



industriales
etsii

Escuela Técnica
Superior
de Ingeniería
Industrial

UNIVERSIDAD POLITÉCNICA DE CARTAGENA

Escuela Técnica Superior de Ingeniería
Industrial

“Técnicas de Inferencia Estadística para percepción del Entorno. Aplicación a Vehículos Autónomos.”

TRABAJO FIN DE GRADO

GRADO EN INGENIERÍA EN TECNOLOGÍAS
INDUSTRIALES



Universidad
Politécnica
de Cartagena

Autor: Pedro Javier Vera Saura
Director: Carlos Fernández Andrés
Codirector: Pedro Javier Navarro Lorente

Cartagena, 2 de octubre de 2015

ÍNDICE

CAPÍTULO 1. OBJETIVOS DEL PROYECTO

- 1.1. INTRODUCCIÓN
- 1.2. OBJETIVOS
- 1.3. DESCRIPCIÓN DE LOS CAPÍTULOS DEL PROYECTO

CAPÍTULO 2. VEHICULOS AUTÓNOMOS

- 2.1. EVOLUCIÓN DE LA ROBOTICA Y SU ESTADO ACTUAL
- 2.2. MOVILIDAD
- 2.3. AUTONOMÍA
- 2.4. INFORMACIÓN SENSORIAL
- 2.5. NAVEGACIÓN
 - 2.5.1. LOCALIZACIÓN GLOBAL
 - 2.5.2. PLANIFICACIÓN DE TRAYECTORIAS
 - 2.5.3. LOCALIZACIÓN LOCAL
 - 2.5.4. EVASIÓN DE OBSTÁCULOS
- 2.6. PROTOTIPOS EN FUNCIONAMIENTO

CAPÍTULO 3. SISTEMA LIDAR

- 3.1. SISTEMA LIDAR
- 3.2. APLICACIONES SISTEMA LIDAR
- 3.3. LIDAR SICK LMS 111
- 3.4. CONEXIONES ELÉCTRICAS DEL SICK LMS 111
- 3.5. INSTALACIÓN DEL SICK LMS 111
- 3.6. TOMA DE MEDIDAS DEL SICK
 - 3.6.1 PUERTA CERRADA
 - 3.6.2 PUERTA ABIERTA
 - 3.6.1 CRUCE CON PASILLO

CAPÍTULO 4. TÉCNICAS DE INFERENCIA PROBABILÍSTICA

- 4.1. INTRODUCCIÓN
- 4.2. REGLA DE BAYES
- 4.3. INFERENCIA BAYESIANA
- 4.4. FUSIÓN SENSORIAL MEDIANTE LA REGLA DE BAYES

- 4.5. ROBÓTICA PROBABILÍSTICA
- 4.6. LOCALIZACIÓN BAYESIANA O DE MARKOV
- 4.7. APLICACIONES LOCALIZACIÓN DE MARKOV
- 4.8. FUSIÓN SENSORIAL MEDIANTE FILTRO DE KALMAN

CAPÍTULO 5. IMPLEMENTACIÓN DE UN SISTEMA DE AUTOLOCALIZACIÓN MEDIANTE MARKOV

- 5.1. INTRODUCCIÓN
- 5.2. ENFOQUE MÉTRICO
- 5.3. IMPLEMENTACIÓN DE LOCALIZACIÓN DE MARKOV
 - 5.3.1. CONJUNTO FINITO DE ESTADOS
 - 5.3.2. CREENCIA $BEL(x_t)$
 - 5.3.3. CONJUNTO FINITO DE ACCIONES
 - 5.3.4. FUNCIÓN DE TRANSICIÓN DE ESTADOS
 - 5.3.5. CONJUNTO FINITO DE OBSERVACIONES Y EL MODELOS DE OBSERVACIÓN

CAPÍTULO 6. CONCLUSIONES Y FUTUROS PROYECTOS

- 6.1. CONCLUSIONES
- 6.2. FUTUROS PROYECTOS

BIBLIOGRAFÍA

CAPÍTULO 1

OBJETIVOS DEL PROYECTO

1.1. Introducción

La robótica está cada vez más presente en nuestras vidas, y no sólo en las industrias como se suele pensar, también existen muchos robots destinados a fines domésticos, como los robots-aspiradores o los robots que limpian el fondo y las paredes de las piscinas, entre muchos otros. Aunque realmente, donde mayor densidad de robots se puede encontrar a día de hoy es en las industrias.

Los robots cada vez son más precisos, más rápidos y son capaces de realizar trabajos más complejos, por lo que está incrementando el número de robots en las industrias, mejorando el rendimiento de estas. Los robots cada vez son más autónomos, de forma que puedan responder correctamente a imprevistos, y así poder realizar mayor cantidad de trabajos y no sólo un mismo trabajo realizado miles de veces como el típico brazo robótico de las empresas automovilísticas.

En el siguiente gráfico se puede observar como incrementaron los envíos anuales de robots industriales en todo el mundo desde el año 1995 hasta el 2013, por lo que estamos ante un campo emergente.

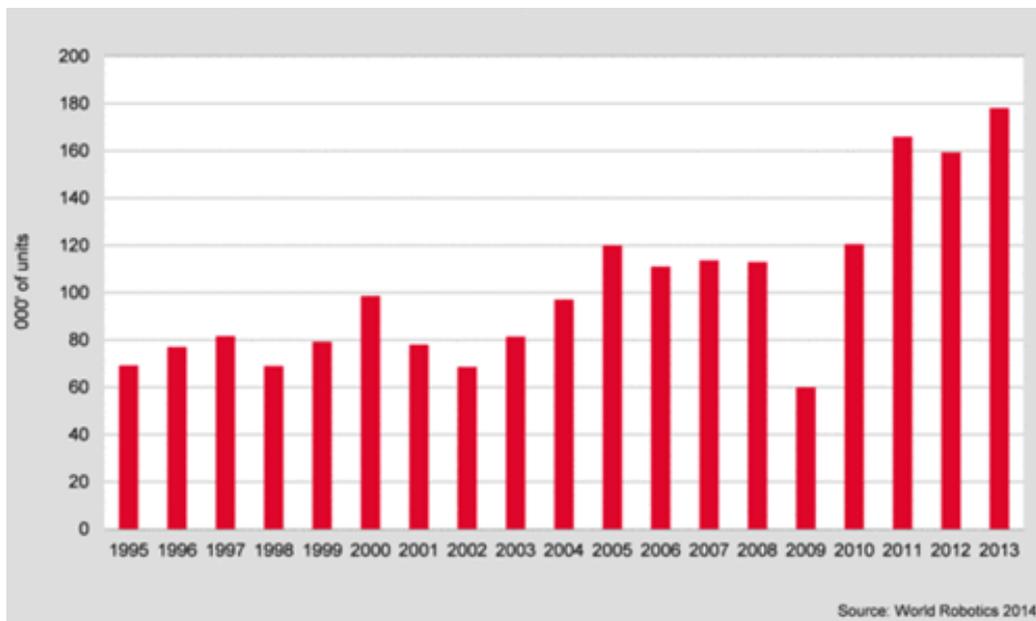


Figura 1.1: envíos anuales de robots industriales en todo el mundo en miles de unidades

Desde hace algo más de dos décadas, se están realizando numerosas investigaciones con el objetivo de conseguir un vehículo que sea totalmente autónomo, bien para entornos interiores (como es el caso de este proyecto fin de grado) o bien para entornos exteriores. Obviamente, las

Capítulo 1: Objetivos del proyecto

mayores inversiones en estos proyectos son realizadas por las grandes marcas de automóviles, que tienen como objetivo poner en el mercado vehículos autónomos que puedan realizar las mismas funciones que un vehículo tradicional pero sin la necesidad de un conductor. Pero también existen numerosas investigaciones sobre robots o vehículos autónomos enfocados a entornos interiores, además de muchos prototipos en funcionamiento, ya que pueden ser de gran ayuda en muchos ámbitos, como por ejemplo en industrias, para transportar material peligroso o pesado o para atravesar zonas de alta peligrosidad para las personas.



Figura 1.2 a)



Figura 1.2 b)

En la figura 1.2 a) aparece el vehículo autónomo de google para entornos exteriores, mientras que en la figura 1.2 b) se ve el robot "in-situ fabricator", que es un robot autónomo capacitado para la construcción de paredes en entornos interiores.

1.2. Objetivos del proyecto

Como ya hemos comentado anteriormente, este proyecto se centra en vehículos autónomos en entornos interiores, bien en oficinas, hospitales, industrias, etc.

El principal problema de la navegación de vehículos autónomos es conseguir procesar correctamente la gran cantidad de datos obtenidos mediante los sensores de los que está provisto el vehículo. Esto se debe a que pese a contar en la actualidad con sensores de muy buenas características, no resulta sencillo conseguir que los datos obtenidos por los sensores sean procesados de forma adecuada para percibir el entorno que le rodea.

De forma resumida estos van a ser los objetivos del presente proyecto:

1. Plantear las distintas técnicas utilizadas.
2. Obtener mapas con los datos captados por el sensor en el PC.
3. Diseñar un algoritmo con el que procesar estos datos mediante inferencia estadística.

1.3. Descripción de los capítulos

El proyecto fin de grado consta de los siguientes capítulos:

- Capítulo 1: Objetivos del proyecto

En este capítulo hemos realizado una breve introducción sobre el tema a tratar y se han marcado unos objetivos.

- Capítulo 2: Vehículos autónomos

Capítulo 1: Objetivos del proyecto

En este capítulo se ha realizado una recopilación de información acerca del estado de la técnica de los vehículos autónomos (problemas a resolver, tecnologías empleadas, prototipos en funcionamiento).

- Capítulo 3: Sistema LIDAR

En este capítulo se ha realizado una recopilación de información acerca del sistema LIDAR y una descripción detallada del LIDAR SICK LMS111.

- Capítulo 4: Técnicas de inferencia probabilística

En este capítulo se han descrito las técnicas de inferencia probabilística y se han recopilado aplicaciones de las mismas en el campo del reconocimiento del entorno.

- Capítulo 5: Implementación de un sistema de autolocalización mediante Markov

En este capítulo se ha realizado una implementación de un sistema de autolocalización basado en la localización de Markov.

- Capítulo 6: Conclusiones y futuros proyectos

En este capítulo se han ofrecido las conclusiones del presente proyecto y se dejan abiertos los futuros proyectos.

CAPÍTULO 2

VEHÍCULOS AUTÓNOMOS

2.1. Evolución de la robótica y estado actual

Los robots se quieren básicamente para facilitarnos la vida, de hecho, la palabra 'robot' viene del checo 'robota', que viene a significar 'trabajo forzado' o 'siervo'. El término Robot fue utilizado por primera vez por el dramaturgo checo Karel Kapek en su obra "R. U. R." (Rossum's Universal Robots). En esta obra los Robots eran máquinas inteligentes que servían a los humanos hasta que sucede una rebelión que aniquila a la raza humana.

La robótica como ciencia siempre ha estado relacionada con la inteligencia artificial. Los primeros robots móviles capaces de aprender tareas como detectar obstáculos datan de los años 50, y llamaron la atención enseguida de los pioneros de la inteligencia artificial. Los robots de aquella época, estaban limitados por el hardware, ya que su capacidad de movimiento estaba muy reducida y los sensores de los que se disponían dejaban mucho que desear. Entre los años 50 y 70, la limitación pasó a estar en las teorías de inteligencia artificial. Se intentó asemejar la inteligencia de los robots a la de los seres humanos, para esto se pretendió dotarlos, en primer lugar de redes neuronales y después de lógicas complejas y abstractas. Estas teorías vieron su fracaso a finales de los años 80, con el surgimiento de otras teorías como la robótica autónoma de Brooks, que proponía acabar con las teorías extremadamente complicadas de la inteligencia artificial clásica y empezar por lo más sencillo. Acorde con esto, se pretendió dotar a los robots de una inteligencia básica, semejante a la de los insectos, que les permitiera reaccionar frente a su entorno e ir avanzando poco a poco.

Esto dio lugar a las teorías multiagentes. Aunque no existe una definición formal y precisa de lo que es un agente, éstos suelen ser vistos como entidades inteligentes, equivalentes en términos computacionales a un proceso del sistema operativo. Siguiendo esta analogía, cada proceso se encarga de una tarea específica, y la unión de todos los procesos forma el sistema operativo. La teoría, en el campo de la robótica, sostiene que para hacer un robot autónomo, debemos de ir solucionando los problemas que surgen uno a uno, y una vez resueltos, combinarlos para formar la inteligencia del robot. Era el nacimiento de los robots autónomos, tal y como los conocemos en la actualidad.

