



Projecte Final de Carrera

Enginyeria Tècnica de Telecomunicacions.

Especialitat en sistemes Electrònics

Mapeo con robot móvil:
Caracterización y modelado

Jorge Martín Pérez

Director: Joan Oliver i Malagelada

Departament de Microelectrònica i Sistemes Electrònics

Escola d'Enginyeria
Universitat Autònoma de Barcelona (UAB)

Febrero 2012

13 de febrero de 2012



El sotasignat, Joan Oliver i Malagelada, Professor de l'Escola Tècnica Superior d'Enginyeria (ETSE) de la Universitat Autònoma de Barcelona (UAB),

CERTIFICA:

Que el projecte presentat en aquesta memòria de Projecte Fi de Carrera ha estat realitzat sota la seva direcció per l'alumne Jose López Morales.

I, perquè consti a tots els efectes, signa el present certificat.

Bellaterra, 13 de Febrer de 2012

Signatura: Joan Oliver i Malagelada

AGRADECIMIENTOS

Me gustaría mostrarle a mi familia, en especial a mi mujer, mi agradecimiento por el soporte y comprensión que me han demostrado durante la elaboración de este proyecto.

A mi compañero Jonathan García, darle mi gratitud por haberme acompañado en esta aventura y decirle que a sido un placer trabajar a su lado.

Dar las gracias a nuestro tutor Joan Oliver por habernos recibido siempre con una sonrisa en la cara y hacer un verdadero placer trabajar a su lado. Cuando a una persona le gusta lo que hace es capaz de transmitir su energía a los que le rodean.

ÍNDICE

Capítulo 1. INTRODUCCIÓN	8
1.1. OBJETIVOS.....	10
1.2. PLANIFICACIÓN DEL PROYECTO	11
Capítulo 2. ROBÓTICA	16
2.1. ANTECEDENTES HISTÓRICOS [1].	16
2.2. ROBOT MÓVIL [3]......	18
2.3. ROBÓTICA SOCIAL. LA NUEVA GENERACIÓN[4][5].	22
Capítulo 3. MODELOS DE MEDICIÓN. CARACTERIZACIÓN.	25
3.1. MODELO DE DETECCIÓN POR RANGO DE HAZ.	26
3.1.1. AJUSTE DE LOS PARÁMETROS INTRÍNSECOS DEL MODELO.....	31
3.2. MODELO POR RANGO DE PROBABILIDAD DEL CAMPO.	34
Capítulo 4. CINEMÁTICA DEL ROBOT Y ODOMETRÍA	37
4.1. ODOMETRÍA[7].....	37
4.2. CINEMÁTICA EN ROBOTS MÓVILES CON RUEDAS [3] [8] [9]	38
4.3. CINEMÁTICA DIFERENCIAL. MODELO MATEMÁTICO	40
4.4. VARIACIÓN DEL MODELO DIFERENCIAL.....	41
Capítulo 5. MAPEO [12].	42
5.1. ALGORITMIA Y REPRESENTACIÓN DE MAPAS [10][18].....	43
Capítulo 6. CARACTERIZACIÓN DE LOS SENSORES.	48
6.1. GRÁFICAS PROBABILÍSTICAS.	48
6.2. ESTUDIO DE LOS SENSORES CUANDO EL ROBOT ESTÁ PARADO.....	54
6.3. ESTUDIO DE LOS SENSORES. ROBOT ESTÁ EN MOVIMIENTO.....	61
6.4. ESTUDIO DEL SENSOR IR DE NUEVA INCORPORACIÓN.	64
Capítulo 7. RECORRIDO Y MAPEO	67

7.1. RECORRIDO	68
7.2. MAPEO	72
Capítulo 8. CONCLUSIONES	82
Capítulo 9. BIBLIOGRAFÍA:	85

ÍNDICE DE ILUSTRACIONES

Ilustración 1: Hombre de palo de Juanelo Turriano.....	17
Ilustración 2: Robot industrial de la marca alemana KUKA	19
Ilustración 3: Walter elaborando un robot móvil en los años 40	20
Ilustración 4: Parámetros para odometría.....	21
Ilustración 5: Robot ASIMO de Honda [4].....	23
Ilustración 6: Robonaut2 construido por la NASA [5].....	24
Ilustración 7: Componentes del modelo de medición. Imagen extraída de [6].....	26
Ilustración 8: Combinación lineal de densidades. Imagen extraída de [6]	30
Ilustración 9: Algoritmo "Beam_range_finder_model" [6].....	31
Ilustración 10: Mediciones típicas de sensores de ultrasonido y láser. Imagen extraída de [6]	32
Ilustración 11: Algoritmo para estimación de máxima verosimilitud [6].....	33
Ilustración 12: Distribuciones de "Likelihood field range finder model" [6]	35
Ilustración 13: Algoritmo "Likelihood_field_range_finder_model" [6].	36
Ilustración 14: Ejemplo de cinemática diferencial	38
Ilustración 15: Modelo síncrono	38
Ilustración 16: Modelo triciclo	39
Ilustración 17: Modelo Ackerman.....	39
Ilustración 18: Modelo omnidireccional	40
Ilustración 19: Mapa de rejilla de ocupación.....	46
Ilustración 20: Mapas topológicos	46
Ilustración 21: Mapa vector de percepción general	47
Ilustración 22: Medidas del ultrasonido	Ilustración 23: Robot en movimiento a 20 cm ... 49
Ilustración 24: Robot en movimiento a 30 cm	Ilustración 25: Robot en movimiento a 40 cm ... 50
Ilustración 26: Robot en movimiento a 50 cm	Ilustración 27: Robot en movimiento a 60 cm ... 50

Ilustración 28: Medidas del infrarrojos	Ilustración 29: Robot en movimiento a 20 cm... 51
Ilustración 30: Robot en movimiento a 30 cm	Ilustración 31: Robot en movimiento a 40 cm ... 51
Ilustración 32: Robot en movimiento a 50 cm	Ilustración 33: Robot en movimiento a 60 cm ... 51
Ilustración 34: Medidas del ultrasonido	Ilustración 35: Robot en movimiento a 20 cm ... 52
Ilustración 36: Robot en movimiento a 30 cm	Ilustración 37: Robot en movimiento a 40 cm ... 52
Ilustración 38: Robot en movimiento a 50 cm	Ilustración 39: Robot en movimiento a 60 cm ... 52
Ilustración 40: Medidas del infrarrojos	Ilustración 41: Robot en movimiento a 20 cm.... 53
Ilustración 42: Robot en movimiento a 30 cm	Ilustración 43: Robot en movimiento a 40 cm ... 53
Ilustración 44: Robot en movimiento a 50 cm	Ilustración 45: Robot en movimiento a 60 cm ... 53
Ilustración 46: Robot parado a 35 cm	Ilustración 47: Robot parado a 36 cm 55
Ilustración 48: Robot parado a 37 cm	Ilustración 49: Robot parado a 38 cm 55
Ilustración 50: Robot parado a 39 cm	Ilustración 51: Robot parado a 40 cm 56
Ilustración 52: Robot parado a 41 cm	Ilustración 53: Robot parado a 42 cm 56
Ilustración 54: Robot parado a 43 cm	Ilustración 55: Robot parado a 44 cm 56
Ilustración 56: Robot parado a 45 cm..... 57	
Ilustración 57: Valores proporcionados por el IR a distancias concretas 57	
Ilustración 58: Comportamiento del sensor IR 58	
Ilustración 59: Robot parado a 35 cm	Ilustración 60: Robot parado a 36 cm 58
Ilustración 61: Robot parado a 37 cm	Ilustración 62: Robot parado a 38 cm 59
Ilustración 63: Robot parado a 39 cm	Ilustración 64: Robot parado a 40 cm 59
Ilustración 65: Robot parado a 41 cm	Ilustración 66: Robot parado a 42 cm 59
Ilustración 67: Robot parado a 43 cm	Ilustración 68: Robot parado a 44 cm 60
Ilustración 69: Robot parado a 45 cm..... 60	
Ilustración 70: Valores del ultrasonido a distancias concretas..... 61	
Ilustración 71: Robot en movimiento a 30 cm	Ilustración 72: Robot en movimiento a 35 cm... 61
Ilustración 73: Robot en movimiento a 40 cm	Ilustración 74: Robot en movimiento a 45 cm... 62
Ilustración 75: Robot en movimiento a 50 cm..... 62	
Ilustración 76: Robot en movimiento a 30 cm	Ilustración 77: Robot en movimiento a 35 cm... 62
Ilustración 78: Robot en movimiento a 40 cm	Ilustración 79: Robot en movimiento a 45 cm... 63
Ilustración 80: Robot en movimiento a 50 cm..... 63	
Ilustración 81: Robot parado a 35 cm	Ilustración 82: Robot parado a 36 cm 64
Ilustración 83: Robot parado a 37 cm	Ilustración 84: Robot parado a 38 cm 64
Ilustración 85: Robot parado a 39 cm	Ilustración 86: Robot parado a 40 cm 65
Ilustración 87: Robot parado a 41 cm	Ilustración 88: Robot parado a 42 cm 65

Ilustración 89: Robot parado a 43 cm	Ilustración 90: Robot parado a 44 cm	65
Ilustración 91: Robot parado a 50 cm		66
Ilustración 92: Traslación de coordenadas		68
Ilustración 93: Primeras pruebas, graficando recorrido robot		69
Ilustración 94: Diagrama de flujo del algoritmo para el recorrido		70
Ilustración 95: Recorrido y traslación		71
Ilustración 96: Gráfica del recorrido del robot		71
Ilustración 97: Primer intento de mapeo		72
Ilustración 98: Resultado erróneo de mapeo		73
Ilustración 99: Operaciones para mapeo		73
Ilustración 100: Recorrido, traslación y mapeo		74
Ilustración 101: Diagrama de flujo. Recorrido y Mapeo		75
Ilustración 102: Mapeo 1		76
Ilustración 103: Comprobación correcto mapeo de esquinas		77
Ilustración 104: Mapeo completo de un comedor		77
Ilustración 105: Estrado de una clase de la ETSE		78
Ilustración 106: Recorrido y mapeo del estrado de una clase		79
Ilustración 107: Improvisación de un entorno cerrado		79
Ilustración 108: Recorrido y mapeo (1) en entorno cerrado		80
Ilustración 109: Recorrido y mapeo (2) en entorno cerrado		81

Capítulo 1. INTRODUCCIÓN.

Una labor en la que actualmente se dedican muchos esfuerzos de investigación es la de realizar mapas con robots móviles. Al ser el robot el que realiza el mapa, permite su elaboración en todo tipo de entornos, incluso en los que el ser humano no puede acceder, tales como cuevas, zonas radioactivas, e incluso siendo capaces de realizar un mapa en Marte. Las técnicas que se utilizan para la realización de mapas, aparte de tener solamente esa finalidad, se hacen indispensables para dar al robot los conocimientos de su posición y entorno. Estos, con el tratamiento adecuado aportaran al robot una mayor autonomía, aunque su fin sea otro.

La realización de mapas con robots móviles no es una tarea sencilla. Implica la construcción de un robot con todos los elementos sensoriales y actuadores necesarios para captar el entorno en el que se mueve. Por otro lado, tampoco son sencillos los algoritmos que determinan tanto su navegación como la detección del entorno. El tratamiento de los datos obtenidos tampoco es fácil, es por eso que se suelen utilizar algoritmos probabilísticos con este fin.

Dado que las dimensiones del proyecto son elevadas, se ha dividido el trabajo a realizar en dos tareas, una más técnica que abarca el montaje, calibración de

componentes y obtención de datos, y una segunda destinada a la caracterización de componentes, modelado del recorrido y entorno.

En esta memoria, se muestra más la parte de tratamiento de datos y en una segunda memoria, "*mapeo con robot móvil: construcción y seguimiento*", la parte de calibración y test. Aún así, y como el trabajo ha sido realizado de una modo muy cooperativo, se ha hecho difícil realizar un buen fraccionamiento de las dos memorias. Debido a esto se encontraran capítulos similares en ambas.

- **Distribución de la memoria**

A partir de este capítulo 1, donde se encuentra una pequeña introducción, y se explican tanto los objetivos planteados, como la planificación del proyecto, se irán exponiendo los capítulos siguientes:

- **Capítulo 2:** en él se localizan una serie de puntos sobre el mundo de la robótica, pasando desde sus inicios hasta la actualidad.
- **Capítulo 3:** se expone la teoría utilizada para conseguir realizar los modelos de medición, utilizados para realizar una caracterización de los sensores.
- **Capítulo 4:** introducción a la cinemática del robot, en el que se incluye el modelo matemático utilizado para el caso particular del robot construido.
- **Capítulo 5:** teoría sobre la creación de mapas y de los algoritmos más utilizados, donde se realiza una comparativa entre todos ellos.
- **Capítulos 6,7:** engloban toda la realización práctica del proyecto, tanto la caracterización de los sensores como las pruebas realizadas para realizar el mapeo.
- **Capítulo 8 y 9:** conclusiones y bibliografía.

1.1.OBJETIVOS.

El objetivo principal de este proyecto consiste en la realización de un robot autónomo, dotado de los componentes necesarios con los que poder realizar el mapeo de una habitación o recinto, sin necesidad de ser guiado ni corregido por acciones externas.

Para conseguir el objetivo principal se tendrán que usar distintos tipos de sensores/actuadores, de los cuales se tendrá que realizar un estudio para determinar su funcionamiento, consiguiendo observar in situ los conocimientos adquiridos en estos años de universidad. Gracias a los sensores se podrá realizar un mapeo con los resultados obtenidos y de esa forma poder representarlos de forma gráfica. Por lo tanto, para realizar el objetivo principal se tendrán que llevar a cabo objetivos secundarios.

Uno de estos objetivos secundarios, será el de adquirir conocimientos de los componentes disponibles en el mercado, y mediante su análisis poder llegar a determinar la mejor elección. Además se tendrá que desarrollar e implementar los drivers de control para dichos componentes, ya que por sí solos carecen de utilidad.

Otro objetivo secundario consistirá en el aprendizaje de los distintos programas que se necesitarán para la programación de los componentes o de la comunicación, así como para la realización de gráficos del espacio mapeado. Se tendrá que elaborar un software de comunicación con el robot, que permitirá ejecutar acciones y extraer los datos que se usarán para realizar el objetivo principal, es decir, el de mapear una superficie.

Por tanto, podríamos decir que aparecen objetivos secundarios en base a las tareas que se deben realizar, estos son principalmente:

- **Adquisición de conocimientos y capacidad de análisis**
 - De todos y cada uno de los componentes utilizados.
 - De los entornos de programación.

- **Creación de programas**
 - Programación de drivers de control.
 - Programa de captura de medidas.
 - Programa de caracterización de componentes.
 - Programa de comunicación. Robot \leftrightarrow Pc.
 - Programa de navegación.
 - Programa de mapeo.

- **Elaboración de una memoria**
 - Capacidad de transmitir los conocimientos adquiridos.

Para finalizar, hay que destacar un último objetivo secundario, y no por ello el menos importante, el de seguir aprendiendo y adquiriendo los conocimientos que permitirán abarcar el mundo laboral con una mínima solidez y confianza. Debido a lo interesante que parece el tema, se puede asegurar que aprender y disfrutar irán de la mano.

1.2.PLANIFICACIÓN DEL PROYECTO

En este apartado se explica la planificación del proyecto y las tareas realizadas para poderlo llevar a cabo. Pretendiendo describir brevemente en qué consiste cada una de las tareas, a la vez que se identifican los recursos utilizados.

- **T.1. Especificaciones del proyecto y reuniones con el tutor**

Como su nombre indica, son todas las reuniones que se han llevado a cabo con el tutor, bien para corregir errores en el trabajo realizado, como para solucionar los diversos problemas que han ido surgiendo e intercambiar impresiones.

- Miembros involucrados:
 - Jorge Martín
 - Jonathan García

- **T.2. Construir el robot**

Esta tarea consiste en el montaje de todos los componentes que forman el robot, a la vez que se realizaban las conexiones necesarias de cada uno de ellos con la placa de control. Una vez realizada esta tarea se puede avanzar en el proyecto, por ese motivo, al ser la primera, es la más importante.

- Miembros involucrados:

- Jorge Martín
- Jonathan García

- **T.3. Crear los programas de control del robot**

Consiste en realizar un estudio y lograr gobernar todos los posibles movimientos del robot. Para ello se profundizó en la teoría del pulso PWM y en las posibilidades que el microcontrolador ofrece para poder implementarlo. Una vez conseguido, se procedió a la realización de los algoritmos de movimiento.

- Miembros involucrados:

- Jorge Martín
- Jonathan García

- **T.4. Establecer comunicación con el robot**

Para poder controlar el robot y así conseguir calibrar y estudiar los componentes, hace falta establecer una comunicación con el PC. En una primera aproximación se hizo uso de *Hyperterminal*, con el que conseguir una comunicación estable. Posteriormente se realizó una aplicación, mediante el entorno de programación *Python*, con el que lograr, aparte de la comunicación, realizar un tratamiento óptimo de los datos.

- Miembros involucrados:

- Jorge Martín
- Jonathan García

- **T.5. Estudio y programación de los *Encoders***

Consiste en estudiar el funcionamiento y obtener una correcta calibración de los *encoders*.

- Miembros involucrados:
 - Jorge Martín
 - Jonathan García

- **T.6. Estudio, programación y calibración de la Brújula**

Estudio de las posibles formas de conexionado que ofrece la brújula (PWM, I2C), a la vez que se investiga su funcionamiento y la manera de realizar una correcta calibración para disminuir los posibles errores.

- Miembros involucrados:
 - Jorge Martín
 - Jonathan García

- **T.7. Estudio, programación y calibración de los Sensores**

Se realiza un exhaustivo estudio de los sensores involucrados en la detección de obstáculos, caracterizándolos y calibrándolos correctamente.

Para ello se precisa de la realización de algoritmos con los que poder obtener las conclusiones oportunas.

- Miembros involucrados:
 - Jorge Martín
 - Jonathan García

- **T.8. Crear los algoritmos necesarios para el mapeo**

Esta tarea está distribuida en dos partes. La primera parte consiste en la realización del algoritmo de control del robot con la que realizar el seguimiento de la pared, y la segunda parte consiste en la elaboración del algoritmo con el que poder graficar a partir de los datos obtenidos de los sensores y del robot.

- Miembros involucrados:
 - Jorge Martín (algoritmo para graficar)
 - Jonathan García (algoritmo de control)

- **T.9. Realización de las pruebas y corrección de los errores**

Realización de todas las pruebas con las que determinar los errores. Permitiendo variar los componentes y/o los códigos por tal de solucionarlos. Consiste en una tarea conjunta por ambos miembros del proyecto.

