiente. En la figura 2.4, se muestran los estados en círculos etiquetados con  $x_k$ , y en cada estado el robot tomana una lectura del ambiente, denotada con  $z_k$ ; posteriormente se aplica un control ( $u_k$ ) que hará que el robot pase al siguiente estado; y esto se repite hasta que se ha completado el mapa.

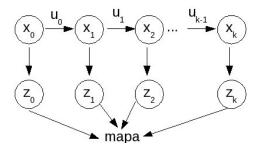


Figura 2.4. Modelo gráfico de SLAM incremental.

Entre los primero trabajos relacionados en este método, se encuentra el propuesto en [Leonard92], en el cuál se define como vector de estado a uno que contiene las posiciones del robot y las posiciones de los landmarks que forman el mapa. El vector de estado se estima mediante un filtro de Kalman extendido (Extended Kalman Filter, EKF), y el vector de la última estimación contiene una lista de posiciones de los landmarks que constituyen el mapa.

## 2.1.2.2 Soluciones numéricas

Por soluciones numéricas, se tienen a todas aquellas soluciones que no utilizan el EKF, sin embargo utilizan alguna técnica numérica o algoritmo computacional. Dentro de este conjunto de soluciones se encuentran por ejemplo, las propuestas hechas en [Thrun98], [Yamauchi98a] y [Castellanos99]. En el trabajo de Thrun, se propone la generación de un mapa utilizando técnicas bayesianas para la construcción de mapas sin tener la necesidad de asumir una distribución de probabilidad Gaussiana como en caso del EKF. Sin embargo en esta técnica, no provee un solución incremental al problema de SLAM, es decir el robot se mueve y se registran todas las lecturas y movimientos del robot durante la exploración y al final con toda la información reunida se calculan el mapa del ambiente, como aquél mapa que maximiza el siguiente valor de probabilidad: p(mapa, x[0:k] + u[0:k-1], z[0:k]). Por otra parte en el trabajo de Yamauchi, se utiliza una malla de ocupación que se va actualizando incrementalmente, asignado a cada celda de la malla un valor de probabilidad que indica la presencia de obstáculo. Conforme el robot va

construyendo el mapa, cada determinado tiempo se utiliza ese mapa parcial que se ha construido hasta cierto momento para relocalizarse y corregir su posición en el ambiente.

En el trabajo desarrollado en [Castellanos99], se utiliza directamente la propuesta hecha en [Cheeseman86] a cerca de las transformaciones con incertidumbre entre dos marcos de referencia, junto con su propia representación de segmentos con pertubaciones (es decir segmentos con incertidumbre asociada). En este trabajo, para poder generar el mapa de manera incremental, se considera que en cada paso en el tiempo se tiene el mapa global hasta el momento construido, el mapa local obtenido a partir de la última lectura de los sensores y la posición del robot. Con esta información se busca realizar un *acoplamiento* (matching) entre el mapa global y el mapa local conociendo el desplazamiento relativo entre las últimas dos posiciones del robot y así obtener el nuevo mapa global. Finalmente con este nuevo mapa global se relocaliza al robot.

La solución al problema de SLAM, implementado en este trabajo de tesis, corresponde a este tipo de soluciones, es decir se tiene un solución que corresponde a un algoritmo computacional. Debido a que la aportación principal del trabajo de tesis es en la parte de exploración, la construcción del mapa se lleva a cabo simplemente alineado la representación obtenida con cada lectura con el mapa global que se lleve construido hasta determinado momento; posteriormente utilizando dicho mapa se vuelve a localizar al robot antes de continuar la exploración.

# 2.1.2.3 Soluciones cualitativas

En este tipo de solución se propone no contar estimaciones absolutas de posición, ni medidas precisas de incertidumbre; en lugar de cantidades numéricas, se propone utilizar representaciones cualitativas que no requieran de modelos exactos. La ventaja de no contar con modelos exactos se traduce generalmente en un menor costo computacional en los algoritmos de navegación y en

contar con representaciones más "humanas". Un ejemplo de este tipo de soluciones se tiene en el trabajo de Tovar [Tovar07], donde el mapa del ambiente es una representación mínima utilizando un grafo, específicamente se tiene un árbol que codifica información topológica del ambiente. Suponiendo que el robot se encuentra en un ambiente de interior delimitado por una curva cerrada, la representación con la que cuenta en determinado momento corresponde a los espacios hacia los cuáles se puede dirigir el robot, más que a coordenadas específicas.

# 2.2 REPRESENTACIÓN DEL AMBIENTE

Ya sea que que el robot utilice la información sensada como una herramienta para resolver una tarea específica (por ejemplo, localizarse), o bien que lo sensado sea la tarea final del robot (por ejemplo, hacer un mapa); es necesario que se defina un modelo o representación del ambiente con el cuál posteriormente el robot puede realizar distintas tareas. En esta sección se describen las formas más comunes que se han propuesto para representar un ambiente de interior, y posteriormente se detalla la opción seleccionada para este trabajo de tesis.

# 2.2.1 TIPOS DE REPRESENTACIÓN

Se han propuesto diferentes clasificaciones para los diferentes tipos de representaciones del ambiente que se han utilizado como mapas. En [Dudek00], se identifican tres tipos de mapas o representaciones: mapas de descomposición espacial, mapas geométricos y mapas topológicos. Por otra parte en [Chatila85] se clasifican los mapas como geométricos, topológicos y semánticos. Considerando estas dos clasificaciones en esta sección se describen cada uno de los tipos de mapas comentando algunas ventajas y desventajas de los mismos. En la figura 2.5, se muestra un ambiente de interior, como si fuera un plano de una planta de un edificio, el cúal se utilizará como

ejemplo para ilustrar los diferentes tipos de representaciones.

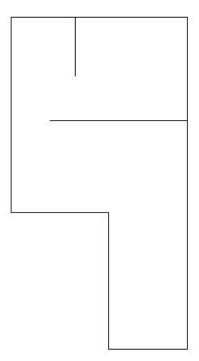


Figura 2.5. Plano de un ambiente de interior.

La representación por *descomposición espacial*, consiste en dividir el espacio utilizando una malla con cierto número de celdas ya sean de tamaño regular o no. Cada celda contiene información de lo que hay en ese lugar en el ambiente. En el caso más sencillo todas las celdas son del mismo tamaño y simplemente se especifica si cada una se encuentra ocupada por un obstáculo o bien si se encuentra vacía. En el trabajo desarrollado en [Elfes87], se tienen celdas de tamaño regular y cada una tiene la probabilidad de esa celda se encuentre o no ocupada. La principal ventaja que tiene este modelo del ambiente, es su generalidad, ya que no hace ninguna suposición a cerca de la forma que debe tener los objetos del ambiente para poder ser representados. La principal desventaja de este modelo depende de la resolución de la malla, que a mayor resolución se tiene una mayor fidelidad en la representación sin embargo se requiere una mayor cantidad de recursos de almacenamiento, independientemente de cuántas celdas se encuentren ocupadas o vacías. Una solución propuesta para reducir el espacio de almacenamiento consiste en definir celdas de distintos tamaños de tal manera que espacio libres amplios se

representen con una sola celda grande y en espacios donde haya más información de obstáculos se utilicen celdas más pequeñas para tener un mejor detalle. En la figura 2.6 se muestra un ejemplo de una malla de ocupación, en esta malla simplemente se muestran las celdas como ocupadas o vacías.

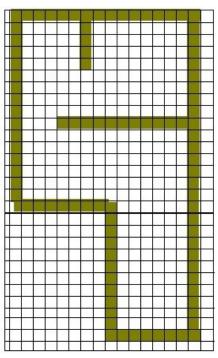


Figura 2.6. Ejemplo de una malla de ocupación.

La representación *geométrica* utiliza alguna o algunas primitivas geométricas para la representación del ambiente, como pueden ser puntos, líneas, polígonos, funciones polinomiales, poliedros, superficies planas, cubos, elipsoides, etc.; dependiendo de si el modelo del ambiente es en dos o tres dimensiones. La primitiva geométrica más utilizada es el segmento de línea [Zhang00, Pfister03], por lo que este modelo supone un proceso de ajuste de curvas previo a la construcción del modelo, para la obtención de las primitivas geométricas necesarias a partir de los datos obtenidos por los sensores. La principal ventaja de estos modelos es la representación compacta que se maneja del ambiente, ya que grandes regiones se pueden representar con unas cuantas primitivas geométricas definidas por algunos parámetros. Una desventaja puede ser la

falta de poder expresivo, ya que la primitiva seleccionada puede ser una dificultad para representar ciertos objetos que se encuentren en el ambiente (por ejemplo si se selecciona como primitiva el segmento de línea, se pueden tener dificultades para representar obstáculos curvos). Esto también puede ocasionar que diferentes ambientes generen la misma representación. En la figura 2.7, se muestra un ejemplo de un mapa de segmentos para el ambiente de la figura 2.5; como se puede observar, los segmentos no se encuentra alineados perfectamente, esto puede presentarse debido al ruido durante el sensado o errores en la localización del robot al momento de construir la representación.

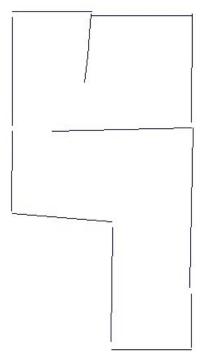


Figura 2.7. Ejemplo de un mapa geométrico utilizando segmentos de línea.

En la representación *topológica* se trata de capturar la información de conectividad entre los lugares de un ambiente, sin importar las medidas exactas. La representación está basada en una abstracción del ambiente en términos de lugares discretos unidos por segmentos. Por lo tanto la representación consiste de un grafo, en el cuál los nodos representan los lugares del ambiente y las aristas las conecciones entre éstos. En un mapa topológico, un lugar o nodo es una unidad

funcional seleccionada dependiendo del ambiente. Por ejemplo, en [Huang04], cada nodo representa esquinas y en [Korten94] representa intersecciones en pasillos o puertas. La principal ventaja de la representación topológica, es que es un modelo compacto, sin embargo puede ser difícil utilizar este tipo de representaciones en el reconocimiento de lugares para la localización del robot. En la figura 2.8, se muestra un posible mapa topológico para el ambiente de la figura 2.5.

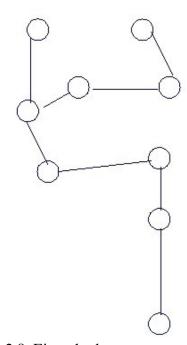


Figura 2.8. Ejemplo de un mapa topológico.

La representación *semántica*, es una representación simbólica de alto nivel que da un significado a partes del mapa y puede estar formada por información que esté relacionada con objetos, propiedades del espacio y relaciones entre los objetos o el espacio [Chatila85]. Un ejemplo de una representación semántica sería por ejemplo definir un mapa etiquetando los cuartos de un ambiente de interior por colores o funciones; o bien definir el mapa de una pasillo de acuerdo a los objetos que se irán encontrando sucesivamente: un cuadro, una escultura, una maceta, etc. La manera más sencilla de construir un mapa de este tipo, probablemente consista en etiquetas los nodos de un mapa topológico. La principal ventaja que presenta este tipo de mapas

es que su representación puede llegar a ser más cercana a cómo los seres humanos guardamos la información sobre determinados lugares, sin embargo supone que el robot debe contar con algoritmos sofisticados capaces de identificar el significado de los lugares, o la identificación de objetos.

## 2.2.1.1 Representación utilizada

Para este trabajo de tesis se seleccionó la representación geométrica utilizando como primitiva al segmento de línea. De manera que los mapas que serán obtenidos son semejantes a los planos de los edificios. La información para la obtención de estos mapas es obtenida a partir de un sensor de rango que entrega lecturas de un plano paralelo al piso.

En las primeras simulaciones y experimentos con robots reales (para la estregia de exploración basada en fronteras), el mapa construido solamente está formado por segmentos. Sin embargo para las simulaciones que se realizaron para las estrategias de movimiento óptimas, se supone que el robot es capaz de reconocer marcas sobresalientes en el ambiente; de manera que el mapa final consiste de un conjunto de segmentos y un conjunto de marcas que se han identificado. Además de los segmentos y las marcas, durante la exploración del ambiente se va guardando cada una de las posiciones en las cuales algún robot tomó una lectura del ambiente, así como la ruta que siguió para llegar a tal posición; de manera que se va creando un *mapa de caminos* (roadmap) que puede ser utilizado en etapas posteriores durante la exploración.

## 2.2.2 EXTRACCIÓN DE SEGMENTOS

Debido a que se utiliza un sensor de rango para la obtención de información del ambiente, es

necesario pasar de los datos obtenidos por el sensor a los segmentos que formarán parte del mapa. Durante los experimentos realizados con robots reales, se utilizó como sensor un LRF que entrega un conjunto de puntos que representan un barrido realizado por el láser. La extracción de segmentos consiste en tomar el conjunto de puntos obtenidos por el LRF y encontrar uno o varios segmentos que formarán parte del mapa.

El ajuste de segmentos de línea a partir de los puntos entregados por el sensor, se realiza en tres etapas:

- Determinar los conjuntos de puntos cercanos, es decir se obtienen grupos de puntos cuya separación entre sí no excede determinado valor.
- Ajustar cada conjunto de puntos cercanos utilizando el método de mínimos cuadrados.
   Puede ser que para cada conjunto se obtengan varios segmentos debido al ajuste.
- Utilizando los puntos de inicio y fin de cada segmento obtenido por mínimos cuadrados, se utiliza la técnica de divide y vencerás [Aho83] para reducir el número de puntos.

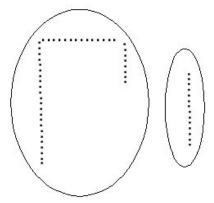


Figura 2.9. Dos conjuntos de puntos formados con los datos de un LRF.

Para determinar los conjuntos de puntos cercanos, se calcula la distancia entre dos puntos consecutivos, si ésta es menor a un valor establecido, se considera que ambos puntos pertenecen al mismo conjunto, de lo contrario se forman un nuevo conjunto. En la figura 2.9, se muestran los datos obtenidos por un LRF; utilizando el criterio de la distancia entre puntos, se han formado dos

conjuntos.

En el segundo paso, para cada conjunto de puntos, se encuentran los segmentos que deben formarse utilizando el algoritmo de mínimos cuadrados. La idea consiste en primero intentar formar un solo segmento con todos los puntos del conjunto, aplicar el método mínimos cuadrados para ajustar los puntos a la mejor línea y determinar cuál es el punto (del conjunto original) más alejado de la línea. Si el punto más alejado se encuentra a una distancia mayor de un umbral específicado, entonces se divide la lista de puntos en dos lista a la cuáles se aplicará el método de mínimos cuadrados por separado, y así sucesivamente hasta que las listas cumplan con el umbral dado. Este algoritmo se ilustra con un ejemplo en la figura 2.10, la salida de este paso, es ahora un conjunto de segmentos que se obtienen como puntos inicial y final de las líneas que se obtuvieron por mínimos cuadrados. Se puede notar en la figura 2.10, que el punto que se encuentra más alejado del conjunto no aparece como parte del segmento obtenido a partir de la línea ajustada, esto es debido a que con el método de mínimos cuadrados se eliminó.

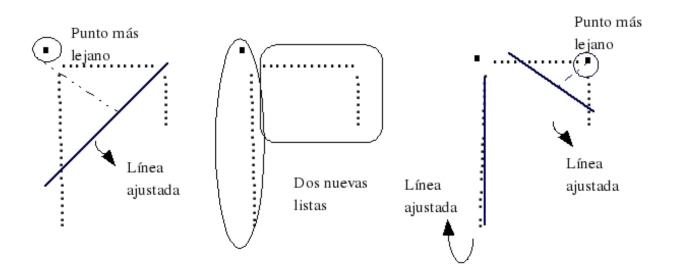


Figura 2.10. Ajuste de líneas utilizando mínimos cuadrados.

Cómo resultado del paso dos, se obtienen varios segmentos al aplicar sucesivamente el método de mínimos cuadradros. Sin embargo se pueden obtener gran cantidad de segmentos

definidos por los puntos terminales de las rectas correspondientes. En la figura 2.11 se indica un ejemplo, en donde la primera imagen de la figura es la salida del paso dos. Para reducir el número de puntos que se convertirán en los vértices del mapa, se utiliza un algoritmo de divide y vencerás, aplicado a los puntos que indican el inicio y fin de cada segmento. Con este algoritmo se trata de formar un solo segmento con los puntos, y se calcula la distancia al punto más lejano, si esta distancia excede determinado umbral, se divide el conjunto en dos listas de puntos y ahora se intentan formar dos segmentos. Esto se repite de manera sucesiva hasta que se obtienen los segmentos finales. Combinando así el método de mínimos cuadrados con el algoritmo de divide y vencerás, se obtiene un método robusto a ruido y se generan pocos vértices en el mapa.

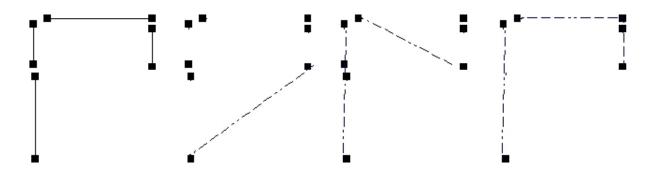


Figura 2.11. Ejemplo donde se aplica el algoritmo de divide y vencerás.

# 2.2.3 FUSIÓN DE LA INFORMACIÓN

Debido a que el problema de SLAM es resuelto de manera incremental, cada vez que se tiene nueva información a cerca del ambiente, es necesario incorporar dicha información con el mapa global que se haya construido hasta determinado momento; por lo tanto se debe fusionar la nueva información con el mapa ya existente para obtener un mapa actualizado del ambiente.

A la representación del ambiente que se ha construido hasta determinado momento, se le

llamará mapa global y a la representación del ambiente recientemente adquirida se le llamará mapa local. Como su nombre lo indica el mapa global se encuentra en el marco de referencia global que se ha definido para el mundo, y el mapa local se encuentra definido en el marco de referencia del robot que tomó la lectura; por lo tanto tiene que buscarse una transformación que acople el mapa local en el mapa global y así obtener el mapa actualizado. Esta idea se representa en la figura 2.12, donde se representa el mapa global por medio de líneas sólidas y el mapa local se representa con líneas punteadas. Cuando se encuentra la transformación apropiada, se acoplan ambos mapas y se muestra el mapa completado hasta el momento.

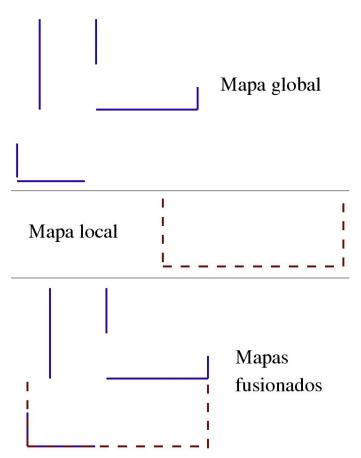


Figura 2.12. Fusión de información durante el proceso incremental de SLAM.

La transformación entre ambos mapas puede ser encontrada considerando que se conoce la trayectoria que ha recorrido el robot; con lo cuál, se puede tener un estimado de la posición y orientación del robot en el marco de referencia global, entonces se puede conocer la transformación que se debe aplicar al mapa local para que se localice en el marco de referencia global. Sin embargo debido a que existen errores en el control de los motores que generan el movimiento del robot y errores en los sensores, puede ser que la transformación calculada para acoplar ambos mapas no sea la que produzca el mejor alineamiento; y por lo tanto se puede probar con ligeras variaciones de la transformación hasta que se obtenga un mejor alineamiento. Como consecuencia de esto, es necesario contar con una medida que indique cómo determinar que una transformación es mejor que otra en cuando al resultado del alineamiento. En este trabajo de tesis, la medida seleccionada fue la distancia parcial de Hausdorff. Esta idea de acoplar información proveniente de sensores de rango para formar un solo mapa se había realizado ya en trabajos previos a esta tesis en [Tovar02].

La *distancia de Hausdorff* entre dos conjuntos de puntos, se define por la ecuación 2.1, donde *P* y *Q* representan los dos conjuntos respectivamente [Huttenlocher93].

$$H(P,Q) = max(h(P,Q),h(Q,P))$$
 (2.1)

Donde:

$$h(P,Q) = \max_{p \in P} \min_{q \in Q} ||p-q||$$
 (2.2)

La ecuación 2.2 representa la distancia directa del conjunto P al conjunto Q y  $\|p-q\|$  es la distancia entre los puntos p y q. La distancia directa es una medida que indica que tan cerca está un punto del conjunto P a un punto del conjunto Q. Un valor pequeño de h(P, Q) implica que cada punto de P se encuentra cercano a algún punto del conjunto Q.