En la actualidad, un robot autónomo cuenta con muchos procesos independientes, como por ejemplo las distintas fases de la navegación que veremos más adelante en este capítulo. Por ejemplo, dos de estas fases son la evasión de obstáculos y la planificación de trayectorias. Estas fases se consideran dos agentes distintos, que realizan su tarea de forma independiente, pero que se combinan entre ellas, y a la vez con muchos otros agentes, para formar la inteligencia del robot. De forma que si el robot detecta un obstáculo, deberá de planificar una nueva trayectoria, por lo que aunque las tareas se realizan de forma independiente, se combinan para que el funcionamiento del robot sea el deseado.

2.2. Movilidad

La movilidad de los robots es el grado que tienen de libertad para poder desplazarse por el mundo. Uno de los primeros usos prácticos que se le dieron a los robots fue industrial, con los llamados robots manipuladores. Estos robots están programados de tal manera que repiten la misma secuencia de acciones una y otra vez, de forma rápida, barata y más exacta que los humanos. Normalmente consisten en un brazo móvil fijado a un punto del suelo, de modo que se le permite el giro alrededor de ese punto pero no el desplazamiento. El más claro ejemplo de estos brazos robóticos son los usados en las industrias automovilísticas.



Figura 2.1: Brazos robóticos típicos de la industria automovilística

La necesidad o no de movilidad depende de la tarea que vaya a realizar el robot. Para los robots manipuladores poder desplazarse no es necesario, en cambio, sí lo es para un robot cuya tarea sea, por ejemplo, limpiar el suelo.

2.3. Autonomía

La autonomía de los robots depende del conocimiento a priori o la información del entorno donde desarrollan sus tareas. Los robots se pueden clasificar en función de su grado de autonomía: no autónomos, semiautónomos, y totalmente autónomos.

Los robots no autónomos son completamente dirigidos por humanos. La inteligencia de estos robots se limita a interpretar y ejecutar las instrucciones recibidas.

Los robots semiautónomos pueden navegar por sí mismos o dirigidos por humanos. También se pueden combinar los comandos de los humanos con decisiones autónomas para controlar los robots, un claro ejemplo de esto son los robots que han navegado por la Luna y Marte. Otra forma de semiautonomía se logra mediante un área o un mapa del entorno donde el robot tiene que navegar.

Los robots totalmente autónomos no requieren la interacción humana para completar sus tareas. Estos son capaces de realizar acciones y movimientos "inteligentes", sin ninguna guía o control externo.

El deseo o no de autonomía, al igual que ocurría con la movilidad, depende de la situación. Para los robots manipuladores no es un problema no tener autonomía, ya que interesa que hagan exactamente lo que se les pide. No tienen que tomar decisiones por sí mismos. En cambio, los robots que tienen que cumplir unos objetivos pero no se les dice como alcanzarlos, tienen que

Capítulo 2: Vehículos autónomos

poder decidir qué acción es la correcta en cada momento. En esta situación los robots serán semiautónomos o totalmente autónomos.

En las últimas décadas se ha constatado el creciente interés que presentan los robots o vehículos semiautónomos y autónomos, tanto desde el punto de vista de la investigación básica como de sus potenciales aplicaciones. La principal característica de los robots autónomos o semiautónomos, es su capacidad para poder desplazarse de un lugar a otro que le hayamos indicado sin la necesidad de ninguna acción por nuestra parte. En el caso de los que están enfocados en entornos interiores, esta particularidad los hace muy atractivos e interesantes para su aplicación práctica en oficinas, industrias u hospitales.

La principal dificultad de los vehículos o robots autónomos es la fase de localización del robot tanto local, como global, y es en la que más se va a centrar el presente proyecto.

2.4. Información sensorial

El robot móvil percibe el entorno a través de la gran cantidad de datos obtenidos por medio de los sensores. Para poder utilizar estos datos, el robot móvil debe de procesarlos adecuadamente para transformar la sensación del entorno (datos recibidos) en percepción del entorno (datos procesados).

Hay gran variedad de sensores, en robótica móvil se pueden clasificar en dos, internos y externos. Los internos son aquellos que nos proporcionan datos de las variables cinemáticas del robot y de su variación, es decir, aquellas que no interactúan con el entorno del robot. Algunos ejemplos de sensores internos son: potenciómetros, codificadores rotativos y acelerómetros. Los externos son aquellos que interactúan con el entorno del robot, varios ejemplos de sensores externos son los ultrasonidos, infrarrojos, visión, compás y laser. A estos sensores externos también se les denomina sensores exteroceptivos.

Una de las premisas más importantes que debe cumplir un robot autónomo, es que debe de conocer en todo momento el lugar en el que se encuentra, para ello, se hace uso de la odometría. A lo largo de este proyecto se va a hacer referencia muchas veces a la palabra odometría, que básicamente es una técnica encargada de estimar la posición relativa del robot respecto al punto de inicio del que ha partido el robot. Los sensores más utilizados para realizar esta estimación, son los codificadores rotativos, que se basan en medir la velocidad de giro de las ruedas a partir de contar el número de vueltas que realizan en un determinado periodo de tiempo.

A continuación vamos a conocer algunas de las características más relevantes de estos sensores:

-Potenciómetros: son sensores analógicos, los hay para medir posición lineal y angular. En esencia, son una resistencia variable en los que el terminal móvil se une al objeto cuya posición se quiere medir, por lo que son muy baratos. Además, tienen una gran resolución. Los inconvenientes son que se desgastan con el uso, que necesita una alimentación de tensión muy estable y que si el movimiento es rápido, la señal puede tener un ruido importante.

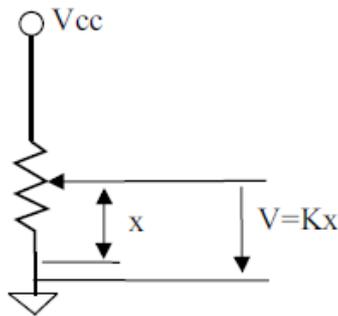


Figura 2.2 a)

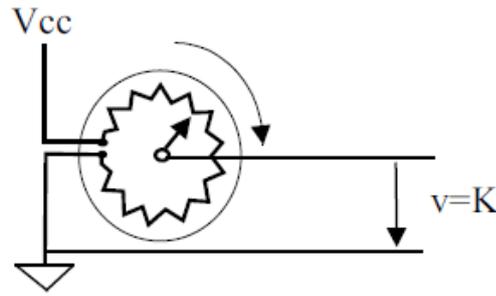


Figura 2.2 b)

En la Figura 2.2 a) vemos un potenciómetro para medir posición lineal, mientras que en la Figura 2.2 b) vemos un potenciómetro para medir posición angular

-Codificadores rotativos: Estos sensores son digitales y miden la posición angular de un eje, existen dos tipos básicos de codificadores rotativos, los incrementales y los absolutos. Los incrementales constan de un disco ranurado que se coloca solidario al eje que gira, de un fotoemisor y de un fotoreceptor colocados a ambos lados del eje que gira. Contando el número de pulsos de luz que llegan al fotoreceptor, se obtienen el número de ranuras que han pasado y por tanto el ángulo girado, y además, se puede obtener la velocidad contando el número de pulsos en un tiempo determinado. Mientras que en los absolutos, el disco ranurado tiene varias franjas de ranuras, cada una de las cuales detectadas por un par emisor-receptor. Cada franja tiene un espaciado de ranuras diferente, de forma que se codifica en binario la posición absoluta del eje. En aplicaciones de robots autónomos se prefiere el uso de los codificadores incrementales ya que el mayor interés está en la velocidad de las ruedas y no en su posición angular. Si el disco ranurado es de un solo canal, no se puede determinar en qué sentido está girando el eje, por lo que no sería aplicable en el caso de robots autónomos. Lo que se suele hacer es poner otro canal desplazado 90° del primero, obteniendo la señal mostrada en la Figura 2.2.

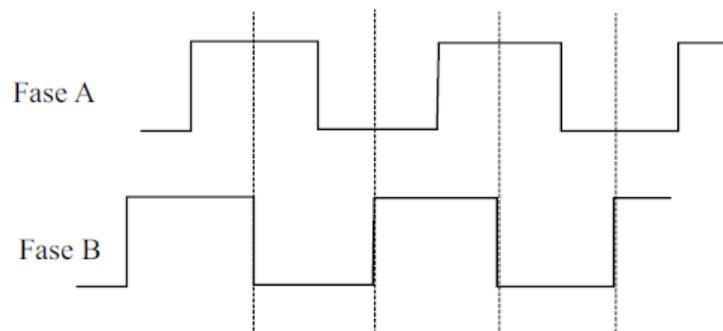


Figura 2.2: señal de un codificador de dos canales girando en sentido horario

- Acelerómetros: Se basan en la utilización de una masa conocida fijada a un elemento elástico. La aceleración produce sobre la masa una fuerza igual al producto de la masa por la aceleración. El sensor mide esa fuerza, con la que se obtiene una medida de la aceleración.

- Ultrasonidos: Sirven para detectar todo tipo de objetos. Constan de un emisor de ondas de frecuencia superior a 20 KHz (máximo audible por los seres humanos) y de un receptor que percibe el rebote de la onda contra los objetos próximos, suelen tener distancias de detección entre 40 cm y 10 m. Permiten la medida de la distancia a la que se encuentra midiendo el tiempo

que tarda en recibir el pulso de ultrasonido enviado. Suelen tener una precisión en torno a un centímetro. Los inconvenientes de estos sensores son que dependen de la temperatura del ambiente, de movimientos de aire y que en la presencia de fuentes acústicas de alta frecuencia aparece mucho ruido en las mediciones. Además presentan un comportamiento erróneo frente a objetos con poca reflexión o con demasiada reflexión.

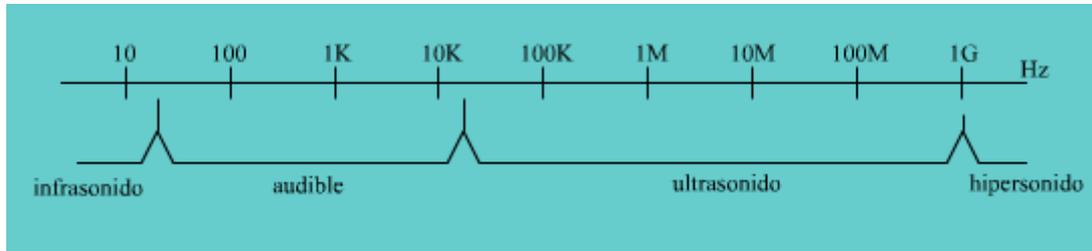


Figura 2.3: Clasificación del sonido en función de su frecuencia

- Infrarrojos: Se basan en el mismo principio que los ultrasonidos pero emitiendo un infrarrojo que permite medir objetos que se encuentren entre 10 y 80 cm. Se obtienen medidas con una precisión de ± 2 mm. Estas características los hacen útiles para maniobras de proximidad o para la evasión de obstáculos en la navegación. La principal desventaja de estos sensores es que su transductor es sensible a la intensidad de una señal luminosa, por lo que señales luminosas como la luz solar o luces artificiales afectan en las mediciones.
- Bumpers o sensores de choque: la mayoría de los robots móviles incorporan sensores de choque, dado que en los entornos reales existen gran cantidad de objetos que, ya sea por su tamaño o posición (patas de mesas, sillas, obstáculos del terreno) los sensores de ultrasonidos o infrarrojos son incapaces de detectar.
- Sensores piroeléctricos: estos sensores son capaces de detectar el movimiento de cuerpos que emiten radiación infrarroja, y su aplicación es la detección de personas en movimiento. Este tipo de sensores son utilizados para seguir movimientos y detectar presencia, distinguiendo a personas de objetos en un entorno.
- Láser: detecta medidas de distancia a objetos opacos. Poseen una buena precisión y velocidad de adquisición de datos. La ventaja de este tipo de sensores frente a los ultrasonidos, es que estos son más rápidos y precisos, pero no permite detectar objetos transparentes, ya que el haz emitido atraviesa el objeto transparente y por tanto no lo detecta.

2.5. Navegación

La navegación de cualquier robot o vehículo autónomo consta de varias fases imprescindibles para poder conseguir con éxito que el robot llegue de un lugar a otro que se le haya indicado sin que se produzca ningún choque y además lo haga de la forma más eficiente posible cumpliendo con las condiciones que se le hayan implementado en el diseño. En la Figura 2.4 se puede observar el proceso general formado por las distintas fases de las que consta la navegación del robot.

Voy a explicar brevemente el proceso que sigue la navegación del robot y en los siguientes apartados realizaré una explicación más exhaustiva de cada fase.

En primer lugar, el robot necesita un mapa del lugar. Este mapa puede ser obtenido por el propio robot en una fase previa de exploración, o puede ser suministrado directamente en formato CAD

o similar. También hay robots que van formando este mapa, conforme van explorando la zona, pero es un reto más complicado que no se va a abordar en este proyecto.

Obtenido este mapa, lo primero que debe de hacer el robot es ubicarse dentro de él. Esta fase es conocida como localización global. Una vez ubicado, el robot tiene que planear la trayectoria más adecuada. Esta fase es conocida como planificación de la trayectoria. Además, se debe realizar un seguimiento de la posición, para que el robot sea consciente de su posición en todo momento. Esta fase es denominada localización local. Por último, el robot debe detectar y evitar cualquier objeto imprevisto o persona que se encuentre por el camino.