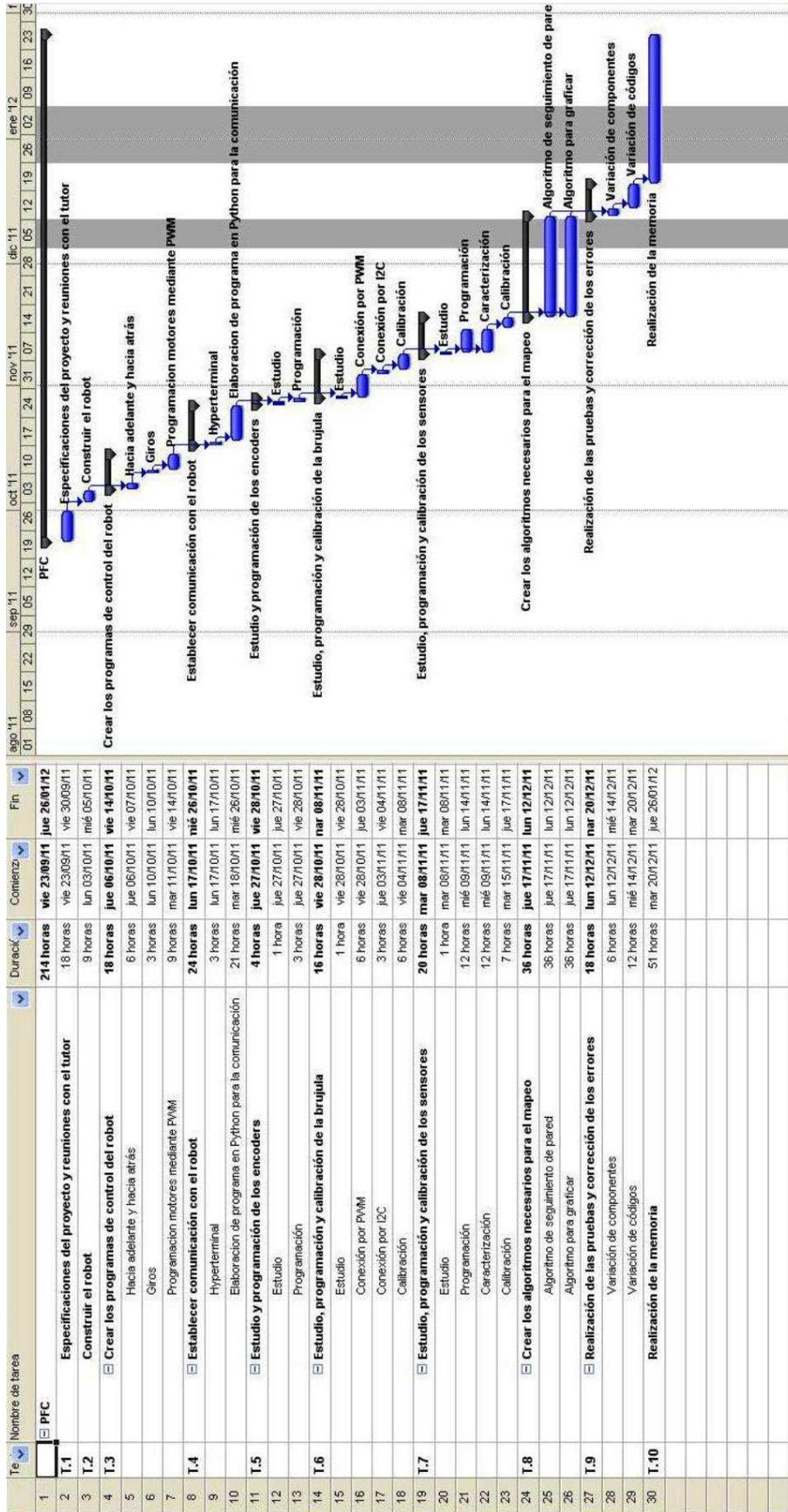
- Miembros involucrados:
 - Jorge Martín
 - Jonathan García

- **T.10. Realización de la memoria**

Tal y como su nombre indica, consiste en la realización de la memoria del proyecto, y por ese motivo se reparte entre los miembros del proyecto, realizando así cada componente su propia memoria.

- Miembros involucrados:
 - Jorge Martín
 - Jonathan García

Todas las tareas anteriormente nombradas y explicadas se pueden apreciar en la captura 1, donde se observa la planificación del proyecto, realizada con la herramienta que proporciona *MS office Project*. Cabe destacar que aunque se aprecia una elaboración muy continua, el resultado final no ha sido exactamente igual. Principalmente porque se ha tenido que volver sobre los pasos realizados, al encontrar en los estudios, componentes que no funcionaban correctamente y que se han tenido que sustituir por otros. De igual manera destaca la implementación y uso de la brújula, que una vez implementado su uso mediante PWM, se tuvo que volver a realizar mediante I2C, ya que se conseguían los datos de una manera más rápida y fluida.



Captura 1: Diagrama Gantt

Capítulo 2. ROBÓTICA.

2.1.ANTECEDENTES HISTÓRICOS [1].

A lo largo de la historia, el hombre ha intentado emular con máquinas la capacidad de autonomía y movimientos de los seres vivos. Para denominar a este tipo de máquinas los griegos usaron la palabra “*automatos*”, de la cual deriva la actual autómatas.

La cultura árabe (siglos VII a XV) aprovechó los conocimientos que los griegos les habían aportado para realizar todo tipo mecanismos con diferentes propósitos, entre los que destaca el de darles una aplicación práctica y no usarlos solo como instrumentos de diversión. Ejemplos de estos sistemas son por ejemplo los dispensadores de agua, destinados tanto a beber como para lavarse, que fueron introducidos entre la realeza.

España (siglos XV y XVI) durante la época del renacimiento, mostro gran interés por este tipo de mecanismos. Un caso particular de la época es el Hombre de palo, construido por Juanelo Turriano para el emperador Carlos V. Poseía la capacidad de andar, mover tanto la cabeza, como los ojos, boca o brazos. Para la época, fue tal la

expectación, que fue mas motivo de terror que de admiración, prueba de ello se encuentra es este gracioso fragmento extraído de una leyenda toledana [2].



Ilustración 1: Hombre de palo de Juanelo Turriano

“Turriano, ajeno a la perplejidad de sus vecinos, anduvo su camino por la calle estrecha que conducía hasta el Palacio Episcopal, asiendo de vez en cuando a su acompañante por un brazo para que éste no diera de bruces con el suelo. El personaje tan enigmático que había causado el terror, más que la admiración, entre la población no era otra cosa que un autómeta de madera que, según los presentes, se movía con tal garbo y destreza que en nada tenía que envidiar a los agüeros que desde el Puente de San Martín al Zoco acarreaban todas las mañanas el agua que refrescaba los gznates de los curtidores, plateros y alfareros, que a grito pelado vendían en este mercado sus artesanías manufacturas. Al día siguiente, Juanelo Turriano repitió el mismo paseo acompañado de su autómeta, y aunque la expectación fue la misma el recelo de la muchedumbre, sin embargo, se convirtió, una vez más, en admiración hacia el ingeniero al que les tenía acostumbrados el relojero italiano de Carlos V.”

Como Turriano, los artesanos que creaban estos ingenios mecánicos, en aquella época, pertenecían al gremio de la relojería. Creaban todo tipo de encargos para la corte, principalmente con el único propósito de servir de atracción en ferias.

No es hasta finales del siglo XVIII y principios del XIX, que este tipo de mecanismos fueron destinados hacia la industria, dando lugar a la automatización industrial. La industria más afectada fue la textil, en la que destacan aportaciones, tales como hiladoras o telares mecánicos.

Se otorga el uso de la palabra robot por primera vez al escritor checo Karel Capek, introduciéndola en su obra "*Rossum's Universal Robot*", estrenada en el teatro nacional de Praga. La palabra robot proviene de la palabra eslava "*robota*", refiriéndose al trabajo realizado de manera forzada.

Alejándose de los orígenes de la palabra robot, en los escritores de ciencia-ficción, encontramos al máximo impulsor de esta, Isaac Asimov (1920 - 1992), que publicó en 1945 una historia (*Runaround*) en la que introducía por primera vez sus tres leyes de la robótica.

1. Un robot no puede perjudicar a un ser humano, ni con su inacción permitir que un ser humano sufra daño.
2. Un robot ha de obedecer las órdenes recibidas de un ser humano, excepto si tales órdenes entran en conflicto con la primera ley.
3. Un robot debe proteger su propia existencia mientras tal protección no entre en conflicto con la primera o segundo ley.

Los progenitores de los robots como maquinas fueron los tele-manipuladores. Estaban destinados a ofrecer seguridad a los operarios cuando estos debían de acceder a zonas de alto riesgo. Eran dispositivos maestro-esclavo en los que el operario gobernaba, desde una zona segura, con el maestro, los movimientos que realizaba el esclavo.

La sustitución del operador por un programa de ordenador que controlase los movimientos del manipulador dio paso al concepto de robot.

2.2.ROBOT MÓVIL [3].

En la robótica industrial, encontramos multitud de robots compuestos de una estructura anclada a uno de sus extremos. Estos insinúan o pretenden simular, la apariencia de un brazo móvil, que gracias a sus seis grados de libertad (en algunos casos ya incorporan un 7º grado con el que dar movilidad, en una recta, a la base del robot), son capaces de realizar todo tipo de operaciones y/o trabajos previamente programados.



Ilustración 2: Robot industrial de la marca alemana KUKA

El desarrollo de robots móviles manifiesta la necesidad de extender el campo de la robótica industrial, incrementando la autonomía a la vez que restringe todo lo posible la mediación con el ser humano.

Profundizando en el concepto de autonomía para robots móviles, y como se vio en el apartado anterior, sus precedentes fueron los dispositivos electromecánicos, tales como los “*micro-mouse*” (años treinta) capaces de descubrir caminos en laberintos. Cabe destacar la tortuga de Walter (1948), capaz de reaccionar ante obstáculos, subir pendientes e incluso dirigirse a una posición concreta de recarga, cuando lo requería.

Durante los años setenta se siguió intentando aumentar la autonomía de estos robots, pero las condiciones no eran precisamente óptimas, ya que el desarrollo tecnológico no permitía lograr la navegación autónoma de forma eficiente.



Ilustración 3: Walter elaborando un robot móvil en los años 40

No es hasta los años ochenta, gracias a un aumento espectacular de la capacidad computacional y al desarrollo de nuevas tecnologías, que permite dotar a los robots móviles de una mayor autonomía. Destacan, entonces, construcciones destinadas tanto a la navegación interior, como a la exterior (Bares y otros, 1988; Thorpe, 1990), realizados en la Carnegie Mellon University (Pittsburgh, EE.UU.).

Se pretende dotar al robot con la suficiente agudeza y capacidad de reaccionar y tomar decisiones, que determinará mediante la interpretación de su entorno, sin ser este necesariamente conocido. Para lograrlo se hace imprescindible dotar al robot de un sistema de navegación, que incluya tareas tales como planificación, percepción y control.

La planificación en robots móviles se puede desglosar en planificación de la misión, de la ruta, de la trayectoria y finalmente, evitar obstáculos. La misión consistiría en establecer a donde hay que desplazarse, mientras que ruta sería el camino desde la situación inicial hasta la final.

Existen métodos de planificación que buscan caminos libres de obstáculos y minimizan la distancia recorrida, haciéndose valer de un entorno simulado mediante polígonos, pero también podemos encontrar planificaciones que tratan de encontrar caminos por

el centro del mismo, por tanto la planificación dependerá del objetivo planteado con anterioridad o de la función que ha de realizar el robot.

Otra posibilidad es realizar la planificación de la trayectoria de forma dinámica, con la que se precisará del conocimiento de la posición actual y de los puntos intermedios determinados en la planificación de la ruta. Hay que tener en cuenta que para realizar la definición de la trayectoria se deben considerar tanto las características cinemáticas del robot, como los modelos dinámicos de conducta del vehículo (interacción entre robot – terreno o velocidad).

Tras realizar la planificación de la trayectoria, se debe planificar el tipo de movimientos que va a realizar el robot y controlarlos para que se mantengan en la misma. En el caso de robots con ruedas se trata de determinar el ángulo de dirección respecto a la posición actual, teniendo en cuenta la trayectoria que debe seguir.

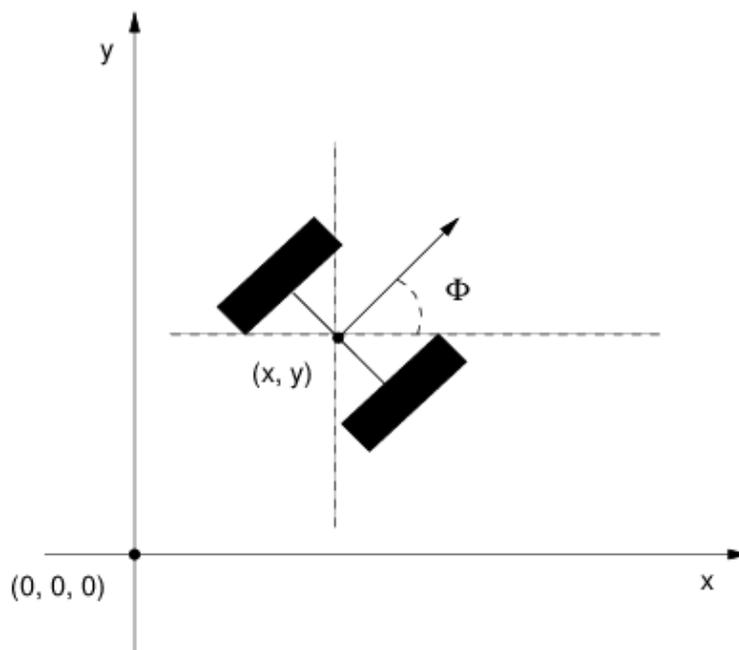


Ilustración 4: Parámetros para odometría

Visto esto, se aprecia que para el control del robot se hace preciso disponer de medidas que determinen su posición y orientación. Entre las técnicas más comunes se encuentra la odometría, en la que se profundizará más adelante, pero también es posible la utilización de sistemas de navegación inercial (giroscopios y acelerómetros).

Para aplicaciones exteriores, sin embargo, se hace muy común la utilización de sistemas de posicionamiento global mediante satélites, cuando las distancias a recorrer son considerables.

Para detectar obstáculos y evitar al robot situaciones peligrosas se hace preciso dotarlo de un sistema de percepción, que de forma análoga permitirán modelar el entorno construyendo un mapa o realizando una representación del mismo. Respecto a los sensores específicos que formen el sistema de percepción, se hace necesario poseer un buen conocimiento de ellos, y así poder determinar cómo y en qué circunstancias son capaces de ofrecer unos resultados óptimos, o en caso contrario, desestimarlos a favor de otros diferentes. Un ejemplo de sensor para navegación sería el de ultrasonido, basado en la determinación del tiempo de vuelo de un pulso de sonido (entre 30 kHz y 1 MHz), pero se debe conocer de él características como:

- Se ve influenciado por condiciones ambientales. (Posible solución mediante una buena calibración).
- Relación señal/ruido normalmente inferior a otros sensores. (Necesidad de empleo de múltiples frecuencias o técnicas de filtrado).
- Resolución lateral mediocre. (Necesidad de técnicas de enfoque).

2.3.ROBÓTICA SOCIAL. LA NUEVA GENERACIÓN[4][5].

La robótica social, busca construir robots con una serie de cualidades, atribuibles a los humanos, como localizar, expresarse, prestar atención o dirigirse a las personas. De esa manera hacer posible que humanos y robots interactúen entre ellos.

El uso al cual puede ser destinado un robot social, puede ser de lo más diverso. Se pueden encontrar desde juguetes y herramientas didácticas, hasta plataformas de desarrollo o sistemas de asistencia médica.

Si se quiere definir a un robot social, se debería profundizar en cuatro elementos que lo definen correctamente, autonomía, inteligencia, interacción y apariencia.

- **Autonomía:** el robot debe ser capaz de realizar su actividad bajo un control no humano, de forma libre.
- **Inteligencia:** de la mano de la programación, el robot debe tomar decisiones y a su vez aprender de ellas, intentando que sus interacciones se asemejen, cada vez más, a las del ser humano.
- **Interacción:** el robot debe comunicarse con los humanos, siendo capaz de percibirlos, entenderlos (voz, gestos, expresiones, etc.) y expresarse de forma similar.
- **Apariencia:** el hecho de dotar al robot con una apariencia humanoide, dotará al robot de mayor aceptación, facilitando la interacción con las personas.

Uno de los robots más populares hoy día es el robot ASIMO, fabricado para prestar asistencia a los humanos en sus tareas cotidianas. Es capaz de evitar a los humanos y darles la bienvenida si se encuentran en su camino, comunicarse por lengua de signos, correr, saltar y hasta ofrecer una bebida.



Ilustración 5: Robot ASIMO de Honda [4]

Muy similar a ASIMO, se puede mencionar a Robonaut2, proyecto elaborado por la NASA que constituye la elaboración y desarrollo de un robot astronauta. Cabe destacar que en esta ocasión, la intención no ha sido otra que la de ayudar en tareas realizadas en el espacio que pudieran constituir un peligro para los humanos.

Robonaut2 a diferencia de ASIMO carece de piernas con las que poder desplazarse de forma análoga a como lo haría un humano, pero entre sus cualidades posee visión dual, que le permite distinguir objetos situados a diferentes distancias entre ellos. Otra característica importante, debido a la tarea a la que se está destinado, es la telepresencia, que permite manipularlo por una persona que realiza los movimientos que quiere que desarrolle.



Ilustración 6: Robonaut2 construido por la NASA [5]

Reflexión: toda la documentación examinada sobre robótica social, expone que este tipo de robots debe estar dotado de apariencia humanoide. Analizando la palabra “social” adecuadamente y el primer objetivo a la hora de construir un robot, que no es otro que liberar al ser humano de trabajos con los que hacerle la vida un poco más fácil, no se debería excluir de este apartado a todos los robots domésticos que cumplen estas condiciones. Como gran ejemplo de ellos podemos mencionar al robot aspirador, de la casa iRobot.

Capítulo 3. MODELOS DE MEDICIÓN. CARACTERIZACIÓN.

Los modelos de medición comprenden el segundo dominio más importante de los modelos probabilísticos en robótica, siendo los primeros los modelos de movimiento. Los modelos de medición responden a la pregunta: ¿Cómo se han formado las medidas que ha realizado el sensor?

Los robots usan una gran variedad de sensores, y por lo tanto el modelo dependerá directamente del tipo de sensor que se utilice. La robótica probabilística pretende modelar el ruido en las mediciones del sensor, teniendo en cuenta la incertidumbre característica de cada sensor. La medición del modelo se define como una distribución de probabilidad condicional $p(z_t | x_t, m)$, donde x_t es la posición del robot, z_t es la medición en el tiempo t , y m es el mapa del entorno.

Se pueden encontrar diversos algoritmos en el mundo de la robótica probabilística, destacando dos:

- **Modelo de detección por rango del haz.** *“Beam models of range finders”.*
- **Modelo por rango de probabilidad del campo.** *“Likelihood fields for range finders”.*

Sebastian Thrun en su libro *“Probabilistic Robotics”* introduce las notaciones necesarias para su elaboración [6]. Respecto a la elaboración práctica se intentará emular el primero, con el que extraer la caracterización de los diferentes sensores de que se dispone.

Para una correcta implantación práctica, primero se debe realizar un exhaustivo estudio con el que aclarar conceptos y algoritmos. Por tanto, los apartados que siguen, mostrarán la teoría con los que realizar su correcta creación.