La distancia de Hausdorff (ecuación 2.1), es el máximo de las dos distancia directas (de *P* a *Q* y de *Q* a *P*) e indica el punto que se encuentra más alejado de cualquier otro punto entre los dos conjuntos comparados. Por lo tanto la distancia de Hausdorff calculada de esta manera es muy

sensible a la presencia de puntos lejanos en el conjunto y esto se puede llegar a presentar en el caso de alineamientos de mapas debido a errores durante el sensado. Por esta razón es más apropiado utilizar una medida que no presente este problema, en donde se cambia la operación del máximo por el cálculo de la media de los valores. A esta medida se le conoce como la *distancia de Hausdorff parcial* y se encuentra definida por las ecuaciones 2.3 y 2.4.

$$H_k(P,Q) = max(h_k(P,Q), h_k(Q,P))$$
 (2.3)

Donde:

$$h_k(P,Q) = Media_{p \in P} \min_{q \in Q} ||p - q||$$
(2.4)

Es importante hacer notar que la distancia de Hausdorff (en cualquiera de sus versiones), es inherentemente asimétrica en su cálculo; esto significa que el hecho de que cada punto de P esté cercano a algún punto de Q no dice nada a cerca de que cualquier punto de Q esté o no cerca a cualquier punto de P. Es decir el cálculo de la distancia no implica que exista una correspondencia entre los puntos de ambos conjuntos.

Para el caso del alineamiento entro los mapas global y local, se trabaja directamente con el conjunto de puntos obtenido por el LRF. El conjunto P es el conjunto de puntos correspondientes al mapa global, mientras que el conjunto Q es el conjunto de puntos correspondientes al mapa local. Después de encontrar una transformación que acople ambos mapas, se calcula la distancia parcial de Hausdorff; de esta manera se selecciona aquella transformación que haga que la distancia parcial de Hausdorff entre los dos conjuntos de puntos sea la menor distancia posible.

# CAPÍTULO 3. EXPLORACIÓN PARA LA CONSTRUCCIÓN DE MAPAS

Para que uno o varios robots puedan ir construyendo un mapa del ambiente, es necesario que se desplacen por ellos mismos, a esta tarea se le conoce como *exploración*. La exploración se considera como una tarea de navegación en un ambiente desconocido o parcialmente desconocido que tiene como propósito descubrir nuevos lugares del ambiente mediante el uso de sensores para percibir la información. En este capítulo, se describe una estrategia de exploración basada en fronteras utilizando un Proceso de Decisión de Markov para la planificación de las trayectorias. Así mismo se describe cómo se utiliza dicha aproximación en el caso de múltples robots.

# 3.1 EXPLORACIÓN BASADA EN FRONTERAS

Cuando un robot se encuentra explorando, es necesario que decida cuál es el siguiente punto o posición que debe alcanzar para adquirir la siguiente lectura; y llegar a esa posición posiblemente utilizando alguna técnica de planificación de trayectorias. En la exploración basada en fronteras, el robot se dirige a los límites del mapa que lleva construido hasta determinado momento; es decir el robot va a las fronteras entre lo que conoce del ambiente y lo que no conoce.

Una frontera se forma debido a paredes u obstáculos que le impiden al robot "ver" ciertas partes del ambiente, como se muestra en la figura 3.1a; donde la pared impide que el robot tenga información a cerca del área que se encuentra sombreada. En esa imagen en particular, se muestra la frontera con una línea roja que divide lo que conoce con lo que no conoce (area sombreada). En una exploración basada en fronteras, el robot se dirige a la línea imaginaria que se forma entre el área sombreada y la no sombreada, de manera que pueda "descubrir" lo que la pared ocultaba, tal como se muestra en la figura 3.1b.

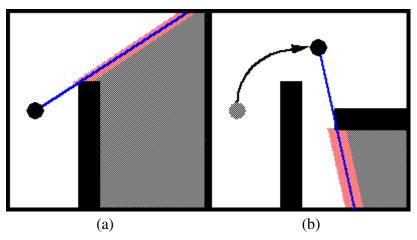


Figura 3.1 Exploración basada en fronteras. (a) El robot descubre una frontera. (b) El robot acude a una posición sobre la frontera.

La primera estrategia de exploración que se definió para este trabajo de tesis [Muñoz04] es una exploración basada en fronteras, en la cuál se realiza lo siguiente:

- El robot toma una lectura del ambiente con un LRF y se obtienen los segmentos a partir de los puntos.
- Se acopla el mapa global con los segmentos obtenido a partir de la última lectura.
- Considerando las posibles fronteras; se selecciona las dos fronteras con mayor longitud, se calcula el punto medio de ambas fronteras y se define como siguiente meta a aquél punto medio que se encuentre más cerca a la posición actual del robot. Al seleccionar una frontera larga, se está asumiendo que posiblemente se encuentre una mayor cantidad de información al acudir a la misma. Dicho de otra manera, si la frontera es muy larga se

asume que el área nueva por descubrir es grande si el robot acude hasta ahí. Por lo tanto si el robot visita las fronteras más largas primero entonces realizará una exploración del ambiente con menos movimientos. En este punto cabe aclarar que esta asumción no siempre es cierta, por lo que se considera que la exploración basada en fronteras es un tipo de exploración sistemática.

- Con la siguiente meta definida para que el robot acuda a explorar, se utiliza un Proceso de Decisión de Markov que calcula la trayectoria que debe seguir el robot para llegar a la frontera seleccionada.
- Cuando el robot llega a la nueva posición, nuevamente toma una lectura y el ciclo se repite hasta que no se encuentren fronteras por explorar.

# 3.1.1 PROCESOS DE DECISIÓN DE MARKOV

Un Proceso de Decisión de Markov (Markov Decision Process, MDP), es un modelo matemático de un problema de decisión secuencial en tiempo discreto [Puterman94]. Dicho de otra manera un MDP es un modelo que sirve para representar problemas donde se tienen que ir tomando una serie de decisiones, cada una de las cuáles se debe tomar en un paso en el tiempo. La decisión a tomar es una acción que depende del estado en que se encuentre un sistema. Por ejemplo, si se considera como sistema a un robot móvil en un ambiente de interior donde se desea que el robot llegue a determinada posición, los posibles estados son las posiciones del robot en el ambiente; por lo tanto el robot debe decidir a donde moverse (acciones) en cada paso en el tiempo hasta que llegue a la meta. Los elementos de un MDP son los siguientes:

- Un conjunto de posibles estados del sistema: S.
- Un conjunto finito de acciones: *A*.
- Un modelo de transición que indica la probabilidad de pasar de un estado a otro cuando se aplica determinada acción:  $P(s \mid s', a)$  donde  $s, s' \in S$  y  $a \in A$ .
- Una función de recompensa R que indica para cada estado que valor de recompensa se

tiene por haber alcanzado cierto estado: R(s).

Es importante notar que al tener un modelo de transición definido por una función de probabilidad, se está asumiendo que no existe certeza en el resultado de las acciones del sistema. Dicho de otra manera, si el sistema se encuentra en un estado s', y se aplica una acción a, no es completamente seguro que se llegue al estado s, por eso se asigna un valor de probabilidad. Por otra parte al decir que se tiene un MDP, se asume también que se tiene un proceso markoviano, es decir que el estado del sistema en un momento del tiempo determinado, solamente depende del estado anterior y de la última acción aplicada y no de toda las historia de acciones y estados previos.

Una vez modelado un problema de decisión como un MDP, se busca encontrar lo que se conoce como *política óptima*. Una política en general, es una función que asigna acciones a los estados, es decir es una especie de tabla que indica que acción se debe tomar cuando el sistema se encuentra en cierto estado. Y una política óptima es aquella función que asigna acciones a los estados de manera que esta asignación represente la mayor utilidad posible, para lo cuál se debe definir una función de utilidad.

La función de utilidad U determina qué tan útil es un estado s suponiendo que las acciones se escogen según una política en particular. Se trata de un concepto distinto al de recompensa. La función de recompensa asigna, para cada una de las acciones que pueden ejecutarse en cada estado, un valor numérico que representa la utilidad inmediata de dicha acción. Sin embargo, la función de utilidad asigna a cada estado un valor numérico que representa la utilidad de dicho estado a largo plazo. Este valor numérico no es la suma exacta de recompensas futuras, que no se puede predecir, sino una aproximación probabilística y se calcula con la siguiente ecuación:

$$U(s) = R(s) + \max_{a \in A} \sum_{s' \in S} P(s'|s, a) U(s')$$
 (3.1)

Uno de los métodos más comunes para poder obtener la política óptima, es el método de Iteración por Valor. Este método se basa en modificar las utilidades de los estados vecinos y repetir este proceso de modificación en cada uno de los estados y acciones durante varias iteraciones hasta que los valores de utilidad converjan. El algoritmo es el siguiente [Russell03]:

Inicializar: 
$$U_0(s) = R(s) \forall s \in S$$

Repetir:

- Incrementar t
- Repetir  $\forall s \in S \ y \ \forall a \in A$ :

$$\quad \quad \quad \quad U_{\scriptscriptstyle t}({\bf s}) \! = \! R({\bf s}) \! + \! \max_{\scriptscriptstyle a} \sum_{{\bf s}' \in {\bf S}} P({\bf s}' | {\bf s}, a) U_{\scriptscriptstyle t-1}({\bf s}')$$

Accion(s)=a

 $|U_t(s)-U_{t-1}(s)| < \epsilon \forall s \in S$ Hasta que

Cuando finaliza la ejecución del algoritmo, se obtiene una política óptima que indica qué acción se debe tomar para cada estado en que se encuentre el sistema. Esto se indica en el algoritmo anterior en la variable *Accion(s)*.

# 3.1.2 EXPLORACIÓN BASADA EN FRONTERAS UTILIZANDO MDPs

En la figura 3.2, se muestra los pasos del ciclo que ejecuta un robot cuando se encuentra explorando el ambiente y construyendo una representación del mismo. El ciclo se ejecuta repetidamente hasta que se completa el mapa del ambiente. Como se observa en la figura, se tiene un servidor central al cuál se le envían los datos del ambiente (puntos, segmentos y celdas libres) de manera que cuando se trabaja con múltiples robots, es este servidor central el que se encarga de recopilar la información proveniente de cada robot.

Como ya describió al inicio de la sección 3.1, el robot toma un lectura con el LRF, se encuentran los segmentos, se ajusta la información al mapa y se determina a qué frontera y a qué punto sobre la frontera debe acudir para continuar la exploración; con la posición sobre la frontera como la siguiente meta durante la exploración, se debe encontrar una trayectoria que conduzca al robot a esa posición y posteriormente efectuar los movimientos necesarios para al llegar a la posición deseada, y tomar una nueva lectura. En la figura 3.2 se observa que antes de definir la frontera a la cuál va a acudir el robot, hay un paso en cuál se calculan las *celdas libres* del ambiente, es decir se tiene una representación adicional del ambiente que es una malla de ocupación para indicar celdas ocupadas y celdas vacías. Esta representación adicional se utiliza para poder modelar el problema de la planificación de la trayectoria hacia la frontera seleccionada por medio de un MDP como se explicará más adelante en esta sección.

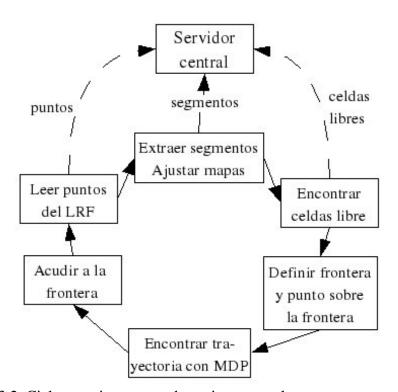


Figura 3.2. Ciclo que ejecuta un robot mientras explora y construye un mapa.

Los primeros pasos del ciclo, la lectura de los puntos del LRF, la extracción de segmentos y la fusión de la información se realizan tal como se explicó en el capítulo anterior, en las secciones 2.2.2 y 2.2.3, después de esto el robot cuenta con un mapa segmentos parcial como el que se muestra en la figura 3.3. En esta figura los segmentos que forman parte del mapa se indican con líneas completas y las fronteras existentes se muestran con líneas punteadas. Tal y como se explicó al inicio de la sección 3.1; se seleccionan las dos fronteras de mayor longitud, es decir los dos segmentos punteados más largos, y se calcula el punto medio de ambos segmentos, seleccionando aquel que se encuentre más cercano a la posición actual de robot. Suponiendo por un momento que ya se ha realizado el proceso necesario para encontrar las celdas libres, cuando se ha seleccionado el punto sobre la frontera al cuál va a ir el robot, se encuentra en qué celda está ese punto y esa celda se convierte en la meta para poder encontrar una trayectoria que conduzca al robot de su posición actual a la siguiente frontera que explorará. La trayectoria se encuentra utilizando un MDP con una malla de ocupación.

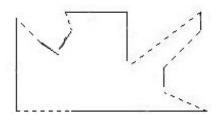


Figura 3.3. Mapa parcial del ambiente y fronteras indicadas con líneas punteadas.

Para obtener la malla de ocupación, se divide el ambiente conocido por el robot en celdas de tamaño regular. Para saber si una celda se encuentra ocupada (por un obstáculo) o vacía (libre), se considera que cada punto obtenido por el LRF define un segmento que sale del láser y que termina cuando el rayo de luz choca con un obstáculo en el punto entregado durante la lectura. Al conocer el segmento que va el sensor láser hasta donde se encuentra el obstáculo definido por el punto, se sabe que todas las celdas por las cuáles atraviesa el segmento se

encuentran libres y que la celda en donde se encuentra el punto se encuentra ocupada. Este procedimiento se muestra en la figura 3.4.

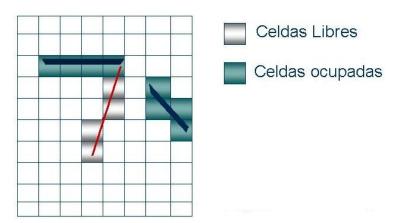


Figura 3.4. Celdas libres y celdas ocupadas.

Con la malla de ocupación obtenida, el MDP que se utiliza para encontrar la trayectoria se define por lo siguiente:

- El conjunto S de posibles estados del sistema, es el conjunto de todas las celdas de la malla de ocupación que se tenga. En realidad el conjunto de celdas representa un conjunto discreto de posibles posiciones en la cuáles se puede encontrar el robot.
- El conjunto A de las posibles acciones que le permitirán pasar al robot de una celda a otra se definió como sigue: si el robot se encuentra en un celda determinada, se puede mover a cualquiera de las ocho celdas vecinas que se encuentran alrededor. Por lo tanto el conjunto de las posibles acciones es: arriba (celda de enfrente), abajo, derecha, izquierda, arriba a la derecha, arriba a la izquierda, abajo a la derecha y abajo a la izquierda. Sin embargo cada posible acción puede significar uno o más movimientos del robot: por ejemplo, si el robot quiere ir a la celda de arriba, basta con que avance una distancia determinada por el tamaño de la celda en línea recta; pero si el robot quiere ir a la celda derecha, debe girar 90° en el sentido de las manecillas del reloj y desplazarse en línea recta.
- El modelo de transición indica la probabilidad de pasar de una celda a otra dependiendo de la acción. En este caso se considera que si al robot se le pide ir a una celda en particular

llamada celda objetivo, existe una probabilidad alta de que llegue a esa celda; sin embargo puede ser que debido a derrapamientos de las ruedas, el robot llegue a alguna de las celdas que se encuentren al lado de la celda objetivo. En la figura 3.5 se muestran dos casos donde el robot quiere ir a la celda de arriba (a) y donde el robot quiere ir a la celda arriba a la izquierda (b). Debido a experimentos realizados, se observó que hay una pequeña probabilidad de que el robot llegue arriba a la derecha o arriba a la izquierda cuando la acción realizada es arriba; lo mismo cuando la acción es arriba a la izquierda, existe una posibilidad de que el robot llegue a la celda de arriba o a la celda de la izquierda. Por lo tanto se definió que el robot llega a la celda objetivo con 0.8 de probabilidad y con una probabilidad de 0.1 a cada una de las celdas contigüas a la celda objetivo.

• La función de recompensa R, se define como sigue: a las celdas ocupadas se les asigna un valor de recompensa igual a 2, a las celdas libres se les asigna un valor de -0.25 y la celda que contiene la meta un valor de 1, simplemente para diferenciarla de las demás celdas.

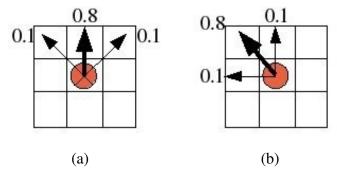


Figura 3.5. Probabilidades de transición de una celda a otra.

Una vez planteado el MDP, se utiliza el algoritmo de iteración por valor para encontrar una política óptima que en este caso le indicará al robot que acción es la más conveniente a tomar si se encuentra en una celda determinada; con el objetivo de ir a la celda meta. Para las celdas ocupadas, no se calcula el valor de utilidad, esto debido a que no son posibles posiciones en donde se puede localizar el robot. La razón principal de utilizar valores negativos, es debido a que en el algoritmo de iteración por valor se considera una suma que va acumulando los valores de

recompensa y utilidad por las celdas por donde el robot podría ir pasando, de tal manera que si los valores de recompensa son positivos, la utilidad de las celdas aumentaría en cuanto más larga fuera la trayectoria a seguir, y el robot finalmente ejecutaría trayectoria muy largas. En cambio, al definir el valor de recompensa negativo, entre más larga sea la trayectoria, menor será la utilidad, ya que por cada celda visitada al sumar la recompensa, en realidad se está disminuyendo el valor de la utilidad. Teniendo en cuenta esto, el robot se va moviendo a la celda vecina que tiene un mayor valor de utilidad en cada paso en el tiempo. Eventualmente el robot será conducido hasta la celda con valor 1, que es la celda en donde se encuentra el punto medio de la frontera que se seleccionó para continuar la exploración. Un ejemplo de una trayectoria generada utilizando un MDP se muestra en la figura 3.6.

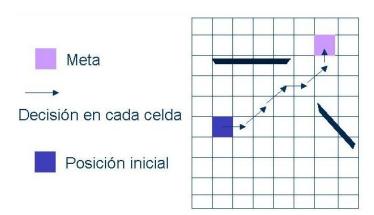


Figura 3.6. Ejemplo de trayectoria del robot utilizando un MDP.

Un aspecto importante de utilizar un MDP para encontrar la trayectoria del robot hacia la siguiente frontera, radica en el hecho de que se está considerando que existe incertidumbre en el resultado del movimiento del robot cuando pasa de una celda a otra. Esta incertidumbre se modela con la probabilidad de transición. Cuando el robot se encuentra moviéndose hacia la frontera, en cada paso después de moverse a la celda siguiente, el robot revisa su posición y se ubica en la celda correspondiente. Si la celda en la que se encuentra no es la celda a la que esperaba llegar, simplemente revisa en la política óptima y se decide a ejecutar la mejor acción para la celda en la que se encuentra realmente.

# 3.2 EXPLORACIÓN CON MÚLTIPLES ROBOTS

En la sección anterior se mencionó que existe un servidor central al cuál se conecta el robot para ir enviando la información del mapa. De manera que la extensión de la exploración basada en fronteras para más de un robot se lleva a cabo teniendo en cuenta que dicho servidor central recibirá información proveniente de varias fuentes. En esta primera aproximación con múltiples robots se considera que la exploración se lleva a cabo de acuerdo a las siguientes características:

- Cada robot explora de manera independiente sin tener conocimiento de la existencia de otros robots que se encuentran explorando.
- Cada robot ejecuta y calcula lo necesario en su propia computadora a bordo, para los pasos que se muestran en la figura 3.2.
- El servidor central recibe la información de puntos, segmentos y celdas libres que le envía cada uno de los robots.
- El servidor central conoce las posiciones iniciales relativas de los robot que van a construir la representación del ambiente.
- El servidor central se encarga de acoplar los mapas que le envían cada uno de los robots y
  formar un mapa global unificado. Esta mapa global se obtiene acoplando los mapas
  locales utilizando como medida para determinar el mejor acoplamiento a la distancia de
  Hausdorff parcial.
- El servidor central se encarga de identificar cuando ya no existen más fronteras que visitar y envía una señal a los robots para que detengan la exploración del ambiente.

El implementar una exploración múlti-robot con las características antes mencionadas, al no ser completamente centraliza en cuanto la coordinación de los robots, permite que se incorporen varios robots al proceso de exploración; así mismo permite que si alguno de los robots

deja de funcionar los robots restantes terminen de construir la representación del ambiente. Sin embargo el mayor inconveniente que se presenta es que debido a que cada robot decide de manera independiente a dónde ir, sin tener conocimiento de los otros robots, pueden presentarse casos en los cuáles los robots interfieran entre sí las trayectorias que ejecutan durante la exploración ya pueden estarse moviendo más de un robot a la vez. Para solucionar este problema en la pruebas experimentales realizadas, simplemente se colocaron lo suficientemente separados para que no interfirieran entre sí o bien se les colocó apuntando sus LRF en direcciones opuestas (como se muestra en la figura 3.7) de manera que cada robot al conocer lo que se encuentra enfrente de sí mismo va a avanzar a fronteras que lo van alejando de otros robots.



Figura 3.7. Configuración inicial de dos robots antes de iniciar la exploración.

# CAPÍTULO 4. EXPLORACIÓN BAJO MÚLTIPLES CRITERIOS

Dentro de la planificación de movimientos de robots móviles existen diferencias en cómo es utilizada la información obtenida del sensado. En algunos casos el sensado es utilizado como una herramienta para resolver una tarea específica, en otros casos el sensado es incorporado como una restricción que se debe satisfacer al momento de generar un plan de movimientos. Por ejemplo, en el caso de la navegación de robots, la información obtenida por los sensores se puede emplear como una herramienta que permita detectar y esquivar obstáculos; en cambio en la exploración para generar un mapa, el sensado de información debe cubrir todo el ambiente para poder obtener una representación completa del ambiente. Por lo tanto en este último caso, se está planificando la forma en que se va a ir percibiendo el ambiente.