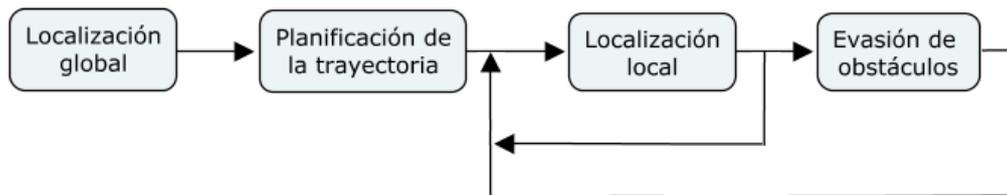


Figura 2.4: Proceso de navegación del robot.

2.5.1. Localización global

Esta es la fase que presenta mayores dificultades en el proceso de navegación ya que se pretende ubicar al robot en el mapa del que se dispone sin ningún tipo de información sobre su ubicación. Tradicionalmente, se había supuesto que para solucionar este problema eran necesarias técnicas basadas en búsqueda de similitudes, en las que a partir de las lecturas de los sensores del robot, se obtenía un conjunto de características y se buscaba el mejor emparejamiento de éstas con el modelo del entorno. Sin embargo, el uso de estas técnicas no era lo suficientemente eficiente para el funcionamiento continuo de un robot móvil. El enfoque de la búsqueda de similitudes fue abandonado por la comunidad de robótica móvil, y el problema de la localización global ha permanecido sin solución hasta la utilización de métodos probabilísticos (bayesianos). Estos métodos se pueden dividir en dos grandes grupos: los que utilizan celdas de probabilidad y los que utilizan modelos topológicos.

Un caso más extremo que se puede incluir en este apartado son los secuestros. Esto se da cuando el robot es consciente de su posición, pero de repente es trasladado a otra posición, por lo que el robot debe darse cuenta que ha sido cambiado de sitio y volver a ubicarse en el entorno. Para solucionar esto, se usarán los mismos métodos probabilísticos.

2.5.2. Planificación de la trayectoria

En esta fase el robot debe elegir la mejor trayectoria teniendo en cuenta el tamaño y las capacidades motrices del robot o cualquier otra condición que se quiera implementar, por ejemplo que se mantenga cerca de las paredes o que se mueva por el centro de las habitaciones. Otras condiciones usuales son las de evitar que se acerque a más de cierta distancia de cualquier obstáculo, para tener más margen de error o la de limitar el número de giros, ya que los giros son una de las mayores fuentes de errores en el odómetro. Existen dos tipos de mapas, los métricos y los topológicos.

2.5.2.1. Mapas métricos

Los mapas métricos son representaciones de las propiedades métricas o de las coordenadas de los objetos del entorno. Este tipo de representaciones se suelen realizar en el mismo sistema de coordenadas de dos dimensiones en el que se representa el robot, facilitando la fusión de los datos obtenidos por el sensor con los propios del robot. Dentro de los mapas métricos, estos se pueden clasificar en mapas geométricos o mapas de ocupación.

Los mapas geométricos representan el entorno mediante un conjunto de características geométricas tales como puntos, líneas y esquinas. La ventaja principal de estos modelos es que, si se utilizan con un buen modelo del sensor, es posible simular los datos que los sensores del robot obtendrían en cualquier posición del entorno. Esto permite comparar los datos percibidos por el robot con los que se obtendrían en posiciones candidatas, para actualizar la posición del mismo.

En cambio, los mapas de ocupación representan el entorno como un conjunto de celdas cuadrículadas. Siendo cada celda una pequeña área o volumen del entorno donde se encuentra el robot. Este tipo de mapas se suelen utilizar para asignarle a cada celda un valor que exprese la probabilidad de que el robot se dirija hacia la zona del entorno que representa esta celda. Este valor se le suele asignar teniendo en cuenta las condiciones que hemos comentado al principio de este apartado. Si por ejemplo, no queremos que el robot se acerque bajo ningún concepto a menos de 20 cm de ningún obstáculo, a las celdas que representen zonas que se encuentren a menos de 20 cm de los obstáculos, se les asignaría un valor muy alto, asegurándonos que el robot nunca se dirija a estas celdas. El problema de este tipo de mapas es que se considera que el robot solo puede realizar giros de 45° , al solo poder dirigirse a alguna de las celdas que le rodea. Una solución a este problema es, una vez se haya calculado la trayectoria, reducir el número de giros siempre que sea posible sin incumplir las condiciones de diseño.

2.5.2.2. Mapas topológicos

Los mapas topológicos representan el entorno como un gráfico en donde cada nodo representa lugares de marcada importancia y las interconexiones o ramas representan la forma en la que se puede ir de un nodo a otro. Un inconveniente común a los métodos de representación anteriores es que se requiere mantener la consistencia métrica global, lo cual resulta difícil cuando el espacio de movimiento es grande. En muchas aplicaciones, y especialmente en las de navegación global, no se requiere una precisión métrica que justifique el uso de los métodos anteriores, que para entornos amplios requieren mucha memoria de almacenamiento de datos. En estos casos, se puede utilizar una discretización más gruesa que, junto con comportamientos de navegación local robustos, permitan conocer de forma aproximada la zona en la que se encuentra el robot y desplazarse localmente por ella.

2.5.3. Localización local

Para que el robot pueda realizar la trayectoria planificada correctamente, debe de saber donde se encuentra desde que parte del punto de inicio hasta que llega a su destino. Para esto, los robots autónomos deben de ir provistos de un odómetro, que como hemos comentado mide la velocidad de rotación, y con ello, la distancia recorrida, en principio se podría pensar que es suficiente para saber dónde se encuentra el robot en cierto momento conociendo el punto de partida. Pero por muy bueno que sea el odómetro, lleva asociado un error que se va acumulando en el tiempo y puede hacer que la posición que cree tener el robot diste mucho de la real, pudiendo provocar un choque contra una pared o persona. Por otra parte, cuando el robot

desliza, el espacio recorrido durante el deslizamiento no es registrado por el odómetro. Además de esto, en los giros, el error del odómetro es mucho mayor, por lo que para seguir la posición del robot debemos de complementar la información que nos proporciona el odómetro con otra proporcionada por sensores exteroceptivos (sensor láser normalmente).

Hay dos formas de realizar un seguimiento de la posición mediante el sensor láser, la primera está basada en la correlación y la segunda en características. El modelo basado en la correlación es muy utilizado debido a su precisión, pero se tarda más tiempo debido a que el robot debe de procesar todos los datos sensados para compararlos con el mapa del entorno del que dispone y poder así determinar su posición encontrando similitudes entre los datos obtenidos y el mapa del que dispone. El modelo basado en correlaciones, se basa en identificar unas marcas en el ambiente, que pueden ser naturales o artificiales, y a partir de estas marcas determinar su posición. Este método es mucho más rápido ya que se procesan muchos menos datos. La desventaja de este último es que la identificación de estas marcas debe de ser muy precisa para realizar una buena estimación.

Bien se haya usado un método u otro, se debe de determinar una correspondencia entre la medición del odómetro y los datos obtenidos por los sensores, bien sea el grueso de datos medidos o simplemente las marcas encontradas.

2.5.4. Evasión de obstáculos

Es muy común que las personas dejemos objetos en lugares en los que no deberían estar, como por ejemplo sillas, cajas, etc. Esta fase empieza en el momento que el robot detecta por medio de sus sensores un obstáculo en la ruta que había planificado con anterioridad. Una vez ha sido detectado el obstáculo, el robot deberá de calcular una nueva ruta y de no ser posible, determinar la imposibilidad de llegar a la meta.

Para resolver esta cuestión lo primero que debe hacer el robot es determinar si el obstáculo es dinámico o estático, ya que en función de esto, deberá de actuar de una forma o de otra. Si el obstáculo es estático, el robot deberá de incorporar este obstáculo a su mapa momentáneamente y volver a planificar la trayectoria con este nuevo mapa momentáneo para llegar al destino fijado inicialmente.

Si el obstáculo es dinámico es más complicado de solucionar, ya que depende de muchos factores. Hay mecanismos basados en técnicas de control reactivo, que se caracterizan por el rápido tiempo de respuesta del que disponen, ya que se ofrece una reacción en tiempo real frente a los estímulos aportados por los sensores. La fusión de la información aportada por los sensores determina la acción a realizarse de forma automática. Por ejemplo, algunas están basadas en campos de potencial, donde los obstáculos son tratados como emisores de energía, y se hace al robot ir acercándose a la meta a la vez que va rodeando el obstáculo (si es posible).



Figura 2.2: Proceso básico en el que se basa el control reactivo.

2.6. Prototipos en funcionamiento

Ahora vamos a ver algunos de los prototipos que están en funcionamiento actualmente, el prototipo más interesante quizás sea el proyecto ROBAUCO, en el que se propone desarrollar un conjunto de tecnologías para obtener robots móviles capaces de realizar tareas complejas con un alto grado de autonomía e interactuar con seres humanos.

En el proyecto ROBAUCO se utilizó un sistema integral de percepción basado en la fusión de distintas fuentes sensoriales con el que se consigue reconocer entornos complejos y se desarrollaron nuevas formas de interacción persona-robot más eficientes y naturales utilizando la voz y gestos. El robot está dotado de un comportamiento autónomo y de la capacidad de navegación en entornos no estructurados. Además, crearon un framework que permite la integración de los sistemas perceptivos y cognitivos en plataformas móviles, tanto terrestres como acuáticas o aéreas y definieron un modelo y herramientas que permitan la simulación de las tecnologías desarrolladas.

Como resultado del proyecto ROBAUCO, se fabricaron con el asesoramiento de la Dirección de Atención de Emergencias del Gobierno Vasco dos prototipos de robot, uno de interiores y otro de exteriores, orientados a colaborar en diferentes escenarios de emergencias para ofrecer apoyo en el reconocimiento y valoración de la situación.



Figura 2.2: Robot del proyecto ROBAUCO

El segundo prototipo que voy a exponer es el robot que está siendo desarrollado por el NCCR Digital Fabrication. El robot es denominado como "in situ fabricator" y el objetivo de este proyecto es desarrollar un robot autónomo que pueda construir con precisión estructuras complejas con tolerancias de construcción milimétricas, y al mismo tiempo ser capaz de hacer frente a las incertidumbres de una obra de construcción.

Uno de los integrantes del proyecto explicó que el sistema debe depender solamente de su propia detección y la informática. Para esto, el robot cuenta con un seguimiento de la posición

Capítulo 2: Vehículos autónomos

muy preciso basado en correlaciones, que como hemos visto antes, se basa en hacer barridos con el láser y obtener una gran cantidad de datos para compararlos con el mapa del que dispone el robot, que como también hemos explicado, podrá haberlo obtenido el propio robot con sus propios sensores en un fase previa de exploración o se le habrá importado directamente.

El problema que tiene este prototipo es que después de haberse movido, el robot no es consciente de su posición por lo que tiene que volver a localizarse. Se pretende conseguir que el robot sea continuamente consciente de su posición, incluso estando en movimiento.



Figura 2.3: datos obtenidos por “in situ fabricator”

CAPÍTULO 3

SISTEMA LIDAR

3.1. Sistema LIDAR

Se puede definir LIDAR (Light Detection And Ranging) como una tecnología que permite medir la distancia entre un sensor y un objeto mediante el empleo de ondas electromagnéticas.

Se trata de un potente sistema de recolección de datos que provee información en 2D ó 3D de un determinado ámbito de estudio. Útil para muchos usos como obtención de la superficie, de la vegetación, de los edificios, aplicaciones de robots autónomos, etc. Dispone de un sensor que utiliza un láser que contiene un transmisor y un receptor. El láser normalmente trabaja en una longitud de onda de 532 a 1550 nm.

El escáner de láser, emite impulsos de alta frecuencia y recoge las reflexiones que se producen en los elementos, midiendo el tiempo desde que el pulso láser es emitido hasta que es detectada su señal reflejada. Conocido el tiempo, mediante un simple cálculo se obtiene la distancia existente al objeto o superficie contra el que se ha reflejado el laser. Un sistema LIDAR básico, cuenta únicamente con un laser trasmisor y un receptor.

Los términos más utilizados en la tecnología LiDAR son los siguientes:

- Pulso de repetición: es el ritmo al que el láser emite pulsos, y se mide en Hercios (Hz). Por ejemplo, un pulso de 200 Khz de un determinado sensor significa que el LiDAR emite pulsos 200.000 veces por segundo (y por tanto el receptor también recibe posteriormente esa información).
- Frecuencia de escaneo: Mientras que el láser emite los pulsos de repetición, el escáner oscila o se mueve hacia delante o hacia atrás en función del uso que se le vaya a dar.
- Ángulo de escaneo: se mide en grados, y es el ángulo que recorre el escáner desde un extremo hasta el otro. Este ángulo se puede ajustar en función de la aplicación que se le quiera dar.
- Rango de detección: es el intervalo de distancia que el sistema LiDAR puede detectar.
- Área de cubrimiento: es la zona que detecta el sistema LiDAR y depende del ángulo y del rango de detección.