3.1. MODELO DE DETECCIÓN POR RANGO DE HAZ. *“BEAM MODELS OF RANGE FINDERS”*.

Para intentar modelar los sensores con los que se realizarán las mediciones, el modelo incorpora cuatro tipos de errores:

- Errores debidos al ruido local.
- Errores debidos a objetos inesperados.
- Errores debidos a fallas en la detección de objetos.
- Errores debidos a causas inexplicables (azar).

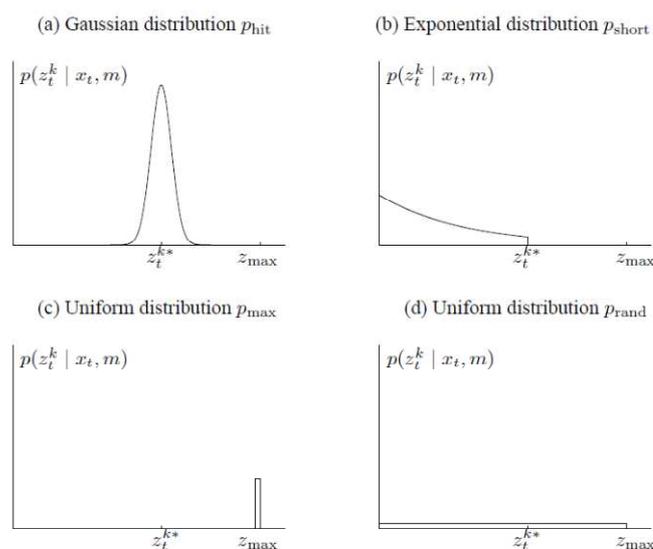


Ilustración 7: Componentes del modelo de medición. Imagen extraída de [6]

Por tanto el modelo deseado $p(z_t | x_t, m)$ es una mezcla de cuatro densidades, y cada una de ellas corresponde a un tipo particular de error:

1. **Rango correcto de medición del ruido local.** Imaginando un mundo ideal, el sensor siempre mediría la distancia correcta hasta el obstáculo encontrado. Sin embargo, aunque el sensor midiera la distancia al obstáculo correctamente, el valor que devolverá está sujeto a error. Este error será debido a la limitada resolución del rango que presente el sensor y al efecto del entorno sobre la señal de medición. Generalmente se obtendrá el rango real hasta el objeto mediante la búsqueda de la medida más característica dentro del cono de medición. La utilización de z_t^{k*} indicará el rango real del objeto medido por z_t^k .

La medición de este ruido se suele modelar con la representación de una estrecha gaussiana con media z_t^{k*} y s_{hit} desviación estándar. La figura de la ilustración 7a muestra esta densidad (con denominación p_{hit}) para un valor específico de z_t^{k*} .

En la práctica, los valores medidos por el sensor están limitados al intervalo $[0; z_{max}]$, donde z_{max} denota el rango máximo del sensor. Por lo tanto, la probabilidad de la medida viene dada por:

$$p_{hit}(z_t^k | x_t, m) = \begin{cases} \eta N(z_t^k; z_t^{k*}, s_{hit}^2) & \text{if } 0 \leq z_t^k \leq z_{max} \\ 0 & \text{otros} \end{cases} \quad (3.1.1)$$

Donde z_t^{k*} se calcula a partir de x_t y $N(z_t^k; z_t^{k*}, s_{hit}^2)$ denota la distribución normal con media z_t^{k*} y la desviación normal p_{hit} :

$$N(z_t^k; z_t^{k*}, s_{hit}^2) = \frac{1}{\sqrt{2\pi s_{hit}^2}} e^{-\frac{1}{2} \frac{(z_t^k - z_t^{k*})^2}{s_{hit}^2}} \quad (3.1.2)$$

El normalizador η se evalúa como:

$$\eta = \left(\int_0^{z_{max}} N(z_t^k; z_t^{k*}, s_{hit}^2) dz_t^k \right)^{-1} \quad (3.1.3)$$

La desviación estándar p_{hit} es un parámetro de ruido intrínseco del modelo de medición. Más adelante se discutirán las estrategias para el establecimiento de este parámetro.

2. **Objetos inesperados.** Los entornos de robots móviles son dinámicos, mientras que los mapas de m son estáticos. Un ejemplo típico de los objetos en movimiento son las personas, que comparten el espacio de funcionamiento del robot y que hacen que el sensor obtenga medidas sorprendentemente cortas. Para tener en cuenta estos objetos hay que tratarlos como parte del vector de estado y calcular su ubicación, o mucho más simple, tratarlos como el ruido del sensor. Al tratarlos como el ruido del sensor, los objetos modelados tienen la propiedad de que la medida va a ser simplemente más pequeña que z_t^{k*} .

La probabilidad de detección de objetos inesperados disminuye con la distancia. Matemáticamente, la probabilidad de que las mediciones varíen en este tipo de situaciones viene descrita por una distribución exponencial, que será denotada como p_{short} y que presentará un parámetro intrínseco del modelo de medición λ_{short} .

De acuerdo con la definición de una distribución exponencial se obtiene la siguiente ecuación para $p_{short}(z_t^k | x_t, m)$:

$$p_{short}(z_t^k | x_t, m) = \begin{cases} \eta \lambda_{short} e^{-\lambda_{short} z_t^k} & \text{if } 0 \leq z_t^k \leq z_t^{k*} \\ 0 & \text{otros} \end{cases} \quad (3.1.4)$$

Como en el caso anterior, es necesario el normalizador η . Esta exponencial se limita al intervalo $[0; z_t^{k*}]$, debido a que la probabilidad acumulada en este intervalo se expresa:

$$\int_0^{z_t^{k*}} \lambda_{short} e^{-\lambda_{short} z_t^k} dz_t^k = -e^{-\lambda_{short} z_t^{k*}} + e^{-\lambda_{short} 0} = 1 - e^{-\lambda_{short} z_t^{k*}} \quad (3.1.5)$$

Podemos obtener el valor de η como:

$$\eta = \frac{1}{1 - e^{-\lambda_{short} z_t^{k*}}} \quad (3.1.6)$$

La figura de la ilustración 7b muestra esta densidad gráfica, que cae exponencialmente en el rango de z_t^k .

3. **Fracasos.** A veces, los obstáculos son omitidos por completo. Un claro ejemplo de ello se encuentra en los sensores laser al detectar algún obstáculo negro que absorba la luz, o bien en los sensores de ultrasonidos como resultado de la reflexión especular. Un resultado típico de fallo del sensor es cuando este proporciona su valor máximo permitido z_{max} .

Dado que estos eventos son muy frecuentes, es necesario incorporarlo en el modelo de medición y se modelará con un punto de distribución centrado en z_{max} :

$$p_{max}(z_t^k | x_t, m) = I(z = z_{max}) = \begin{cases} 1 & \text{if } z = z_{max} \\ 0 & \text{otros} \end{cases} \quad (3.1.7)$$

Aquí la función toma el valor 1 si su argumento es cierto, y 0 en caso contrario.

Técnicamente, p_{max} no posee una función de densidad de probabilidad debido a que p_{max} es una distribución discreta. Esto no es preocupante puesto que el modelo matemático para el modelo de medición no se ve afectado por la inexistencia de una función de densidad. Sin embargo se puede representar gráficamente como una distribución uniforme, muy estrecha y centrada en z_{max} .

4. **Mediciones aleatorias.** Por último, los sensores producen ocasionalmente mediciones inexplicables. Por ejemplo, los sensores de ultrasonidos generan mediciones fantasmas cuando las mediciones provienen de rebotes en paredes, o cuando están en intercomunicación entre diferentes sensores.

Este tipo de mediciones las modelaremos como una distribución uniforme, entre el rango que proporciona el sensor $[0; z_{max}]$.

Estas cuatro diferentes distribuciones están mezcladas en un promedio ponderado, que se define por los parámetros z_{hit} , z_{short} , z_{max} y z_{rand} , los cuales verifican:

$$z_{hit} + z_{short} + z_{max} + z_{rand} = 1 \quad (3.1.8)$$

$$p(z_t^k | x_t, m) = \begin{pmatrix} z_{hit} \\ z_{short} \\ z_{max} \\ z_{rand} \end{pmatrix}^T * \begin{pmatrix} p_{hit}(z_t^k | x_t, m) \\ p_{short}(z_t^k | x_t, m) \\ p_{max}(z_t^k | x_t, m) \\ p_{rand}(z_t^k | x_t, m) \end{pmatrix} \quad (3.1.9)$$

El resultado típico de esta combinación lineal de las densidades individuales se muestra en la figura de la ilustración 8. Como se puede apreciar, las características básicas de los cuatro modelos básicos están presentes en esta densidad combinada.

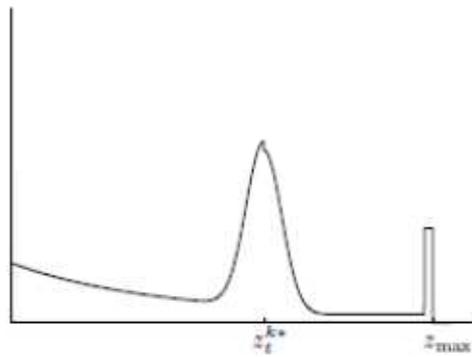


Ilustración 8: Combinación lineal de densidades. Imagen extraída de [6]

El modelo de sensor es implementado por el algoritmo destinado a la detección por rango de haz ("*beam_range_finder_model*"). La ilustración 9 muestra las operaciones e iteraciones necesarias para llevarlo a cabo.

```

1:      Algorithm beam_range_finder_model ( $z_t, x_t, m$ )
2:           $q = 1$ 
3:          for  $k = 1$  to  $K$  do
4:              compute  $z_t^{k*}$  for the measurement  $z_t^k$  using ray casting
5:               $p = z_{hit} * p_{hit}(z_t^k | x_t, m) + z_{short} * p_{short}(z_t^k | x_t, m) +$ 
6:                   $+ z_{max} * p_{max}(z_t^k | x_t, m) + z_{rand} * p_{rand}(z_t^k | x_t, m)$ 
7:               $q = q * p$ 
8:          return  $q$ 

```

Ilustración 9: Algoritmo "Beam_range_finder_model" [6].

La entrada de este algoritmo es z_t que representa una exploración completa del sensor, la posición del robot x_t , y el mapa m . Desglosando se encuentra:

- En su lazo externo (líneas 2 y 7) se multiplica la probabilidad de cada medida individual del sensor z_t^k .
- En la línea 4 se aplica el cómputo necesario para calcular el rango de ruido libre que presenta el sensor utilizado en particular.
- La probabilidad individual de que z_t^k se encuentre en el rango de medición se calcula en la línea 5, que implementa la regla de mezcla de densidades (3.1.9).
- Después de iterar a través de todas las mediciones del sensor z_t^k en z_t , el algoritmo devuelve la probabilidad deseada $p(z_t^k | x_t, m)$.

3.1.1.AJUSTE DE LOS PARÁMETROS INTRÍNSECOS DEL MODELO.

Los parámetros incluyen z_{hit} , z_{short} , z_{max} y z_{rand} , los parámetros de mezcla, pero también incluyen los parámetros s_{hit} y λ_{short} . Para simplificar se denotará al conjunto de todos los parámetros intrínsecos como θ . Debido a que la probabilidad de cada medición es una función de θ , se debe hablar de un algoritmo para el ajuste de los mismos.

Una forma de determinar los parámetros intrínsecos es basarse en los datos. Las figuras de la ilustración 10 muestran dos series de 10.000 mediciones obtenidas con un robot móvil que se desplaza a través de un entorno típico de oficina. Ambas gráficas muestran el conjunto de mediciones tomadas en un rango de aproximadamente 3 metros. La grafica izquierda muestra los resultados obtenidos por un sensor de ultrasonidos mientras que la de la derecha muestra las de un sensor láser. En ambos casos el eje X muestra el numero de la lectura (de 1 a 10000), y el eje Y es el rango medido por el sensor.

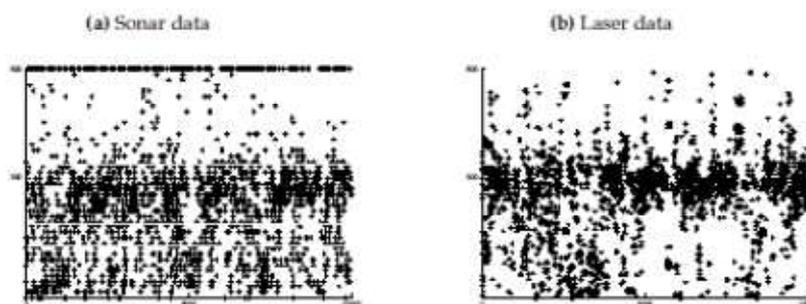


Ilustración 10: Mediciones típicas de sensores de ultrasonido y láser. Imagen extraída de [6]

Mientras que, para los dos sensores, la mayoría de las mediciones se aproximan a una medición relativamente correcta, el comportamiento en ambos es sustancialmente diferente. El sensor de ultrasonidos parece sufrir muchos más errores en las mediciones y con bastante frecuencia, no puede detectar un obstáculo, y en su lugar da resultados de alcance máximo. Por el contrario, el sensor láser es más preciso. Sin embargo, también de vez en cuando proporciona este tipo de valores.

La manera de establecer θ se logra mediante la maximización de la probabilidad de un conjunto de datos de referencia $Z = \{z_i\}$, con las posiciones asociadas $X = \{x_i\}$ y m . Donde cada z_i es una medición real, x_i es la posición en que la medida fue tomada, y m es el mapa.

La probabilidad de los datos Z viene dada por:

$$p(Z | X, m, \theta) \tag{3.1.1.1}$$

Cualquier algoritmo que maximice la probabilidad de los datos se conoce como un estimador de máxima verosimilitud, o estimador ML.

La ilustración 11 muestra el algoritmo de “*learn_intrinsic_parameters*”, que es un algoritmo para el cálculo de la estimación de máxima verosimilitud de los parámetros intrínsecos.

```

1:      Algorithm learn_intrinsic_parameters (Z, X, m)
2:      repeat until convergence criterion satisfied
3:      for all  $z_t$  in Z do
4:           $H = [ p_{hit}(z_t|x_t, m) + p_{short}(z_t|x_t, m) +$ 
               $+ p_{max}(z_t|x_t, m) + p_{rand}(z_t|x_t, m) ]^{-1}$ 
5:          calculate  $z_t^*$ 
6:           $e_{i,hit} = \eta p_{hit}(z_t|x_t, m)$ 
7:           $e_{i,short} = \eta p_{short}(z_t|x_t, m)$ 
8:           $e_{i,max} = \eta p_{max}(z_t|x_t, m)$ 
9:           $e_{i,rand} = \eta p_{rand}(z_t|x_t, m)$ 
10:          $z_{hit} = |Z|^{-1} \sum_i e_{i,hit}$ 
11:          $z_{short} = |Z|^{-1} \sum_i e_{i,short}$ 
12:          $z_{max} = |Z|^{-1} \sum_i e_{i,max}$ 
13:          $z_{rand} = |Z|^{-1} \sum_i e_{i,rand}$ 
14:          $s_{hit} = \sqrt{\frac{1}{\sum_i e_{i,hit}} \sum_i e_{i,hit} (z_i - z_i^*)^2}$ 
15:          $\lambda_{short} = \frac{\sum_i e_{i,short}}{\sum_i e_{i,short} z_i}$ 
16:         return  $\theta = \{z_{hit}, z_{short}, z_{max}, z_{rand}, s_{hit}, \lambda_{short}\}$ 

```

Ilustración 11: Algoritmo para estimación de máxima verosimilitud [6].

Inicialmente, el algoritmo “*learn_intrinsic_parameters*” requiere de una buena inicialización de los parámetros intrínsecos s_{hit} y λ_{short} . Desglosando se encuentra:

- En las líneas 3 y 9, se estiman variables auxiliares: Cada $e_{i,xxx}$ es la probabilidad de que la medida z_i es causada por “xxx”, donde se elige “xxx” de los cuatro aspectos del modelo de sensor, *hit*, *short*, *max* y *random*.
- Posteriormente, se estima los parámetros intrínsecos en las líneas 10 a 15. Los parámetros intrínsecos, sin embargo, están en función de las expectativas calculadas antes.

Para un buen ajuste de los parámetros intrínsecos, el algoritmo tiene que ser repetido. Sin embargo, en la práctica la iteración converge rápidamente, y una docena de repeticiones suelen ser suficientes para cinco buenos resultados.

3.2.MODELO POR RANGO DE PROBABILIDAD DEL CAMPO. “LIKELIHOOD FIELD RANGE FINDER MODEL”.

Al igual que en el modelo discutido antes, se supone tres tipos de fuentes de ruido e incertidumbre:

1. **Medición del ruido.** Producido por el proceso de medición, se basa en gaussianas al igual que en el modelo anterior. En el espacio x - y , se trata de encontrar el obstáculo más cercano en el mapa y se toma la variable “*dist*” para denotar la distancia euclídea entre la medición de coordenadas $(x_{z_t^k} \ y_{z_t^k})^T$ y el objeto más cercano en el mapa m .

$$p_{hit}(z_t^k | x_t, m) = e^{-s^2(dist)^2} \quad (3.2.1)$$

La función resultante es la mostrada en las Figuras de la ilustración 12.

2. **Fracasos.** Como antes, se supone que el alcance máximo de las lecturas tienen una distinta probabilidad. Ésta es modelada por una distribución p_{max} .
3. **Mediciones aleatorias.** Por último, una distribución uniforme p_{rand} , que se utiliza para modelar el ruido aleatorio en la percepción.

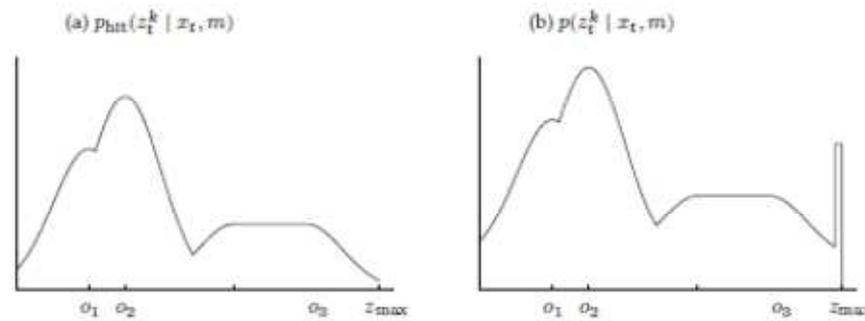


Ilustración 12: Distribuciones de "Likelihood field range finder model" [6]

Al igual que para el modelo anterior, basado en la probabilidad deseada $p(z_t^k | x_t, m)$, se integran las tres distribuciones, utilizando los pesos conocidos de z_{hit} , z_{rand} y z_{max} :

$$z_{hit} * p_{hit} + z_{rand} * p_{rand} + z_{max} * p_{max} \tag{3.2.2}$$

Las figuras de la ilustración 12 muestran un ejemplo de la distribución resultante $p(z_t^k | x_t, m)$. Es fácil ver que en esta distribución se combina p_{hit} y las distribuciones p_{max} y p_{rand} . Una representación como esta constituye la probabilidad de una detección de obstáculos en función de las coordenadas x-y mundiales, recibe el nombre de probabilidad del campo.