Existen diversas estrategias de exploración, cada una de las cuáles responde a los propósitos de construcción del mapa. Por ejemplo, puede ser que se requiera un mapa muy preciso del ambiente con lo cuál la estrategia de exploración probablemente considere acudir a aquellos sitios donde tiene información más incierta. Otro ejemplo puede ser que se quiera obtener un mapa lo más rápidamente posible con la menor cantidad de movimientos del robot, por lo que probablemente la estrategia de movimientos conduzca al robot a aquellos lugares donde se espera obtener una gran cantidad de información nueva. Por lo tanto la información que se obtiene a partir del sensado será utilizada como una restricción que deberá tomarse en cuenta para la

planificación de los movimientos que efectuará el robot durante la exploración.

En este capítulo se describen estrategias de exploración que toman en cuenta información, ya sea del ambiente o del robot mismo, para definir cómo se debe mover el robot en el momento que está construyendo un mapa. La primera estrategia propuesta considera información tal como el número de esquinas percibidas, las marcas sobresalientes, el número de veces que el robot cambiará de ángulo, la distancia que debe recorrer el robot, etc.; y teniendo en cuenta toda esa información se decide cuál es el mejor lugar al que el robot debe ir para obtener la siguiente lectura del ambiente. Esta primera estrategia se presenta tanto para un robot como para múltiples robots y se consideran dos variantes en cada caso: cuando el robot planifica movimientos solamente para llegar al punto donde tomará la siguiente lectura y cuando el robot planifica movimientos para varias configuraciones a las que deberá acudir de manera sucesiva. Esta estrategia de exploración fue reportada durante el desarrollo de esta tesis en [Tovar06].

En la segunda estrategia propuesta se considera fundamentalmente información a cerca de los robots para decidir a qué posición debe acudir cada uno para continuar la exploración. Se considera que puede tenerse un grupo de robots con diferentes características motrices y con sensores de diferentes alcances. Por lo tanto en este caso se toma en cuenta la información de cada robot, es decir qué tan exacto es un sus movimientos y qué tan buenos son sus sensores, para determinar a donde es más conveniente que cada robot se dirija durante la exploración.

# 4.1 EXPLORACIÓN BASADA EN LA GENERACIÓN DE MUESTRAS

En la sección 3.1 del capítulo anterior, se definió la exploración basada en fronteras. En las dos estrategias de exploración definidas en este capítulo, se sigue utilizando la idea de que el robot debe acudir a las fronteras para continuar conociendo el ambiente. Sin embargo al utilizar la

información sensada por el robot para determinar cuál es el mejor lugar para que el robot tome la siguiente lectura, se deben considerar distintas posiciones cercanas a la frontera. Esto debido a que no se obtiene la misma información dependiendo de dónde se localice el robot al sensar el ambiente. Para ilustrar la idea anterior, suponer por ejemplo que el robot debe acudir a determinada frontera; si el robot se acerca a los extremos de la frontera y se posiciona muy cerca de una pared, puede ser que la distancia que tenga que recorrer el robot sea más corta pero puede ser también que sus sensores capten una menor cantidad de información por la oclusión ocasionada por la pared. En cambio, si el robot se dirige al punto medio de la frontera puede ser que tenga que recorrer una mayor distancia, pero tal vez puede captar una mayor cantidad de información nueva con los sensores.

En la primera estrategia de exploración presentada en este capítulo se genera un conjunto de muestras en cada frontera; donde cada muestra representa una posible localización del robot (coordenadas x y y del robot y ángulo de orientación, es decir una configuración). Posteriormente se evalúa cada muestra con una función de utilidad y aquella muestra que haya obtenido la calificación más alta define la posición a donde el robot acudirá en el siguiente paso de la exploración. Esta idea se muestra en la figura 4.1, donde las muestras se presentan como pequeños círculos cercanos a las fronteras y el robot es representado por el círculo más grande. Después de evaluar las muestras de la figura 4.1(a), se selecciona una en particular y en la figura 4.1(b) el robot se ha dirigido a esa posición. Este proceso se repite de manera sucesiva hasta que ya no se tienen fronteras por descubrir.

Para poder generar las muestras, se sigue el siguiente proceso por cada una de las fronteras que se tienen hasta determinado momento:

- Se calcula el punto medio de la frontera.
- Se generan un círculo imaginario teniendo como centro al punto medio y como radio un valor arbitrario que se encuentre del alcance del sensor LRF.

- Se generan puntos aleatorios dentro de la círculo definido.
- Se determina si los puntos generados se encuentran del lado de la frontera que pertence al mapa conocido. Si los puntos se encuentran dentro del mapa conocido se guardan de lo contrario se descartan.
- Para cada punto se generan también de manera aleatoria ángulos que definirán la orientación del robot en dicha muestra.

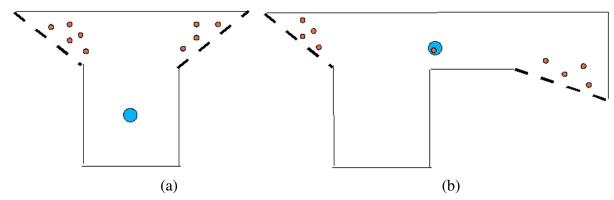


Figura 4.1. Exploración basada en la generación de muestras cercanas a las fronteras.

Para cada frontera se determina un número máximo de muestras que se deben generar, el número máximo se define de manera arbitraria. Estas muestras serán utilizadas como posibles configuraciones del robot en donde se colocará para obtener una nueva lectura del ambiente ya sea que se considere visitar una sola frontera en un paso de la exploración o una secuencia de fronteras.

# 4.2 DEFINICIÓN DE LA FUNCIÓN DE UTILIDAD A EVALUAR

En esta sección se describe una función de utilidad que puede ser utilizada para evaluar las muestras que definen las posibles localizaciones siguientes para el sensado durante la exploración. La funcion seleccionada se desea que tome en cuenta las siguientes características para determinar

que tan útil es una localización dada:

- Que selecciona aquellas localizaciones en donde se puedan reconocer landmarks que faciliten la localización del robot.
- Que se consideren posiciones donde se pueda facilitar el proceso de alineamiento entre el mapa global y la última lectura adquirida.
- Que se construya el mapa lo más rápidamente posible tratando de sensar en cada paso grandes áreas inexploradas.
- Que se tenga la menor incertidumbre en cuanto a la posición del robot considerando que algunos movimientos pueden incrementar el error en cuanto a su posición reportado por los sensores odométricos.
- Que la trayectoria que debe seguir el robot hasta la siguiente localización sea lo más corta posible.
- Que la siguiente localización para el robot no se encuentre demasiado cerca de una pared o algún otro objeto.

En la función de utilidad, se considera que las posiciones y orientaciones del robot desde las cuáles se pueden percibir uno o varios landmarks del ambiente; representan posiciones que pueden ser útiles para reducir la incertidumbre en la localización del robot o bien ayudar en la tarea de navegación [Lazanas95]. Un landmark puede ser cualquier objeto sobresaliente en el ambiente y que pueden ser definidos como marcas artificiales. Para este trabajo, se asume que los landmarks pueden ser reconocidos con cierto porcentaje de error, y que se utiliza el método de clasificación bayesiana que provee un valor de probabilidad de que determinado landmark sea identificado desde determinada posición. Por lo tanto se desea que aquellas muestras que representan localizaciones desde las cuáles se pueden ver un mayor número de landamarks y con alta probabilidad de ser identificados, tenga un mayor valor de utilidad. Esto se representa mediante la siguiente expresión:

$$\frac{1}{n_i}\sum_{k=1}^{n_i}p_k \tag{4.1}$$

Donde:

 $n_i$  es el número total de landmarks que se pueden ver desde la posición y orientación que definen a la muestra i.

 $P_k$  Es la probabilidad de que el landmark k sea identificado desde la muestra i.

Además de los landmarks, se consideran algunas características que pueden ayudar a un mejor alineamiento de los mapas parciales con el mapa global. Por ejemplo, considérese el caso en el que un robot se encuentra construyendo el mapa de un pasillo muy largo. Cuando se tome la primera lectura probablemente se obtendrán dos segmentos paralelos que representan una parte del pasillo, cuando se tome la siguiente lectura nuevamente se obtendrán dos segmentos paralelos. La dificultad en alinear estos dos mapas resulta debido que no se sabe a partir de donde se traslapan ambas representaciones o si son completamente independientes o bien son exactamente las mismas, ya que el robot puede haber derrapado al moverse o avanzar más de lo estimado.

Teniendo en cuenta lo anterior, se considera que las esquinas del ambiente que puedan ser detectadas desde cierta posición, pueden ayudar a facilitar el alineamiento de partes del mapa. De manera que en a la expresión 4.1 se le añade el término  $Ne_i$ , que representa el número de esquinas que pueden ser detectadas desde la posición y orientación definidas por la muestra i.

$$Ne_i + \frac{1}{n_i} \sum_{k=1}^{n_i} p_k$$
 (4.2)

Si se quiere que uno o varios robots construyan una representación del ambiente lo más rápidamente posible, es necesario que en cada lectura se trate te abarcar la mayor cantidad de

nueva información que se debe incorporar al mapa. Sin embargo, debido que los robots se encuentran en un ambiente desconocido, es imposible saber en qué lugar se va poder recopilar más información. Nuevamente al igual que lo señalado en el capítulo 3, se asume que la longitud de una frontera puede dar una idea a cerca de la información que se va a descubrir si el robot acude a la frontera. Como resultado de esto, si se utiliza la expresión 4.3, se tendrá un mayor valor si la frontera es más larga ( $lv_i$ , denota la longitud de la frontera) y si la muestra se encuentra más cerca de la frontera ( $sv_i$ , distancia desde la muestra a la frontera) como se muestra en la figura 4.2.

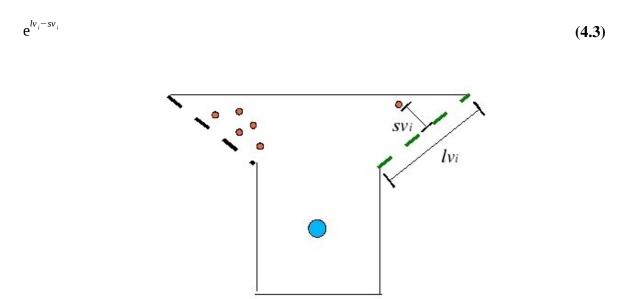


Figura 4.2. Longitud de la frontera y distancia de la muestra a la frontera.

Otro factor importante para que la construcción del mapa sea realizada rápidamente, es que el robot se desplace la menor distancia posible. De esta manera se está considerando indirectamente que el consumo de energía sea también el menor posible. Como consecuencia de esto se debe dar un valor de mayor utilidad a aquellas muestras que involucren un menor desplazamiento del robot cuando se mueva de su última posición a la siguiente donde continuará la exploración.

Sin embargo, no solamente la distancia de la trayectoria que ejecutará el robot debe tomarse en cuenta, sino también el tipo y el número de movimientos que efecturá. Ya que se sabe que los movimientos rotacionales generan una mayor incertidumbre en cuanto a la localización del robot que los movimientos de desplazamiento en línea recta. Se utilizó un modelo simple de incertidumbre, en el cuál la incertidumbre en la localización del robot crece en proporción a la raíz cuadrada de la distancia viajada. El modelo también incluye una penalización por los movimientos de rotación de manera que se evalúen mejor las trayectorias rectas. Por lo tanto se tiene un modelo en el cuál la cantidad de incertidumbre varía dependiendo de cómo se mueva el robot, y para evaluar la utilidad de una trayectoria que conduce a una muestra se considera que entre menos rotaciones y menor sea la distancia viajada es mejor. La expresión que se utiliza para evaluar la utilidad una trayectoria es la siguiente:

$$\prod_{j=1}^{q_j} \frac{\mathrm{e}^{-|\theta_j|/\lambda}}{\sqrt{s_j} + 1} \tag{4.4}$$

## Donde:

j es el índice de un paso en la trayectoria.

 $q_i$  es el número total de pasos.

 $\theta_i$  es el ángulo (en radianes) que el robot debe girar en cada paso.

 $s_i$  es la distancia el robot debe avanzar en línea recta en el paso j.

 $\lambda$  es una constante.

En la figura 4.3, se muestra una trajectoria con varios pasos donde se indican los ángulos y distancia que debe avanzar el robot. Como se puede observar, para el trabajo desarrollado en esta tesis el robot siempre avanzará girando y avanzando en línea recta.

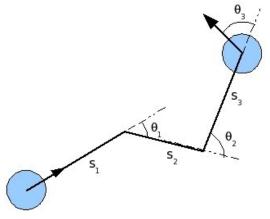
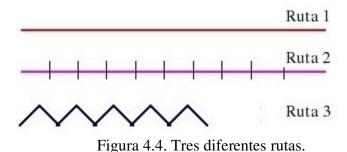


Figura 4.3. Una posible trayectoria para el robot durante la exploración.

Para ejemplificar como se utiliza la expresión 4.4, se presentan en la figura 4.4 tres diferentes trayectorias. La primera ruta es simplemente un desplazamiento en línea recta desde el punto inicial hasta el final. La segunda ruta también es una línea recta, sin embargo en este caso el robot se detendrá en cada una de las líneas verticales marcadas, es decir el robot avanza en línea recta, luego se detiene y nuevamente vuelve avanzar y así sucesivamente; así recorre en total la misma distancia que en la ruta 1. Finalmente en la ruta 3, el robot avanza con movimientos en zig-zag, avanza y gira alternativamente.



En la gráfica que se muestra en la figura 4.5, se tienen los valores parciales de evaluar la expresión 4.4 que se van calculando en cada paso de las rutas. Como se puede observar, debido a que la primera ruta es realizada por el robot en un solo paso, en la gráfica se muestra una línea que indica un valor de utilidad después de realizar ese único paso. En cuanto a la ruta 2, conforme el robot avanza más, el valor de utilidad va disminuyendo con cada paso. Y finalmente en el caso

de la ruta 3, debido a qué el robot rota y avanza, el valor de utilidad disminuye aún más rápidamente. Por lo tanto se está considerando con la expresión 4.4 no sólo el valor de utilidad de acuerdo con la distancia que viaja el robot sino también de acuerdo al tipo de movimientos o controles que se aplican al robot.

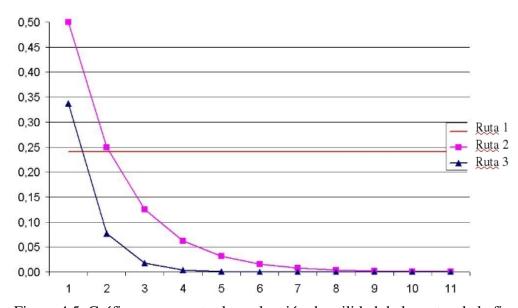


Figura 4.5. Gráfica que muestra la evaluación de utilidad de las rutas de la figura 4.4.

La última característica que se definió para la función de utilidad, es a cerca de la localización del robot con respecto a las paredes y a los obstáculos. Debido a que si el robot se acerca demasiado a las paredes u obstáculos, éste puede colisionar o bien al encontrarse demasiado cerca alguno de sus sensores puede volverse "ciego" (al no captar información alguna), es necesario que se defina una distancia mínima a la cuál se puede acercar el robot. Aquellas muestras que se encuentren a una distancia menor a esa distancia mínima, deben ser tener un menor valor de utilidad que aquellas muestras que se encuentren más alejadas de las paredes. Teniendo en cuenta esto, en la figura 4.6 se muestra una gráfica que indica el valor de utilidad para diferentes distancias a las paredes, es simplemente una función que a partir de la distancia mínima asigna un valor constante de utilidad igual a uno. Esta idea se representa con la expresión 4.5, donde  $d_1$  representa la distancia que hay desde la muestra hasta la pared u

obstáculo más cercano.



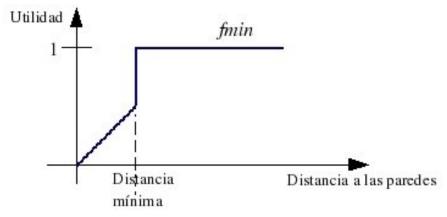


Figura 4.6. Función *fmin* que penaliza las posiciones muy cercanas a paredes u obstáculos.

Hasta este el momento, con las expresiones que van de la 4.1 a la 4.5 se han definido formas de medir el valor de utilidad para cada una de las características que se mencionaron al inicio de esta sección. Para tener una sola función de utilidad, se ha seleccionado una función multiplicativa; de manera que una configuración (representada por una muestra) en la cuál al menos una de las características tenga una evaluación muy baja, será descartada a pesar de que en alguna otra característica esté bien evaluada. Por ejemplo, si existe una configuración o muestra cercana a una frontera y con la posibilidad de que se obtenga gran cantidad de información, deberá ser descartada si no existen suficientes esquinas o landmarks que ayuden a alinear la información con el mapa global. La función de utilidad final que se utiliza para evaluar una configuración o muestra es la que se indica en la ecuación 4.6, donde el índice *i* indica la i-ésima configuración generada de manera aleatoria.

$$Utilidad_{i} = e^{(lv_{i} - sv_{i})} \prod_{j=1}^{q_{j}} \frac{e^{-|\theta_{j}|/\lambda}}{\sqrt{s_{i}} + 1} \left( Ne_{i} + \frac{1}{n_{i}} \sum_{k=1}^{n_{i}} p_{k} \right) fmin_{i}(d_{l})$$
(4.6)

Como se explicará en las secciones 4.3.2 y 4.4.2, se considera la idea de que el robot genere configuraciones y visite más de una frontera conocida antes de volver a generar muestras para continuar la exploración. En estos casos, se debe contar con una medida de utilidad que evalúe una secuencia de configuraciones que tendrá el robot al ir visitando varias fronteras antes de volver a analizar y generar nuevas muestras. La utilidad de la secuencia de estas configuraciones se definió simplemente como la suma de la utilidad de cada configuración, como se muestra en la expresión 4.7; donde *m* representa el total de las muestras que visitará el robot al acudir a varias fronteras de manera sucesiva en una trayectoria *T*.

$$Utilidad_{T} = \sum_{i=1}^{m} Utilidad_{i} = \sum_{i=1}^{m} \left| e^{(lv_{i} - sv_{i})} \prod_{j=1}^{q_{j}} \frac{e^{-|\theta_{j}|/\lambda}}{\sqrt{s_{j}} + 1} \left( Ne_{i} + \frac{1}{n_{i}} \sum_{k=1}^{n_{i}} p_{k} \right) fmin_{i}(d_{l}) \right|$$

$$(4.7)$$

Otra posible expresión para poder evaluar la utilidad de una trayectoria consiste en calcular el promedio de utilidad para las muestras que visitará el robot. Dicho de otra manera, en vez de tener la sumatoria definida en 4.7, el resultado de la sumatoria se tendría que dividir por el número total de muestras m. La idea de utilizar el promedio en vez de la suma es debido a que es posible que en trayectorias largas, la evaluación de la función de utilidad dé como resultado un valor alto, porque se están sumando las utilidades individuales de varias muestras; dando como resultado trayectorias bien evaluadas debido a la gran cantidad de muestras individuales que se incluyeron en la suma. Esto podría originar que al comparar trayectorias de distinta longitud (en cuanto al número muestras de cada trayectoria), es muy probable que la trayectoria más larga tenga un mayor valor de utilidad. Sin embargo, para los casos que se exponen en las secciones 4.3.2 y 4.4.2 este problema no se presenta, ya que si el robot tiene que visitar 3 fronteras por ejemplo, todas las trayectorias que se van a generar y a comparar tendrán una longitud de 3, ya que se deberán visitar todas las fronteras.

## 4.3 EXPLORACIÓN CON UN ROBOT

En esta sección se describe cómo se utiliza la función de utilidad definida en el punto 4.2 de este capítulo para el caso de un sólo robot, tanto considerando que visitará una sola frontera como en el caso en que acude a visitar varias fronteras antes de generar nuevas muestras.

### 4.3.1 CONSIDERANDO UN PASO ADELANTE EN EL TIEMPO

Este es el caso más sencillo de exploración propuesto con la función de utilidad dada. Fundamentalmente se siguen los pasos descritos en la sección 4.1: se identifican las fronteras del mapa obtenido hasta determinado momento, se generan muestras de posibles configuraciones para el robot cercanas a las fronteras, se evalúan todas las muestras con la ecuación 4.6, se selecciona la muestra mejor evaluada para que el robot llegue a esa configuración para obtener la siguiente lectura. Y este proceso se repite hasta que no existan nuevas fronteras por explorar.

En este caso la estrategia de exploración considera un paso adelante en el tiempo, es decir cada vez que se generan y evalúan posibles configuraciones para el siguiente sensado, el robot solamente acude a una de éstas y se repite nuevamente el proceso de generación de muestras en las fronteras. Dicho de otra manera, solamente se visita una frontera y el algoritmo de exploración solamente obtiene un plan para el siguiente sensado y nada más.