La principal ventaja del sistema LIDAR frente al radar (mide distancias a través de ondas de radio), es que el sistema LIDAR tiene una mayor capacidad para reflejar imágenes, haciendo más objetos visibles.

3.2. Aplicaciones sistema LiDAR

El sistema LIDAR tiene su principal aplicación en la generación de Modelos Digitales del Terreno (MDT). Esencialmente consiste en un sensor láser instalado en una plataforma móvil (helicóptero o avión) que barre las superficies que se encuentran entre el sensor y el terreno obteniendo una

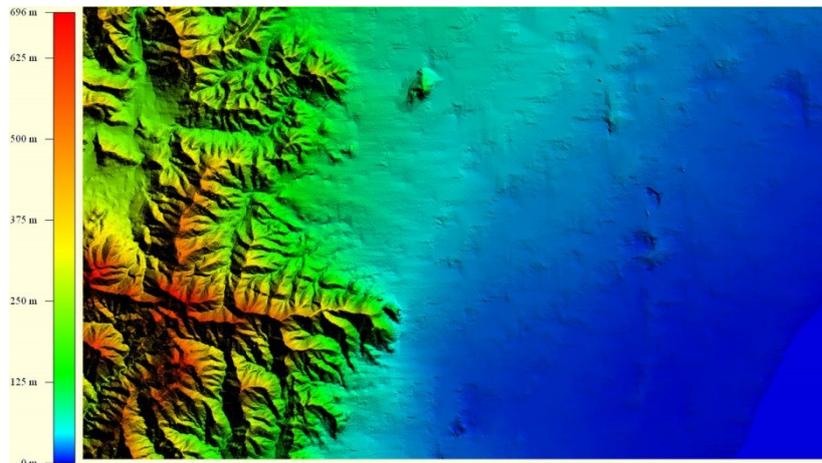
Capítulo 3: Sistema LIDAR

nube de puntos de coordenadas precisas. El equipo está compuesto por un sensor láser, un sistema inercial de navegación (INS) y un receptor GPS en una plataforma.

El INS tiene como objetivo orientar la plataforma tomando como punto fijo la antena GPS, el sistema está compuesto en general por tres acelerómetros y giróscopos.

El objetivo del GPS es determinar la posición de la plataforma. Para alcanzar las máximas precisiones en el avión se sitúa un receptor GPS de doble frecuencia, mientras que en tierra se establece una red que realiza una observación simultánea con el medio móvil.

El levantamiento LIDAR tiene ventajas sobre la captura con métodos convencionales: requiere de mínimo control geodésico en tierra, los datos tienen una mayor densidad y una mayor precisión.



Figuro 3.1: MDT generado mediante LIDAR

También se suele usar en los estudios de la atmósfera ya que permite detectar las partículas atmosféricas, como por ejemplo nubes, lluvia o humo. Además, es posible medir la velocidad del viento o la densidad de algunos componentes de nuestra atmósfera como oxígeno o nitrógeno.

Ahora vamos a ver algunas aplicaciones que se han realizado con un sistema LIDAR y qué están relacionadas en cierta medida con este proyecto.

El sistema LIDAR es utilizado en los vehículos autónomos, para realizar un reconocimiento del entorno o simplemente para detectar obstáculos y poder así evitarlos. Un claro ejemplo del uso del sistema LIDAR para percibir obstáculos es el robot guepardo desarrollado por Sangbae Kim, del Instituto Tecnológico de Massachusetts (MIT), en Cambridge, Estados Unidos.

Cuando el sistema LIDAR instalado a bordo del robot detecta un obstáculo en su trayectoria, se obtiene la altura y la distancia hasta dicho objeto. El robot entonces calcula la mejor posición desde la que saltar, y ajusta sus zancadas para aterrizar justo antes del obstáculo y ejercer suficiente fuerza para empujarse hacia arriba y pasar por encima del objeto. En función de la altura del obstáculo, el robot aplica una cierta cantidad de fuerza para aterrizar de forma segura sobre sus patas, antes de retomar su modo normal de correr.



Figura 3.2: Robot guepardo saltando un obstáculo.

3.3. LIDAR SICK LMS 111

El sensor que voy a utilizar en el desarrollo de este proyecto es el láser tipo SICK LMS 111, que garantiza un rendimiento fiable y es ideal para aplicaciones en interiores, ya que tiene una detección de objetos excelente y un ángulo de barrido de 270° de alta capacidad de detección. Además, puede cambiar dinámicamente los campos supervisados, tiene una amplia variedad de interfaces y cuenta con un dispositivo de ahorro de energía.



Figura 3.3: SICK LMS 111

Vamos a ver algunas de las características más importantes del láser SICK LMS 111:

- Rango de temperaturas del ambiente: puede operar a una temperatura mínima de -30 °C y a una temperatura máxima de 50 °C . Teniendo en cuenta que el presente proyecto se centra en entornos interiores, el rango de temperaturas es más que suficiente.
- Dimensiones: el láser tiene 162 mm de alto, 102 mm de ancho y 106 mm de profundidad.
- Color: gris (RAL 7032)
- Masa: aproximadamente tiene una masa de 1,1 kg, por lo que es adecuado para ir equipado en cualquier robot de cierta envergadura.
- Voltaje: 24 V. El sensor puede trabajar entre 10,8 y 30 V.

- Enclosure rating: IP 67.
- Conexiones: consta de 4 conectores plug-in redondos M12 y un M8.
- Máximo rango: 20 m, 18m (10% reflectividad). Para entornos interiores como puede ser un ambiente de oficinas, el rango de 18 metros debe de ser más que suficiente.
- Ángulo de escaneo: 270° con resolución angular ajustable entre 0,25 y 0,5 °. El rango del ángulo de escaneo y la resolución resultan idóneos para esta aplicación, ya que se pueden obtener hasta 1080 lecturas por barrido donde cada una indica la distancia y el ángulo al objeto más próximo en esa dirección.

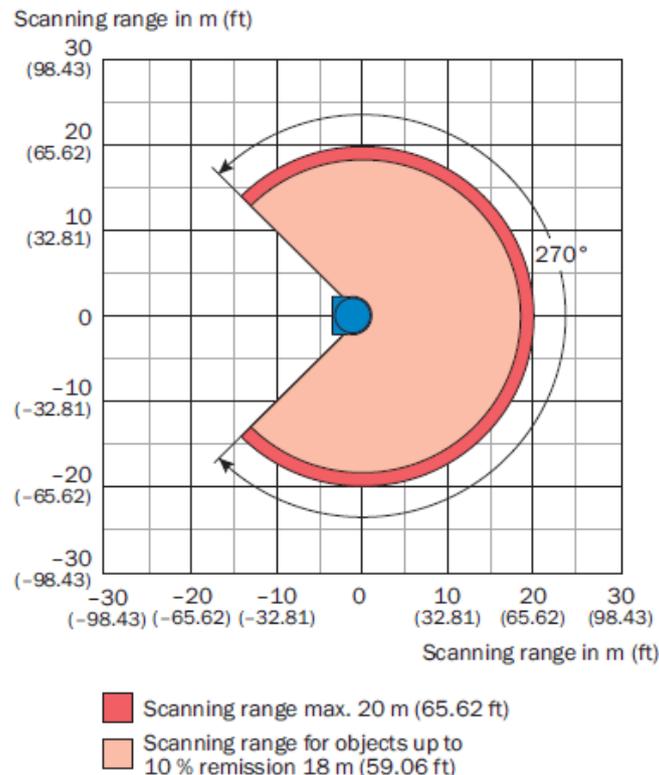


Figura 3.2: Rango de operación LMS111

- Frecuencia de exploración: entre 25 y 50 Hz.
- Tiempo de respuesta: entre 20 y 40 ms. Es un tiempo de respuesta muy bajo, idóneo para este tipo de proyectos.

3.4. Conexiones eléctricas del SICK LMS 111

Como hemos visto en las características de nuestro Láser, contamos con 4 conectores M12. En el siguiente esquema podemos observar estos cuatro conectores.

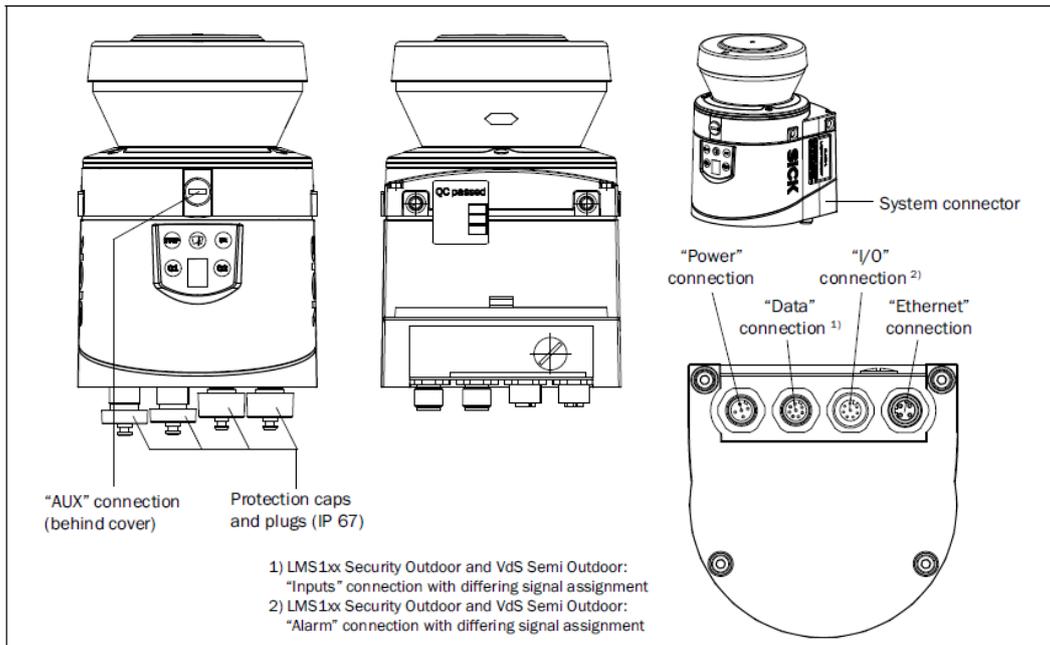


Figura 3.3: esquema de conexiones del Laser SICK LMS 111

En la siguiente imagen podemos observar la función y el color de cada uno de los cinco pines del conector denominado "Power" connection en la figura 3.3.

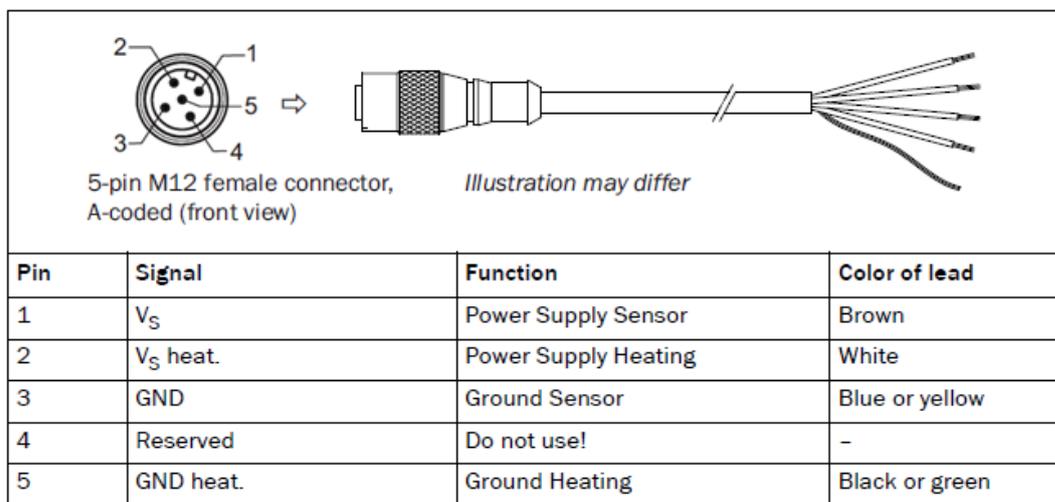


Figura 3.4: función y color de cada pin del conector de energía.

En la siguiente imagen podemos ver la función de cada uno de los 4 pines del conector "Ethernet".