La figura de la ilustración 13 proporciona el algoritmo, donde se observa:

- Circuito exterior que multiplica los valores individuales de $p(z_t^k | x_t, m)$, asumiendo la independencia entre el ruido de las diferentes medidas.
- La línea 4 comprueba si la lectura del sensor es un valor máximo, en cuyo caso simplemente se ignora.
- Las líneas 5-8 manejan un interesante caso: Aquí se calcula la distancia al obstáculo más cercano en el espacio x-y (Línea 7), y el resultado de la probabilidad se obtiene en la línea 8 por mezcla de una distribución normal y otra uniforme. (la función " $prob(dist^2, s_{hit}^2)$ " calcula la probabilidad de " $dist^2$ " bajo una distribución gaussiana, centrada en cero y con varianza s_{hit}^2).

- La operación más costosa del algoritmo es el buscar el vecino más cercano en el mapa (Línea 7). Para acelerar esta búsqueda, es ventajoso pre-calcular la probabilidad de campo.

```

1:      Algorithm likelihood_field_range_finder_model ( $z_t, x_t, m$ ):
2:       $q = 1$ 
3:      for all  $k$  do
4:          if  $z_t^k \neq z_{max}$ 
5:               $x_{z_t^k} = x + x_{k,sens} \cos \theta - y_{k,sens} \sin \theta + z_t^k \cos(\theta + \theta_{k,sens})$ 
6:               $y_{z_t^k} = y + y_{k,sens} \cos \theta - x \sin \theta + z_t^k \sin(\theta + \theta_{k,sens})$ 
7:               $dist^2 = \min_{x',y'} \{(x_{z_t^k} - x')^2 + (y_{z_t^k} - y')^2 | (x', y') \text{ occupied in } m\}$ 
8:               $q = q * (z_{hit} * prob(dist^2, s_{hit}^2) + \frac{z_{random}}{z_{max}})$ 
9:      return  $q$ 

```

Ilustración 13: Algoritmo "Likelihood_field_range_finder_model" [6].

Capítulo 4. CINEMÁTICA DEL ROBOT Y ODOMETRÍA

4.1. ODOMETRÍA[7].

La odometría se utiliza para estimar la posición relativa de un robot en el plano x - y . Para lograrlo se debe disponer de los valores del vector (x, y, θ) .

Se habla de estimación debido a que saber a ciencia cierta la posición del robot es prácticamente imposible, los métodos que se utilizan para calcular la posición no tienen una precisión absoluta. Las medidas son aportadas por los sensores y estos nunca se librarán de errores, que serán inversamente proporcionales al coste invertido en los sensores.

En odometría es preciso hablar de posiciones relativas, pues no podemos hablar de coordenadas absolutas, o coordenadas del mundo. Con la odometría las posiciones que se dan, son relativas al punto de inicio, que será donde el robot comience a tomar las mediciones.

4.2. CINEMÁTICA EN ROBOTS MÓVILES CON RUEDAS [3] [8] [9]

La cinemática del robot vendrá determinada por la composición y características del medio de locomoción que incorpore. En robots con ruedas, la cinemática estará ligada a la ubicación y número de estas. Las composiciones más conocidas son:

- **Diferencial:** siendo el más sencillo, se compone de dos ruedas de tracción unidas en un eje común y otras dos que le dan estabilidad. La traslación y rotación viene determinada por las ruedas de tracción, que se controlan de forma independiente.

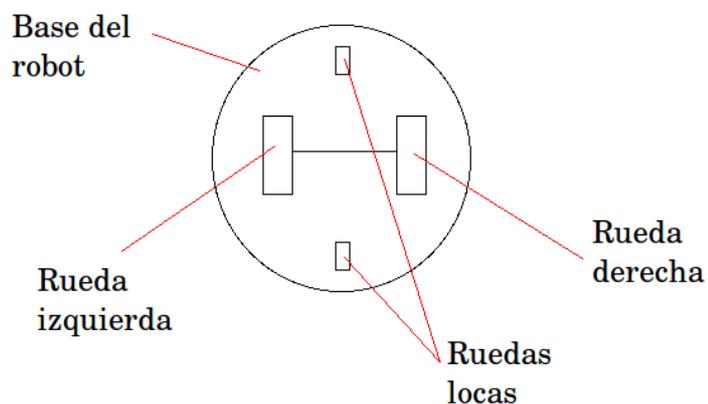


Ilustración 14: Ejemplo de cinemática diferencial

- **Síncrono:** las ruedas se mueven al mismo instante, es un caso particular del diferencial. Las ruedas están ligadas de tal manera que siempre apuntan en la misma dirección y para girar, lo hacen sobre el eje vertical, dejando la estructura fija.

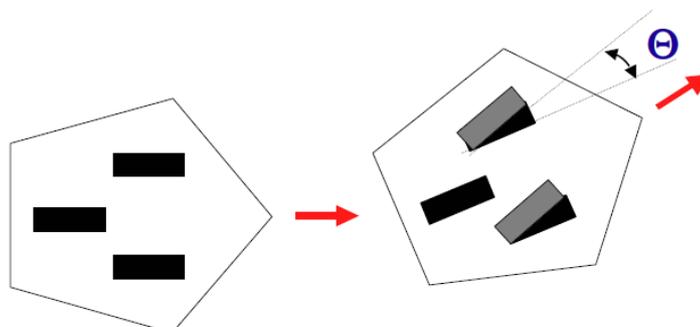


Ilustración 15: Modelo síncrono

- **Triciclo:** compuesto, normalmente, por dos ruedas de tracción y una tercera de dirección. Se caracterizan por disponer de buena estabilidad y simplicidad mecánica, pero su cinemática es más compleja.

También se puede usar la otra configuración posible, donde la única rueda de tracción es la delantera. Su cinemática resultará más simple, determinada por el número de pulsos de la rueda de tracción y la dirección de la misma. Esta configuración presenta un problema, ya que cuando el robot se encuentra en subidas, su centro de gravedad tiende a alejarse de la rueda de motriz y provoca una pérdida de tracción.

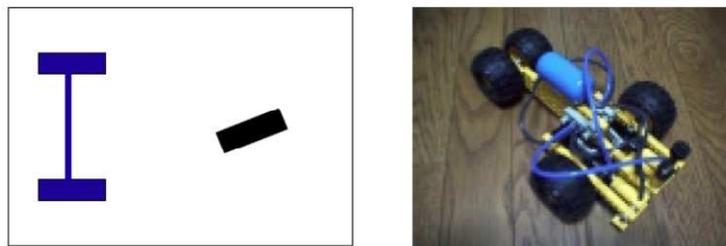


Ilustración 16: Modelo triciclo

- **Ackerman:** similares al triciclo, pero el modelo cuenta con dos ruedas de tracción y dos de dirección. El modelo Ackerman se utiliza casi exclusivamente en la industria del automóvil (ruedas traseras de tracción y dos ruedas de dirección delanteras), permitiendo que la rueda interior tenga un ángulo ligeramente más agudo que la exterior, evitando posibles derrapes.

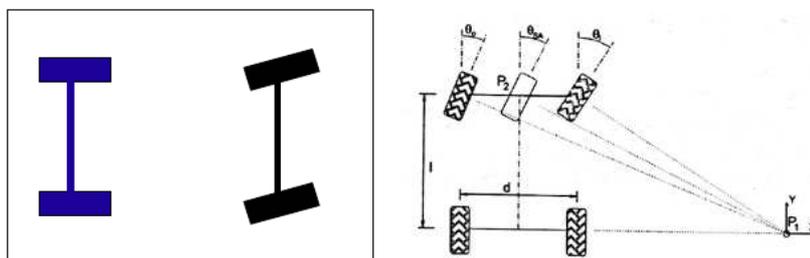


Ilustración 17: Modelo Ackerman

- **Omnidireccional:** cuentan con 3 ruedas, colocadas a 120 grados, que tienen la capacidad de girar en ambos lados, logrando un control lineal mas simplificado que en

el caso diferencial. Gracias a sus ruedas omnidireccionales son capaces de desplazarse desde un punto inicial a uno final en línea recta, sin necesidad de realizar una rotación antes de desplazarse.

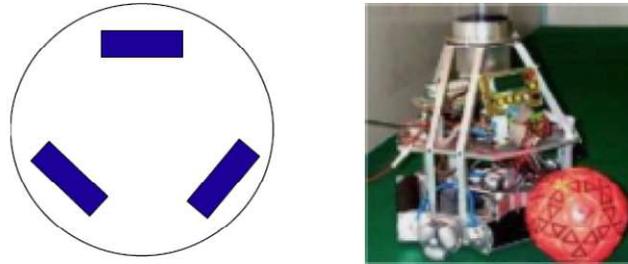


Ilustración 18: Modelo omnidireccional

Una vez vistos los diferentes modelos que se pueden implementar, y limitados por los componentes que se poseían para la realización del proyecto, se optó por una composición de tipo diferencial. Es por ello que en el apartado siguiente se muestra el modelo matemático con el que se debe trabajar.

4.3.CINEMÁTICA DIFERENCIAL. MODELO MATEMÁTICO

Para implementar este modelo, se deberán seguir una serie de pasos, con los que poder estimar la posición en el siguiente estado.

- **Paso 1.** Considerando la resolución de los encoders, r , la distancia entre ruedas, d , y el valor acumulado de los encoders izquierdo y derecho, E_l y E_r respectivamente, se puede realizar el cálculo del desplazamiento de cada rueda:

$$D_l = E_l * r$$

$$D_r = E_r * r$$

- **Paso 2.** El desplazamiento lineal realizado por el punto que se encuentra justo en medio del eje se puede calcular como:

$$D_c = (D_l + D_r) / 2$$

- **Paso 3.** Cuando las ruedas del robot giran a diferentes velocidades, hacen que el robot gire, para determinar el valor del ángulo (en radianes) de giro realizamos:

$$D_{\theta} = (D_r - D_l) / d$$

- **Paso 4.** Se conoce la situación del instante actual, t , y partir de esta se ha de determinar la posición en el instante siguiente, $t+1$:

$$\theta_{t+1} = \theta_t + D_{\theta}$$

$$X_{t+1} = X_t + D_c * \cos(\theta_{t+1})$$

$$Y_{t+1} = Y_t + D_c * \sin(\theta_{t+1})$$

4.4.VARIACIÓN DEL MODELO DIFERENCIAL

Debido a que el montaje del robot incorpora una brújula magnética, y a que el modelo diferencial simple, puede generar unos errores acumulativos excesivamente elevados, se tomo la decisión de utilizar el valor de la brújula para dar la orientación adecuada que se debe aplicar al algoritmo.

Como veremos en el capítulo siguiente, donde se mencionan los principales algoritmos destinados para realizar mapas, en este proyecto se intenta emular el denominado vector de percepción general. En él se incorpora el modelo de cinemática diferencial, donde el robot ofrece un vector, para cada posición que ha recorrido en 40 ms, con el valor de la brújula, encoders y los sensores.

En este caso, el valor de los sensores no es necesario, estos son incorporados al vector con la función de posteriormente realizar el mapa del entorno. El modelo cinemático diferencial solamente aportara la localización del robot, en cada punto, con la que poder incorporar al grafico el recorrido que realiza en todo momento.

Capítulo 5. MAPEO [12].

La creación de mapas resulta muy útil para interactuar con el entorno. Los ejemplos más claros se pueden obtener de los animales, que examinan su entorno para realizar sus acciones más cotidianas, o lo que sería lo mismo, realizan un mapa de su entorno valiéndose de sus cinco sentidos. Incluso se podría afirmar que algunos de ellos guardan en su memoria los mapas de recorridos y lugares por los que han pasado a lo largo de su vida. Avanzando hasta los seres humanos, se aprecia la capacidad de interpretar, generar y utilizar cualquier mapa creado.

En los sistemas robóticos se plantea emular esta capacidad, puesto que si se les introduce un plano o mapa de su entorno en la memoria, le será de gran ayuda a la hora de posicionarse y/o moverse.

La confección de mapas por robots se emplea como una importante capa de apoyo en tareas como la navegación y planificación de rutas, para bordear/pasar eficazmente obstáculos, como guías de museos o robots sirvientes y domésticos

Las principales tareas de reconocimiento de terrenos desconocidos son la localización y el mapeo del entorno, si se trata de cartografiar el entorno, se hará conociendo a

priori la localización del autómata, y si se trata de localizar la máquina, será conociendo previamente el mapa del lugar. Pero resolver ambos problemas a la vez es bastante más complicado. A esta tarea conjunta se le conoce como SLAM (*Simultaneous Localization And Mapping*) [17]. SLAM puede ser resuelto mediante aproximaciones basadas en filtro de Kalman, donde el mapa que se genera, muestra la probabilidad posterior de localización de alguna característica o baliza que puede ser detectada por el robot mientras explora el entorno. También se usan algoritmos basados en la técnica de *Extimiation-Maximation* (EM), pero ambas aproximaciones tienen sus desventajas, sobre todo en el hecho de tener que conocer posiciones relativas.

5.1.ALGORITMIA Y REPRESENTACIÓN DE MAPAS [10][18].

Los robots almacenan en su memoria todos los datos necesarios que han ido captando al explorar el terreno formando, datos como localización, orientación o distancias que acaban formando un segmento. Al formar este conjunto de datos incluso pueden llegar a guardar relaciones con otros segmentos próximos. De este modo, al final se obtiene una lista que posee toda la información con la que generar el mapa.

Los dos paradigmas más estudiados para el aprendizaje en realización de mapas, que dan lugar a los algoritmos más usados se conocen como:

- **Paradigma Geométrico**

Los algoritmos que se encuentran dentro de este grupo, hacen uso de relaciones geométricas absolutas entre los obstáculos que el robot va encontrando. Entre ellos podemos encontrar:

- Mapas de Rejilla de Ocupación.
- Mapas Difusos.
- Espacio de Configuración.

- **Paradigma Topológico**

Los algoritmos que se ubican en este grupo, generan un mapa en forma de grafo, este se realiza con nodos que representan localizaciones clave y que se unen entre sí mediante arcos de grafo.

- Mapas Topológicos.
- Mapas Basados en Características.
- Vector de Percepción General.
- Esquema de Seguimiento de Camino.

A mitad de camino entre un tipo y otro, o mediante la unión de ambos, se pueden encontrar otro tipo de algoritmos con los que adaptar, según las necesidades, las características de uno con el otro, dando lugar a otro tipo de algoritmos.

- Mapas Híbridos.

- **COMPARATIVA DE ALGORITMOS**

La tabla 1 [18], muestra las principales características de los algoritmos de representación, junto con sus principales ventajas e inconvenientes.

Sistemas de Representación	Resumen características	Ventajas	Inconvenientes
Rejilla de Ocupación	Matriz de celdas que contienen la probabilidad de estar en el estado ocupado o vacío.	Facilidad de uso, construcción y actualización.	Ocupa mucho espacio en memoria y consume mayor tiempo.
Basados en Características	Representación del entorno por características paramétricas como líneas, puntos, arcos...	Más información métrica, idóneo para entornos estáticos. Menor espacio en memoria.	Difícil concretar obstáculos con exactitud. Difícil uso y actualización. No usable para entornos desestructurados.

Topológico	Grafo de nodos representando lugares del espacio, conectados por arcos si hay un camino entre ellos.	Menor espacio en memoria, compactado. Bueno para planificación de rutas.	Menor información métrico-espacial. Ambigüedad de reconocimiento de lugares.
Difusos	Conjunto difuso compuesto por la unión de conjunto vacío y ocupado de puntos del espacio.	Ventajas de Rejilla de Ocupación. Mejora de estimación de ocupación.	Desventajas de Rejilla de Ocupación. Cálculos de intersección de conjuntos.
Vector de Percepción General	Vector con información de los alrededores cercanos que se actualiza en cada movimiento fuera de éste. Mapa parcial.	Poco espacio en memoria. Decisión de camino más rápida. Generación más rápida.	Mayor incertidumbre. No tiene memoria. Recalculado para zonas ya visitadas.
Esquema de Seguimiento de Camino	Genera hitos de referencia entre dos puntos. Almacena tuplas de posición relativa entre hitos.	Fácilmente escalable, bueno para planificación de rutas. Poco espacio en memoria.	Necesario definir hitos en el sendero. No información global.
Híbridos	Combinación de métodos. Típicamente entre uno topológico y otro métrico.	Se compensan las desventajas de los métodos.	Doble computación y mantenimiento. Dificultad de relación entre representaciones.
Espacio de Configuración	Lista de pares vértice/borde que definen el contorno del obstáculo	Estructurado. Solo representa el espacio ocupado. Alta precisión.	Dificultad para trazar rutas, computación más costosa.

Tabla 1: Comparativa de algoritmos de mapeo. Tabla extraída de [18]

Entre los más usados destacan:

- **MAPAS DE REJILLA DE OCUPACIÓN:**

0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5
0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5
0,5	0,5	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	1,0	0,5
0,5	0,9	0,3	0,3	0,3	0,3	0,3	0,3	0,3	0,3	0,3	1,0	0,5
0,5	0,9	0,3	0,3	0,3	0,3	0,3	0,3	0,3	0,3	0,3	0,9	0,5
0,5	1,0	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,8	0,5
0,5	0,9	0,1	0,1	0,1	0,9	0,8	0,8	0,7	0,1	1,0	0,5	
0,5	0,9	0,2	0,2	0,2	0,9	0,2	0,2	0,2	0,2	1,0	0,5	
0,5	1,0	0,0	0,0	0,0	1,0	0,0	0,0	0,0	0,0	0,9	0,5	
0,5	0,9	0,1	0,1	0,1	1,0	1,0	0,1	0,1	0,1	1,0	0,5	
0,5	0,9	0,0	0,0	0,0	0,0	1,0	1,0	0,0	0,0	0,9	0,5	
0,5	0,9	0,0	0,0	0,0	0,0	0,0	1,0	0,0	0,0	0,9	0,5	
0,5	0,8	0,1	0,1	0,1	0,1	0,1	1,0	0,1	0,1	1,0	0,5	
0,5	0,9	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,2	1,0	0,5	
0,5	0,9	0,2	0,2	0,2	0,9	1,0	0,9	0,9	0,9	0,5	0,5	
0,5	0,5	1,0	1,0	1,0	0,5	0,5	0,5	0,5	0,5	0,5	0,5	
0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	

Ilustración 19: Mapa de rejilla de ocupación

Este método describe una matriz de NxM celdas que representa una cierta región cuadrada del entorno. Así mismo, almacena la probabilidad de que exista o no un obstáculo. Con probabilidades entre 1 y 0, 1 representa la seguridad de que la celda se encuentra ocupada, y 0 muestra la certeza de que la celda se encuentra vacía. El resto de valores darán una idea, según la probabilidad que almacenen, de la situación de encontrarse en uno u otro estado. Inicialmente las celdas deben partir de una probabilidad intermedia para demostrar la incerteza del estado de esa celda. El robot al realizar el recorrido irá incrementando o decrementando el valor de las celdas demostrando la situación de cada una de ellas.