### 4.3.2 CONSIDERANDO VARIOS PASOS ADELANTE EN EL TIEMPO

A diferencia del caso descrito en la sección 4.3.1, cuando se consideran varios pasos adelante en el tiempo, el algoritmo de exploración deberá obtener un plan en el cuál se tenga una lista de varias configuraciones del robot. La lista con las configuraciones del robot, incluyen una

configuración por cada frontera diferente que el robot deberá visitar; en cada frontera se considera que el robot realiza una operación de sensado y luego se dirige a la frontera siguiente. Solamente cuando el robot ha visitado todas las fronteras del plan obtenido, entonces se generan nuevamente muestras aleatorias para las nuevas fronteras. Por lo tanto cuando se generan las configuraciones, en vez de seleccionar sólo la muestra mejor evaluada, se selecciona una secuencia de configuraciones (una por frontera) que deberán ser visitadas por el robot; de manera que para seleccionar la secuencia de mayor utilidad es necesario utilizar la ecuación 4.7.

Para ejemplificar la idea de cómo funciona la estrategia de exploración considerando varios pasos adelante en el tiempo, se presenta el caso que se muestra en la figura 4.7. Se supone que durante algún momento en la exploración, el robot se encuentra en una configuración denotada como  $r_{ini} = (x_{ini}, y_{ini}, \theta_{ini})$  y existen tres fronteras que deberán ser visitadas denotadas con las letras A, B y C. Para este ejemplo en particular, suponemos que se generan dos configuraciones (muestras) por cada frontera. En la figura 4.7, cada muestra en este caso se encuentra representada por una flecha para indicar que incluye posición en (x, y) y la orientación  $\theta$ . Además cada muestra está etiquetada con una letra minúscula que indica a qué frontera está asociada y un índice que indica el número de muestra de esa frontera en particular.

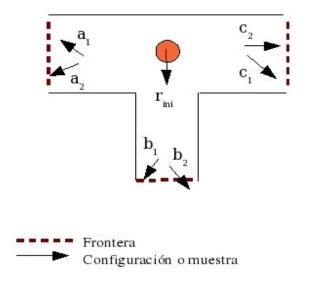


Figura 4.7. Robot enfrente de tres fronteras por explorar.

El algoritmo de exploración, en el caso descrito en el párrafo anterior, deberá encontrar en qué orden y en cuál configuración el robot deberá visitar cada una de las tres fronteras. Cuando el robot se encuentra en  $r_{ini}$ , para el primer paso tiene como posibles opciones acudir a cualquiera de las seis configuraciones existentes (desde  $a_1$  hasta  $c_2$ ). Suponiendo que en el primer paso el robot acude a la frontera B, en la configuración  $b_1$ , para el segundo paso el robot tiene como opciones las configuraciones:  $a_1$ ,  $a_2$ ,  $c_1$ ,  $c_2$ . Esto debido a que la configuración  $b_2$ , ya no es una opción para el segundo paso, ya que la frontera B ya ha sido previamente visitada. Y así de manera sucesiva se pueden ir planteando las opciones para cada paso.

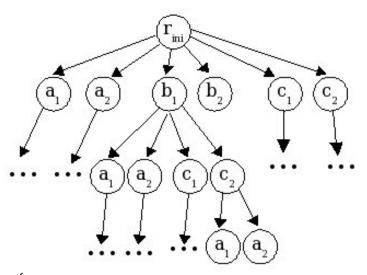


Figura 4.8. Árbol de búsqueda para visitar las tres fronteras de la figura 4.7.

Como se desea evaluar cuál es la mejor trayectoria que debe seguir el robot, es decir cuál es el mejor orden para visitar las fronteras y dentro de cada frontera cuál es la configuración adecuada, entonces se necesitan modelar todas las posibles trayectorias que el robot podría seguir para explorar las fronteras conocidas. En la figura 4.8, se muestra un árbol que modela todas las posibles trayectoria que pueden generar a partir del caso mostrado en la figura 4.7. Cada nodo del árbol representa una configuración. Los hijos de cada nodo representan las configuraciones que pueden ser visitadas considerando que previamente ya se ha visitado el nodo padre. Y cada nivel

del árbol indica un paso en el tiempo. Por lo tanto si existen tres fronteras por visitar, el árbol tiene tres niveles a partir del nodo raíz.

Se puede notar que en el árbol anterior, se ha impuesto una restricción al momento de construir el árbol: el robot no puede regresar a una frontera que ya ha sido visitada en un paso previo. Esto significa que si un nodo tiene la etiqueta p en el nivel 1, todas las ramas que penden de dicho nodo no tendrán ningún nodo p en los niveles subsecuentes. Además las ramas que surgen a partir del nodo p, no tendrán nodos que representen configuraciones que se encuentren en la misma frontera a la que pertenece p.

Para construir un árbol como el que se muestra en la figura 4.8, para cualquier número de fronteras y cualquier número de muestras por frontera, se deben obtener todas las trayectorias posibles permutando las configuraciones sin permitir que se repitan configuraciones que pertenecen a la misma frontera. Para cada posible trayectoria, que es representada por una rama del árbol desde la raíz hasta la hoja, se puede calcular la utilidad con la ecuación 4.7 y así se puede determinar cuál de las trayectorias es la de mayor utilidad y por lo tanto la que ejecutará el robot.

Se considera que durante el proceso de exploración el robot se puede encontrar en dos cualquier de dos estados posibles: viajando a una configuración o bien sensando en una configuración previamente alcanzada. Por ejemplo, considerando el ejemplo de la figura 4.7, después de evaluar todas las posibles trayectorias, se puede haber seleccionado la siguiente:  $b_1$ ,  $a_2$ ,  $c_1$ ; por lo tanto el robot se encontrará durante la trayectoria en los siguientes seis estados de manera consecutiva: viajando a  $b_1$ , sensando en  $b_1$ , viajando a  $a_2$ , sensando en  $a_2$ , viajando a  $c_1$  y finalmente sensando en  $c_1$ . Teniendo los estados en cuenta y comparando con los nodos del árbol de la figura 4.8, se puede ver que por cada nodo en la trayectoria, se tienen dos estados (viajando y sensando) excepto para el nodo raíz. Por lo que si se construye un árbol con los estados del

robot, cada nodo del árbol de la figura 4.8 se divide en dos nodos uno para indicar el estado viajando y otro para indicar el estado sensando como se muestra en la figura 4.9. Cada etiqueta de los nodos antepone una letra S (sensing, es decir sensando) o una letra T (traveling, es decir viajando) a las etiquetas de las configuraciones para indicar los estados. Por ejemplo si el nodo tiene una etiqueta T- $a_2$ , significa que el robot está viajando a la configuración  $a_2$ .

Esta representación de estados en el árbol puede parecer innecesaria, pero este esquema se utiliza para extender la exploración basada en muestras para varios pasos adelante en el tiempo para múltiples robot como se verá en la sección 4.4.2.

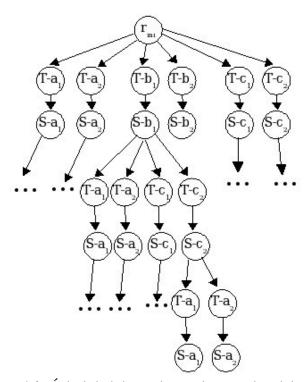


Figura 4.9. Árbol de búsqueda con los estados del robot.

Para el caso general, suponiendo que en determinado momento el robot tienes n fronteras por visitar y que hay exactamente m muestras aleatorias por cada frontera, el árbol de búsqueda tendrá n niveles. Además en el primer nivel del árbol habrá n.m nodos hijos del nodo raíz. Para el segundo nivel habrán en total (n-1).m nodos por cada nodo del primer nivel; esto debido a que

una de las fronteras ya habrá sido visitada y el robot deberá seleccionar entre las n-1 fronteras que no se han visitado. Para el tercer nivel por lo tanto, habrá (n-2).m hijos por cada nodo del segundo nivel; y así sucesivamente para los subsecuentes niveles.

A partir de las observaciones anteriores, se puede notar que el número de nodos del árbol crece exponencialmente dependiendo de n y m. Si los valores de n y m son muy grandes, la búsqueda de la trayectoria con la mayor utilidad en el árbol puede llegar a ser intratable. Debido a esto se propuso un reducción al árbol de búsqueda limitando el número de ramas que se desarrollan completamente. El algoritmo utilizado para la reducción es el llamado branch and bound (limitar ramas) [Aho83]. La idea de este algoritmo consiste en desarrollar el árbol completo hasta cierto nivel w < n. Cuando se llega al nivel w, se selecciona al nodo hoja con la mayor utilidad hasta ese momento y se continúa expandiendo solamente la rama del nodo hoja mejor evaluado. Nuevamente se desarrolla el árbol hasta cierto nivel w y se vuelve a realizar otra evaluación para determinar la rama que se continuará expandiendo. Esto se repite hasta que se llega al último nivel del árbol. La idea de esta reducción en el árbol de búsqueda se muestra en la figura 4.10, donde el árbol completo se encuentra representado por el tríangulo grande al fondo; mientras que los tríangulos pequeños y sombreados indican la parte del árbol total que se desarrolla para la búsqueda.

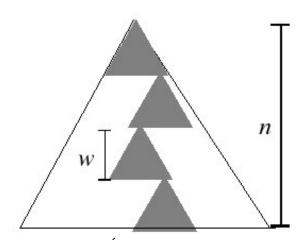


Figura 4.10. Árbol de búsqueda reducido.

## 4.4 EXPLORACIÓN CON MÚLTIPLES ROBOTS

La estrategia de exploración propuesta en la sección anterior para un sólo robot se extendió para el caso de realizar una exploración con múltiples robots. En este caso, se definió la utilización de un planificador centralizado que calcula las configuraciones para todos los robots que se encuentran construyendo el mapa. Esta es una diferencia importante con la estrategia de exploración que se definió en el capítulo 3, en la cuál cada robot se encargaba de planificar su propia trayectoria. Otra diferencia es que durante la exploración propuesta en el capítulo 3, se permitía que más de un robot se estuviera moviendo a la vez; lo que ocasionaba que se tenía que tener cuidado al seleccionar las posiciones iniciales de los robots de manera que ningún robot interfiriera con otro. En cambio, en la estrategia de exploración propuesta ahora, se definió que solamente un robot se puede mover a la vez; es decir a cada robot le corresponde un turno para poder ejecutar sus movimientos. Si bien es cierto que esto hace más lenta la construcción del mapa, evita el tener que realizar cálculos para poder coordinar el movimiento simultáneo de varios robots.

En esta sección nuevamente se presentan dos versiones para la exploración con múltiples robots: cuando cada robot visita una sola frontera en un paso de la exploración (un paso adelante en el tiempo) y el caso en el que cada robot visitará más de una frontera antes de definir nuevas muestras para continuar la exploración (varios pasos adelante en el tiempo).

### 4.4.1 CONSIDERANDO UN PASO ADELANTE EN EL TIEMPO

Este es el caso más sencillo para la exploración con múltiples robots, se puede considerar un

variante del caso mostrado en la sección 4.3.1 que no requiere muchas modificaciones para funcionar con más de un robot a la vez. Se puede decir que al algoritmo presentado en la sección 4.3.1 lo único que hay que cambiarle es que cada vez que se generan muestras y se evaluán, se considera que es el turno de uno de los robots en particular. Teniendo en cuenta esto, la exploración se lleva a cabo de la siguiente manera:

### Repetir para cada robot:

- Generar configuraciones aleatorias en las fronteras conocidas hasta el momento.
- Evaluar cada configuración utilizando la ecuación 4.6. Para esto hay que tomar en cuenta la posición actual de robot en cuestión, para poder evaluar la parte correspondiente a la expresión 4.4.
- Seleccionar la configuración mejor evaluada.
- Realizar los movimientos necesarios para que robot llegue a la configuración deseada.
- Sensar el ambiente y construir un mapa local.
- Alinear el mapa local con el mapa global y actualizar el mapa global.

Se puede observar a partir de los pasos descritos anteriormente, que no se presenta restricción alguna para que el robot acuda a una frontera en particular. Sin embargo debido a la expresión 4.4 (la referente al movimiento del robot), serán mejor evaluadas aquellas configuraciones que se encuentren más cercanas al robot; por lo tanto generalmente cada robot acude a la frontera que se encuentra "en frente" de él y que seguramente ese mismo robot descubrió en el paso anterior.

### 4.4.2 CONSIDERANDO VARIOS PASOS ADELANTE EN EL TIEMPO

El proceso de exploración explicado en la sección 4.3.2 se presenta a continuación adaptado para el caso de múltiples robots. En el caso general se pueden tener n robots y m fronteras por visitar,

de manera que a cada robot se le puede asignar una o más fronteras por visitar como resultado de ejecutar el algoritmo de búsqueda que se presentará en esta sección. Las consideraciones para llevar a cabo la exploración multi-robot son las siguientes:

- Solamente uno de los robots se encuentra moviéndose a la vez. Es decir se asume que un robot se encuentra en el estado viajando y el resto de los robots se encuentra en estado sensando.
- El estado del sistema multi-robot está definido como un vector que tiene los estados de cada uno de los robots. Por ejemplo un posible estado del sistema puede ser. (robot 1 viajando, robot 2 sensando, robot 3 sensando).
- Una vez que una frontera es explorada por alguno de los robots, dicha frontera no volverá a ser visitada por ningún otro robot.

En este caso también se construye un árbol de búsqueda en donde los nodos representan estados del sistema multi-robot. Para explicar cómo se genera el árbol en el caso de múltiples robots, se considera el ejemplo mostrado en la figura 4.11. En este ejemplo, se tienen tres fronteras por explorar y dos robots, y en cada frontera se ha generado solamente una configuración.

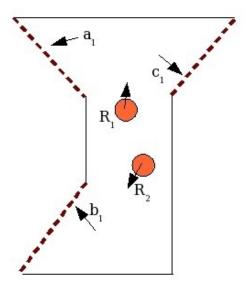


Figura 4.11. Dos robots con tres fronteras por explorar.

Para poder desarrollar el árbol de búsqueda para el ejemplo de la figura 4.11, el estado del sistema multi-robot será denotado por vectores con dos entradas: la primera indicando el estado del robot 1 y la segunda para indicar el estado del robot 2. Un ejemplo de estado para este caso sería el siguiente:  $(R_1-T-a_1, R_2-S-c_1)$ , donde se está indicando que el robot 1 está viajando a la configuración a1 y el robot 2 se encuentra sensando en la configuración c1. Estos estados del sistema serán los nodos del árbol. El nodo raíz en este caso indica la configuración inicial de ambos robots antes de continuar con la exploración; esta configuración inicial se denota con el vector  $(R_1-S-I_1, R_2-S-I_2)$ .

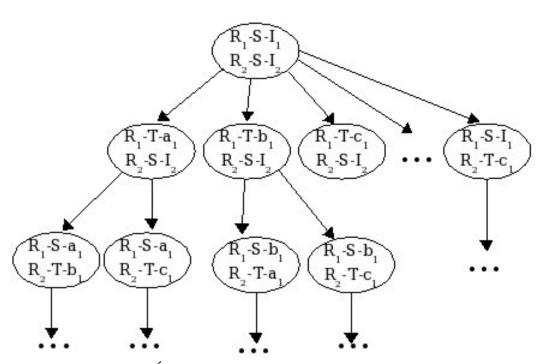


Figura 4.12. Árbol de búsqueda para ejemplo de la figura 4.11.

En la figura 4.12, se muestra el árbol desarrollado para el ejemplo presentado en la figura 4.11. Se puede observar que el nodo raíz tiene la configuración inicial del sistema, donde los dos robots se encuentran en su posición inicial. Posteriormente, existen seis posibles nodos hijos del nodo raíz: tres estados donde el robot 1 se encuentra en su posición inicial mientras que el robot 2

se encuentra viajando a cualquiera de las tres configuraciones posibles  $(a_l, b_l \circ c_l)$ , y otros tres estados en donde el robot 2 es el que permance en la posición inicial mientras que el robot 1 está viajando a cualquiera de las tres fronteras. Nótese que en la figura 4.12 en el primer nivel después de la raíz solamente se muestran cuatro de los seis posibles estados.

Considerando que los robots van a ir alterando sus movimientos, es decir primero se mueve uno y luego otro, en el segundo nivel del árbol se pueden observar que si en el nodo padre (nodo correspondiente al primer nivel) el robot 1 estaba viajando, entonces los nodos del segundo nivel tendrán como viajando al robot 2.

Para poder asignar un valor de utilidad que defina qué trayectoria de los robots de todas las posibles desarrolladas en el árbol es la mejor, es necesario definir un valor de utilidad por pasar de un estado a otro en el árbol. Debido a que solamente un robot se mueve a la vez, el valor de utilidad asignado para cada arco del árbol corresponde al valor de utilidad del robot que se mueve de una configuración a otra. Por ejemplo, si el sistema se encuentra en el estado ( $R_1$ -S- $a_1$ ,  $R_2$ -T- $b_1$ ) y el estado siguiente es ( $R_1$ -T- $c_1$ ,  $R_2$ -S- $b_1$ ), entonces el valor asignado al arco correspondiente es la utilidad calculada con la ecuación 4.6 cuando el robot 1 se mueve la configuración  $a_1$  a  $c_1$ . Teniendo en cuenta esto, la utilidad para una trayectoria está dada por la suma de las utilidades de los arcos por donde se definió dicha trayectoria; y así al desarrollar el árbol completo se puede determinar cúal es la mejor trayectoria. Para el caso de la figura 4.11 se puede obtener como un ejemplo de trayectoria a la siguiente: ( $R_1$ -T- $c_1$ ,  $R_2$ -S- $I_2$ ), ( $R_1$ -S- $c_1$ ,  $R_2$ -T- $b_1$ ) y ( $R_1$ -T- $a_1$ ,  $R_2$ -S- $b_1$ ); lo cuál define que primero se moverá el robot 1 a la configuración  $c_1$ , luego se moverá el robot 2 a  $b_1$  y finalmente el robot 1 visitará a la configuración  $a_1$ . De esta manera entre los dos robot visitarán las 3 fronteras.

Nuevamente se observa que el desarrollo del árbol para el caso de múltiples robots puede ser muy extenso dependiendo de número de fronteras, del número de configuraciones por frontera y el número de robots, por lo tanto en este caso también se puede aplicar el algoritmo de branch and bound para reducir el árbol de búsqueda.

# 4.5 EXPLORACIÓN CON MÚLTIPLES ROBOTS CON DIFERENTES CARACTERÍSTICAS

En la estrategia de exploración mencionada hasta ahora en este capítulo, hay un aspecto que no se tomó en cuenta y que pueden llegar a presentarse sobre todo cuando se trabaja con varios robots reales. Este aspecto, a partir del cuál surgió la idea de esta sección, tiene que ver con el hecho de que se pueden tener robots con diferentes características de movimiento. Dicho de otra manera, el modelo de incertidumbre propuesto para poder definir la utilidad de los movimientos del robot con la expresión 4.4, puede no ser válido para todos los robots o puede presentar variantes; debido por ejemplo a que un robot puede ser más preciso que otro en sus movimientos rotacionales o en su movimiento en trayectorias rectas; o bien debido a que un robot derrapa más que otro dependiendo del tipo de superficie del piso.

En la ecuación para definir la utilidad de una muestra, como se indicó en la sección 4.2, se tiene una expresión en la cuál se consideran los landmarks y esquinas que se pueden "ver" desde la configuración que se está evaluando. De esta manera se está considerando que los robots pueden tener diferentes características de sensado; es decir si se tiene un robot con un sensor de poco alcance, cuando se evalúe una configuración como posible localización siguiente del robot, se debe tomar en cuenta cuántas esquinas y landmarks podrá ver el sensor de ese robot si acude a esa configuración. Esta idea de que los robots puedan tener diferentes características de sensado se retoma también en la estrategia de exploración que se define en esta parte de la tesis.

En esta sección en particular, se hace referencia a robots con distintas características para indicar que se tienen robots con diferencias en cuanto a la precisión de sus movimientos y además diferencias en cuanto a la percepción del ambiente por medio de sus sensores. Por lo tanto, se toma en cuenta que existe incertidumbre tanto en el sensado como en el control de los movimientos, y que cada robot tendrá diferentes "grados" de incertidumbre en sus movimientos y en el sensado. De manera particular para el trabajo desarrollado en esta sección, se asume que las diferentes características de los robots solamente son de acuerdo al "grado" de incertidumbre que cada uno tiene tanto en sensado como en control. Cabe aclarar que no se toman en cuenta otras diferencias que podrían presentarse como pueden ser el tamaño y el tipo de sensores que tiene cada robot.

## 4.5.1 DEFINICIÓN DEL PROBLEMA

Nuevamente se considera una situación similar a la presentada anteriormente; varios robots se encuentran construyendo un mapa de un ambiente originalmente desconocido, y en determinado momento, existen varias fronteras que los robots deben explorar para descubrir nuevas áreas del ambiente. Si se tienen dos robots o más con diferentes características, se necesita determinar a qué frontera es más conveniente que cada robot vaya tomando en cuenta sus diferencias. Para esta parte se asume que cada robot visitará solamente una frontera a la vez, es decir el algoritmo de exploración funcionará solamente para un paso adelante en el tiempo. De manera que la salida del algoritmo que defina la exploración será la mejor asignación de una frontera a cada robot.

Como ya se mencionó, cada robot puede tener distintas capacidades motrices y sensoriales. Se asume que cada robot cuenta con un modelo de control que indica cuál es la probabilidad de que el robot alcance determinada posición dado que aplicó determinados controles para moverse de una posición a otra. También cada robot cuenta con un modelo para sus sensores que indica cuál es la probabilidad de que el robot sense su ubicación dado que se

encuentra en determinada configuración.