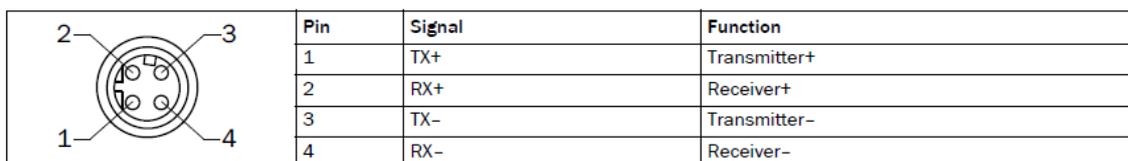


Figura 3.5: función de cada pin del conector Ethernet.

3.5. Instalación del SICK LMS 111

Para poder operar con el Láser SICK LMS 111, debemos contar con un PC que tenga las siguientes características:

- Mínimo 256 MB de RAM, se recomienda 512 MB.
- Interfaz de datos RS-322 o Ethernet.
- Sistema operativo: MS Windows 2000, XP o VISTA.
- Monitor con mínimo 256 colores, se recomienda 65536 colores. Resolución de pantalla mínimo 800 x 600.
- Disco duro: mínimo 170 MB libres.
- CD-ROM drive.

En primer lugar, abrimos el archivo "start.exe" que encontraremos en el CD que acompaña al SICK. Tras esto, se iniciará la instalación del programa SOPAS en nuestro PC. Una vez tengamos instalado el programa, debemos conectar correctamente el Láser tanto a una fuente de alimentación (10,8 V – 30 V), como al PC.

La conexión a la fuente de alimentación, se realizará con el conector M12 de 5 pines que hemos visto en el apartado anterior, el cable marrón al positivo y el azul al negativo de una fuente de alimentación. Mientras que la conexión al PC, se realiza con el conector M12 de 4 pines que también hemos visto en el apartado anterior ("Ethernet"), que se conecta al PC mediante una conexión RJ45.

Una vez instalado SOPAS, y conectado el Láser, abrimos el programa y buscamos nuestro dispositivo. En unos instantes lo tendremos conectado. Debemos registrarnos en el dispositivo para poder cambiar su configuración, para ello debemos de elegir un nivel de usuario de la siguiente tabla (extraída del manual de SICK) y poner la contraseña que tiene asignado ese usuario. En nuestro caso, el nivel de usuario es "Authorised client" y la contraseña "client", como podemos observar en la figura 3.6.

User level	Password
Maintenance personnel	main
Authorised client	client

Figura 3.6: nivel de usuario y contraseña.

Una vez registrados, podremos cambiar la configuración de nuestro láser, como por ejemplo la frecuencia, el ángulo de resolución, el ángulo de inicio y final del barrido del láser, etc. Cuando hemos configurado el Láser según nuestras necesidades debemos de tener a nuestra disposición los datos leídos. Para el presente proyecto no vamos a usar los datos en tiempo real, por lo que vamos a obtener datos de la función "grabador de datos" del Programa SOPAS. Para una accesibilidad a los datos a tiempo real, tendríamos que hacer uso de OPC server, o librerías con muchas funciones relacionadas con la localización de robot autónomos creadas en C++ (como por ejemplo MRTP) o C#.

Para grabar los datos haciendo uso de SOPAS, debemos pulsar en extras, que aparece en la barra de tareas. Al pulsar, se abrirá un desplegable, y en este desplegable debemos seleccionar grabador de datos. Hecho esto, debemos elegir el formato en el que guardar los datos y la carpeta donde guardarlos. Pulsando en configuración, se nos abrirá una ventana como la de la

figura 3.7, en la que podemos seleccionar el intervalo de grabación y el tiempo de grabación máximo. Para obtener los datos, debemos seleccionar del desplegable inferior el que pone ScanData en la columna “Denominación”.

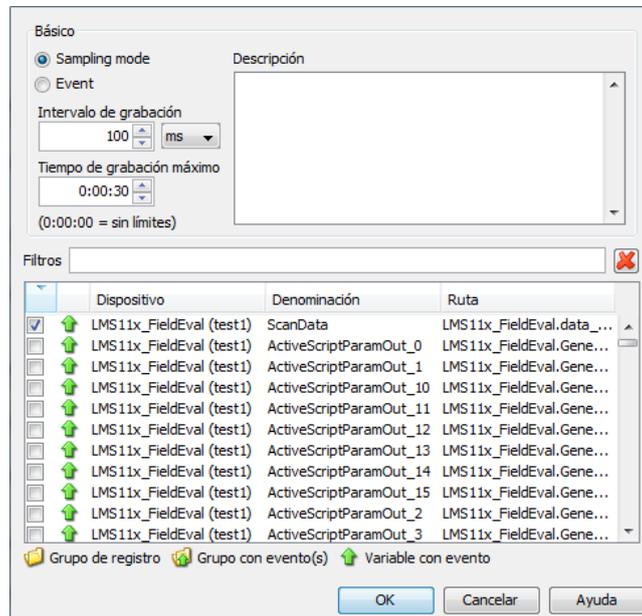


Figura 3.7: ventana de configuración del grabador de datos.

3.6. Toma de medidas del SICK

En este apartado vamos a ver como toma las medidas nuestro láser con la siguiente configuración:

- Ángulo de barrido: 270°.
- Resolución angular: 0,5°.
- Tiempo entre medidas: 1 segundo.
- Número de medidas: 5.

Para este experimento, las medidas fueron tomadas desde la ubicación marcada en la figura 3.8 en color rojo, apuntando hacia la derecha

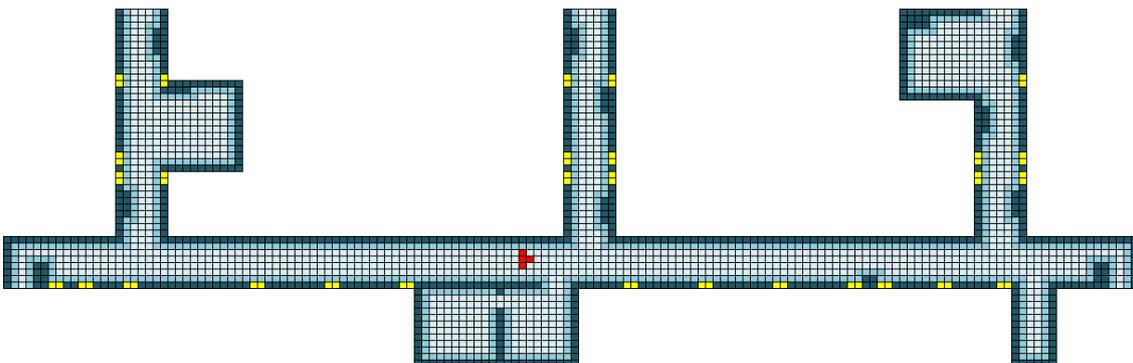


Figura 3.8: Ubicación del SICK

Las medidas del láser son extraídas en un excell a través del programa SOPCAS de la forma que dije en el capítulo 3, donde se recogen para cada tiempo, tantas medidas como hayamos configurado con el ángulo de barrido y la resolución angular. En este caso, se configuró con el máximo ángulo de barrido (270 °) y con resolución angular de 0,5 grados, de forma que en el excell aparecen 540 medidas. Estas 540 medidas se han tomado 5 veces, con un segundo de separación entre cada una de ellas. Haciendo una media entre las cinco y representado dicha media en coordenadas cartesianas obtenemos la figura 3.9, donde se puede observar el pasillo en el que está situado, una puerta cerrada (lo primero que encontramos a la derecha), una puerta abierta (lo segundo que nos encontramos a la derecha) y un cruce de un pasillo perpendicular (lo que se observa a la izquierda).

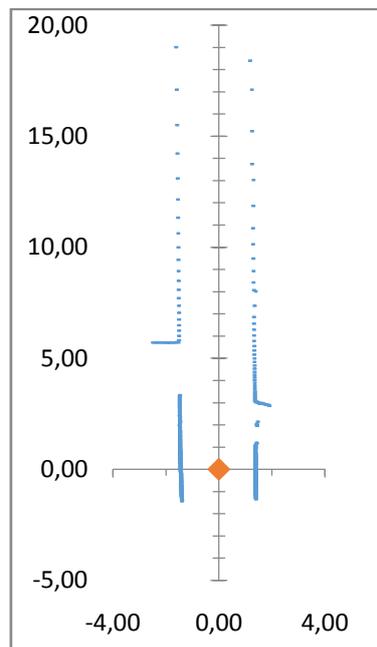


Figura 3.9: representación de las medidas del láser, siendo el rombo naranja la ubicación del láser, y los puntos azules cada una de las medidas

Ahora vamos a ver un poco más en detalle cada uno de los elementos que ha detectado.

3.6.1. Puerta cerrada

En primer lugar, las puertas cerradas se ven representadas como un hueco, esto es debido a que dichas puertas son translúcidas, por lo que el haz de luz atraviesa la puerta cambiando su dirección y no es detectado por nuestro láser. Las medidas que devuelve el láser cuando se encuentra con una puerta cerrada son las siguientes:

Y (m)	X (m)
0,98	1,33
1,00	1,33

1,02	1,33
1,04	1,33
1,06	1,33
1,07	1,33
1,12	1,35
1,16	1,38
1,18	1,38
1,20	1,38
0,00	0,00
...	...
0,00	0,00
1,96	1,40
1,97	1,38
2,01	1,38
2,04	1,38

Puerta cerrada

Como se puede observar, la X permanece constante (1,33) hasta que aumenta levemente (1,38) al detectar el marco de la puerta, justo antes de detectar el cristal de la puerta cerrada, que como he comentado anteriormente es 0, ya que no recibe el haz enviado. Desde la Y igual a 1,2 no se detecta nada, hasta que en la Y igual a 1,96 se vuelve a detectar el marco de la puerta. Cualquier puerta cerrada del sótano donde se van a realizar los experimentos se detectaría aproximadamente de esta forma.

3.6.2. Puerta abierta

Al estar la puerta abierta, el sensor detecta el marco de la puerta, hasta que deja de detectar marco y detecta en interior de la habitación. Debido a la distancia e inclinación a la que se encuentra de la puerta, en este caso lo único que detecta es la puerta abierta.

Y (m)	X (m)
1,96	1,40
1,97	1,38
2,01	1,38
2,04	1,38
2,14	1,42
2,86	1,86
2,88	1,84
2,89	1,80

2,90	1,78
2,91	1,75
2,92	1,72
2,93	1,69
2,94	1,66
2,95	1,63
2,96	1,61
2,97	1,58
2,99	1,55
2,94	1,50
2,96	1,47
2,97	1,45
3,00	1,43
3,03	1,41
3,01	1,37
3,04	1,35
3,04	1,32
3,08	1,31
3,16	1,31
3,24	1,31
3,32	1,31
3,41	1,31
3,50	1,31

Puerta abierta

Cuando la Y es igual a 2,14 se detecta el marco de la puerta por última vez antes de pasar a detectar la puerta abierta. Cuando se está detectando la puerta, vemos como la Y aumenta levemente debido a que la puerta está abierta algo menos de 90 grados con respecto al pasillo, mientras que la X va disminuyendo, hasta que llega un punto en el que acaba la puerta y la Y vuelve a aumentar manteniéndose constante la X, ya que se está detectando la pared del pasillo.

3.6.3. Cruce con pasillo

Un cruce con otro pasillo se detecta de la siguiente forma, mientras se detecta el pasillo en el que estamos, la X se mantiene constante y la Y va disminuyendo, hasta que llega un punto en el que la Y se mantiene constante y la X empieza a aumentar (en valor absoluto), ya que se está detectando el pasillo perpendicular al pasillo en el que estamos situados. Esto sigue así hasta que en un determinado punto ya no se detecta el pasillo que cruza, sino que se vuelve a detectar

Capítulo 3: Sistema LIDAR

la pared del pasillo en el que estamos situados, por lo que la X vuelve a mantenerse constante y la Y sigue disminuyendo.

Y (m)	X (m)
7,36	-1,56
7,04	-1,56
6,74	-1,56
6,48	-1,56
6,24	-1,56
6,01	-1,55
5,81	-1,56
5,71	-1,58
5,72	-1,64
5,71	-1,69
5,71	-1,74
5,71	-1,80
5,71	-1,86
5,71	-1,91
5,70	-1,96
5,71	-2,02
5,71	-2,08
5,71	-2,14
5,71	-2,19
5,70	-2,25
5,71	-2,31
5,71	-2,36
5,71	-2,42
5,70	-2,48
5,70	-2,54
3,34	-1,52
3,29	-1,53
3,22	-1,53
3,14	-1,53
3,07	-1,53
3,01	-1,53

Cruce con pasillo

CAPÍTULO 4

TÉCNICAS DE INFERENCIA PROBABILÍSTICA

4.1. Introducción

Como se comentó en el segundo capítulo del proyecto, la mayor dificultad se encuentra en conseguir que el robot sea capaz de ubicarse en el mapa sin tener información de donde se encuentra. El éxito de los algoritmos de los modelos clásicos depende del uso de sensores precisos y la obtención de modelos exactos. Pese a contar con estas condiciones, el éxito no está garantizado, ya que siempre existen errores e incertidumbres. Los principales factores que causan incertidumbre son:

- El ruido en los sensores: las medidas de los sensores son ruidosas y la distribución estadística de este ruido no es sencilla de modelar.
- El ruido en la detección de la posición: los movimientos del robot no suelen ser exactos, y además no se detectan de forma precisa mediante la odometría, como se comentó en el capítulo 2.
- Entornos complejos y dinámicos: se hace complicado mantener modelos consistentes en entornos interiores, ya que estos suelen ser complejos y dinámicos.