- **MAPAS TOPOLÓGICOS.**

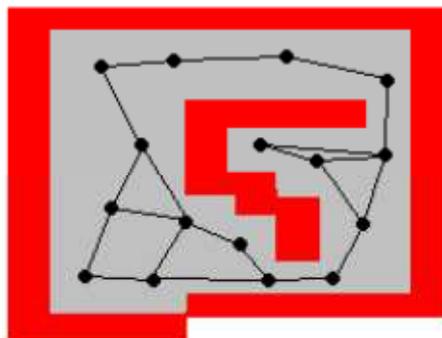


Ilustración 20: Mapas topológicos

Este algoritmo trata de localizar lugares clave del entorno, que sean navegables, para almacenar su posición. Si la navegación entre los puntos que almacena es posible, se interconectarán dichas posiciones por una serie de arcos. La forma de representar el mapa de este algoritmo lo hace óptimo si se desea localizar el mejor camino, siendo la mejor forma para trazar rutas entre dos puntos.

- **VECTOR DE PERCEPCIÓN GENERAL**

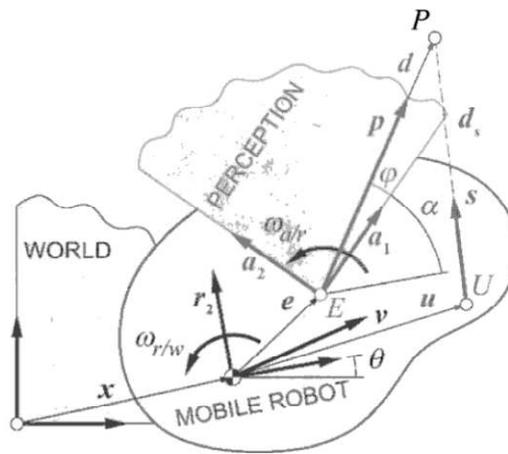


Ilustración 21: Mapa vector de percepción general

La técnica de mapeo basada en vector de percepción general, almacena los datos del entorno más próximo al robot en un solo vector, lo que hace que el cómputo de los datos se agilice de manera elevada, penalizando el grado de incertidumbre. Normalmente es usado en robots con un sistema de percepción completa, o con robots omnidireccionales, capaces de almacenar todos los datos de su entorno más próximo. Con todos los datos extraídos, estima una probabilidad entre 0 y 1 que estima la posibilidad de percibir un obstáculo a una distancia concreta, haciendo que este avance de forma segura hacia otra posición donde recalculará la estimación para seguir progresando.

***Capítulo 6.* CARACTERIZACIÓN DE LOS SENSORES.**

Para mayor comprensión y poder realizar un mejor trabajo con los sensores, se ha realizado la caracterización de los mismos, con la que se pretende descubrir todos los parámetros que puedan ser útiles a la hora de realizar cualquier acción con ellos. Por ejemplo, puede ser de gran ayuda determinar su grado de fiabilidad, rango de trabajo, conversión de voltaje a cm, etc.

Se han realizado dos conjuntos de pruebas distintas para caracterizar correctamente, una donde el robot se encontraba parado mientras realizaba las medidas y otra con el robot en movimiento.

Por otro lado, antes de empezar a caracterizar, se ha realizado la programación del algoritmo probabilístico, teniendo que realizar, como se muestra en el apartado 6.1, varias pruebas con el fin de conseguir las gráficas deseadas.

6.1.GRÁFICAS PROBABILÍSTICAS.

Para realizar una primera implementación del algoritmo y como este se basa en funciones de probabilidad, se realizó una primera toma de contacto haciéndonos valer de las funciones que MATLAB lleva incorporadas.

A través de los sensores ubicados en la parte frontal del robot y enfocados hacia una superficie plana (como la pared de una habitación), se implemento un programa que realizaba movimientos perpendiculares a distintas distancias, las cuales se encuentran comprendidas entre los 20 y 60 centímetros, en intervalos de 10 centímetros. En total el robot realizó un conjunto de 5500 medidas (aproximadamente) para cada distancia, donde realizaba un desplazamiento de 10 cm, se paraba, y tomaba la medición de ambos sensores (IR y Ultrasonido) en la misma posición.

Los resultados obtenidos quedan reflejados en las siguientes graficas. Aunque estas no resultaron demasiado vistosas, aportaron algunas conclusiones que hicieron que el conjunto del proyecto fuera afinándose, realizando mejoras en los códigos y en el sistema de captación de medidas.

- **SENSOR ULTRASONIDOS.**

En la gráfica de la ilustración 22 se observan el conjunto de todas las medidas realizadas por el sensor de ultrasonidos (nº de medida en eje de abscisas y valor en eje de ordenadas). En las ilustraciones posteriores se localizan los primeros intentos de conseguir las gráficas probabilísticas, en ellas se distinguen las diferentes densidades correspondientes a cada rango de medida.

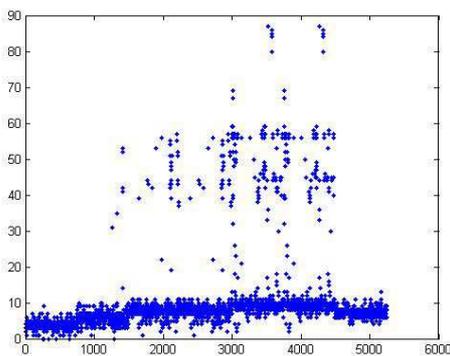


Ilustración 22: Medidas del ultrasonido

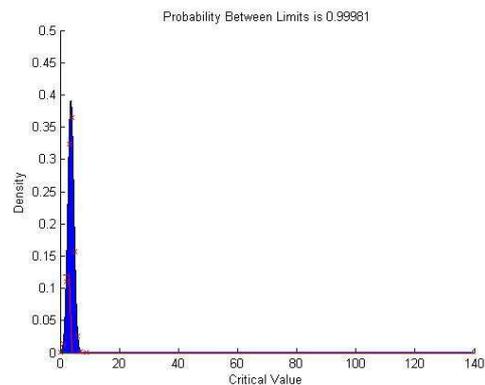


Ilustración 23: Robot en movimiento a 20 cm

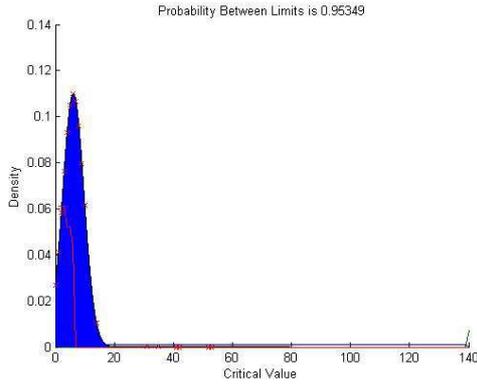


Ilustración 24: Robot en movimiento a 30 cm

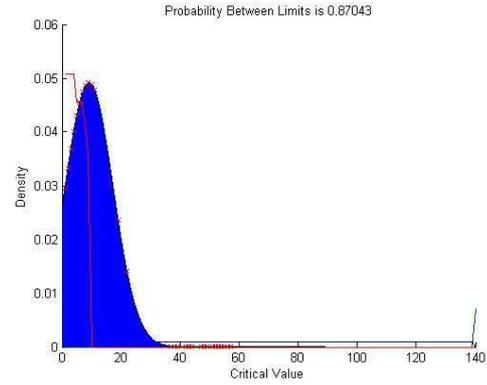


Ilustración 25: Robot en movimiento a 40 cm

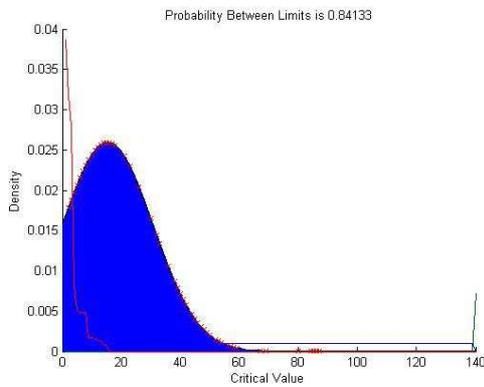


Ilustración 26: Robot en movimiento a 50 cm

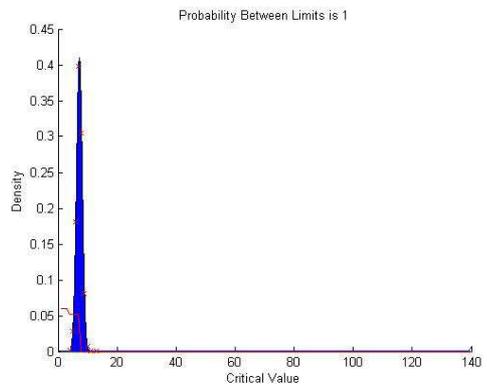


Ilustración 27: Robot en movimiento a 60 cm

- **SENSOR INFRARROJOS (IR).**

De forma análoga localizamos el mismo conjunto de gráficas, pero en este caso, las resultantes para el sensor de infrarrojos. Al ser el algoritmo utilizado exactamente igual, los resultados gráficos, sin interpretación, resultan exactamente iguales.

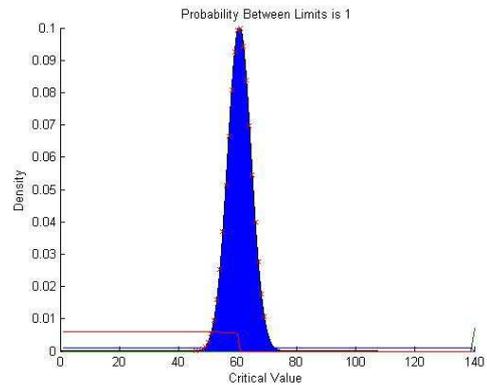
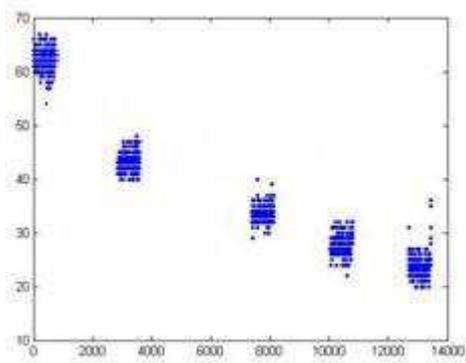


Ilustración 28: Medidas del infrarrojos

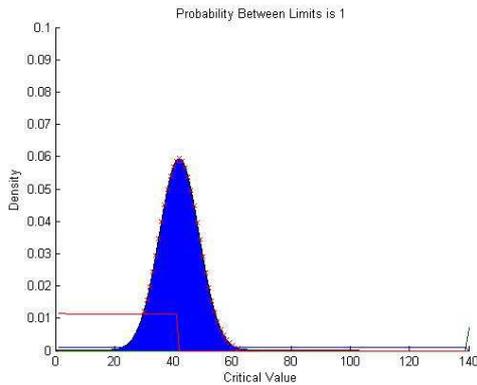


Ilustración 29: Robot en movimiento a 20 cm

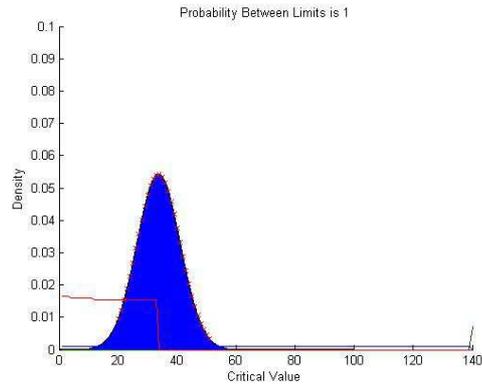


Ilustración 30: Robot en movimiento a 30 cm

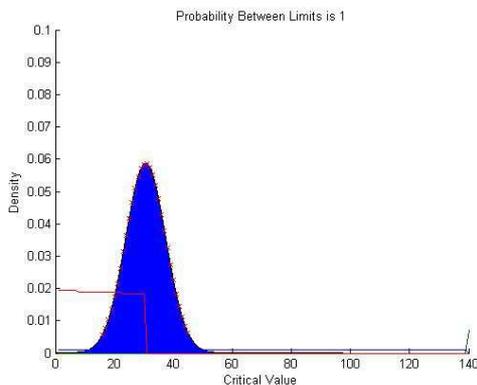


Ilustración 31: Robot en movimiento a 40 cm

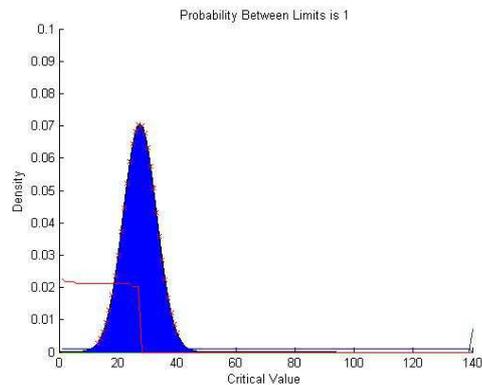


Ilustración 32: Robot en movimiento a 50 cm

Ilustración 33: Robot en movimiento a 60 cm

Como se comento con anterioridad, estas primeras pruebas, aunque semejantes a los resultados gráficos que esperábamos, no presentaban la continuidad de una grafica única. El tratamiento de los datos, sobre todo en relación a la exponencial (obstáculos inesperados), no se acaba de definir como la propia función a la que pertenecen. Es por ello que se convirtió en una obligación el continuar mejorando el algoritmo, dando como resultado unas gráficas más definidas y esclarecedoras.

- **SENSOR ULTRASONIDOS.**

El siguiente grupo de graficas, muestra los resultados obtenidos al optimizar el algoritmo probabilístico. Como se puede observar en la figura de la ilustración 34, las

mediciones son exactamente las mismas, pero dando en las graficas posteriores un resultado totalmente diferente.

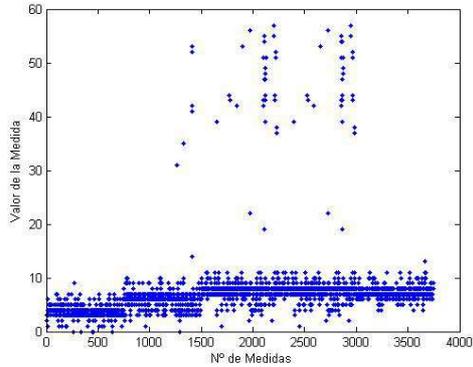


Ilustración 34: Medidas del ultrasonido

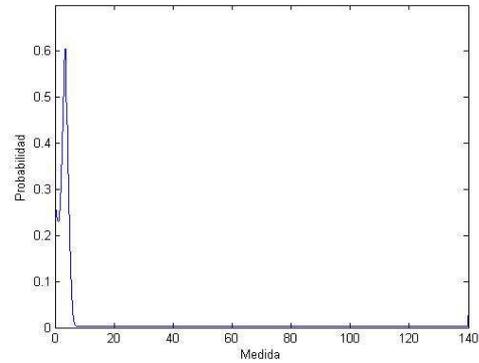


Ilustración 35: Robot en movimiento a 20 cm

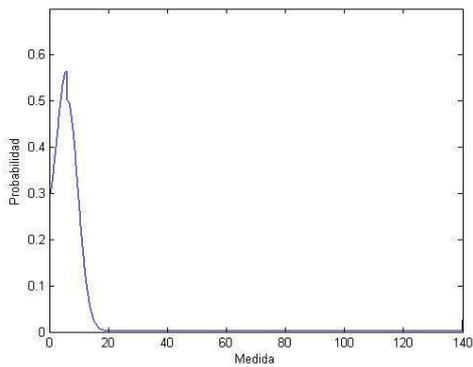


Ilustración 36: Robot en movimiento a 30 cm

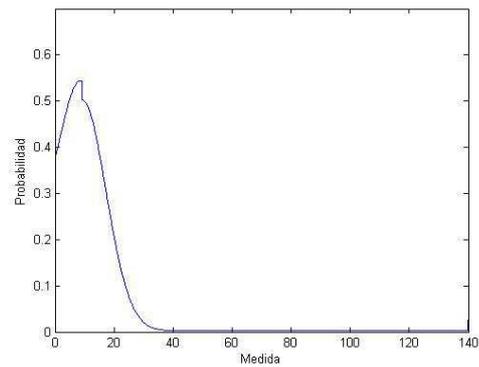


Ilustración 37: Robot en movimiento a 40 cm

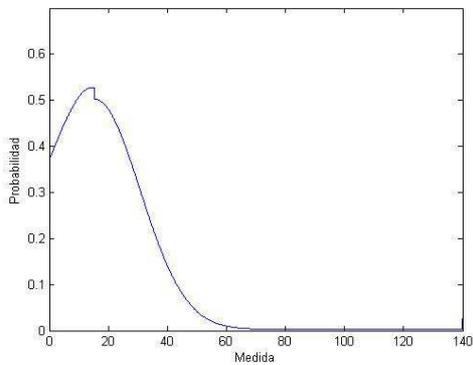


Ilustración 38: Robot en movimiento a 50 cm

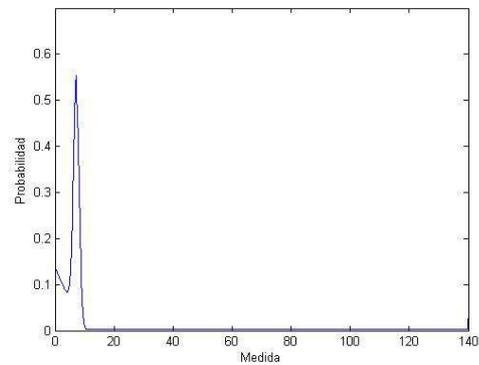


Ilustración 39: Robot en movimiento a 60 cm

- **SENSOR INFRARROJOS (IR).**

De igual manera, se muestran ahora las graficas obtenidas con la modificación del algoritmo pero en este caso, para el sensor de infrarrojos.