Por otra parte, debido a que se desea la mejor asignación una frontera para cada robot, entonces es necesario contar con una función de utilidad que permita evaluar cuál es la mejor localización para cada robot. Se asume también que la asignación de fronteras es realizada en varias etapas, considerando de manera sucesiva el caso de cada robot. Por lo tanto se plantea la exploración como un problema en donde se tiene un sistema dinámico que pasa de un estado a otro (los robots se van desplazando a las fronteras) y en el cuál se deben ir tomando decisiones (cuál robot va a qué frontera) para llegar a la mejor configuración final con la mayor utilidad posible.

En la figura 4.13, se muestra un ejemplo de los estados por los que fue pasando el sistema cuando se tiene el caso de dos robot y tres fronteras. En la figura 4.13 (a) se tiene el sistema en el estado inicial, es decir los dos robots en sus posiciones iniciales. En la figura 4.13 (b) después de tomar una primera decisión, el robot 1 se encuentra en la frontera C. Finalmente, después de una segunda decisión, en la figura 4.13 (c) se muestra el robot 2 en la frontera A. Para este ejemplo el resultado de aplicar el algoritmo de exploración consistió en asignar la frontera A al robot 2 y la frontera C al robot 1.

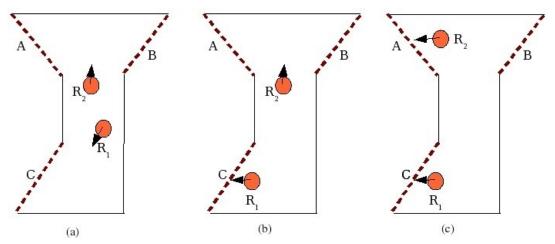


Figura 4.13. Secuencia de decisiones para asignar los robots a sus respectivas fronteras.

## 4.5.2 SISTEMA DINÁMICO CON ESTADO DE INFORMACIÓN IMPERFECTA Y PROGRAMACIÓN DINÁMICA

Como se mencionó en la sección anterior, se puede considerar el problema de exploración con múltiples robots con diferentes capacidades como un sistema dinámico que pasa de un estado a otro en un paso en el tiempo. Los estados estarán definidos por las configuraciones de los robots en el ambiente y para que se pueda pasar de un estado a otro se tienen que aplicar controles a los robots. Sin embargo al considerar que el modelo de sensado y el modelo de control están dados por una probabilidad, se está tomando en cuenta que existe incertidumbre en el resultado de las acciones y en la información que se está sensando; por lo tanto la información del sistema es imperfecta. También se está asumiendo que las decisiones de "acomodar" a los robots en sus respectivas fronteras se están tomando en varias etapas (una por cada robot), de manera que se puede utilizar la técnica de *Programación Dinámica* para resolver el problema.

La técnica de Programación Dinámica considera que las decisiones que se deban ir tomando en el sistema no se pueden ver de manera aislada ya que debe haber un balance entre la utilidad presente y la utilidad futura de una decisión [Bertsekas00]. En cada etapa las decisiones son evaluadas con base en la suma de la utilidad presente y la utilidad esperada en el futuro, asumiendo la toma de decisiones óptima en las etapas subsecuentes.

Para poder aplicar la técnica de programación dinámica, se considera un sistema dinámico discreto definido por la siguiente ecuación:

$$x_{k+1} = f(x_k, u_k) \tag{4.8}$$

donde  $x_k$  representa el estado actual del sistema,  $u_k$  es el control o acción tomada para hacer

que el sistema pase al estado  $x_{k+1}$  en el tiempo k+1, y la función f, es la define esos cambios de estado. Dicha función puede ser una distribución de probabilidad que indique el valor de probabilidad de pasar de un estado a otro cuando se aplica determinada acción. La función de utilidad debe estar definida de forma aditiva, y la utilidad total del sistema después de N pasos en el tiempo está definida por:

$$J_{\pi} = g(x_N) + \sum_{k=0}^{N-1} g(x_k, u_k)$$
(4.9)

donde  $g(x_k, u_k)$  es la utilidad de aplicar el control o acción  $u_k$  cuando el sistema se encuentra en el estado  $x_k$  y  $g(x_N)$  es la utilidad que se tiene por permanecer en el estado final, aunque en algunos casos esta utilidad es nula. Por lo tanto el algoritmo de programación dinámica debe maximizar la ecuación 4.9 para una secuencia de acciones  $\pi = \{u_0, u_1, ..., u_{N-1}\}$  que se deberán ir tomando. A la secuencia de acciones  $\pi^*$  que maximiza 4.9, se le conoce como *política óptima*.

Para poder calcular la política óptima, se utiliza el siguiente algoritmo, que va calculando la utilidades empezando por el paso final; a la ecuación 4.10, se le conoce como la ecuación de programación dinámica:

- Calcular  $J_N(x_N) = g(x_N)$
- Calcular para k=N-1,....,1,0

o La acción óptima para  $x_k$  es aquella  $u_k$  que hizo a la ecuación 4.10 máxima.

En el caso de que se tenga un sistema con estado de información imperfecta, en vez de conocer directamente los estados del sistema  $x_k$ , se tiene acceso a observaciones  $z_k$  que son adquiridas dependiendo del estado. Esto se encuentra definido en general por la siguiente

ecuación:

$$z_{k} = h(x_{k}, u_{k-1}) \tag{4.11}$$

A esta función h, se le conoce generalmente como el modelo de sensado y es cualquier función que indique qué lectura u observación se espera obtener cuando el sistema se encuentra en determinado estado, después de aplicar cierto control. La función h, al igual que la función f, puede ser una distribución de probabilidad que indique la probabilidad de obtener una observación dado el estado y la última acción aplicada.

También debido a que se desconoce con exactitud el estado del sistema, lo único que se sabe con seguridad son las lecturas que se toman del sistema y las acciones que se van aplicando, de manera que en vez de utilizar el estado  $x_k$  se define  $I_k$  como el *vector de información*. Este vector de información va "creciendo" en dimensión conforme pasa el tiempo. Es decir para el tiempo k=0, la única información que se tiene es la primera lectura; por consiguiente el vector de información está definido por  $I_0=\{z_0\}$ . En el tiempo k=1, ya se habrá aplicado una acción y se ha adquirido una nueva lectura por lo que el vector de información es  $I_1=\{z_0,u_0,z_1\}$ . De manera general el vector de información está dado por la siguiente ecuación:

$$I_{k} = \{I_{k-1}, Z_{k}, u_{k-1}\}$$
(4.11)

Debido a que generalmente la función de utilidad se encuentra definida en términos de  $x_k$  y  $u_k$ , al tener ahora un vector de información, entonces la función de utilidad se debe definir como el valor esperado para todos los posibles valores de  $x_k$  dado el vector de información, como se muestra en la ecuación 4.12.

$$g(I_k, u_k) = E\{g(x_k, u_K) | I_k\} = \sum_{x_k} p(x_k | I_k) g(x_k, u_K)$$
(4.12)

Nótese que en la ecuación 4.12, aparece el término  $p(x_k|I_k)$  para indicar la probabilidad de que el sistema se encuentre en determinado estado, dado el vector de información conocido. El cuál se calcula con la ecuación 4.13. La demostración de cómo se obtiene dicha ecuación se encuentra en el apéndice A de este escrito.

$$p(x_{k+1}|I_{k+1}) = \frac{\sum_{i=1}^{n} p(x_k = i|I_k) p(x_{k+1}|x_k = i, u_k) p(z_{k+1}|x_{k+1})}{\sum_{s=1}^{n} \sum_{i=1}^{n} p(x_k = i|I_k) p(x_{k+1} = s|x_k = i, u_k) p(z_{k+1}|x_{k+1} = s)}$$

$$(4.13)$$

Finalmente teniendo definido el vector de información y las ecuaciones 4.12 y 4.13, el algoritmo y la ecuación de programación dinámica para el caso de información imperfecta se definen como sigue:

- Calcular  $J_N(I_N) = g(I_N)$
- Calcular para k=N-1,....,1,0

o La acción óptima para  $I_k$  es aquella  $u_k$  que hizo a la ecuación 4.14 máxima.

## 4.5.3 PLANTEAMIENTO UTILIZANDO PROGRAMACIÓN DINÁMICA

Para ilustrar la idea de cómo utilizar la técnica programación dinámica para robots con distintas capacidades, en esta sección se recurre a un ejemplo sencillo para mostrar el planteamiento. El ejemplo es para dos robots con dos fronteras por explorar. Sin embargo se puede generalizar a más robots y más fronteras y en general no es necesario que el número de robots sea igual al número de fronteras.

Se tienen dos robots denotados como  $R_1$  y  $R_2$ , y se tienen dos fronteras por explorar denotadas como A y B. El problema es determinar qué robot debe ir a qué frontera. Se asume nuevamente que un sólo robot se mueve a la vez y que cada robot visitará solamente una frontera, como ya se había mencionado anteriormente. Para este problema en particular, cada robot puede encontrarse en cualquiera de 4 estados posibles. El estado de  $R_1$  en el tiempo k se denota como  $rI_k$  y de manera análoga  $r2_k$  denota el estado para  $R_2$ . Los valores que puede tomar cada estado son:

- La posición inicial de cada robot que se denota como *i*.
- La posición correspondiente a la frontera *A* que se denota como *a*.
- La posición correspondiente a la frontera *B* que se denota como *b*.
- Cualquier otra posición que no sea estar en i, a, b, y que será denotada como o.

En cuanto a las acciones o controles que se pueden aplicar a cada robot se denotan como  $a1_k$  y  $a2_k$  respectivamente y pueden tomar alguno de los siguientes valores:

- El robot no se mueve, es decir la acción es permanecer en el mismo lugar y se denota como s.
- El robot se dirige a la frontera A, y se denota como ga.
- El robot se dirige a la frontera B, y se denota como gb.

Finalmente también se sabe que cada robot puede tomar lecturas del ambiente, la lectura que adquiere el robot  $R_1$ , se denota como  $sI_k$ ; lo mismo para  $R_2$  con  $s2_k$ . Para este ejemplo, se asume que cada robot puede obtener alguna de las siguientes lecturas:

- Sensar su posición inicial, denotada con si.
- Sensar que el robot se encuentra en la frontera A, denotada con sa.
- Sensar que el robot se encuentra en la frontera B, denotada con sb.
- Sensar que robot se encuentra en cualquier otro lugar diferentes a las fronteras y a su posición inicial, denotada con *so*.

Cada robot tiene diferentes capacidades de sensado y de control de movimiento que se representan por medio de distribuciones discretas de probabilidad. Para expresar las diferentes capacidades de control de movimiento, se asume que cada robot puede alcanzar con cierta probabilidad una posición después de que se ejecutó un control. Esta probabilidad se denota como  $p(r1_{k+1}|r1_k,a1_k)$  es decir la probabilidad de que el robot  $R_I$  llegue al estado  $r1_{k+1}$  en el tiempo k+1 después de ejecutar el control  $a1_k$  a partir de la posición  $r1_k$  en el tiempo k. Y de manera similar para  $R_2$  se tiene  $p(r2_{k+1}|r2_k,a2_k)$ .

Para expresar las distintas capacidades de sensado se tiene en cuenta dos aspectos por separado; el primero cuando el robot sensa su posición generalmente por medio de la odometría o utilizando algún algoritmo de localización y el segundo cuando el robot sensa el ambiente para descubrir nueva información para el mapa. Con una distribución discreta de probabilidad se pueden modelar las diferentes capacidades de sensado con respecto al caso de la posición del robot. La probabilidad se define como  $P(s1_k|r1_k)$ , es decir la probabilidad de que el robot  $R_I$  sense  $s1_k$  dado que se encuentra en el estado  $r1_k$ . Y de manera similar para  $R_2$  se tiene  $p(s2_k|r2_k)$ .

En cuando a la capacidad de sensado de cada robot para descubrir nueva información del ambiente, se representa por una función de utilidad. En este caso suponemos que dicha función relaciona el área nueva que descubre el robot con la distancia que viajó para llegar a la nueva posición de sensado. La función de utilidad se expresa como el área nueva descubierta dividida por la distancia que el robot viajó; de manera que si el área nueva a descubrir es muy pequeña y el camino muy largo, entonces es de poca utilidad que el robot acuda a determinada posición. Esto se expresa como  $c(r1_k,a1_k)=area/distancia$  para  $R_1$  y como  $c(r2_k,a2_k)$  para  $R_2$ . El área nueva que descubre el robot tiene que ver con el rango y ángulo de sensado de cada robot; es decir entre mayor sea el rango y el ángulo del sensor utilizado por el robot, mayor será el área

que se puede descubrir en una lectura.

Una vez que se tienen los datos para cada uno de los robots, se modela el problema de exploración como un sistema dinámico con estado de informacín imperfecta. Para esto se necesitan los siguientes datos:

- El conjunto X de los posibles estados, donde el estado en el tiempo k, es denotado como  $x_k$ .

  Para el ejemplo definido, cada posible estado es un vector de 2 variables, cada una indicando la posición de un de los robot, denotado como  $x_k = (r1_k \cdot r2_k)^T$ .
- El conjunto de posibles controles U. El control para cada tiempo k es denotado como  $u_k$ . De la misma manera que en el caso de los estados, cada posible control está definido por un vector que incluye una acción o control para cada robot. Debido a que solamente se mueve un robot a la vez, en el vector de los controles para cada robot solamente uno de los robots tendrá un control para dirigirse a una frontera y el resto de los robots tendrá asignado el control s ("stop"). Para el ejemplo que se está desarrollando, el vector de control es denotado como  $u_k = (a1_k, a2_k)^T$  debido a que son dos robots. El conjunto de posibles controles es  $U = \{(ga, s), (gb, s), (s, ga), (s, gb)\}$ , recordando que sólo un robot se mueve a la vez .
- El conjunto de posibles lecturas Z. La lectura o sensado de cada tiempo k es denotada como  $z_k$ . Para el ejemplo dado, el vector de lo sensado indica las lecturas de cada robot, y se denota como  $z_k = (s1_k, s2_k)^T$ .
- La función de utilidad de llegar a cierto estado después de aplicar cierto control se denotada como  $g(x_k, u_k)$ . Dicha función, en términos de la utilidad definida para cada robot se puede expresar como sigue:

$$g(x_k, u_k) = \begin{cases} c(r2_k, a2_k) & \text{si } a1_k = s \\ c(r1_k, a1_k) & \text{si } a2_k = s \end{cases}$$

$$(4.15)$$

• El modelo del sistema que indica la probabilidad de pasar de un estado a otro despúes de aplicar un control, denotada como  $p(x_{k+1}|x_k,u_k)$ . Esta probabilidad se debe expresar en términos de las probabilidades de cada robot, ya que tanto la variable de estado como la variable de que indica el control son vectores con los datos de cada robot. Para el caso de los dos robots se tiene lo siguiente:

$$p(x_{k+1}|x_k,u_k) = p(r1_{k+1},r2_{k+1}|r1_k,r2_k,a1_k,a2_k) = \frac{p(r1_{k+1},r2_{k+1},r1_k,r2_k,a1_k,a2_k)}{p(r1_k,r2_k,a1_k,a2_k)}$$
(4.16)

Desarrollando la ecuación 4.16 se tiene lo siguiente:

$$\frac{p(a1_k)p(r1_{k+1}|a1_k,r1_k)p(a2_k|r1_{k+1},a1_k,r1_k)p(r2_k|a2_k,r1_{k+1},a1_k,r1_k)p(r2_k|a2_k,r1_{k+1},a1_k,r1_k)p(r2_{k+1}|r2_k,a2_k,r1_{k+1},a1_k,r1_k)}{p(a1_k)p(r1_k|a1_k)p(a2_k|r1_k,a1_k)p(r2_k|a2_k,r1_k,a1_k)}$$
(4.17)

Si se asume que las probabilidades de cada robot son independientes entre sí, es decir que las variables del robot  $R_1$  no dependen de las variables del robot  $R_2$  y visceversa la expresión anterior se reduce a lo siguiente:

$$p(x_{k+1}|x_k,u_k) = \frac{p(a1_k)p(r1_k|a1_k)p(r1_{k+1}|a1_k,r1_k)p(a2_k)p(r2_k|a2_k)p(r2_{k+1}|r2_k,a2_k)}{p(a1_k)p(r1_k|a1_k)p(a2_k)p(r2_k|a2_k)}$$

$$(4.18)$$

Es decir que el modelo del sistema, en el caso de contar con dos robots se define como se muestra con la ecuación 4.19.

$$p(x_{k+1}|x_k,u_k) = p(r1_{k+1}|r1_k,a1_k) p(r2_{k+1}|r2_k,a2_k)$$
(4.19)

• El modelo de sensado. La probabilidad tomar cierta lectura dado que se encuentra el sistema en cierto estado, denotada como  $p(z_k|x_k)$ . Utilizando el mismo razonamiento que en el modelo del sistema. El modelo de sensado expresado en términos de las

probabilidades de cada robot, se puede definir como sigue:

$$p(z_k|x_k) = p(s1_k|r1_k) p(s2_k|r2_k)$$
(4.20)

Dado que en este caso se tienen dos fronteras y dos robots, el problema es encontrar la política para 2 periodos de tiempo. En otras palabras lo que se quiere saber qué acción se debe tomar después de estar en la posición inicial, esa acción será  $u_0$ . En el siguiente paso, después de tomar una lectura  $z_1$ , se debe determinar cuál es la segunda acción  $u_1$  para el sistema. La primera acción es para mover al uno de los robots, y la segunda acción para mover al otro robot. Para encontrar las acciones óptima para este ejemplo se debe utilizar el último algoritmo presentado en la sección 4.5.2 utilizando la ecuación de programación dinámica 4.14.

Aunque en esta sección se definieron las ecuaciones para el caso de dos robots, éstas se pueden extender para un número mayor de robots. Si se tienen n robots en el sistema, los vectores que definen el estado del sistema, las posibles acciones y los posibles sensados, serán de dimensión n. Si se tienen m fronteras esto generará m+2 posibles estados para cada robot (un estado por cada frontera, más el estado inicial, más el estado que indica que está en cualquier otro lado), m+1 acciones posibles (una para ir a cada frontera y una para permanecer en un lugar) para cada robot y m+2 posibles lecturas (una por cada posible estado). Por lo tanto se puede observar que conforme aumente el número de robots y el número de fronteras se necesitarán gran cantidad de operaciones para encontrar la estrategia de exploración utilizando esta técnica.

## CAPÍTULO 5. IMPLEMENTACIÓN Y RESULTADOS

En este capítulo se muestran los resultados para los estrategias de exploración propuestas en este trabajo y se describen los programas implementados. Los resultados presentados son tanto a nivel simulación, como con robots reales en algunos casos.

## **5.1 ROBOTS UTILIZADOS**

Para las pruebas experimentales, se utilizaron dos robot móviles de ActivMedia Robotics: el Pioneer 3-AT que es un robot de exterior y el Pioneer 3-DX que es un robot de interior, sin embargo ambos robots fueron utilizados en ambientes de interior. Los robots se mostraron en el capítulo uno, en la figura 1.1, cada uno posee en sistema de visión y en LRF, sin embargo en los experimentos realizados para la construcción de mapas, solamente se utilizó el sensor de rango.

Cada robot cuenta con una computadora a bordo con un procesador Pentium III a 850 Mhz con sistema operativo Linux Red-Hat 7.3. Los LRF, con que cuenta cada robot, son láser SICK que tienen un ángulo de captura 180°, con un rango de lectura que va de 10 cm a 10 m. Cada sensor láser entrega en cada lectura entre 120 y 160 puntos aproximadamente en un plano paralelo al piso.

## 5.2 CONSTRUCCIÓN DE MAPAS GEOMÉTRICOS USANDO MDPs

En esta sección se describen los resultados simulados y experimentales de la primera estrategia de movimiento que fue descrita en el capítulo 3. Se describe cómo se implementó el sistema y algunos experimentos que se realizaron tanto con robots reales como con un simulador.

## 5.2.1 DESCRIPCIÓN DEL SISTEMA IMPLEMENTADO

El sistema computacional para implementar la estrategia de exploración utilizando MDPs, fue realizado utilizando C/C++, las librerías de OpenGL para el despliegue gráfico en 3D y las librería de Aria [ActivMedia04] para mover a los robots y obtener la información del láser. El sistema completo se encuentra corriendo en las computadoras de cada uno de los robots y en una laptop o máquina de escritorio que se encarga de reunir el mapa global.

Cada robot, utilizando la computadora que tiene a bordo, efectúa la lectura de los puntos del LRF, calcula los segmentos, calcula las celdas libres, alínea los segmentos obtenidos con el mapa local que cada robot posee, calcula las fronteras y elige en cuál frontera continuar la exploración y finalmente calcula la trayectoria para llegar a su nueva posición utilizando el algoritmo de iteración por valor.

Por otra parte, la laptop funciona como un servidor central que se encarga de recibir las listas de puntos que envía cada robot, los segmentos y las celdas libres. Con esta información que recibe, se encarga de alinear los mapas parciales enviados por cada robot y desplegar la representación del ambiente que se va construyendo poco a poco. Si bien la representación del ambiente seleccionada es un mapa geométrico formado por segmentos de línea, en el despliegue

del sistema computacional se optó por mostrar los segmentos como pequeñas "paredes" con una altura predefinida en un ambiente tridimensional.