Ante esto, los enfoques probabilísticos están cobrando importancia, ya que se consiguen métodos más robustos. Esto se debe básicamente a que estos enfoques siguen funciones de probabilidad, lo que permite esta robustez ante las incertidumbres comentadas anteriormente. La principal desventaja de estos enfoques probabilísticos es la ineficiencia probabilística al tener que considerar todas las distribuciones de probabilidad del espacio de posiciones del robot.

El principio básico de todo algoritmo probabilístico es la regla de Bayes, así que vamos a explicar y deducir esa regla en el siguiente apartado.

4.2. Regla de Bayes

La regla de Bayes determina la probabilidad de una causa sabiendo el efecto que ha producido. Sea un espacio muestral E , compuesto de un determinado número de sucesos disjuntos A_i , de forma que:

$$E = \bigcup_{i=1}^n A_i$$

La probabilidad de que ocurra un suceso B perteneciente al espacio muestral E se puede expresar de la siguiente forma:

$$P(B) = P\left(\bigcup_{i=1}^n B \cap A_i\right) = \sum_{i=1}^n P(B \cap A_i) \quad (1)$$

Siendo $P(B \cap A_i)$ la probabilidad conjunta de A y B , es decir, es la probabilidad de que se cumpla tanto A como B .

Capítulo 4: Técnicas de inferencia estadística

Si reducimos el espacio muestral E a A, llegamos a la expresión (2), esto que acabamos de desarrollar es el teorema de la probabilidad total.

$$P(B) = \sum P(A_i) P(B | A_i) \quad (2)$$

Igualando (1) y (2):

$$P(B | A_i) = \frac{P(B \cap A_i)}{P(A_i)} \quad (3)$$

De forma análoga:

$$P(A_i | B) = \frac{P(A_i \cap B)}{P(B)} \quad (4)$$

Combinando (3) y (4) llegamos a la regla de Bayes:

$$P(B | A_i) P(A_i) = P(A_i | B) P(B)$$

$$P(A_i | B) = \frac{P(B|A_i) P(A_i)}{P(B)} \quad (5)$$

Siendo:

$P(A_i | B)$ = probabilidad a posteriori de que ocurra el evento A_i sabiendo que ha ocurrido B. También llamada posterior.

$P(A_i)$ = probabilidad a priori de que ocurra la causa A_i . También llamada prior.

$P(B | A_i)$ = probabilidad condicional de que ocurra el suceso B, dado que existe la causa A_i . También llamada función de verosimilitud.

En definitiva, con la regla de Bayes se provee una forma de computar la probabilidad a posteriori en función de la probabilidad condicional y la probabilidad a priori.

4.3. Inferencia Bayesiana

La incertidumbre es natural en el proceso de razonamiento donde se pueden establecer reglas para inferir de manera deductiva una proposición determinada que puede ser verdadera o falsa, dependiendo del límite de esta estimación. Dentro de los métodos de razonamiento se encuentran los Modelos Bayesianos, que simulan diferentes condiciones de incertidumbre cuando no se conoce si es verdadera o falsa la hipótesis enunciada en un rango de variación.

Todos los modelos bayesianos tienen en común la asignación de la probabilidad como medida de creencia de una hipótesis, por lo que, la inferencia es un proceso de reajuste de medidas de creencia al conocerse nuevos axiomas.

Cuando se utilizan evidencias y observaciones para establecer que una suposición sea cierta, es lo que se denomina como Inferencia Bayesiana. La inferencia bayesiana observa la evidencia y calcula un valor estimado según el grado de creencia planteado en la hipótesis. Esto implica que al tener mayor cantidad de datos disponibles se podrá obtener resultados más satisfactorios.

Siendo "H1" la hipótesis 1, "H2" la hipótesis 2 y "d" la información visual obtenida por el sensor, podemos determinar cuál es la hipótesis más probable haciendo uso de la regla de Bayes.

$$P(H_1 | d) = \frac{P(d|H_1) P(H_1)}{P(d)} \quad (6)$$

$$P(H_2 | d) = \frac{P(d|H_2) P(H_2)}{P(d)} \quad (7)$$

Capítulo 4: Técnicas de inferencia estadística

$P(d)$ es una constante de normalización, que se puede expresar como en la ecuación (8), como voy a demostrar a continuación:

$$P(H_1 | d) + P(H_2 | d) = 1$$

$$\frac{P(d|H_1)P(H_1)}{P(d)} + \frac{P(d|H_2)P(H_2)}{P(d)} = 1$$

$$P(d) = P(d | H_1) P(H_1) + P(d | H_2) P(H_2) \quad (8)$$

Sustituyendo $P(d)$ en las ecuaciones (6) y (7), llegamos a las siguientes expresiones:

$$P(H_1 | d) = \frac{P(d|H_1)P(H_1)}{P(d|H_1)P(H_1) + P(d|H_2)P(H_2)}$$

$$P(H_2 | d) = \frac{P(d|H_2)P(H_2)}{P(d|H_1)P(H_1) + P(d|H_2)P(H_2)}$$

Siendo $P(H_1 | d)$ y $P(H_2 | d)$ las probabilidades posteriores, es decir, las probabilidades de que las hipótesis sean ciertas dada la información d . Las probabilidades a priori de que se cumplan las hipótesis son $P(H_1)$ y $P(H_2)$. Mientras que las probabilidades condicionales o funciones de verosimilitud son $P(d | H_1)$ y $P(d | H_2)$ y expresan la probabilidad de que se de esa información d , sabiendo que la hipótesis es correcta. El denominador de las ecuaciones es simplemente una constante de normalización. La siguiente expresión de proporcionalidad expresa de forma general lo que acabo de desarrollar:

$$P(\text{hipótesis} | \text{información}) \propto P(\text{información} | \text{hipótesis}) P(\text{hipótesis})$$

$$\text{Probabilidad posterior} \propto \text{Función verosimilitud} \times \text{Probabilidad a priori}$$

4.4. Fusión sensorial mediante la regla de Bayes

La fusión de datos consiste en combinar varias fuentes de información sobre un mismo fenómeno, con el objetivo de tomar la mejor decisión con respecto a ese fenómeno. La fusión sensorial es necesaria para resolver el problema de localización local, ya que tenemos que combinar las medidas obtenidas mediante la odometría (x_1), con los datos aportados por los sensores exteroceptivos (x_2).

El fenómeno que vamos a evaluar es el evento A , de forma que cada sensor asigna una probabilidad al evento B para cierta medida x_1 . Esta medida es convertida en probabilidad mediante el modelo inverso del sensor $P(x_1 | B)$. Este modelo se relaciona con las características del sensor mediante la regla de Bayes:

$$P(A | x_1) = \frac{P(x_1|A)P(A)}{P(x_1)}$$

De forma análoga con los datos de los sensores exteroceptivos,

$$P(A | x_2) = \frac{P(x_2|A)P(A)}{P(x_2)}$$

Haciendo uso de la regla del producto, se obtiene la probabilidad de observación del evento A dado x_1 y x_2 :

Capítulo 4: Técnicas de inferencia estadística

$$P(A | x_1, x_2) = \frac{P(x_1|A, x_2) P(A|x_2)}{P(x_1|x_2)}$$

Si asumimos la independencia de las mediciones odométricas (x_1) de las provenientes de los sensores (x_2), la expresión queda de la forma:

$$P(A | x_1, x_2) = \frac{P(x_1|A) P(A|x_2)}{P(x_1)}$$

Aplicando nuevamente la regla de Bayes a $P(x_1 | A)$ y sustituyéndolo en la ecuación, obtenemos la densidad de probabilidad conjunta:

$$P(A | x_1, x_2) = \frac{P(A|x_1)P(A|x_2)}{P(A)}$$

Esta expresión representa los estimados y los valores de certidumbre del evento A a partir de la fusión de las medidas x_1 y x_2 .

4.5. Robótica probabilística

La robótica probabilística busca estimar la posición del robot y el modelo del entorno que lo rodea. En este apartado voy a denominar la posición del robot como $s = (x, y, \Theta)$ y el modelo del entorno que lo rodea como m . A la composición de la posición y del entorno se le va a denominar estado $x = (s, m)$. La evolución del estado (x), el control (u) y la sensorización (z) se puede modelizar mediante una red dinámica de Bayes.

Antes de seguir con el apartado, voy a hacer una breve explicación de lo que es una red dinámica de Bayes. Una red dinámica de Bayes es una red bayesiana que representa una secuencia de variables, es decir, una red bayesiana que se va modificando a lo largo del tiempo. En la figura 4.1 se puede encontrar una forma muy general de representar las redes bayesianas dinámicas. En la figura se representa en cada ventana la red para cada instante de tiempo, donde las flechas representan el flujo de información. De esta forma se tiene una red bayesiana que en cada instante de tiempo recibe información del instante anterior además de las variables observables.

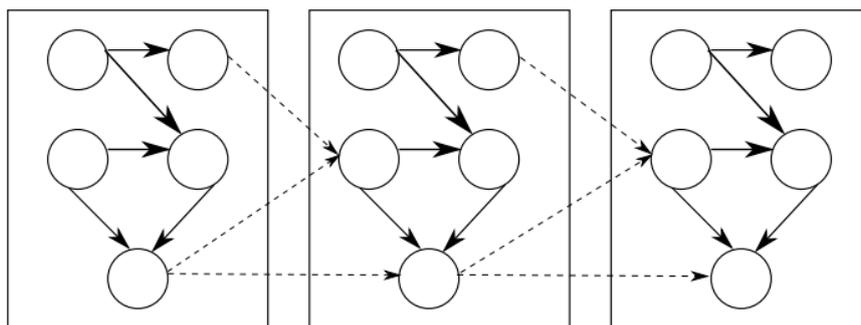


Figura 4.1: Ejemplo típico de una red dinámica de Bayes.

La ecuación de Bayes aplicada a la robótica móvil, se expresa como en la ecuación (9). Donde a es la variable que queremos inferir a partir de b , que serían los datos conocidos.

$$p(a | b) = \eta p(b | a) p(a) \tag{9}$$

Capítulo 4: Técnicas de inferencia estadística

Siendo $p(a)$ la probabilidad a priori, $p(a | b)$ la probabilidad a posteriori, $p(b | a)$ la función de verosimilitud, y η un factor normalizador para asegurar que la expresión se encuentre entre 0 y 1.

La red dinámica en nuestro caso es la expresada en la figura 4.2, donde se puede ver que el estado x_t depende estocásticamente del control u_t y del estado anterior x_{t-1} , mientras que las observaciones z_t , depende estocásticamente del estado x_t y del entorno.

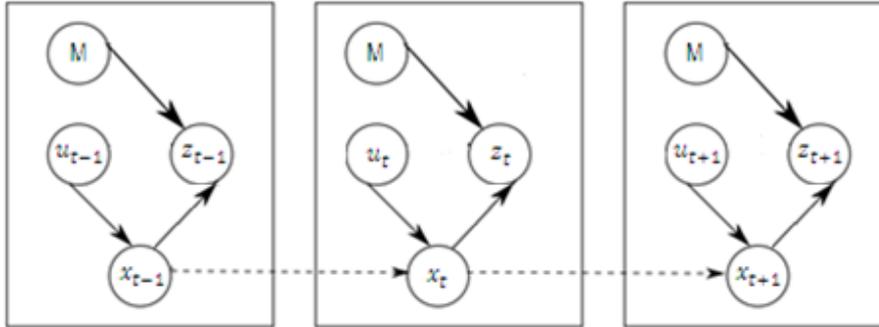


Figura 4.2: Red dinámica que caracteriza la evolución del estado x , la observación z , y el control u .