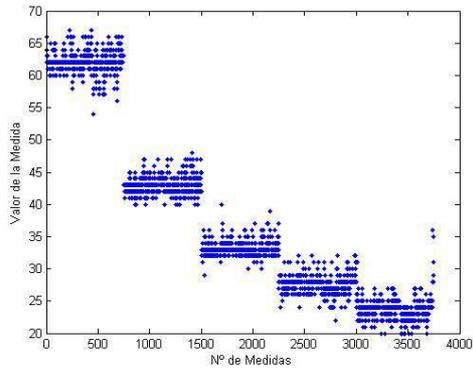


Ilustración 40: Medidas del infrarrojos

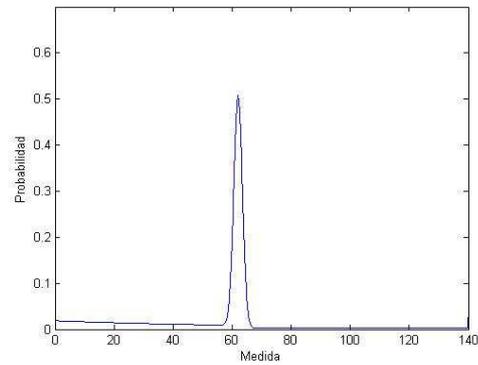


Ilustración 41: Robot en movimiento a 20 cm

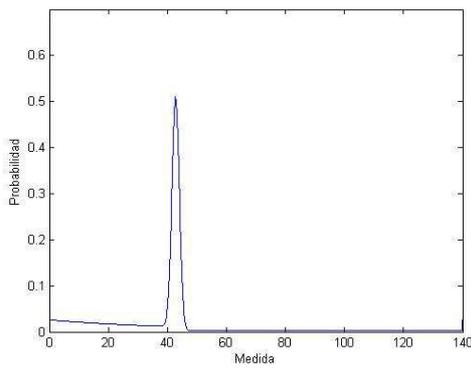


Ilustración 42: Robot en movimiento a 30 cm

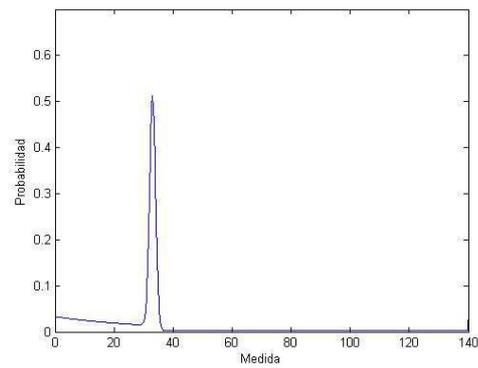


Ilustración 43: Robot en movimiento a 40 cm

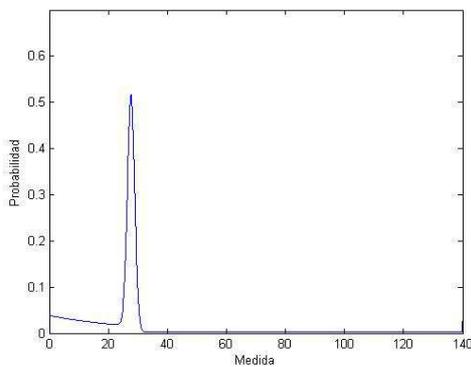


Ilustración 44: Robot en movimiento a 50 cm

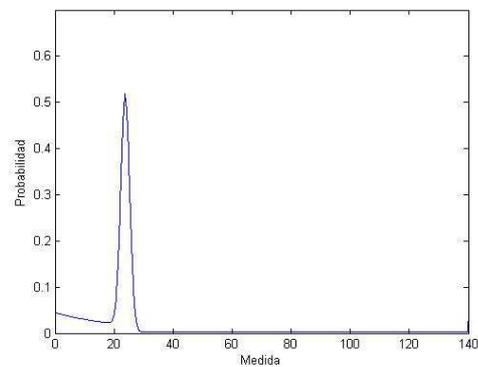


Ilustración 45: Robot en movimiento a 60 cm

Al mismo tiempo que se conseguía optimizar el algoritmo para la representación de las graficas, se pudo ir analizando los resultados que proporcionaban y sacar unas primeras conclusiones de ambos sensores:

- Con relación al sensor de ultrasonidos, y como se puede apreciar en las gráficas, la dispersión que muestran algunas de ellas hace cuestionar su correcto funcionamiento. Al realizar pruebas con el sensor y visualizando los resultados que ofrecía mediante el osciloscopio, se llegó a la determinación de que no funcionaba correctamente y se procedió a la sustitución del mismo por uno nuevo, cuyos resultados analizaremos en su posterior caracterización.
- Por otro lado el sensor IR presentaba los valores obtenidos de forma inversa al Ultrasonido, presentando a mayores distancias valores inferiores. Esto llevo a una modificación del código de captación de medidas con la que ir mostrando los resultados, al igual que en el sensor de ultrasonido, de forma creciente en relación a la distancia a la que se encuentra.

Otro aspecto que se decidió optimizar, fue la calibración del sensor IR. Haciendo que su valor máximo (3v) correspondiera con el valor 255 mediante la regulación de un potenciómetro a su distancia máxima óptima, proporcionada por el fabricante (80 cm).

6.2.ESTUDIO DE LOS SENSORES CUANDO EL ROBOT ESTÁ PARADO.

Para el estudio de los sensores cuando el robot está parado se ha utilizado el algoritmo probabilístico *"beam_range_finder_model"*. Se han realizado un conjunto de medidas comprendidas entre las distancias de 35 a 45 cm con un intervalo de 1 cm, ya que el rango en el que el robot va a trabajar se encuentra entre los 40 cm.

Una vez aplicado el algoritmo, los resultados obtenidos en las diferentes medidas han resultado muy similares, encontrando un ligero desplazamiento hacia la derecha, que como es predecible corresponde a la diferencia de distancia entre las diferentes medidas. Es preciso recordar que los valores del sensor IR han sido obtenidos restando al valor máximo, es decir, a 255 el valor real medido por el sensor, para una mejor

comparación con el sensor de ultrasonido. Así pues, los resultados obtenidos son los mostrados en las figuras siguientes:

○ **RESULTADOS PARA EL IR (Robot parado).**

Realizando una gráfica de las diferentes medidas tomadas, se obtiene el comportamiento del sensor IR. Este demuestra ser bastante preciso debido a que el grado de dispersión de la gaussiana no es nada elevado. Por otro lado el comportamiento que presenta a objetos inesperados (exponencial) y a medidas aleatorias (uniforme) son prácticamente inapreciables, cuyo resultado es muy positivo.

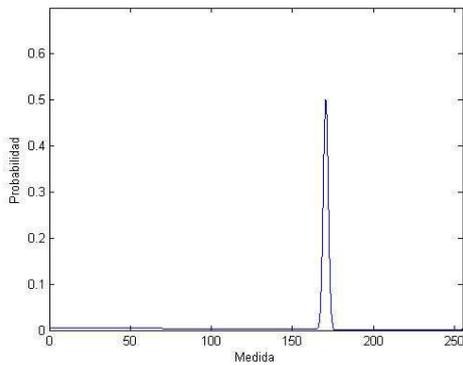


Ilustración 46: Robot parado a 35 cm

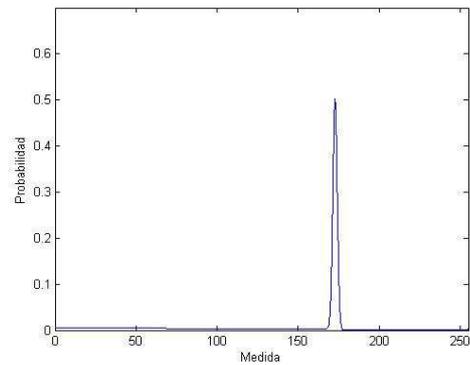


Ilustración 47: Robot parado a 36 cm

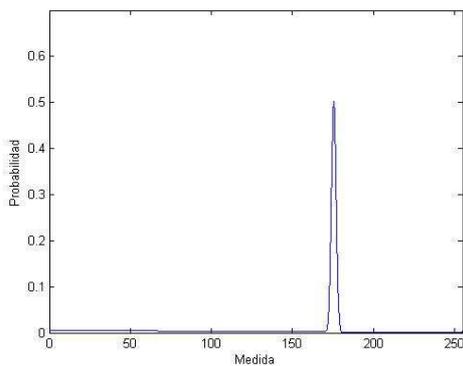


Ilustración 48: Robot parado a 37 cm

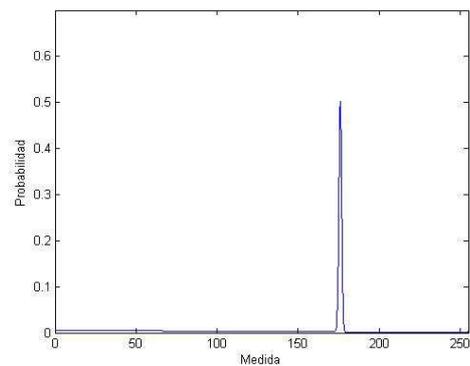


Ilustración 49: Robot parado a 38 cm

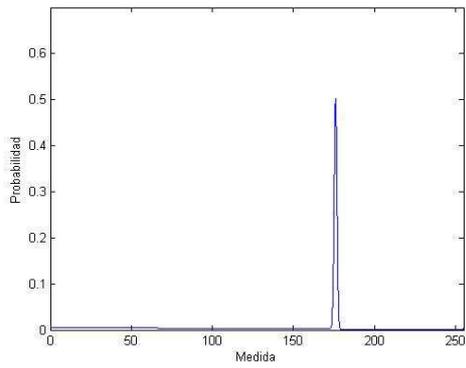


Ilustración 50: Robot parado a 39 cm

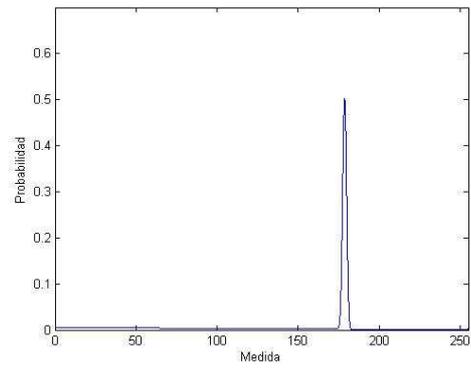


Ilustración 51: Robot parado a 40 cm

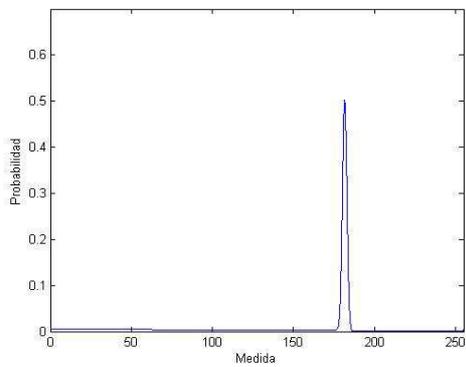


Ilustración 52: Robot parado a 41 cm

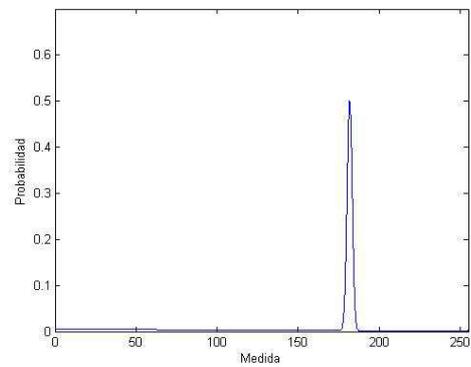


Ilustración 53: Robot parado a 42 cm

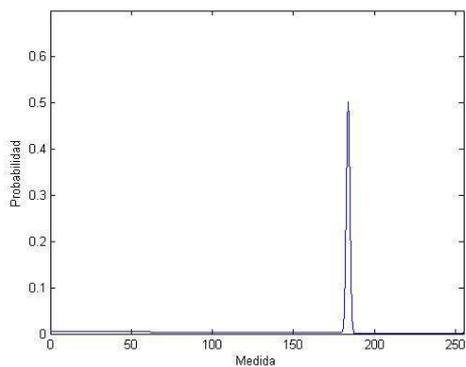


Ilustración 54: Robot parado a 43 cm

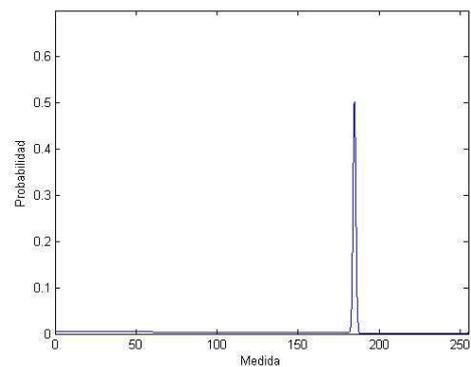


Ilustración 55: Robot parado a 44 cm

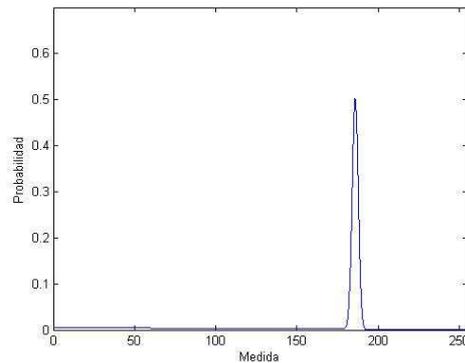


Ilustración 56: Robot parado a 45 cm

Paralelamente, gracias a estos resultados se demostró que el comportamiento de este sensor en el rango de medidas con las se pretendía trabajar, presenta una linealidad aceptable. Esta se hace visible en la figura de la ilustración 57.

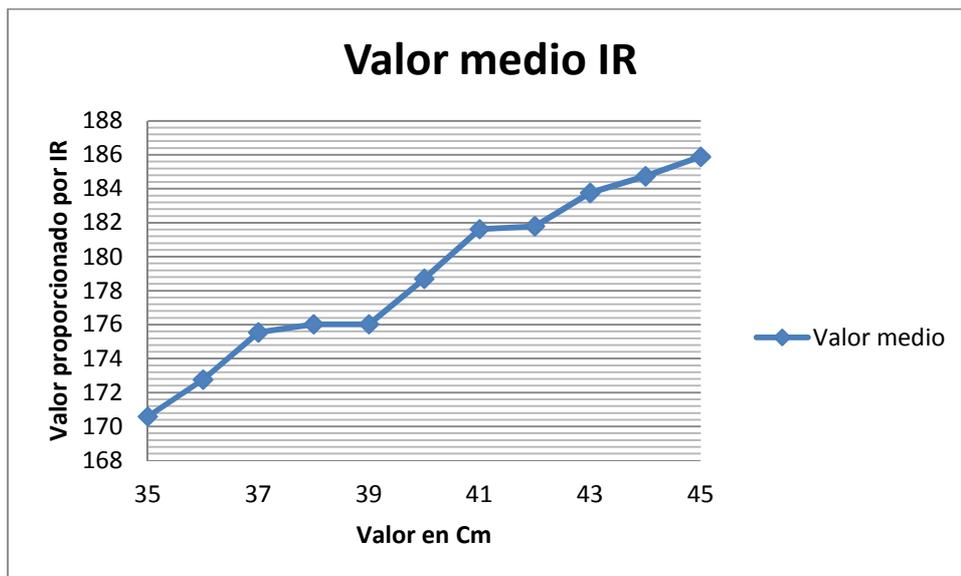


Ilustración 57: Valores proporcionados por el IR a distancias concretas

Los valores obtenidos presentan una linealidad creciente semejante a la esperada, proporcionada por el fabricante (ilustración 58), aportando como primera impresión, unas características óptimas para la utilización del sensor en el proyecto. Posteriormente, junto con el ultrasonido, se verá su comportamiento cuando el robot se encuentra en movimiento, dado que se hace primordial para las circunstancias y condiciones en las que se va a trabajar.

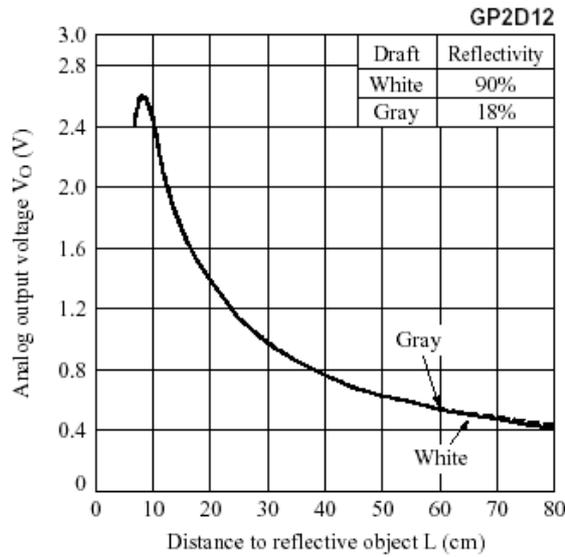


Ilustración 58: Comportamiento del sensor IR

○ **RESULTADOS PARA EL ULTRASONIDOS (Robot parado).**

Realizando el mismo proceso utilizado para el sensor IR, y para las mismas distancias (medidas tomadas al mismo tiempo mediante programa específico de captación de medidas), se obtienen las gráficas referentes al sensor de ultrasonido. Estas demuestran la gran precisión del sensor si se observa la poca dispersión que presenta la gaussiana. Por el contrario, es mucho más apreciable, en comparación con el sensor IR, como el ultrasonido es propenso a los obstáculos inesperados que se puedan presentar dentro de su rango de captación. (Como se vio en la parte de teoría, esto se presenta como una exponencial dentro del algoritmo probabilístico).