### **5.2.2 EXPERIMENTOS SIMULADOS Y REALES**

En esta sección se muestran dos de los experimentos que se realizaron con la estrategia de exploración en la cuál se utiliza un MDP. El primer experimento fue probado con los dos robots reales en una sección de uno de los edificios del campus. El segundo experimento fue realizado utilizando el simulador que provee ActivMedia.

El primer experimento, se llevó a cabo en una área de oficinas de aproximadamente 5 x 8 metros, dos vistas del ambiente se muestra en la figura 5.1, se puede notar un pasillo, un escritorio y una sala donde se encuentran cajas y archiveros. Para este experimento la posición relativa entre los robots era x = 50cm, y = 120cm y un ángulo  $\theta=180^{\circ}$ , lo cuál significa que cada robot se encuentraba apuntado en la dirección contraria del otro robot, tal y como se muestra en la figura 5.2. Esto con la finalidad de que cada robot se mueva alejándose del otro para no interferir en los movimientos del otro tal y cómo se explicó en el capítulo 3.

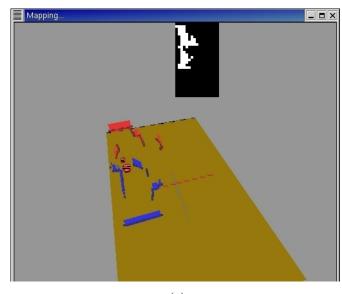


Figura 5.1. Vistas del ambiente de interior.



Figura 5.2. Configuración inicial de los robots antes de comenzar la construcción del mapa.

En la figura 5.3, se muestra una secuencia de imágenes que fueron capturas de la pantalla del servidor central que va desplegando el mapa que se va construyendo. En cada imagen se muestra en color rojo (en gris claro) el mapa que va obteniendo el robot Pioneer 3-DX y en color azul (en gris oscuro) el mapa que va obteniendo el robot Pioneer 3-AT. En la parte superior de la pantalla se encuentra un pequeño rectángulo negro, dentro del cuál se aprecia un área blanca; este recuadro representa la malla de ocupación que se tiene entre los dos robots. Las celdas blancas, representan celdas libres por las cuáles se puede mover un robot y las áreas negras representan ya sea celdas ocupadas o bien celdas desconocidas, para este experimento y el siguiente, se definieron celdas rectangulares de 40 cm de lado.



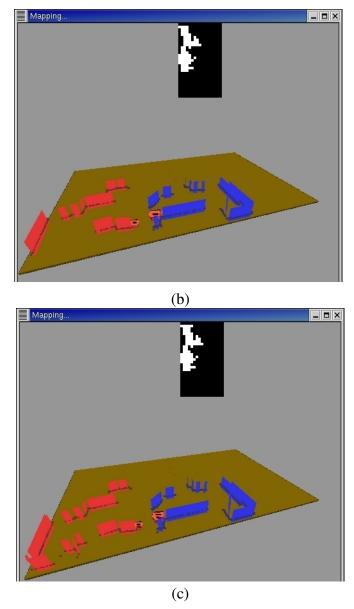


Figura 5.3. Secuencia de imágenes durante la construcción de un mapa.

La secuencia de imágenes anterior, se muestra una parte del proceso de construcción del mapa, en la figura 5.3(a), se tiene el mapa global después de que cada uno de los robots realizó una lectura con el LRF, calcularon los segmentos y enviaron la información al servidor central. En la figura 5.3(b), se muestra el mapa después de que el Pioneer 3-DX realizó dos lectura y el Pioneer 3-AT solamente una; y finalmente en la figura 5.3(c) se muestra el mapa obtenido después de dos lecturas realizadas por cada robot. Debido a que el espacio de pruebas no era demasiado amplio, y a que el alcance del LRF es de 10 m, se puede ver que con pocas lecturas se

obtiene la mayor parte del mapa. En la figura 5.4, se muestran dos imágenes donde cada uno de los robots se encuentra explorando una parte del ambiente para obtener su representación.



Figura 5.4. Los robots durante la exploración. (a) Pioneer 3-AT (b) Pioneer 3-DX.

La segunda prueba fue realizada en un ambiente simulado con dos robots simulados. El ambiente era un espacio de  $18 \times 10$  metros y la posición relativa entre de los robots era x=200cm, y=700cm y un ángulo  $\theta=90^\circ$ . En la figura 5.5 se muestra una secuencia de imágenes que fueron capturadas en la pantalla del servidor central mientras los robots iban contruyendo el mapa. Nuevamente se usan dos colores (dos tonos de gris), para indicar qué parte del mapa fue obtenida por cada uno de los robots. En la figura 5.5(a) se muestra un pequeño mapa obtenido después de que el robot 1 realizara una lectura, en la 5.5(b) se muestra la representación correspondiente a una lectura de cada uno de los robots; como se puede observar hasta esta imagen, los mapas de cada uno de los robots aún no traslapan, esto debido a que fueron colocados muy alejados uno del otro. En la figura 5.5(c) dos lecturas del robot 1 y una lectura del robot 1 y así sucesivamente se muestra el incremento de una lectura para cada una de las figura 5.5(d), 5.5(e) y 5.5(f). Finalmente en la figura 5.5(g), se muestra el mapa casi terminado cuando el robot 1 ha realizado cinco lecturas y el robot 2 ha realizado cuatro lecturas.

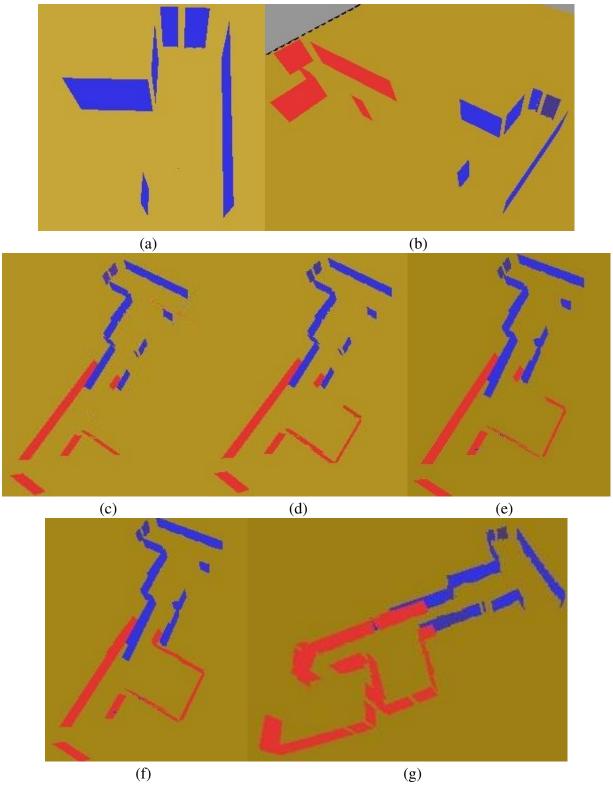


Figura 5.5. Secuencia de imágenes del proceso de construcción de un mapa utilizando el simulador.

## 5.3 EXPLORACIÓN UTILIZANDO LA FUNCIÓN DE UTILIDAD

En esta sección se presentan los resultados obtenidos para el caso de la estrategia de exploración basada en generación de muestras que son evaluadas con la función de utilidad presentada en el capítulo 4. Los resultados se encuentran organizados de la siguiente manera: primero la parte de simulaciones con un sólo robot y después el caso de multirobots; se comparan las estrategias en los casos de un sólo paso y varios pasos adelante en el tiempo y finalmente se muestras resultados experimentales con un robot real.

El sistema computacional que utilizó para probar esta estrategia de exploración fue originalmente desarrollado en [Tovar02], y durante este trabajo se retomó haciendo las modificaciones necesarias. El sistema se encuentra divido en varios módulos que funcionan de manera conjunta comunicándose entre sí por medio de socktes de TCP/IP, de manera que cada uno se puede considerar como un servidor para una tarea en particular. Los módulos son los siguientes: lector de puntos del láser, ajuste de segmentos, alineamiento de mapas con la distancia de Hausdorff, planificador de movimientos y control de movimientos de los robots. El sistema fue implementado utilizando C/C++ y las librerías de Aria para la comunicación con los sensores y los motores del robot y LEDA para cálculos geométricos y parte del despliegue gráfico; otra parte del despliegue gráfico fue realizado utilizando OpenGL. Para el caso de las simulaciones el sistema fue desarrollado en C/C++, utilizando solamente las librerías de LEDA.

### 5.3.1 SIMULACIONES CON UN SOLO ROBOT

En las figuras que se muestran a continuación, se muestran los landmarks como círculos oscuros dentro del ambiente y el robot que se encuentra explorando, se muestra como un círculo sobre el

color de fondo del ambiente. Conforme el robot se va moviendo durante la exploración se va guardando un mapa de las trayectorias que ejecuta el robot; éstas se muestran también en las figuras como los segmentos de línea dentro del ambiente.

En la figura 5.6, se muestra dos veces el mismo mapa final obtenido por un sólo robot; en la figura 5.6(a) se tiene el mapa generado por un robot que tiene un sensor de 360°, mientras que en 5.6(b) el mismo mapa es obtenido por un robot cuyo ángulo de sensado es de 180°. Aunque el mapa final es el mismo, se puede observar que el robot con visión omnidireccional realiza una menor cantidad de movimientos para obtener el mapa completo que el robot que tiene un ángulo de 180°; de manera que la estrategia de movimiento generada para ambos robots fue diferente debido a que la función de utilidad considera el número de esquinas que puede "ver" cada robot así como el número de landmarks.

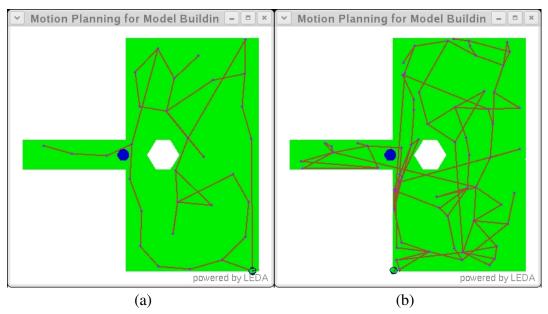


Figura 5.6. Mapa final obtenido por un sólo robot: (a) con sensor de 360°, (b) con sensor de 180°.

A continuación se presenta la generación de un mapa con un mismo robot pero considerando los casos de un sólo paso adelante en el tiempo y de varios pasos adelante en el tiempo. En la figura 5.7, se presenta una secuencia de imágenes que fueron capturadas durante la

exploración cuando el robot está planificando sus movimientos en un sólo paso adelante en el tiempo; y en la figura 5.8 se muestra otra secuencia para el mismo mapa y con el mismo robot solamente que en este caso se consideran varios pasos adelente en el tiempo. En la figura 5.8(a), se pueden ver las tres fronteras que el robot descubrió, en la imagen se encuentra visitando la tercera frontera, esto quiere decir que el número de pasos adelante en el tiempo para los cuáles se hace la planificación depende del número de fronteras. Es decir si hay 4 fronteras la planificación se realiza para 4 pasos adelante en el tiempo. Estos dos ejemplos serán retomados en la sección 5.3.3 para comparar los resultados obtenidos por ambos casos.

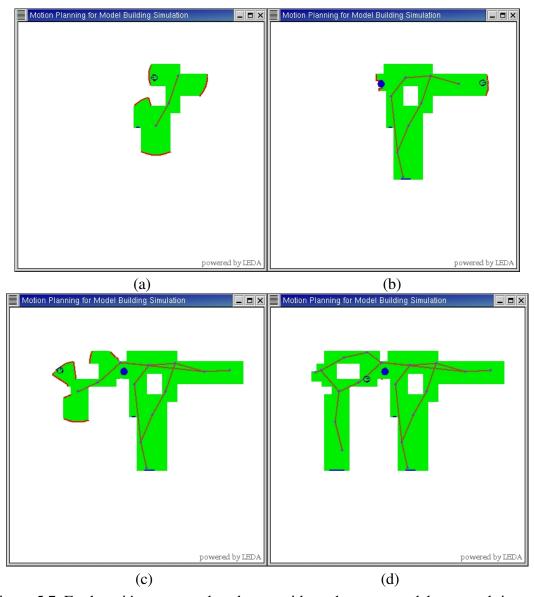


Figura 5.7. Exploración con un solo robot considerando un paso adelante en el tiempo.

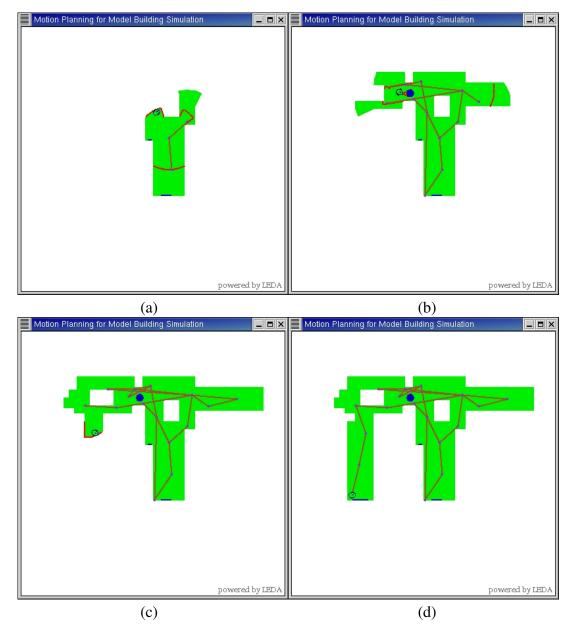


Figura 5.8. Exploración con un solo robot considerando varios pasos adelante en el tiempo.

### 5.3.2 SIMULACION CON MÚLTIPLES ROBOTS

Como primer ejemplo, se tiene como resultado el mapa que se muestra en la figura 5.9. Este mapa fue construido por 4 robots todos con las misma características y en este caso se consideró un solo paso adelante en el tiempo durante la estrategia de exploración. En la figura el mapa aparece

iluminado con distintos colores que indican la región que cada robot descubrió durante la exploración.

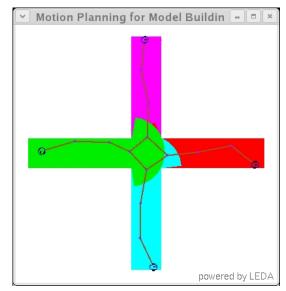
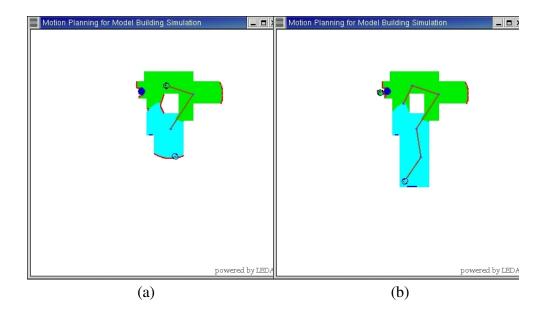


Figura 5.9. Mapa obtenido por 4 robots.

Nuevamente se presenta el mismo mapa de las figura 5.7 y 5.8 solamente que en este caso la representación es construida por dos robots. En las figuras 5.10 y 5.11 se muestran dos secuencias que corresponden al caso de un solo paso y varios en el tiempo respectivamente. A simple vista se puede notar que en el caso de varios pasos adelante en el tiempo, los robots recorren trayectorias más largas para construir el mapa.



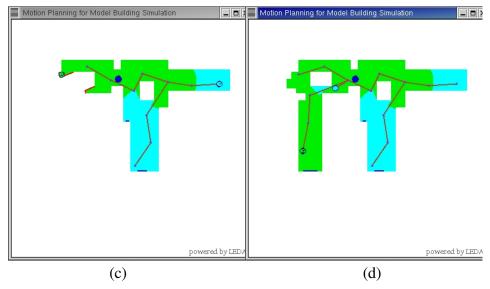


Figura 5.10. Exploración con dos robots considerando un paso adelante en el tiempo.

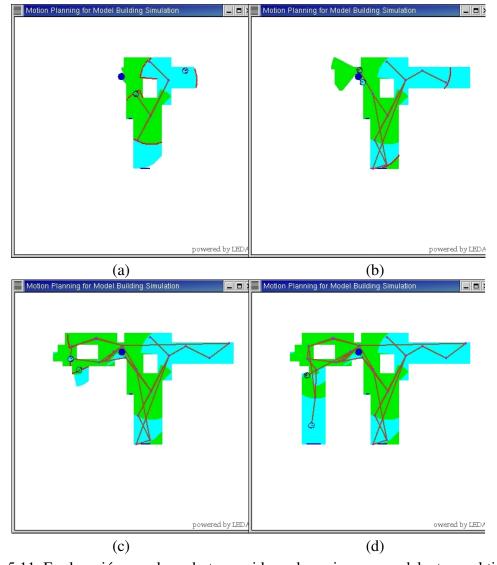


Figura 5.11. Exploración con dos robots considerando varios pasos adelante en el tiempo.

#### 5.3.3 COMPARACIÓN ENTRE UN PASO Y VARIOS PASOS EN EL TIEMPO

Cómo ya se mencionó en la sección anterior, al observar las trayectorias de los robot son más simples en los casos de un sólo paso adelante en el tiempo. Por lo tanto se definieron varios parámetros para comparar el comportamiento de la exploración en ambos casos. Los parámetros utilizados fueron: la longitud de la ruta recorrida por cada robot, el total en grados que gira durante la trayectoria, el número de veces que se detiene a tomar lecturas y el número de veces que se detiene el robot para cambiar de dirección o para sensar.

Parámetros	Un robot / Un paso	Un robot / Varios pasos
Longitud de la trayectoria	517.0367	637.2925
Número de veces que se detiene	51	58
Número de veces que sensa	20	18
Ángulo total que gira	1406.735	879.578
Parámetros	Dos robots / Un paso	Dos robots / Varios pasos
Longitud de la trayectoria	260.71756	461.5974
Número de veces que se detiene	27	57
Número de veces que sensa	15	20
Ángulo total que gira	1323.924	1903.18

Tabla 5.1. Comparación: Un paso adelante vs. Varios pasos adelante.

En la tabla 5.1 se muestran los resultados obtenidos en términos de los parámetros seleccionados. Estos resultados fueron los obtenidos para las secuencias de imágenes de las figuras 5.7, 5.8, 5.10 y 5.11. En general se puede observar que los resultados son mejores con la estrategia de exploración que considera un sólo paso adelante en el tiempo, es decir la longitud de la trayectoria es menor, el número total de grados girados y el número de veces que se detiene el robot también son menores. Debido a que la estrategia que considera varios pasos adelante en el

tiempo considera visitar en cierto orden más de una frontera a la vez y esto se repite hasta que ya no se tengan fronteras nuevas, entonces la estrategia de movimientos genera gran cantidad de movimientos repetitivos o no necesarios.

Tan pronto como una nueva frontera aparece durante la exploración (dicha frontera no se ha considerado en el plan original), el robot deberá regresar a algunas posiciones previamente visitadas. Para aclarar este punto se muestra el ejemplo de la figura 5.12. En la parte 5.12(a) se muestra la trayectoria que siguió un sólo robot con la estrategia de un paso adelante en el tiempo y en 5.12(b) y 5.12(c) se muestran dos imágenes durante el proceso de exploración considerando varios pasos adelante en el tiempo. En la figura 5.12(a), el robot avanzó por un pasillo a la vez hasta terminar de conocerlo y luego regresa a explorar otro pasillo; en cambio en las figuras 5.12(b) y 5.12(c) se puede observar que cada vez que define un plan el robot visita las cuatro fronteras que conoce, lo cuál se traduce en que el robot visite los cuatro pasillos varias veces durante la exploración dando como resultado una trayectoria mucho más larga al ir y regresar varias veces por todos los pasillos. La interpretación de estos resultados es que al tratar de generar un plan con varios pasos con información parcial (porque el mapa aún no está completo) y dinámica el resultado es una pérdida de recursos utilizados.

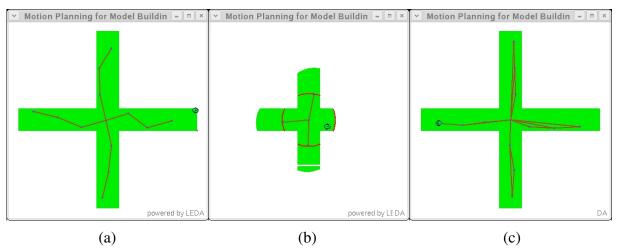


Figura 5.12. Comparación entre uno y varios pasos. Mapa y trayectoria para un sólo robot.

#### 5.3.4 EXPERIMENTOS CON ROBOTS REALES

Se realizaron algunos experimentos con los robots reales, para la estrategia de exploración utilizando la función de utilidad, las pruebas se llevaron a cabo con un sólo robot que utiliza su sensor LRF para construir el mapa. En la figura 5.13, se presentan dos imágenes en las cuáles se puede apreciar el ambiente para el experimento. En las fotografías se pueden apreciar los dos robots dentro del ambiente, sin embargo solamente el Pioneer 3-DX (el robot que cambia de posición en las imágenes) es el robot que se encuentra construyendo el mapa. En la figura 5.13(a) el Pioneer 3-DX aparece al fondo parcialmente ocluído por un obstáculo y en la figura 5.13(b) se ha desplazado hacia el otro robot para continuar la exploración.