El filtro de Bayes extiende la regla de Bayes de la ecuación (9) para tratar problemas de estimación en el dominio del tiempo. Este filtro es un estimador recursivo que permite computar la secuencia de una distribución de probabilidad posterior que no puede ser observada directamente. De esta forma se puede calcular la probabilidad posterior sobre el estado x_t por medio de la ecuación (10) asumiendo que el entorno es estático y que el movimiento del robot es independiente del mapa:

$$p(s_t, m | z_t, u_t) = \eta p(z_t | s_t, m) \int p(s_t | u_t, s_{t-1}) p(s_{t-1}, m | z_{t-1}, u_{t-1}) ds_{t-1} \quad (10)$$

La distribución $p(s_t, m | z_t, u_t)$ se denota como $\text{Bel}(x_t)$, es conocida como la confianza o certidumbre del estado x_t y representa la probabilidad de que el sistema se encuentre en el estado x_t , dados las mediciones z_t y las acciones ejecutadas u_t . Además, la distribución $(s_{t-1}, m | z_{t-1}, u_{t-1})$ es la misma que $\text{Bel}(x_t)$, pero de un instante anterior, por lo que se le puede denominar $\text{Bel}(x_{t-1})$. Esto le da el carácter recursivo a este algoritmo.

$$\text{Bel}(x_t) = \eta p(z_t | x_t) \int p(x_t | u_t, x_{t-1}) \text{Bel}(x_{t-1})$$

Para implementar este estimador, se deben especificar dos densidades generativas: la densidad $p(z_t | x_t)$ conocida como el modelo de observación, y la densidad $p(x_t | u_t, x_{t-1})$ conocida como el modelo de movimiento. La densidad $p(z_t | x_t)$ describe en términos probabilísticos la manera como se generan las distintas observaciones z para diferentes localizaciones y mapas del entorno. La densidad $p(x_t | u_t, x_{t-1})$ describe cómo afecta el control u a la evolución de la posición del robot, dicho de otra forma, describe la probabilidad de que estando el robot en el estado x_{t-1} , alcance el estado x_t al aplicarle el control u_t .

4.6. Localización bayesiana o de Markov

El algoritmo de localización de Markov es un algoritmo probabilístico recursivo que permite estimar una variable dada en función de otras. Es decir, tenemos una variable que no se puede medir directamente (la posición del robot), pero que sí se puede inferir teniendo la información de otras variables (mediciones de sensores, movimientos ejecutados y el mapa del entorno). Este algoritmo se puede utilizar para muchísimas cosas, entre ellas, para estimar la posición de un robot.

La localización bayesiana o de Markov busca estimar la distribución posterior de probabilidad sobre el espacio de posiciones del robot condicionado a los datos disponibles. Esta función de probabilidad es conocida como certidumbre o confianza y se denomina $Bel(x)$.

La distribución de probabilidad que se pretende calcular se puede expresar como $p(x_t | d_{0:t}, M)$, donde $d_{0:t}$ es la secuencia de percepciones sensoriales hasta el instante t y M es un modelo del entorno. La posición del robot en el instante t , está formada por las coordenadas cartesianas del centro del robot (x, y) y la orientación θ de un eje del mismo en ese instante, $x_t = (x, y, \theta)$. Los posibles estados de robot x_t están representados por una variable aleatoria Denominada X .

Los datos sensoriales pueden ser divididos en dos tipos, los datos z que caracterizan la situación momentánea del robot y provienen de las mediciones del entorno de los sensores exteroceptivos y datos u , que están relacionados con el cambio de posición del robot tras realizar una acción y se miden a través de la odometría.

Entonces, los datos se pueden expresar como $d_{0:t} = (z_{0:t}, u_{0:t})$, donde $z_{0:t}$ es la secuencia histórica de observaciones del entorno hasta el instante t , y $u_{0:t}$ representa la secuencia de comandos dados al sistema o los datos odométricos hasta el instante t .

De modo que la distribución para estimar la posición del robot quedaría de la siguiente forma:

$$p(x_t | z_{0:t}, u_{0:t}, M)$$

La idea del algoritmo es que podemos estimar la posición en el instante t , sabiendo la posición del instante $t-1$, el movimiento ejecutado en el instante t y la medición del sensor del mismo instante.

La estimación de $Bel(x)$ para cada instante se hace de forma recursiva en base a la condición de Markov. Esta condición establece que el estado del robot en cualquier instante depende únicamente de la posición y el control del instante anterior, como podemos observar en la figura 4. Esto se puede expresar como:

$$Bel(x_t | x_1, \dots, x_{t-1}, u_1, \dots, u_{t-1}) = Bel(x_t | x_{t-1}, u_{t-1})$$

Siendo x_{t-1} el estado del instante anterior, y u_{t-1} la acción ejecutada en el instante anterior.

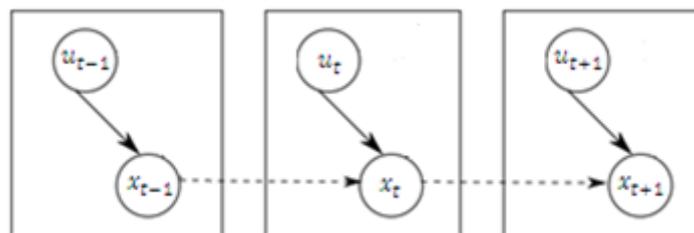


Figura 4.3: Red Bayesiana donde se ve que el estado x_t sólo depende el estado anterior x_{t-1} y de la acción ejecutada u_t

Capítulo 4: Técnicas de inferencia estadística

El algoritmo de correlación de Markov se realiza mediante dos pasos básicos: la predicción y la observación.

En la predicción se actualiza la confianza del instante anterior $Bel(x_{t-1})$ utilizando la información del movimiento ejecutado u_t . Esta información viene dada por el modelo de movimiento descrito por la función $p(x_t | u_t, x_{t-1})$, también conocido como modelo de propagación o de actuación. Este modelo se usa para incorporar el efecto de la acción realizada sobre una certidumbre previa denominada $Bel^-(x_t)$. La operación de actualización que se lleva a cabo en el paso de predicción no es más que una integral que tiene este aspecto:

$$Bel^-(x_t) = \int_{\Omega} p(x_t | u_t, x_{t-1}) Bel(x_{t-1}) dx_{t-1}$$

Siendo $Bel(x_{t-1})$ la confianza del instante anterior y Ω el espacio de las posibles localizaciones del robot. Hay que tener en cuenta que en un espacio discreto donde la posición sólo puede tomar unos valores dados como es nuestro caso, la integral se convierte en una simple suma. Aquí, para calcular la predicción $bel^-(x_t)$, es decir, la probabilidad de estar en la posición x en el instante t , estamos sumando (integrando) el modelo de movimiento para todas las posiciones posibles en el instante anterior ($t - 1$).

Necesitamos algo más que la predicción para poder localizarnos de forma fiable. Es aquí donde entra en juego el segundo paso del algoritmo, el paso de corrección. En este paso se recalcula la distribución de probabilidad para tomar en consideración los nuevos datos observados por los sensores (z_t). La observación z_t es incorporada a la función de localización mediante la regla de Bayes, la cual en general permite reducir la incertidumbre en lo que a la posición del robot se refiere.

$$Bel(x_t) = \frac{P(z_t | x_t) Bel^-(x_t)}{P(z_t | z_1, u_1, \dots, z_{t-1}, u_{t-1})}$$

El factor $P(z_t | x_t)$ que pondera la confianza previa, es el modelo de observación. El denominador corresponde a una constante de normalización que asegura que la densidad de probabilidad resultante $Bel^+(x_t)$ sea como máximo igual a 1.

Aplicando el teorema de la probabilidad total y la condición de Markov, la ecuación a la que llegamos con este modelo de localización bayesiano o de Markov es la siguiente:

$$Bel(x_t) = \eta p(z_t | x_t) Bel^-(x_t)$$

$$Bel(x_t) = \eta p(z_t | x_t) \int_{\Omega} p(x_t | u_{t-1}, x_{t-1}) Bel(x_{t-1}) dx_{t-1}$$

Donde $p(x_t | u_{t-1}, x_{t-1})$ es una generalización del modelo cinemático del robot que caracteriza el efecto de la acción sobre la posición del robot.

A pesar de que estas técnicas probabilísticas surgieron para dar respuesta al problema de la localización global, también han sido utilizadas para resolver el problema de la localización local, también conocida como seguimiento de la posición.

En la siguiente imagen podemos ver la secuencia típica en localización de markov:

Capítulo 4: Técnicas de inferencia estadística

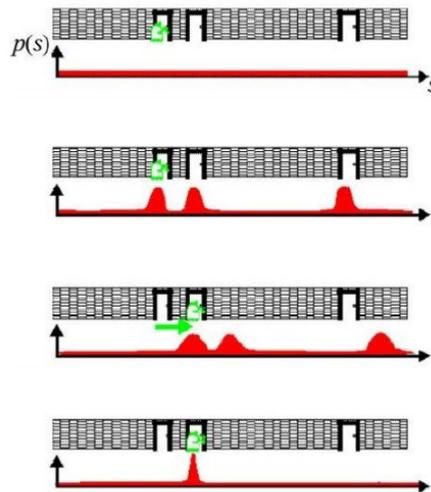


Figura 4.4: localización probabilística

En la primera fase, el robot desconoce su estado inicial. De forma que la distribución de la probabilidad o creencia del estado es uniforme. En la segunda fase, el robot detecta una puerta a través de sus sensores. Se utiliza el modelo de observación en conjunto con un mapa de puertas y pasillos, y se comprueba que existen tres zonas con igual probabilidad de ser la zona donde se encuentra el robot. En la tercera fase, el robot avanza aplicando el modelo de predicción. Se puede observar que las distribuciones de probabilidad se aplanan un poco debido al modelo de incertidumbre en el movimiento del robot. En la última fase, el robot vuelve a detectar con sus sensores la presencia de otra puerta. Esto hace que solo exista una zona posible donde poder encontrarse, finalizando así el proceso de localización global.

4.7. Aplicaciones Localización de Markov

Por muy simple que parezca, esta es la base del método de localización que usa el coche autónomo de Google. Lo que cambia en esencia es la forma de percibir y medir el mundo. El Google Car utiliza láseres de 3 dimensiones que le permiten construir modelos tridimensionales de su entorno. Los modelos de medición que utiliza son muy complejos y costosos de poner en marcha (por no hablar del precio de los sensores). Cabe resaltar que la idea básica de la localización está en el Filtro Bayesiano que acabamos de ver.

4.8. Fusión sensorial mediante filtro de Kalman

Aunque se puede usar la localización de Markov para resolver el problema de la localización local, el método que se suele usar es el filtro de Kalman. El filtro de Kalman es un excelente método recursivo para fusionar información proveniente de múltiples sensores o sistemas lineales. Normalmente se suele utilizar para fusionar la información obtenida por la odometría con la información proveniente de los sensores exteroceptivos.

El filtro de Kalman es básicamente un conjunto de ecuaciones matemáticas mediante las cuales se estima el estado de un sistema lineal en forma eficiente, a través de una solución recursiva del método de mínimos cuadrados. Estas ecuaciones permiten implementar un estimador lineal, insesgado y óptimo del estado anterior, y corrigiendo este estado estimado a partir de la

Capítulo 4: Técnicas de inferencia estadística

información disponible del estado anterior. Este estimador es óptimo ya que minimiza el error cuadrático medio del estado estimado.

Para usar este método, se estima que el estado es un proceso lineal afectado por un ruido gaussiano. Esto se puede expresar de la siguiente forma:

$$x_t = Ax_{t-1} + Bu_{t-1} + s_t \quad (11)$$

Donde se sigue la misma nomenclatura que en el apartado anterior, x_t es el estado actual,

x_{t-1} es el estado en el instante inmediatamente anterior, u_{t-1} es la acción ejecutada en el instante inmediatamente anterior, A y B son matrices de parámetros, y w_t es el ruido del sistema.

La siguiente ecuación es conocida como el modelo de medición. En ella se relaciona la medida del sensor z con el estado x a través de la matriz de parámetros H, y se incluye el ruido de la medición (m_t).

$$z_t = Hx_t + m_t \quad (12)$$

Las variables s y m se asume que no están correlacionadas y que tienen distribuciones de probabilidad normales o Gaussianas de media 0:

$$P(s) \simeq N(0, Q)$$

$$P(m) \simeq N(0, R)$$

Q representa la varianza del ruido en el sistema, mientras que R representa la varianza del ruido de la medición.

El proceso del filtro de Kalman al igual que el modelo de localización de Markov, se divide en dos subprocesos, el de predicción y el de medición, a partir de la estimación del valor previo de la variable fusionada x_{t-1} , se obtiene la predicción del instante siguiente. Además, con la medición sensorial, se genera un término de corrección proporcional al error de predicción. Este término que se genera se añade a la variable pronosticada para proporcionar la variable fusionada corregida en ese instante. Esto es expresado en la ecuación (13), donde K es conocida como la ganancia de Kalman y se determina a través de las matrices Q y R.

$$x_t^{corregida} = x_t^{pronosticada} + K(z_t - Hx_t^{pronosticada}) \quad (13)$$

La ganancia de Kalman se calcula mediante la siguiente ecuación:

$$K = \frac{Q}{Q+R} \quad (14)$$

Analizando la ecuación (14), podemos darnos cuenta que si el ruido de la medición (R) es mucho mayor que el ruido del sistema (Q), K será próxima a cero, lo que implicará que en la ecuación (13), el término que acompaña a K, sea próximo a cero, y solo se tenga en cuenta el valor pronosticado para obtener el valor fusionado corregido. Mientras que si el ruido del sistema (Q) es mucho mayor que el ruido de la medición (R), K será próxima a 1, y el término que acompaña a la k en la ecuación (13), tendrá mayor importancia.