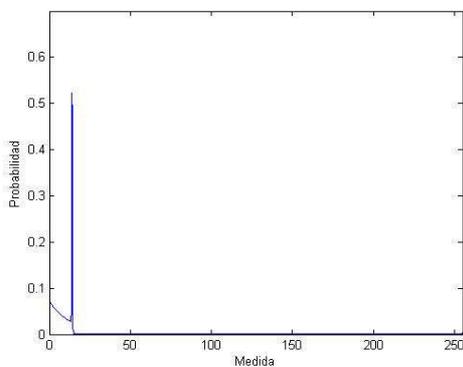


Ilustración 59: Robot parado a 35 cm

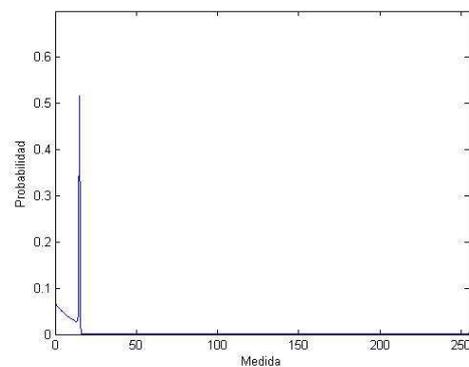


Ilustración 60: Robot parado a 36 cm

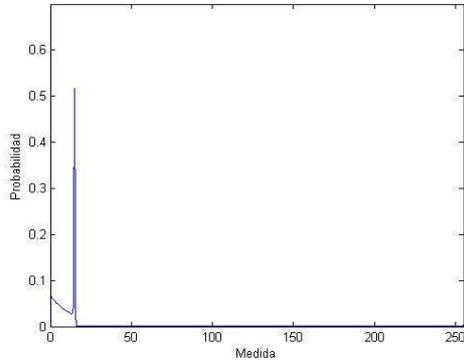


Ilustración 61: Robot parado a 37 cm

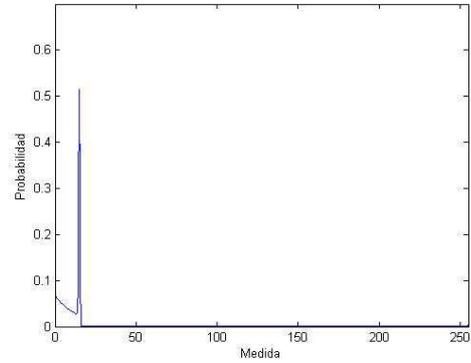


Ilustración 62: Robot parado a 38 cm

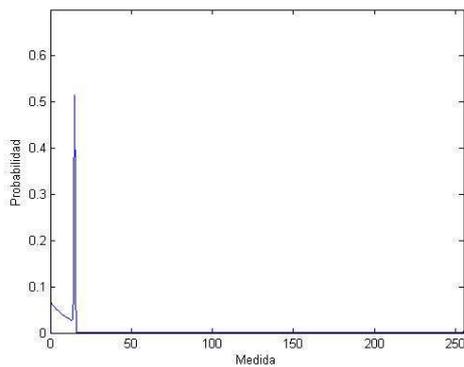


Ilustración 63: Robot parado a 39 cm

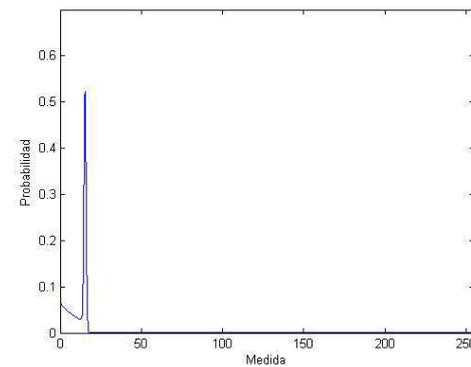


Ilustración 64: Robot parado a 40 cm

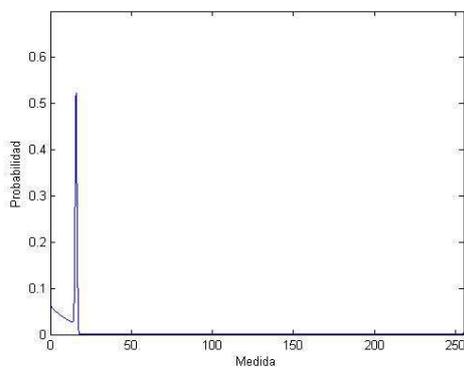


Ilustración 65: Robot parado a 41 cm

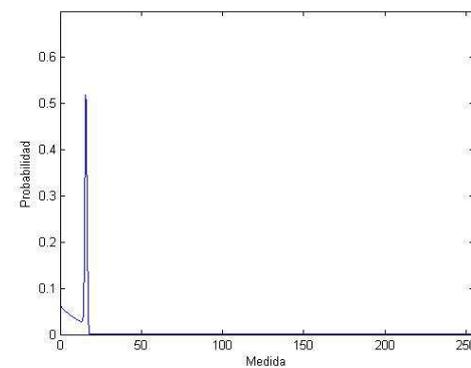


Ilustración 66: Robot parado a 42 cm

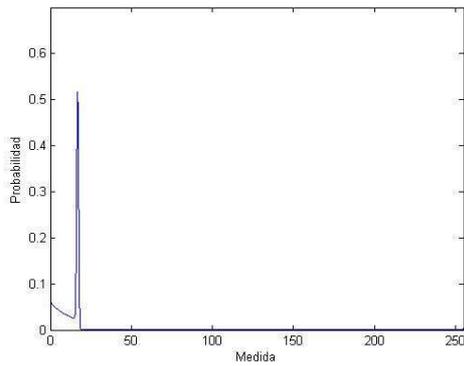


Ilustración 67: Robot parado a 43 cm

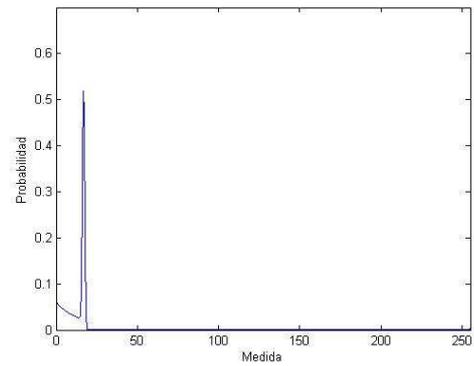


Ilustración 68: Robot parado a 44 cm

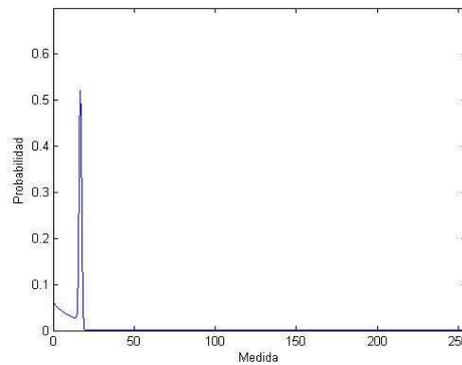


Ilustración 69: Robot parado a 45 cm

Por otro lado, al ser el rango óptimo del sensor de 0 a 6,45 m y el rango de medidas en el que queremos trabajar ligeramente reducido (38 – 42 cm), hace que la diferencia entre las distancias medidas sea mínima (ilustración 70). Esto puede presentar problemas al trabajar con las mediciones tomadas y con la capacidad de reacción del robot, haciendo que el robot reaccione o demasiado pronto o demasiado tarde.

De todas formas se decidió seguir adelante con la posible incorporación de este tipo de sensor dada la forma de su gaussiana, que como se vio con anterioridad, presenta una precisión considerable. Por lo tanto, al igual que el IR, se realizó un estudio de su comportamiento cuando el robot se encuentra en movimiento.

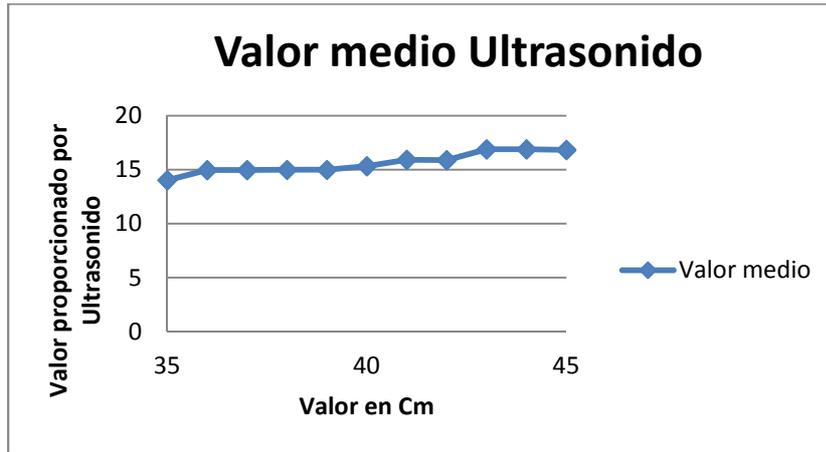


Ilustración 70: Valores del ultrasonido a distancias concretas

6.3.ESTUDIO DE LOS SENSORES. ROBOT ESTÁ EN MOVIMIENTO.

Para el estudio de los sensores cuando el robot está en movimiento se ha puesto el robot a tomar medidas a unas distancias comprendidas entre 30 y 50 cm a intervalos de 5 cm. En cada una de estas medidas, el robot ha ido recorriendo una distancia de 1 metro, de forma ininterrumpida y en paralelo a la pared, intentando simular el recorrido que realizaría cuando se encuentre trabajando autónomamente.

○ **RESULTADOS PARA EL IR (Robot en movimiento).**

Las siguientes gráficas, muestran el comportamiento del sensor IR, cuando el robot se encuentra en movimiento. En ellas se aprecian unos resultados muy semejantes con la única variación de la posición de la gaussiana, que corresponde al valor de la posición donde se encontraba el robot.

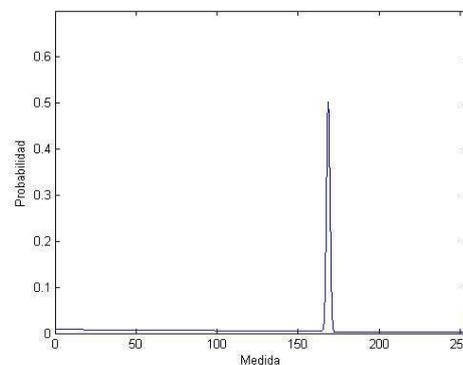
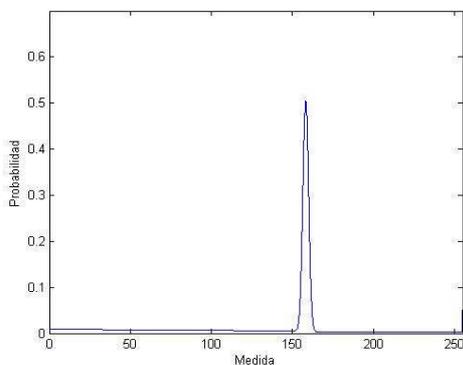


Ilustración 71: Robot en movimiento a 30 cm Ilustración 72: Robot en movimiento a 35 cm

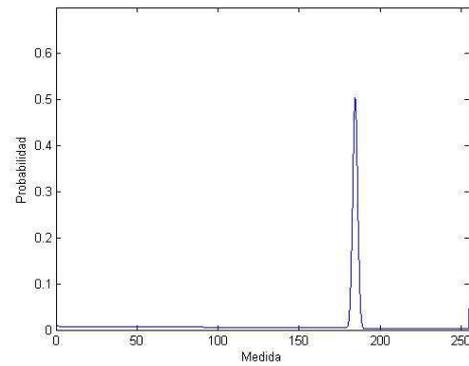
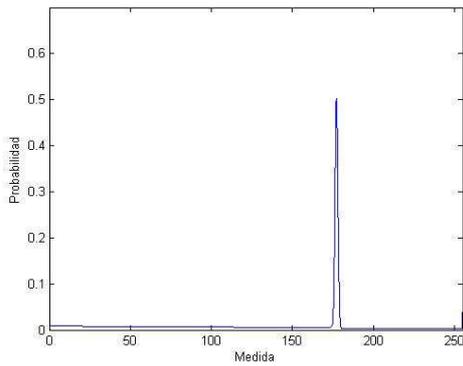


Ilustración 73: Robot en movimiento a 40 cm Ilustración 74: Robot en movimiento a 45 cm

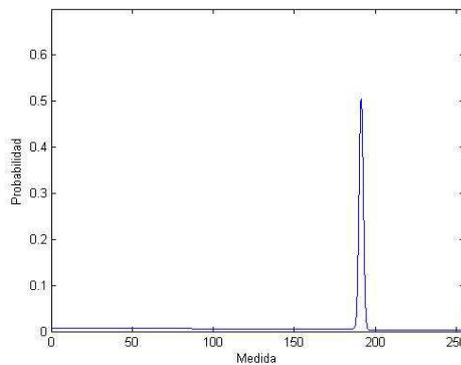


Ilustración 75: Robot en movimiento a 50 cm

○ **RESULTADOS PARA EL ULTRASONIDOS (Robot en movimiento).**

De igual manera, se muestran las gráficas con los resultados obtenidos con el sensor de ultrasonidos.

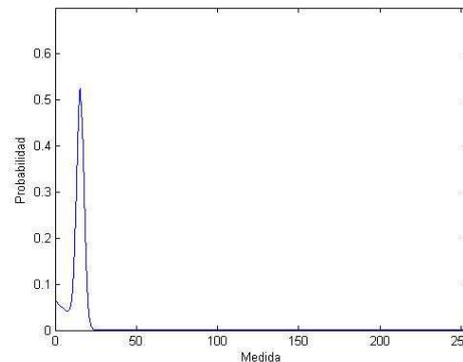
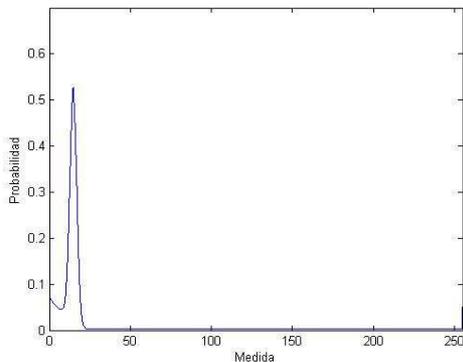


Ilustración 76: Robot en movimiento a 30 cm Ilustración 77: Robot en movimiento a 35 cm

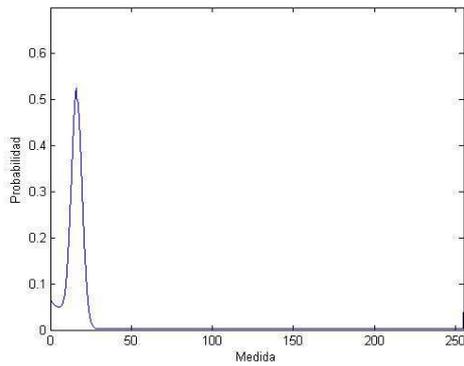


Ilustración 78: Robot en movimiento a 40 cm

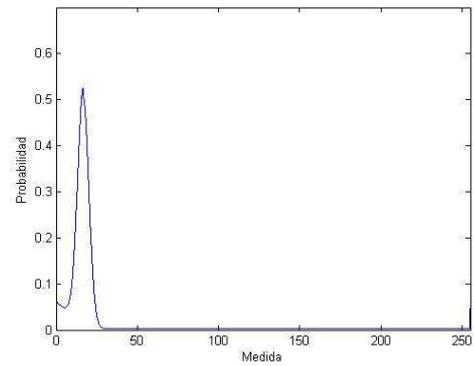


Ilustración 79: Robot en movimiento a 45 cm

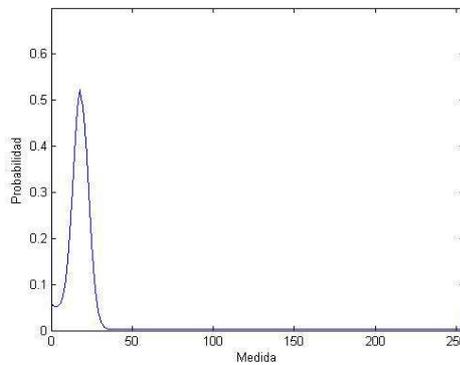


Ilustración 80: Robot en movimiento a 50 cm

Realizando una comparativa entre los resultados con el robot parado y los resultados con el mismo en movimiento, se aprecia claramente que los resultados del sensor IR son muy parecidos. Sin embargo, en los del ultrasonido la gaussiana presenta una dispersión mucho mayor en movimiento que en parado, proporcionando una primera conclusión de la poca fiabilidad del dispositivo.

Teniendo en cuenta estos resultados, se decidió realizar una prueba de campo y poner el robot a trabajar autónomamente, viendo *"in situ"* cuál era el comportamiento de este. Se comprobó con mucha claridad que el robot reaccionaba a distancias no programadas, con ligeros movimientos de corrección de trayectoria, donde incluso la presencia de algún obstáculo próximo era inexistente.

El hecho de que al sensor de ultrasonidos le afectara tanto los errores aleatorios y los errores por obstáculos inesperados del modelo de medición, que a su vez han sido

comprobados con el algoritmo probabilístico *"beam_range_finder_model"*, hizo desestimar al sensor de ultrasonidos en beneficio de un segundo sensor IR. Esto llevó consigo otro pequeño estudio de este nuevo sensor IR para comprobar su correcto funcionamiento, al igual que el comportamiento del mismo.

6.4.ESTUDIO DEL SENSOR IR DE NUEVA INCORPORACIÓN.

Una vez incorporado el nuevo sensor IR, en sustitución del sensor de ultrasonidos que se tenía como primera opción, se le realizó el mismo estudio con el fin de determinar su correcto funcionamiento. Los resultados, mostrados en las ilustraciones 81 a 91, son prácticamente los esperados, comprobando que el grado de dispersión de la gaussiana es ligeramente mayor que en el primer sensor y que las medidas se han desplazado hacia la derecha (en comparación con el primer IR), lo que indica que el rango del sensor es menor pero que igualmente se encuentra dentro de nuestro rango de trabajo.

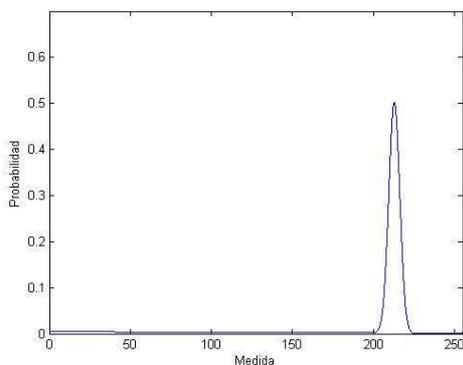


Ilustración 81: Robot parado a 35 cm

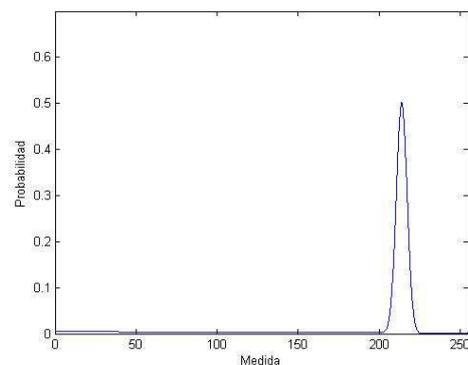


Ilustración 82: Robot parado a 36 cm

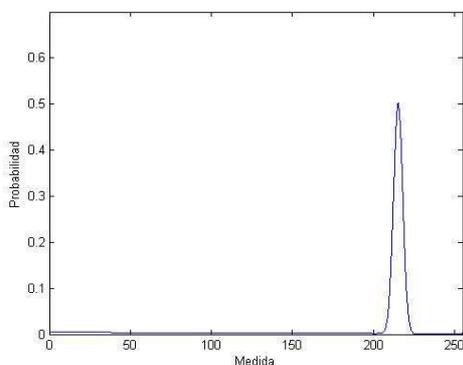


Ilustración 83: Robot parado a 37 cm

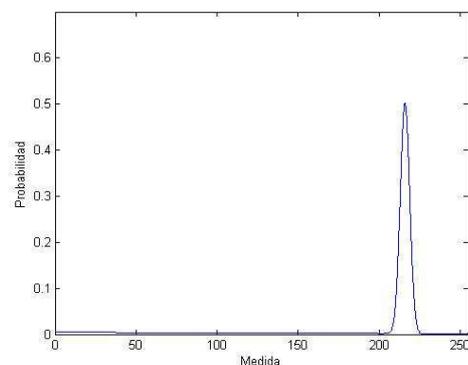


Ilustración 84: Robot parado a 38 cm

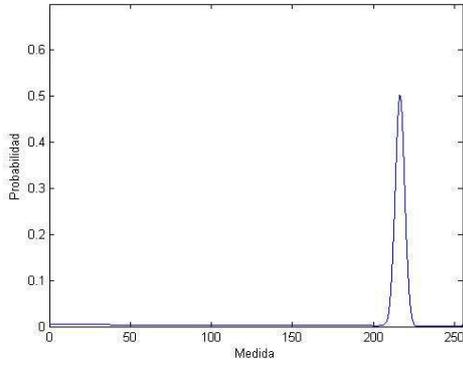


Ilustración 85: Robot parado a 39 cm

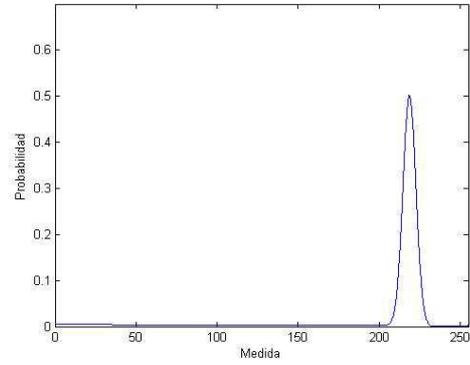


Ilustración 86: Robot parado a 40 cm

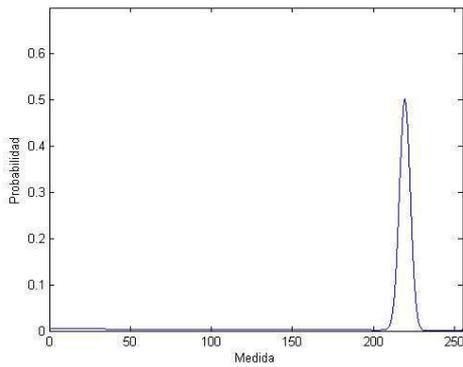


Ilustración 87: Robot parado a 41 cm

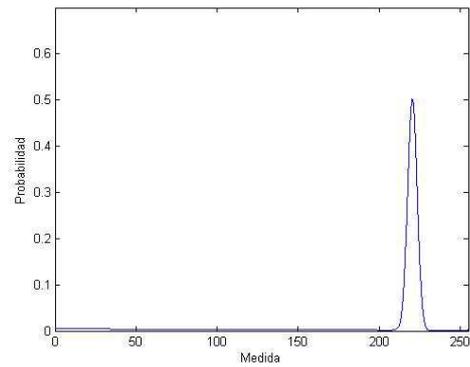


Ilustración 88: Robot parado a 42 cm

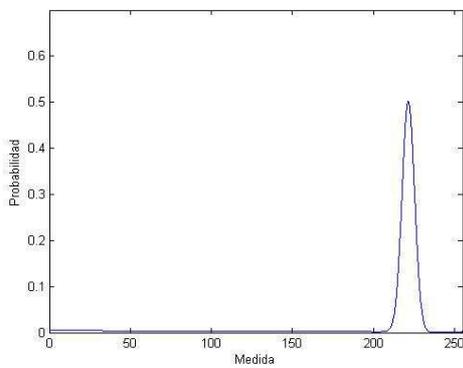


Ilustración 89: Robot parado a 43 cm

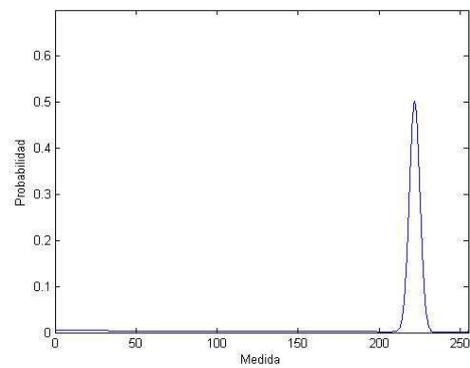


Ilustración 90: Robot parado a 44 cm

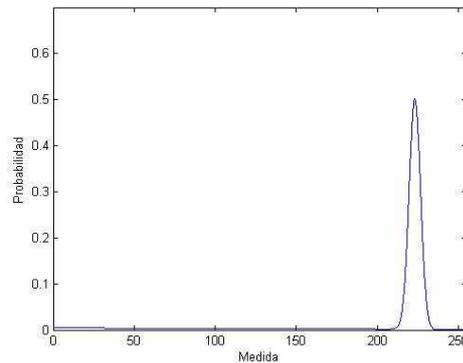


Ilustración 91: Robot parado a 50 cm

También se aprecia que tanto la exponencial, como la uniforme, que proporcionan información sobre los errores por obstáculos inesperados y errores aleatorios que presenta el sensor, son mínimas. Este hecho, al igual que en el sensor de ultrasonidos, llevo a una prueba de campo que, en este caso, produjo un movimiento estable del robot hasta llegar a su obstáculo más próximo, parándose a la distancia programada y haciendo de él un dispositivo correcto para el trabajo a realizar.