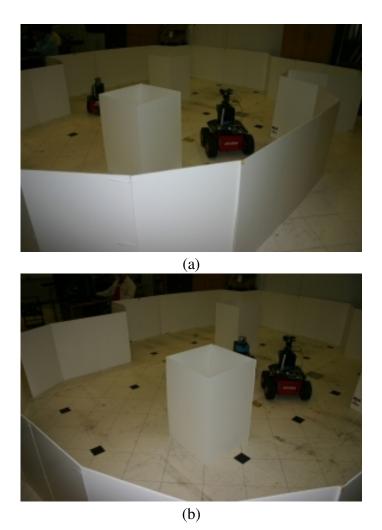


Figura 5.13. Ambiente de pruebas para experimentos con robots reales.

Mientras el robot se encuentra explorando el ambiente, en el ambiente gráfico del sistema computacional se van mostrando el mapa parcial que se está construyendo. En la figura 5.14 se presentan dos mapas parciales obtenidos en diferentes etapas de la exploración. En ambas imágenes se muestra la última posición del robot y las fronteras por explorar aparecen marcadas con líneas rojas. En una de las fronteras en cada una de las imágenes aparecen segmentos de línea sobre los cuáles aparecen puntos; dichos puntos representan las posibles configuraciones siguientes para que el robot continúe la exploración.

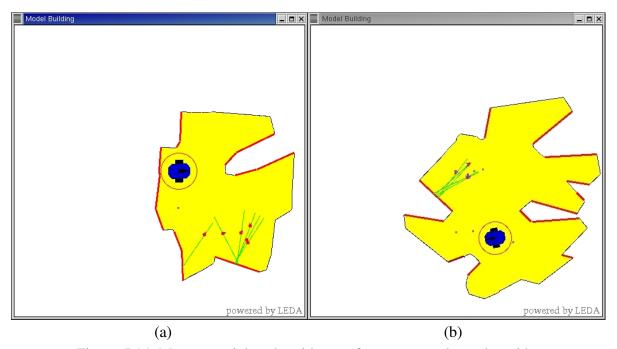


Figura 5.14. Mapas parciales obtenidos conforme avanza la exploración.

Finalmente en la figura 5.15 se observa una secuencia de imágenes que presentan el mapa que se va obteniendo con la información de puntos del LRF. En las imágenes se pueden apreciar puntos rojos y azules, los puntos azules indican el mapa anterior construido hasta determinado momento y los puntos rojos indican la última lectura que se acaban de incorporar al mapa. En la figura 5.15(c) se puede apreciar con mayor claridad el mapa global y la última lectura que son alineadas en la figura 5.15(d) utilizando el algoritmo descrito en el capítulo 2 que emplea la

distancia de Hausdorff.

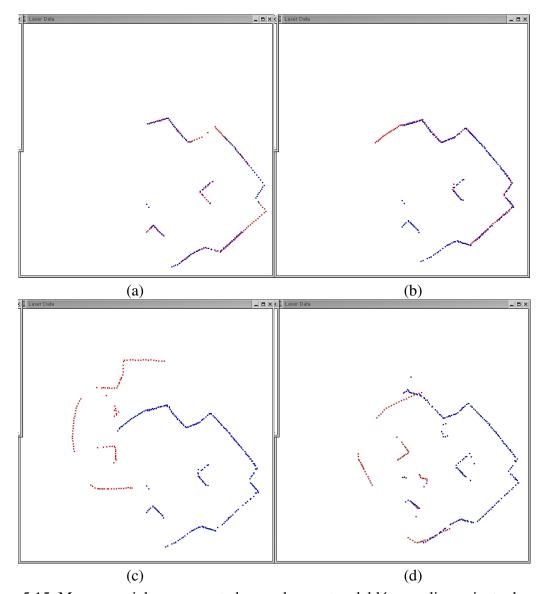


Figura 5.15. Mapas parciales representados por los puntos del láser y alineamiento de mapas.

### 5.4 EXPLORACIÓN CON ROBOTS CON DISTINTAS CARACTERÍSTICAS

En esta sección se describe la prueba que se realizó por medio de un programa computacional a cerca del algoritmo de exploración propuesto utilizando programación dinámica. Para este caso se asumió, tal y cómo se explicó en la sección 4.5.3, que existen dos robots con diferentes

características de sensado y motrices que deberán acudir a fronteras con distintos tamaños.

Cada robot tiene un cierto valor de probabilidad de llegar a la frontera indicada, se asume que los robots deberán llegar al punto medio de la frontera. Para el robot 1 se asumió que tiene una probabilidad de 0.75 de alcanzar la frontera deseada y para el robot 2 esa probabilidad es de 0.82. En cuanto a sus capacidades de sensado, se definió que el robot 1 tiene una probabilidad de 0.8 de reconocer correctamente la frontera en la que se encuentra y puede sensar como máximo un area de 25 unidades cuadradas; mientras que el robot 2 tiene una probabilidad de 0.9 de sensar correctamente la frontera en la que se encuentra y su area de visibilidad es de 34 unidades cuadradas. Dadas estas características se observa claramente que el robot 2 es más preciso en sus movimientos y tiene mejores sensores.

A continuación se muestra el algoritmo general que sigue la implementación de esta estrategia de exploración:

- Entrada: Los dos robots y probabilidades de cada uno, así como su area de visibilidad.
   Mapa parcial donde se pueden identificar los tamaños de las fronteras, la posición de los robots y la distancia a viajar por cada robot a cada frontera.
- Salida: Política óptima que indique que acción tomar para cada uno de los posibles vectores de información.

#### • Inicializar:

- Obtener el conjunto X con los vectores de posibles estados del sistema de dos robots.
- Obtener el conjunto U con todas las posibles acciones.
- Obtener el conjunto Z con todas las posibles lecturas que se pueden adquirir.
- Obtener todos los posibles vectores de información para el tiempo 0,  $I_0$ .
- Obtener todos los posibles vectores de información para el tiempo 1,  $I_1$ .
- O Calculas para todo estado en el tiempo  $0 x_0$  el valor de  $p(x_0|I_0)$ .
- O Calculas para todo estado en el tiempo 1  $x_1$  el valor de  $p(x_1|I_1)$ .

- O Para todos los robots R, para todas las acciones por robot aR y para todos los estados de cada robot rR:
  - $\blacksquare$  Calcular la utilidad c(rR, aR).
- o Para todo estado  $x \in X$  y para toda acción  $u \in U$ :
  - $\blacksquare$  Calcular la utilidad g(x, u) utilizando los valores de c(rR, aR).
- Algoritmo de Programación Dinámica:
  - $_{\circ}$   $J_{2}(I_{2})=0$
  - o Calcular para k=1,0:
    - Para todo vector de información  $I_k$ :

$$\qquad J_{k}(I_{k}) = \max_{u_{k} \in U} \{ (\sum_{x_{k}} p(x_{k}|I_{k})g(x_{k},u_{k})) + \underset{x_{k}}{E} \{ J_{k+1}(I_{k},z_{k+1},u_{k})|I_{k} \} \}$$

• La acción óptima para  $I_k$  es aquella  $u_k$  que dió el valor a  $J_k(I_k)$ .

Como se puede notar en el algoritmo anterior, la mayor parte de los cálculos son realizados en la parte de inicialización, donde se calculan las probabilidades, conjuntos y utilidades que serán necesarios para la parte del algoritmo de programación dinámica propiamente dicho. Debido a que se cuenta con dos robots el algoritmo de programación dinámica se ejecuta para los periodos 0 y 1, que corresponderán a dos movimientos del sistema, uno para cada robot. Se aclara también que se está considerando que no existe una valor de utilidad final al indicar que  $J_2(I_2)$  es cero.

Se realizaron varias corridas del programa con los dos robots con las características antes descritas, pero cambiando el número de fronteras en el mapa, así como la longitud de cada frontera. A partir de dichas corridas se observaron los siguientes hechos:

- Si se tiene un frontera a una distancia grande de los robots, el robot con mayor precisión en sus movimientos acudirá a visitar dicha frontera.
- Si se tiene una frontera de longitud pequeña, el robot con menor capacidad de sensado acudirá a explorar dicha frontera.

- Generalmente se mueve primero en el primer paso aquel robot que tiene mayores capacidades motrices y mejores sensores. Esto debido a que se tendrá mayor certeza de que el robot llegará a su destino.
- La idea general de las políticas obtenidas es enviar al robot mejor equipado a la frontera más lejana y más grande y al robot 1, con menores capacidades, a la frontera más cercana y donde existe una menor cantidad de información por descubrir.

# CAPÍTULO 6. CONCLUSIONES Y TRABAJO FUTURO

A partir de la definición del problema de SLAM, el objetivo principal al resolver dicho problema es integrar la información recolectada durante la exploración en un mapa lo más preciso posible. En este trabajo se presentaron varias estrategias de movimiento para la construcción de mapas en ambientes de interior; la primera estrategia utiliza un MDP para calcular los movimientos del robot durante la exploración, la segunda estrategia se basa en evaluar configuraciones generadas de manera aleatoria por medio de una función de utilidad y la tercera considera que existen robots con distintas capacidades que deben ser asignados a la frontera más conveniente utilizando programación dinámica.

Los mapas obtenidos después de la exploración, fueron representados por medio de segmentos de líneas junto con un mapa de caminos que define la trayectoria que siguieron los robots durante la exploración, así como en el caso de las simulaciones también se toman en cuenta los landmarks existentes. Y para poder ir acoplando las sucesivas lecturas en el mapa que se va obteniendo, se utilizó la distancia de Hausdorff como medida para determinar el mejor alineamiento entre ambas partes.

Las estrategias propuestas tienen en común los siguientes aspectos: se pueden utilizar para la construcción de mapas con más de un robot, son estrategias basadas en la exploración de fronteras y todas ellas consideran la incertidumbre que existe cuando el robot se desplaza durante

la exploración. La construcción de mapas con múltiples robots genera un problema de coordinación de movimientos que en este trabajo no fue abordado; en la primera estrategia de exploración se permitió a ambos robots moverse al mismo tiempo tratando inicialmente de acomodar a los robots de manera que no interfirieran el uno con el otro. En el caso de las otras dos estrategias se optó por permitir solamente el movimiento de un sólo robot a la vez.

En todos los casos, para continuar la exploración, los robots deben desplazarse hacia las fronteras que existen entre lo que ya se conoce del ambiente y la parte desconocida; en la primera estrategia (utilizando MDPs), los robots se dirigen al punto medio de la frontera, en la segunda estrategia (con la función de utilidad), los robots acuden a configuraciones generadas de manera aleatoria cercanas a las fronteras y finalmente en la tercera estrategia (utilizando programación dinámica), los robots deben acercarse simplemente al punto medio de la frontera.

En cuanto al manejo de la incertidumbre, en todos los casos se asume que existe incertidumbre en el control de movimiento de los robots, en el caso de MDPs y de programación dinámica esta incertidumbre se modela por medio de distribuciones discretas de probabilidad; y en el caso de la estrategia basada en la generación de muestras, la incertidumbre se representa en un término de la función de utilidad que considera la longitud de los desplazamientos y los cambios de orientación. Además, en el caso de la estrategia de exploración donde se utiliza programación dinámica y la estrategia basada en la generación de muestras, se considera también una incertidumbre en el sensado modelada por medio de distribuciones de probabilidad que definen la capacidad de los sensores de reconocer landmarks y fronteras.

De manera particular para el caso de la exploración basada en la generación de configuraciones que se evalúan con la función de utilidad se tienen las siguientes conclusiones:

• Desde el punto de vista de la planficación de trayectorias, la originalidad de esta estrategia es debido al hecho de que la meta del robot debe ser determinada en cada iteración del

- algoritmo. En los casos donde se considera más de un paso adelante en el tiempo, consideran que existen varias sub-metas que se deben cubrir en una iteración del algoritmo.
- La función de utilidad fue diseñada para combinar la información geométrica que se tiene, la facilidad de integrar las nuevas lecturas y la información a cerca de los movimientos del robot. Así mismo dicha función trata de balancear el hecho de tratar de ver la mayor área posible no visitada con el hecho de tener el mayor traslape de mapas e información sobre landmarks para asegurar un buen alineamiento del mapa y una mayor localización del robot.
- La función de utilidad toma en cuenta la incertidumbre en la posición del robot de manera que se prefieran las trayectorias con la menor cantidad de movimientos que afecten menos la localización del robot, teniendo en cuenta que no todos los tipos de movimiento afectan por igual en la incertidumbre (Por ejemplo los giros inducen un mayor error en la localización que avanzar en línea recta). Con esto de manera indirecta se está considerando también un menor consumo de energía.
- La función de utilidad toma en cuenta que las capacidades sensoriales que puede tener un robot, tales como distintas rangos de sensado y ángulo de vista. De manera que un robot con mayor rango y ángulo de vista, podrá "ver" más esquinas y landmarks que son parte de un término de la función de utilidad. Por lo tanto el algoritmo de exploración generará una mayor o menor trayectoria dependiendo de las capacidades de sensado de los robots.
- Se propuso también un algoritmo que puede explorar varios pasos adelante en el tiempo sin incurrir en un costo computacional alto. Se comparó esta estrategia con una estrategia que optimiza la función de utilidad con un paso adelante en el tiempo. El resultado es que en general es mejor considerar solamente un paso de anticipación. La interpretación de este resultado es que realizar planes de largo plazo con información dinámica y parcial puede resultar en un desperdicio de recursos.
- En resumen, esta estrategia de exploración cubre algunos vacíos entre los métodos

- geométricos exactos y los métodos que consideran la incertidumbre tomando en cuanta la información percibida.
- Se consideraron también los casos con múltiples robots centralizando el control de movimientos para evitar colisiones y se realizaron pruebas con robots reales.

En cuanto a la estrategia propuesta para considerar que la construcción del mapa se lleve a cabo con robots con distintas características, se tienen las siguientes observaciones y comentarios:

- Modelando el sistema multirobots como un sistema dinámico con estado de información imperfecta, se puede considerar que cada uno de los robots tiene diferentes capacidades motrices, diferentes rangos y ángulos de visión y diferentes capacidades para reconocer si el propio robot ha llegado a una posición en particular.
- Al considerar el método de programación dinámica se puede definir una política que indique la acción más conveniente a ejecutar cuando se han ido aplicando controles previos y se han ido tomando lecturas previas.
- La implementación del método de programación dinámica para el caso de información imperfecta puede resultar muy costosa computacionalmente debido a que se toma en cuenta toda la historia previa de acciones y lecturas. Esto se traduce que entre más robots se tengan y exista también un mayor número de fronteras por explorar mayor será el costo computacional para obtener la política que indique qué robot visitará qué frontera durante la exploración.
- A pesar de la desventaja que se presenta a cerca del costo computacional, al modelar el sistema multirobot como un sistema dinámico con incertidumbre y poder así aplicar el método de programación dinámica, provee de una herramienta formal para abordar el problema de exploración con múltiples robots con distintas capacidades.

Como todo trabajo de investigación, lo realizado en esta tesis permite continuar con diferentes aspectos que pueden ser mejorados y desarrollados a partir del trabajo realizado. A continuación se mencionan algunas cosas que se pueden realizar como trabajo futuro a lo actualmente presentado:

- Se puede considerar un algoritmo de coordinación de movimientos de los robots más sofisticado que permita explorar simultáneamente a los robots ejecutando trayectorias al mismo tiempo sin que interfieran entre sí.
- Se puede considerar que existe incertidumbre en la posición de los landmarks en el ambiente y no solamente incertidumbre en cuanto a que puedan ser reconocidos por los sensores de los robots.
- Se debe completar el sistema computacional que funciona con los robots reales, de manera que también se puedan llevar a cabo experimentos reales con un sistema de visión que identifique landmarks en el ambiente.
- Hace falta llevar a cabo experimentos reales con más de un robot a la vez para el caso de la estrategia de exploración basada en la función de utilidad.
- En cuanto a la exploración con robots con distintas capacidades es necesario mejorar la implementación de manera que se tenga una simulación que permita observar cómo cambian las políticas cuando se varías distintos parámetros como pueden ser el número de fronteras, el número de robots, la probabilidades de sensado y de control, la visibilidad de cada robot, etc.
- Se puede explorar la posibilidad de dividir el espacio conocido en el ambiente en algún tipo de celdas que permitan a los robots con distintas características pasar de un estado a otro sin que esto signifique pasar de una frontera a otra en un solo paso, y tal vez así permitir hacer correcciones en cuanto a la posición de los robots incluso mientras se encuentran desplazándose hacias las fronteras. Esto supone por supuesto un alto costo computacional.
- Para tratar de evitar en parte el alto costo computacional que se menciona en el punto anterior, se puede buscar la manera de evitar el cálculo de la probabilidad  $p(x_k|I_k)$  para todos los posibles vectores de información tratando de aproximar dicha distribución de

probabilidad con una función gaussiana que permita el uso del filtro de Kalman para los cálculos.

Finalmente a lo largo del desarrollo del presente trabajo, también se realizaron algunos artículos de investigación relacionados con la robótica móvil que no estaban en relación directa con el trabajo de tesis presentado. Sin embargo se pueden integrar estos trabajos con la tesis presentada para contar con sistemas multirobots que resuelvan problemas más generales o que permitan contar con sistemas más autónomos. A continuación se mencionan brevemente dichos trabajos y se comentan las posibles maneras de relacionarlos con esta tesis:

- Hasta el momento, se obtiene como salida de la exploración del ambiente un mapa geométrico; sin embargo no se presentó ninguna aplicación en la cuál se utilicen dichos mapas. Uno de los árticulos desarrollados previamente [Murrieta05a] ataca el problema de mantener la visibilidad de un objetivo móvil por medio de un robot no holonómico. Para resolver este problema se necesita que el robot cuente con un mapa del ambiente que puede ser el resultado de una exploración realizada con lo mencionado en la tesis.
- Se trabajó un método basado en eventos críticos para llevar a cabo movimientos reactivos y planificación del sensado [Murrieta05b] para robots móviles que se encuentran solucionando distintos problemas. Un posible trabajo futuro es tratar de encontra la manera de definir alguna de las estrategias de exploración propuestas considerando dicho método.
- Otra extensión natural sería definir las estrategias de exploración pero ahora para el caso de ambiente de exterior. De manera que se podrían utilizar algunos algoritmos de visión desarrollados para la identificación de ambientes naturales en [Alencastre04] y [Alencastre06a].
- Finalmente se ha desarrollado una arquitectura distribuida llamada OODVR
  [Alencastre07], que permite la interacción entre múltiples robots reales y virtuales de
  manera remota. A esta arquitectura se han integrado varias aplicaciones relacionadas con

robótica móvil [Muñoz04b][Muñoz03] o con dispositivos electrónicos de control del ambiente virtual [Díaz06]. Por lo tanto existe la posibilidad de integrar alguna de las estrategias de exploración de manera que puede ser probada utilizando múltiples robots reales y/o virtuales al mismo tiempo.

## **BIBLIOGRAFÍA**

[ActivMedia 04] ActivMedia, Aria Libraries, http://www.activrobots.com

[Aho83] A. Aho, J. Hopcropft and D. Ullman. *Data Structures and Algorithmics*. (Addison Wesley,1983).

[Alencastre04] M. Alencastre-Miranda, L. Muñoz-Gómez, R. Swain-Oropeza, "Monocular Vision-Based Exploration with an Outdoor Nonholonomic Mobile Robot". In proceedings of *10th IASTED International Conference on Robotics and Applications (RA'2004)*, Honolulu, Hawaii, USA, August 23-25, 2004, 6p.

[Alencastre06a] M. Alencastre-Miranda, L. Muñoz-Gomez, R. Swain-Oropeza, C. Nieto-Granda, "Color-Image Classification using MRFs for an Outdoor Mobile Robot". *Journal of Systemics, Cybernetics and Informatics*, International Institute of Informatics and Cybernetics, Vol. 3, No. 1, 2006, ISSN 1690-4524, 8p.

[Alencastre06b] M. Alencastre-Miranda, L. Muñoz-Gómez, R. Murrieta Cid, and R. Monroy. "Local Reference Frames vs. Global Reference Frame for Mobile Robot Localization and Path Planning". In Proceedings of the *IEEE Proceedings of Mexican International Conference on Artificial Intelligence (MICAI'06)*. Apizaco, Tlaxcala, México, November 2006, 10p.

[Alencastre07] M. Alencastre-Miranda, L. Muñoz-Gómez, C. Nieto-Granda, I. Rudomin and R. Swain-Oropeza. "A Multi-Robot-Multi-Operator Collaborative Virtual Environment". In Davide Brugali, *Software Engineering for Experimental Robotics*. Springer Tracts on Advanced Robotics (STAR series), 2007.

[Bemporad97] A. Bemporad and M. DiMarco and A. Tesi, "Wall-Following Controllers for Sonar-Based Mobile Robots", *Conference on Decision and Control*, pp. 3063-3068, San Diego, CA, USA, 1997.

[Bertsekas00] D. Bertsekas. *Dynamic Programming and Optimal Control*. (Athena Scientific, 2000).

[Bourgault02] F. Bourgault and A. Makarenko and S.B. Williams and B. Grocholsky and H.F. Durrant-Whyte , "Information Based Adaptive Robotic Exploration", Proceedings of International Conference on Intelligent Robots and Systems , pp. 540-545, Lausanne, Switzerland, 2002.