CAPÍTULO 5

IMPLEMENTACIÓN DE UN SISTEMA DE AUTOLOCALIZACIÓN MEDIANTE MARKOV

5.1. Introducción

Como hemos visto en el apartado anterior, la localización de Markov es un excelente método para el tema de la localización de robots autónomos. En este capítulo vamos a explicar la técnica con la que implementar este tipo de localización. Esta técnica varía dependiendo de si tenemos un espacio discreto o continuo, de modo que si tenemos un espacio continuo usaremos métodos que representen la certidumbre continuamente (enfoque métrico), mientras que si tenemos un espacio discreto, usaremos métodos que representen la certidumbre discretamente (enfoque topológico). En este proyecto se va a usar un enfoque métrico, así que voy a centrarme en este de aquí en adelante.

5.2. Enfoque métrico

En un enfoque métrico, la posición del robot x_t está definida por las coordenadas cartesianas del centro del robot y la orientación Θ de un eje del mismo, $x_t = (x, y, \Theta)$. Las acciones son movimientos métricos que para realizar este experimento se realizarán manualmente y serán perfectamente conocidos aunque se le asignará un certidumbre, pero que en una aplicación real provendrían de las mediciones odométricas, y las observaciones son las mediciones realizadas con el láser SICK LMS 111. El mapa donde se van a realizar los experimentos es en el sótano de la Universidad. Las celdas amarillas son las puertas, las celdas de color azul más oscuro son las paredes u objetos, las de color el azul con un tono intermedio representa las celdas libres pero próximas a objetos, y las celdas de color azul más claro son las celdas libres que no tienen ninguna pared u objeto en su proximidad

Capítulo 5: Implementación de un sistema de autolocalización mediante Markov

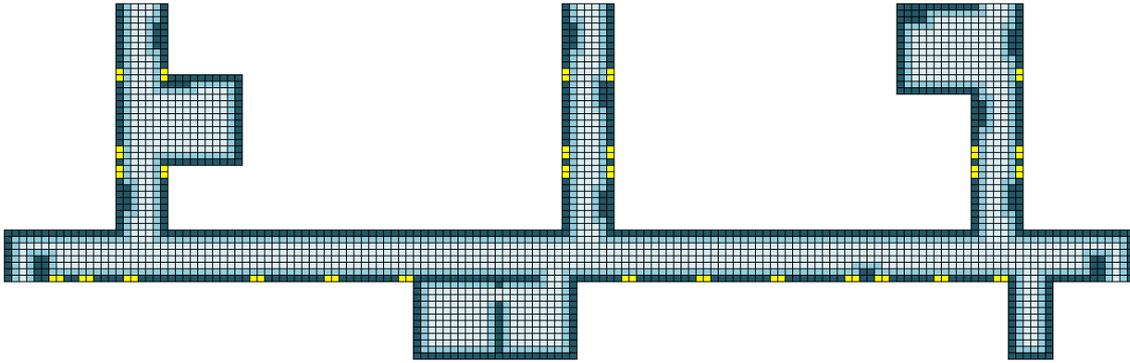


Figura 5.1: plano del sótano dividido en celdas de 0,5 x 0,5 m

5.3. Implementación de localización de Markov

En la localización de Markov, en lugar de tener una única hipótesis, se tiene una creencia $Bel(x_t)$, que es una distribución de probabilidad sobre todas las posibles posiciones del robot. Esta creencia, como vimos en el capítulo 4, se define de la siguiente forma:

$$Bel(x_t) = \eta p(z_t | x_t) \int_{\Omega} p(x_t | u_{t-1}, x_{t-1}) Bel(x_{t-1}) dx_{t-1}$$

En la ecuación anterior se utiliza un modelo de movimiento $p(x_t | u_{t-1}, x_{t-1})$, que expresa la probabilidad de llegar a la posición x_t , dada la posición x_{t-1} y tras realizar la acción u_{t-1} . También se usa un modelo de percepción $p(z_t | x_t)$, que expresa la probabilidad de que una cierta percepción z_t sea obtenida en la posición x_t .

Por lo que para llevar a cabo el cálculo de la creencia hay que definir una serie de elementos:

- El conjunto finito de estados en los que se puede encontrar el robot.
- La creencia $Bel(x_t)$ sobre la posición inicial del robot.
- El conjunto finito de acciones que puede realizar el robot.
- Función de transición de datos que determina la probabilidad $p(x_t | u_{t-1}, x_{t-1})$.
- El conjunto finito de observaciones que el robot puede recibir del entorno y el modelo de observación.

5.3.1. Conjunto finito de estados

Como hemos comentado al principio de este capítulo, se ha representado el entorno del sótano como una rejilla de ocupación. El mapa está representarlo con celdas de 0,5 x 0,5 m, porque las medidas del sótano se adaptan casi de forma perfecta con este tipo de celdas. Pero para simplificar el experimento se ha decidido representar los posibles estados del robot en celdas de 1 x 1 m, dejando únicamente las 10 celdas de color lila como posibles estados iniciales del robot.

Capítulo 5: Implementación de un sistema de autolocalización mediante Markov

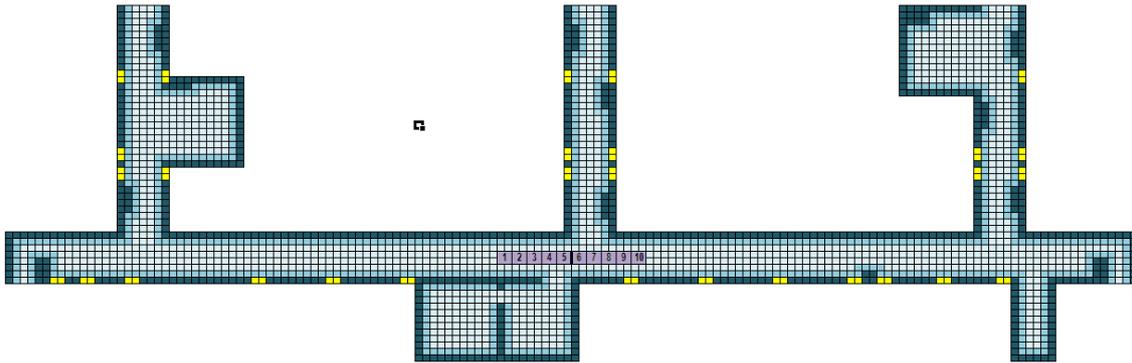


Figura 5.2: Mapa del sótano (celdas 0,5 x 0,5 m) y posibles estados iniciales del robot (celdas de 1 x 1 m)

5.3.2. Creencia $Bel(x_t)$

En un primer momento se propuso que la posición del robot iba a ser totalmente desconocida. De esta forma, el robot tendría 1926 estados iniciales posibles en los que encontrarse, por lo que la creencia sería:

$$Bel(x_0) = \frac{1}{\text{número de estados}} = \frac{1}{1926} = 0,000519$$

Finalmente se decidió que únicamente hubiera 10 estados iniciales posibles, de modo que la creencia inicial es:

$$Bel(x_0) = \frac{1}{\text{número de estados}} = \frac{1}{10} = 0,1$$

5.3.3. Conjunto finito de acciones

Las acciones son comportamientos que permiten al robot desplazarse de un estado a otro. Para nuestro experimento, con las acciones de avanzar y de retroceder es suficiente. Como sabemos, el movimiento realizado tras ejecutar la acción no siempre es el deseado, por lo que hay que tener en cuenta estas incertidumbres en un modelo probabilístico.

Este modelo probabilístico se muestra en la figura 5.3, donde se pueden observar los diferentes resultados posibles tras ejecutar una acción, y a cada uno de estos resultados, se les ha asociado un probabilidad de que ocurran.

	No moverse	Moverse una celda	Moverse dos celdas
Avanzar una celda	10%	80%	10%
Retroceder una celda	10%	80%	10%

Figura 5.3: modelos probabilístico para las acciones

Estos valores pueden ser obtenidos de forma empírica o mediante el aprendizaje, pero se ha demostrado que el sistema no es tan sensible al valor exacto de estas probabilidades, como al hecho de que sean o no nulas.

Capítulo 5: Implementación de un sistema de autolocalización mediante Markov

5.3.4. Función de transición de estados

Conociendo las transacciones ideales y el modelo probabilístico de las acciones, se genera el modelo de movimiento $p(x_t | u_{t-1}, x_{t-1})$. Para tener más accesible estas probabilidades, se ha generado una matriz cuadrada de probabilidades $p(x_t | x_{t-1})$ para cada una de las acciones. Las filas representan el estado de origen (x_{t-1}) y las columnas el estado de destino (x_t). Estas matrices se pueden observar en las figuras 5.4 (acción avanzar) y 5.5 (acción retroceder).

x_{t-1} / x_t	1	2	3	4	5	6	7	8	9	10
1	10	80	10	0	0	0	0	0	0	0
2	0	10	80	10	0	0	0	0	0	0
3	0	0	10	80	10	0	0	0	0	0
4	0	0	0	10	80	10	0	0	0	0
5	0	0	0	0	10	80	10	0	0	0
6	0	0	0	0	0	10	80	10	0	0
7	0	0	0	0	0	0	10	80	10	0
8	0	0	0	0	0	0	0	10	80	10
9	0	0	0	0	0	0	0	0	10	80
10	0	0	0	0	0	0	0	0	0	10

Figura 5.4: Matriz de transacción de la acción avanzar.

x_{t-1} / x_t	1	2	3	4	5	6	7	8	9	10
1	10	0	0	0	0	0	0	0	0	0
2	80	10	0	0	0	0	0	0	0	0
3	10	80	10	0	0	0	0	0	0	0
4	0	10	80	10	0	0	0	0	0	0
5	0	0	10	80	10	0	0	0	0	0
6	0	0	0	10	80	10	0	0	0	0
7	0	0	0	0	10	80	10	0	0	0
8	0	0	0	0	0	10	80	10	0	0
9	0	0	0	0	0	0	10	80	10	0
10	0	0	0	0	0	0	0	10	80	10

Figura 5.5: Matriz de transacción de la acción retroceder

Capítulo 5: Implementación de un sistema de autolocalización mediante Markov

5.3.5 Conjunto finito de observaciones y modelo de observación

Las observaciones corresponden con las distintas percepciones del entorno que el robot puede realizar a través de sus sensores. En este caso, el sensor que se va a utilizar es el láser SICK LMS 111, como se explicó en el capítulo 3. El modelo consiste en tomar y almacenar las medidas obtenidas previamente en cada uno de los estados posibles como distribuciones gaussianas, y comparar las medidas que toma el robot en un estado desconocido con las estas distribuciones a priori, para conseguir la distribución posterior $P(z_t|x_t)$.

CAPÍTULO 6

CONCLUSIONES Y FUTUROS PROYECTOS

6.1. Conclusiones

Como conclusiones, pienso que el presente trabajo ha sido muy teórico debido a que es un proyecto iniciador, que se ha centrado sobre todo en buscar y sintetizar información sobre el campo de los vehículos autónomos y las distintas técnicas de localización. Aún así, se ha realizado un pequeño experimento para mostrar cómo se implementa un modelo de autolocalización mediante Markov.

6.2. Futuros proyectos

En cuanto a futuros proyectos, sería interesante conseguir implementar este método de localización de Markov procesando los datos a tiempo real, con ayuda del software OPC server (como se comentó en el capítulo 3). También sería interesante contar con un robot o automóvil con el que poder determinar un modelo de movimiento real. En definitiva, seguir ampliando y mejorando el modelo.

BIBLIOGRAFÍA

HERNÁNDEZ ALAMILLA, S. F. (2005). *NAVEGACIÓN DE UN ROBOT MÓVIL EN AMBIENTES INTERIORES URBANOS*. TESIS. MONTERREY: INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY.

MICROSOFT RESEARCH. *INFER.NET*.

<[HTTP://RESEARCH.MICROSOFT.COM/EN-US/UM/CAMBRIDGE/PROJECTS/INFERNET/](http://research.microsoft.com/en-us/um/cambridge/projects/infernet/)>

NAVARRO GARCÍA, D. A. (2009). *CONTRIBUCIÓN A LA AUTOLOCALIZACIÓN DE ROBOTS MÓVILES BASADA EN LA FUSIÓN DE INFORMACIÓN MULTISENSORIAL*. TESIS. VALENCIA: UNIVERSIDAD POLITÉCNICA DE VALENCIA

SOLOVEY, G. (2014). "EL CEREBRO ESTADÍSTICO" ESCUELA DE MODELADO EN NEUROCIENCIAS, INSTITUTO BALSEIRO - CENTRO ATÓMICO BARILOCHE.

WINN, J. Y MINKA, T. (2009). "PROBABILISTIC PROGRAMMING" MACHINE LEARNING SUMMER SCHOOL