[Castellanos99] J. Castellanos, J. Montiel, J. Neira and J. Tardós, "The SPmap: A probabilistic framework for simultaneous localization and map building", *IEEE Transactions on Robotics and Automation*, pp. 948-953, 15(5), October, 1999.

[Chatila85] R. Chatila, J.P. Laumond. "Position referencing and consistent world modeling for mobile robots". *International Conference on Robotics and Automation*, San Louis, USA, 1985, 138-145.

[Chatila94] R. Chatila, "Mobile Robot Navigation Algorithms", in *Workshop on the Algorithm Foundations of Robotics* (WAFR), February 1994, 97-108.

[Cheeseman86] R.C. Smith and P. Cheeseman, "On the Representation and Estimation of Spatial Uncertainty", *International Journal of Robotics Research*, pp. 56-68, 5(4), Winter, 1986.

[Choset05] H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L.E. Kavraki and S. Thrun. *Principles of Robot Motion. Theory, Algorithms, and Implementations.* (The MIT Press, Cambridge, Massachusetts, 2005).

[Crowley89] J.L. Crowley. World Modeling and Position Estimation for a Mobile Robot Using Ultrasonic Ranging. *International Conference on Robotics and Automation*, Scottsdale, AZ, 1989, 679-680.

[Díaz06] M. Díaz, M. Alencastre-Miranda, L. Muñoz-Gómez, I. Rudomin. "Multi-user Networked Interactive Augmented Reality Card Game". In Proceedings of the *IEEE International Conference of CYBERWORLDS*, Laussane, Switzerland, November 2006, 4p.

[Dissanayake01] G. Dissanayake, P. Newman, S. Clark, H. Durrant-Whyte, "A solution to the Simultaneous Localization and Map Building (SLAM) Problem", IEEE *Transactions on Robotics and Automation*, pp. 229-241, 17(3), June, 2001.

[Dudek00]G. Dudek, M. Jenkin. *Computational Principles of Mobile Robotics*. (Cambridge University Press, 2000).

[Elfes87] A. Elfes, "Sonar-Based Real-World Mapping and Navigation", IEEE Journal on

*Robotics and Automation*, 3(3), 1987, 249-265.

[Feder99] H. Feder, J. Leonard, C. Smith, "Adaptive mobile robot navigation and mapping", *International Journal of Robotics Research*, pp. 650-668, 18(7), 1999.

[Fenwick02] J.W. Fenwick, P.M. Newman, J.J. Leonard, "Cooperative Concurrent Mapping and Localization". *Proceedings of International Conference on Robotics and Automation*, Washington, DC, 2002, 1810-1817.

[Gonzalez94] J. Gonzalez, A. Ollero, A. Reina, "Map Building for a Mobile Robot equipped with a 2D Laser Rangefinder", *Proceeding of International Conference on Robotics and Automation*, San Diego, CA, 1994, 1904-1909.

[Gonzalez02] H.H. Gonzalez-Baños, J.-C. Latombe, "Navigation strategies for exploring indoor environments", International *Journal of Robotics Research*, pp. 829-848, 21(10/11), 2002.

[Haddad98] H. Haddad, M. Khatib, S. Lacroix, R. Chatila, "Reactive Navigation in Outdoor Environments using Potential Fields", in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA'98)*, Leuven, Belgium, May 16-20, 1998, pp. 1232-1237.

[Huang04] W. H. Huang, K. R. Beevers, "Topological Mapping with Sensing-Limited Robots", Accepted to the *Sixth International Workshop on the Algorithmic Foundations of Robotics* (WAFR 2004).

[Huttenlocher93] D. Huttenlocher, G.Klanderman and J. Rucklidge, "Comparing images using the Hausdorff distance". IEEE *Transactions on Pattern Analysis and Machine Intelligence*, pp. 850-863, 15(9), 1993.

[Konolige04] K. Konolige, D. Fox, C. Ortiz, A. Agno, M. Eriksen, B. Limketkai, J. Ko, B. Morisset, D. Schulz, B. Stewart, R. Vincent. Centibots: Very large scale distributed robotic teams. *Proceedings of the International Symposium on Experimental Robotics*, 2004.

[Korten94] D. Kortenkamp, T.Weymouth. "Topological mapping for mobile robots using a combination of sonar and vision sensing". *Proceedings of the Twelfth National Conference on Artificial Intelligence*, Menlo Park, CA, 1994, 979–984.

[Lazanas95] A. Lazanas, J.C. Latombe. "Landmark based robot navigation". *Algoritmica*, pp. 472-501, 13(1995), 1995.

[Leonard92] J. Leonard, H. Durrant-Whyte, Directed Sonar Sensing for Mobile Robot

Navigation. (Kluwer Academic, Boston, Massachusetts, 1992).

[Makarenko02] A. Makarenko, B. Williams, F. Bourgault, H.F. Durrant-Whyte, "An experiment in integrated exploration", Proceedings of *International Conference on Intelligent Robots and Systems*, pp. 534-539, Switzerland, 2002.

[Marín05] A. Marin-Hernandez, M. Devy, V. Ayala-Ramirez, "Visual Planning for Autonomous Mobile Robot Navigation", in *Mexican International Conference on Artificial Intelligence*, LNAI 3789, Springer-Verlag, pp. 1001-1011, Monterrey, Mexico, November 14-18, 2005.

[Marín06] A. Marin-Hernandez, R. Méndez-Rodríguez, F. M. Montes-González, "Significant Feature Selection in Range Scan Data for Geometrical Mobile Robot Mapping". In Proceedings of 5th International Symposium on Robotics and Automation (ISRA'2006), San Miguel Regla, Hidalgo, Mexico, 2006.

[Muñoz03] L. Muñoz-Gómez, M. Alencastre-Miranda, G. Arechavaleta-Servín, I. Rudomín, R. Swain-Oropeza, "Collaborative Robotics System: Motion Planning, Teleoperation and Programming". Short paper in CD of *Workshop at Fourth Summer School on Image and Robotics (SSIR'03)*, UDLA, Cholula, Puebla, México, July 7-18, 2003.

[Muñoz04] L. Muñoz-Gómez, M. Alencastre-Miranda, R. Swain-Oropeza, "A Multi-Robot Mapping Approach using different Motion Planning Methods". In proceedings of *10th IASTED International Conference on Robotics and Applications (RA'2004)*, Honolulu, Hawaii, USA, August 23-25, 2004, 6p.

[Muñoz04b] L. Muñoz-Gómez, M. Alencastre-Miranda, I. Rudomín, R. Swain-Oropeza, G. Arechavaleta, J. Ramírez-Uresti. "Extending OODVR, a collaborative virtual robotics environment". In Proceedings of *1st Workshop on Virtual Laboratories at IX Iberoamerican Conference on Artificial Intelligence (IBERAMIA'04)*, INAOE, Cholula, Puebla, México, November 22-23, 2004, ISBN 968-863-786-6, pp. 409-418.

[Murrieta05a] R. Murrieta-Cid, L. Muñoz-Gomez, M. Alencastre-Miranda, A. Sarmiento, S. Kloder, S. Hutchinson, F. Lamiraux, J. P. Laumond. "Maintaining Visibility of a Moving Holonomic Target at a Fixed Distance with a Non-Holonomic Robot". Accepted on *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'05)*, Edmonton, Alberta, Canada, August 2-6, 2005, 7p.

[Murrieta05b] R. Murrieta-Cid, A. Sarmiento, T. Muppirala, S. Hutchinson, R. Monroy, M. Alencastre, L. Muñoz and R. Swain. "A Framework for Reactive Motion and Sensing Planning: A Critical Events-Based Approach". *MICAI 2005, Lecture Notes on Artificial Intelligence LNAI* 

3789, Springer-Verlag: Berlin Heidelberg Germany, November 2005, ISSN 0302-9743, pp 990-1000.

[Navarro02] L.E. Navarro-Serment, R. Grabowski, C.J.J. Paredis, P.K. Khosla. Millibots: "The Development of a Framework and Algorithms for a Distributed Heterogenous Robot Team", *IEEE Transactions on Robotics and Automation*, 9(4), 2002, 31-40.

[Newman03] P. Newman, M. Bosse, J. Leonard, "Autonomous feature/based exploration", *Proceedings of International Conference on Robotics and Automation*, pp. 1234-1240, Taiwan, 2003.

[Pfister03] S.T. Pfister, S.I. Roumeliotis, J.W. Burdick, "Weighted Line Fitting Algorithms for Mobile Robot Map Building and Efficient Data Representation". *Proceeding of International Conference on Robotics and Automation*, Taipei, Taiwan, 2003, 1304-1311.

[Puterman94] M.L. Puterman. *Markov Decision Processes Discrete Stochastic Dynamic Programming*. (John Wiley & Sons, Inc., New York, NY. 1994).

[Russell03] S.J. Russell and P. Norving, *Artificial Intelligence: A Modern Approach*. (Prentice Hall, 2003).

[Sim03] R. Sim, G. Dudek, "Effective Exploration Strategies for the Construction of Visual Maps", Proceedings of International Symposium on Intelligent Robots and Systems, pp. 3224-3231, Las Vegas, USA, 2003.

[Sim05] R. Sim, N. Roy, "Global A-optimal Robot Exploration in SLAM", Proceedings of *International Conference on Robotics and Automation*, pp. 661-666, Barcelona, Spain, 2005.

[Simmons00] R. Simmons, D. Apfelbaum, W. Burgard, D. Fox, M. Moors, S. Thrun, H. Younes, "Coordination for multi-robot exploration and mapping", Proceedings of *AAAI National Conference on Artificial Intelligence*, 2000.

[Singh93] K.Singh, K. Fujimura. Map making by cooperating mobile robots. *International Conference on Robotics and Automation*, Tokyo, Japan, 1993, 254-259.

[Stroupe03] A.W. Stroupe, Tucker Balch. "Value-Based Observation with Robot Teams (VBORT) Using Probabilistic Techniques". Proceedings of *International Conference on Advanced Robotics*, Coimbra, Portugal, 2003.

[Thrun98] S. Thrun, W. Burgard and D. Fox, "Probabilistic mapping of an environment by a

mobile robot", Proceedings of *International Conference on Robotics and Automation*, pp. 1546-1551, Belgium, May, 1998.

[Thrun00] S. Thrun, W. Burgard, D. Fox, "A Real-Time Algorithm for Mobile Robot Mapping with Applications to Multi-Robot and 3D Mapping". Proceedings of *International Conference on Robotics and Automation*, San Francisco, CA, 2000, 321-328.

[Tovar02] B. Tovar, R. Murrieta-Cid, C. Esteves, "Robot motion planning for map building". Proceedings of *International Conference on Intelligent Robots and Systems*, pp. 673-680, Switzerland, 2002.

[Tovar06] B. Tovar, L. Muñoz-Gomez, R. Murrieta-Cid, M. Alencastre-Miranda, R. Monroy and S. Hutchinson. "Planning Exploration Strategies for Simultaneous Localization and Mapping". *Robotics and Autonomous Systems*, Elsevier, Vol. 54, No. 4, April 2006, ISSN 0921-8890, pp. 314-331.

[Tovar07] B. Tovar, R. Murrieta-Cid and S.M. LaValle, "Distance-Optimal Navigation in an Unknown Environment without Sensing Distances". *IEEE Transactions on Robotics and Automation*, 23(3), 2007, 506-518.

[Williams02] S. Williams, G. Dissanayake, H. Durrant-Whyte, "Towards Multi-Vehicle Simultaneous Localisation and Mapping", *Proceedings of International Conference on Robotics and Automation*, Washington, DC, 2002, 2743-2748.

[Yamauchi98a] B. Yamauchi, A. Schultz, W. Adams, "Mobile robot exploration and map-building with continuous localization". Proceedings of *International Conference on Robotics and Automation*, 1998, 3715 – 3720.

[Yamauchi98b] B. Yamauchi, "Frontier-Based Exploration using Multiple Robots". Proceedings of the Second International Conference on autonomous Agents, Minneapolis, MN, 1998, 47-53.

[Zhang00] L. Zhang, B. Ghosh, Line Segment Based Map Building and Localization Using 2D Laser Rangefinder. Proceeding of *International Conference on Robotics and Automation*, San Francisco, CA, 2000, 2538-2543.

[Zunino01] G. Zunino, H.I. Christensen, "Simultaneous Localization and Mapping in Domestic Environments", *Multisensor Fusion and Integration for Intelligent Systems*, pp. 67-72, 2001.

## APÉNDICE A.

En este apéndice se presenta la demostración de la ecuación 4.13. Esta ecuación, tal y como se describió en el capítulo 4, indica cómo calcular el valor de probabilidad de que el sistema se encuentre en determinado estado dado que se conoce toda la información de sensado y controles aplicados hasta determinado momento, es decir  $p(x_k|I_k)$ .

Se tiene que el modelo del sistema se encuentra dado por  $p(x_{k+1}|x_k,u_k)$  que indica la probabilidad de pasar de un estado a otro cuando se aplica determinado control. También se tiene el modelo de sensado definido por  $p(z_k|x_k)$ , que indica la probabilidad de que los sensores adquieran determinada lectura dado que el sistema se encuentra en determinado estado. Por lo tanto se debe de expresar  $p(x_k|I_k)$  en términos de  $p(x_{k+1}|x_k,u_k)$  y  $p(z_k|x_k)$ , para poder utilizar el algoritmo de programación dinámica con estado de información imperfecta.

En vez de utilizar la expresión de probabilidad en el tiempo k, se utilizará con el paso en el tiempo k+1, tal y como se muestra en la ecuación A.1. También en la expresión A.1, aplicando la regla de Bayes se obtiene el lado derecho de la igualdad.

$$p(x_{k+1}|I_{k+1}) = \frac{p(x_{k+1}, I_{k+1})}{p(I_{k+1})}$$
(A.1)

Tal y como se expresó en el capítulo 4, el vector de información se puede definir como  $I_{k+1} = (I_k, u_k, z_{k+1})$ , si esta expresión se sustituye en A.1, se obtiene la expresión A.2.

$$p(x_{k+1}|I_{k+1}) = \frac{p(x_{k+1}, I_k, u_k, z_{k+1})}{p(I_k, u_k, z_{k+1})}$$
(A.2)

Si a la expresión A.2, se aplica la regla de probabilidad conjunta, tanto en el denominador como en el numerador, se obtiene la expresión A.3. La regla de probabilidad conjunta, indica que si se quiere calcular la probabilidad de dos o más variables, ésta se puede expresar como una multiplicación de probabilidades condicionales, es decir se puede calcular con la siguiente expresión:  $p(a_1,a_2,a_3,...,a_n) = p(a_1) p(a_2|a_1) p(a_3|a_1,a_2)... p(a_n|a_1,a_2,...,a_{n-1})$ .

$$p(x_{k+1}|I_{k+1}) = \frac{p(I_k)p(u_k|I_k)p(x_{k+1}|I_k,u_k)p(z_{k+1}|I_k,u_k,x_{k+1})}{p(I_k)p(u_k|I_k)p(z_{k+1}|I_k,u_k)}$$
(A.3)

Cancelando los términos que se repiten en el numerador y el denominador, se obtiene lo siguiente:

$$p(x_{k+1}|I_{k+1}) = \frac{p(x_{k+1}|I_k, u_k) p(z_{k+1}|I_k, u_k, x_{k+1})}{p(z_{k+1}|I_k, u_k)}$$
(A.4)

Si se considera que una lectura depende del lugar desde dónde es tomada, es decir el valor de lo que se sensa depende del estado del sistema, entonces se puede decir que la lectura  $z_{k+1}$  es condicionalmente independiente de  $I_k$  y de  $u_k$  dado el estado  $x_{k+1}$ . Por lo tanto se asume la igualdad que se muestra en la expresión A.5 y con esto la ecuación A.4 se puede reescribir como la ecuación A.6.

$$p(z_{k+1}|I_k,u_k,x_{k+1}) = p(z_{k+1}|x_{k+1})$$
(A.5)

$$p(x_{k+1}|I_{k+1}) = \frac{p(x_{k+1}|I_k, u_k) p(z_{k+1}|x_{k+1})}{p(z_{k+1}|I_k, u_k)}$$
(A.6)

Ahora se analiza el único término del denominador, y aplicando la regla de Bayes, se obtiene la expresión que se denota como A.7.

$$p(z_{k+1}|I_k, u_k) = \frac{p(z_{k+1}, I_k, u_k)}{p(I_k, u_k)}$$
(A.7)

Si a esta ecuación se le aplica la marginalización con respecto a  $x_{k+1}$ , se obtiene la ecuación A.8. La marginalización permite calcular un valor de probabilidad para una variable y como una sumatoria en la cuál, para todos los posibles valores de otra variable x, se tiene la probabilidad conjunta de x y la variable y, es decir:  $p(y) = \sum_{\forall x} p(x,y)$ .

$$p(z_{k+1}|I_k, u_k) = \sum_{x_{k+1}} \frac{p(z_{k+1}, I_k, u_k, x_{k+1})}{p(I_k, u_k)}$$
(A.8)

Si a la ecuación A.8, se aplica la regla de probabilidad conjunta se obtiene la siguiente expresión:

$$p(z_{k+1}|I_k, u_k) = \sum_{x_{k+1}} \frac{p(u_k) p(I_k|u_k) p(x_{k+1}|I_k, u_k) p(z_{k+1}|x_{k+1}, I_k, u_k)}{p(u_k) p(I_k|u_k)}$$
(A.9)

Finalmente, simplicando la expresión anterior, y aplicando la suposición mostrada en A.5, se obtiene la ecuación A.10.

$$p(z_{k+1}|I_k,u_k) = \sum_{x_{k+1}} p(x_{k+1}|I_k,u_k) p(z_{k+1}|x_{k+1})$$
(A.10)

Si se sustituye la expresión A.10 en la ecuación A.6 se obtiene la ecuación A.11 que se muestra a continuación:

$$p(x_{k+1}|I_{k+1}) = \frac{p(x_{k+1}|I_k, u_k) p(z_{k+1}|x_{k+1})}{\sum_{x_{k+1}} p(x_{k+1}|I_k, u_k) p(z_{k+1}|x_{k+1})}$$
(A.11)

Ahora se analiza el término  $p(x_{k+1}|I_k,u_k)$  del numerador, y aplicando la regla de Bayes, se obtiene la expresión que se denota como A.12. Si a esta ecuación se le aplica la marginalización esta vez con respecto a  $x_k$ , se obtiene la ecuación A.13.

$$p(x_{k+1}|I_k, u_k) = \frac{p(x_{k+1}, I_k, u_k)}{p(I_k, u_k)}$$
(A.12)

$$p(x_{k+1}|I_k, u_k) = \sum_{x_k} \frac{p(x_{k+1}, I_k, u_k, x_k)}{p(I_k, u_k)}$$
(A.13)

Si a la ecuación A.13, se aplica la regla de probabilidad conjunta se obtiene la ecuación A.14. Y simplificando la expresión se obtiene la ecuación A.15.

$$p(x_{k+1}|I_k, u_k) = \sum_{x_k} \frac{p(u_k) p(I_k|u_k) p(x_k|I_k, u_k) p(x_{k+1}|x_k, I_k, u_k)}{p(u_k) p(I_k|u_k)}$$
(A.14)

$$p(x_{k+1}|I_k, u_k) = \sum_{x_k} p(x_k|I_k, u_k) p(x_{k+1}|x_k, I_k, u_k)$$
(A.15)

Ahora se hacen dos nuevas suposiciones de independencia condicional. Primero para el término  $p(x_k|I_k,u_k)$ , se puede asumir que  $x_k$  es condicionalmente independiente de  $u_k$  dado  $I_k$ . Esto debido a que el estado del sistema en el tiempo k, no depende del control  $u_k$  que se

aplicará para llegar al estado en el tiempo k+I. Por lo tanto se asume la igualdad que se muestra en A.16. Por otra parte, para el término  $p(x_{k+1}|x_k,I_k,u_k)$ , se puede asumir que  $x_{k+1}$  es condicionalmente independiente de  $I_k$  dado  $x_k$  y  $u_k$ . Esto debido a que si se conoce el estado en el tiempo k y el control que se aplicará, entonces se puede no tomar en cuenta todo lo que se encuentra en el vector de información y se puede obtener la igualdad mostrada en A.17.

$$p(x_k|I_k, u_k) = p(x_k|I_k)$$
 (A.16)

$$p(x_{k+1}|x_k, I_k, u_k) = p(x_{k+1}|x_k, u_k)$$
(A.17)

Ahora con las ecuaciones A.16 y A.17 sustituyendo en A.15 se obtiene lo siguiente:

$$p(x_{k+1}|I_k, u_k) = \sum_{x_k} p(x_k|I_k) p(x_{k+1}|x_k, u_k)$$
(A.18)

Ahora finalmente se sustituye la expresión A.18 en la ecuación A.11 con lo cuál se obtiene la ecuación A.19, que es la misma que la ecuación 4.13. Así se ha logrado expresar la probabilidad del estado del sistema dado el vector de información en términos del modelo de sensado y del modelo del sistema.

$$p(x_{k+1}|I_{k+1}) = \frac{\sum_{x_k} p(x_k|I_k) p(x_{k+1}|x_k, u_k) p(z_{k+1}|x_{k+1})}{\sum_{x_{k+1}} \sum_{x_k} p(x_k|I_k) p(x_{k+1}|x_k, u_k) p(z_{k+1}|x_{k+1})}$$
(A.19)