Vistas las pruebas realizadas hasta ahora, se puede determinar, que para el proyecto que se pretende realizar, no es tan importante la dispersión que pueda tomar la gaussiana en torno al valor medio, como la capacidad que pueda poseer el sensor de dar medidas erronas a objetos inesperados o medidas “*fantasmas*” debidas a los errores aleatorios. Presentando el robot un comportamiento inestable con estos tipos de medidas, donde variaba su trayectoria sin motivo alguno.

Capítulo 7. RECORRIDO Y MAPEO

Una vez determinados los componentes que incorpora el robot para la adquisición de datos, al que debemos añadir la brújula magnética, que proporcionara la posición del robot con respecto al norte magnético y el valor de los encóders (distancia recorrida), se dispone de los elementos necesarios para comenzar a graficar. Para poder mapear el entorno se hace imprescindible realizar con anterioridad la localización del robot, permitiendo tomar a esta como referencia.

Disponemos pues de la brújula, los encóders y dos sensores IR. Con respecto a los sensores, uno se encuentra en el frontal del robot, enfocado en el sentido de avance de las ruedas y otro de manera perpendicular, a 90 grados de las mismas. De estos sensores solo interesará, como se verá más adelante, el situado a 90 grados (proporcionará la distancia entre el robot y el obstáculo o pared), el delantero solo será usado para evitar la colisión del robot con cualquier tipo de elemento que pueda aparecer en su trayectoria. En el caso particular de graficar el recorrido, tan solo se harán necesarios los datos suministrados por los encóders y la brújula.

Una vez planteados los elementos de los que disponemos es preciso aclarar mínimamente el concepto de traslación de los ejes de coordenadas, pues es el elemento básico que será utilizado en el algoritmo para poder graficar correctamente.

- **TRASLACIÓN**

Sea un punto P de coordenadas (x,y) con respecto de los ejes rectangulares X e Y , se pueden obtener las ecuaciones que relacionan las coordenadas (x',y') del mismo punto P con respecto al nuevo referencial también rectangular X' e Y' .

Sean los nuevos ejes X' e Y' obtenidos por una traslación paralela y en el mismo sentido con respecto a los ejes X e Y , al nuevo origen (h,k) . (Ilustración 92)

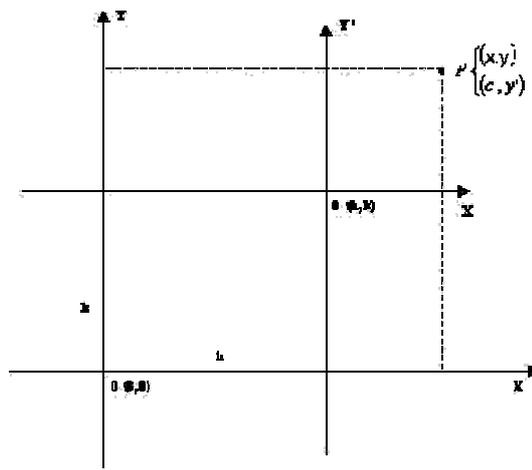


Ilustración 92: Traslación de coordenadas

De la ilustración 92 se obtiene de inmediato: $x = x' + h$; $y = y' + k$

7.1.RECORRIDO

Como primer intento de graficar el recorrido del robot se realizó una primera aproximación. Esta no proporciono el resultado esperado pero fue el determinante del algoritmo final. En ella básicamente se intento graficar el recorrido del robot realizando un cuadrado, pero como se puede apreciar en la ilustración 93, el resultado fue algo parecido a una estrella.

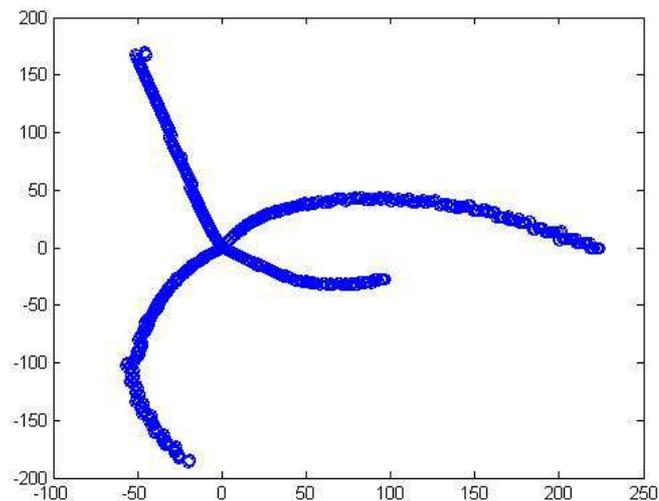


Ilustración 93: Primeras pruebas, graficando recorrido robot

No obstante, de la figura se extrajo la conclusión de que, aunque ligeramente curvadas, se disponía de dos trayectorias horizontales y otras dos verticales, con las que (con un poco de imaginación) formar algo parecido a un cuadrado, haciendo que el final de una trayectoria coincidiera con el principio de la otra. El algoritmo, por tanto, no contemplaba el concepto de traslación explicado un poco más arriba y representaba los puntos obtenidos (pasando de coordenadas polares a cartesianas) desde el origen de coordenadas.

Con esta conclusión se llegó al algoritmo presentado en la ilustración 94. En él se aprecia un bucle que no dará fin hasta que no hayamos tratado todas las medidas obtenidas. Dentro de este se realizan los tratamientos necesarios como la conversión de coordenadas y la traslación de las mismas, que conformarán el resultado deseado.

No obstante, hay que destacar el tratamiento de los valores no como se obtienen directamente, sino como el incremento que hay desde el último valor tomado hasta el actual. Esto es así ya que el valor de los encóders es monótonamente creciente hasta que encuentra un obstáculo, se detiene y realiza un giro. Este incremento es el estrictamente necesario al haber incorporado las ecuaciones de traslación.

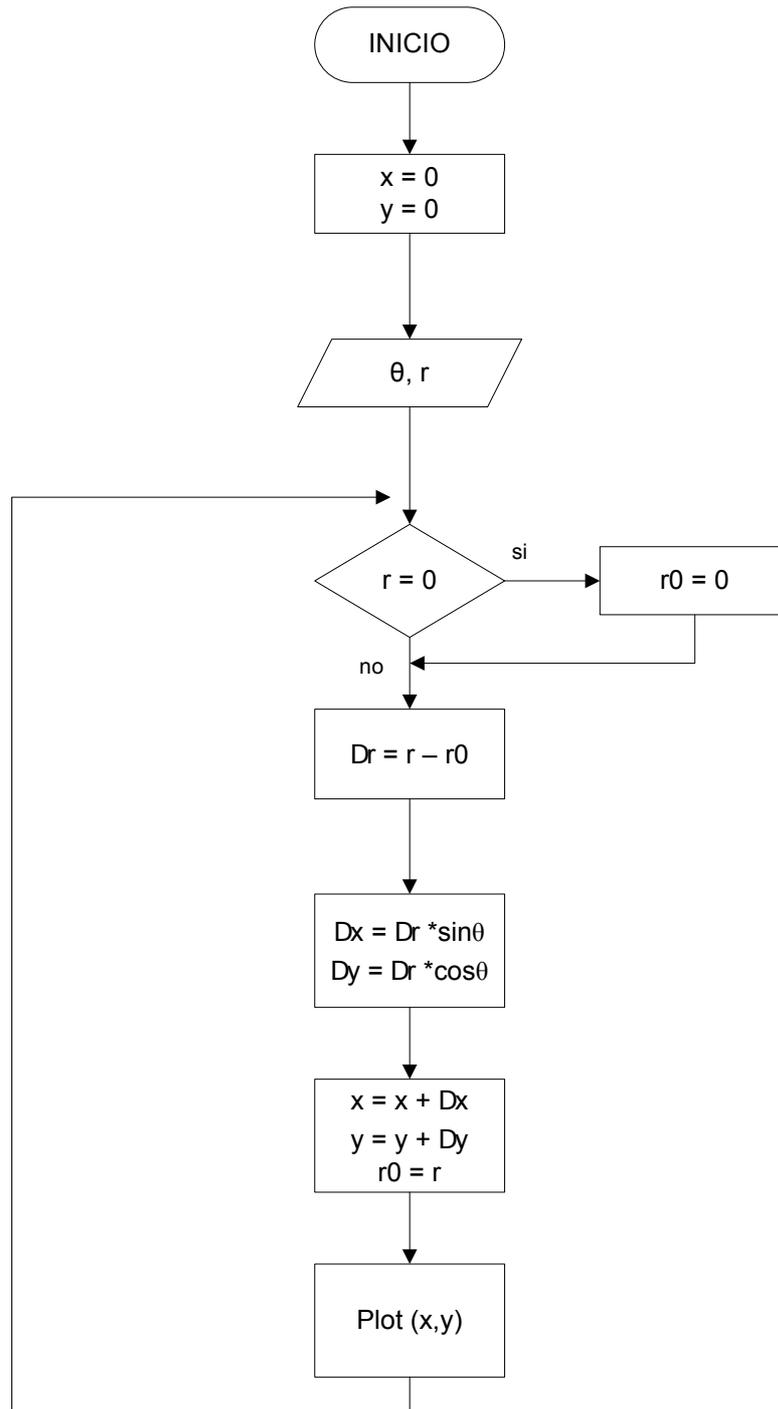


Ilustración 94: Diagrama de flujo del algoritmo para el recorrido

En la ilustración 95 se observa lo anteriormente comentado. Disponemos de los valores de los encóders (en aumento) pero al realizar la traslación, solo necesitamos la distancia recorrida desde el punto final al inicial, siendo el final de este el inicial del siguiente.

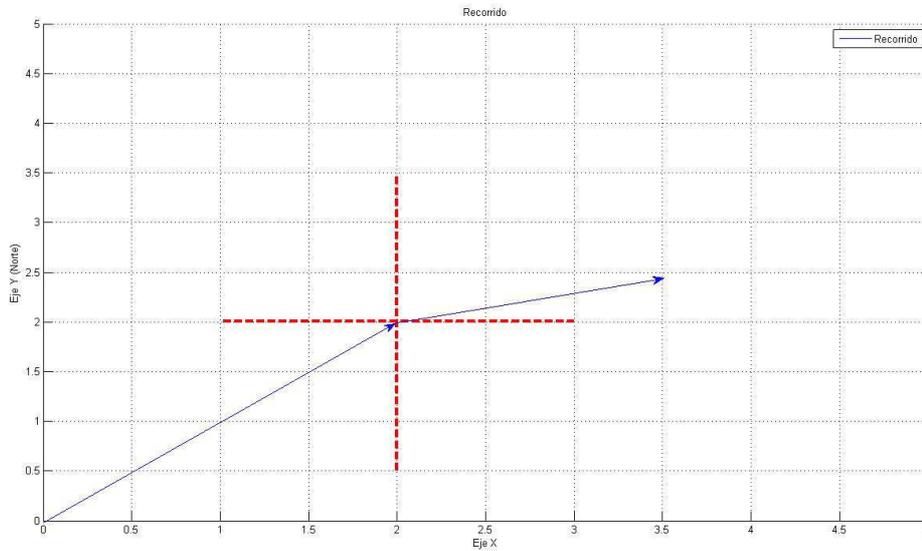


Ilustración 95: Recorrido y traslación

Como resultado del algoritmo se obtuvo la siguiente grafica (ilustración 96), en la que se muestra el recorrido del robot con un grado de similitud elevado (se pretendió que el robot realizara un recorrido en forma de cuadrado). Teniendo en cuenta que los errores en la captación de medidas o en su posterior tratamiento, se pueden minimizar pero casi imposible de acabar con ellos por completo.

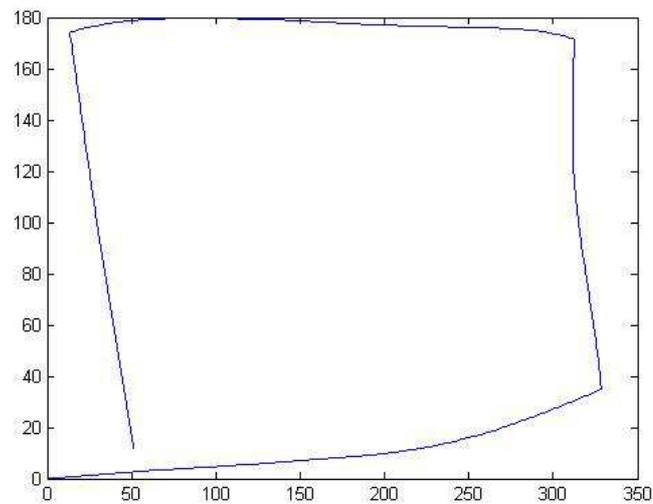


Ilustración 96: Gráfica del recorrido del robot

7.2.MAPEO

A la hora de realizar el mapeo, entra en juego el valor del sensor, incluyendo un tercer valor con el que trabajar. La primera idea adoptada fue que como se estaba trazando un cuadrado, el que formaba la habitación donde se encontraba el robot, y sabiendo que el sensor forma un ángulo de 90 grados con la trayectoria que sigue el robot, simplemente se debían desplazar las trazadas horizontales, una distancia proporcional al valor del sensor IR en el eje de ordenadas, y la misma operación para las trazadas verticales, pero en este caso en el eje de abscisas.

Los resultados obtenidos, mostrados en la figura de la ilustración 97, fueron bastante prometedores ya que mostraba correctamente los obstáculos que se había encontrado el robot en el transcurso del trayecto.

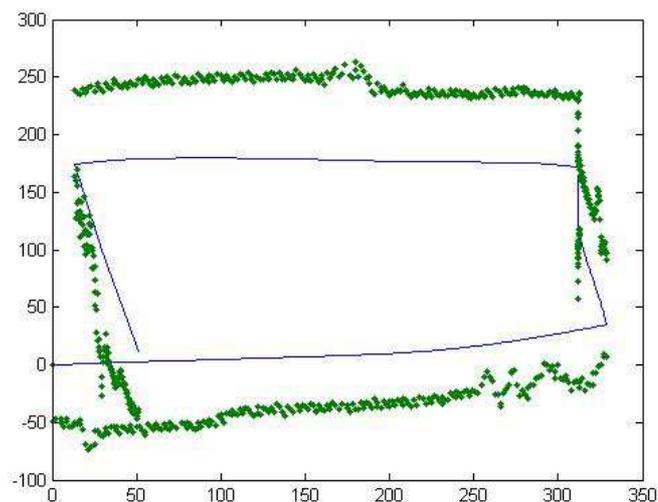


Ilustración 97: Primer intento de mapeo

Pero estudiando de nuevo la gráfica, se concluye, que si se pretendía dibujar trayectorias completamente horizontales o verticales sería correcto, pero no así cuando el robot tomara trayectorias diferentes, puesto que la coordenada necesita de la variación (proporcionalmente) de sus dos componentes (tanto en el eje de abscisas como en el de ordenadas).

La ilustración 98 muestra como si se modifican los parámetros en “x” se obtiene un ajuste de los resultados obtenidos pero perjudica a los parámetros en componente “y”.

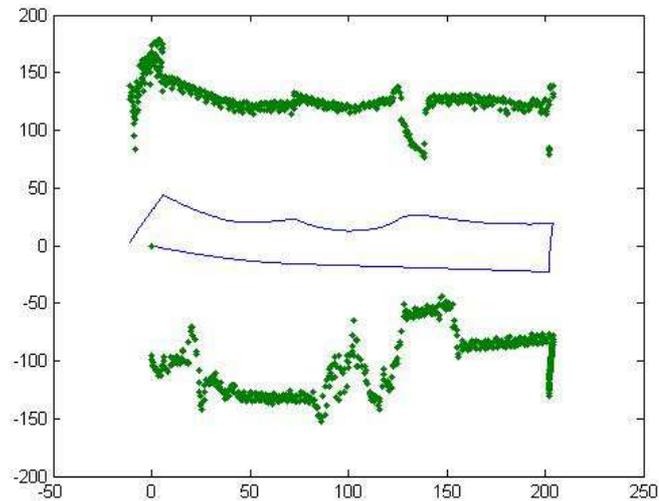


Ilustración 98: Resultado erróneo de mapeo

Al volver sobre los pasos realizados, se encontraba la solución al problema, el algoritmo aplicado para graficar el recorrido era el mismo que se necesitaba para mapear correctamente, ya que la medida que proporciona el sensor no deja de ser una distancia. Para ello solo se debe aplicar el mismo concepto de traslación, pero en vez de tomar la siguiente medida del encoder tomar la del sensor.

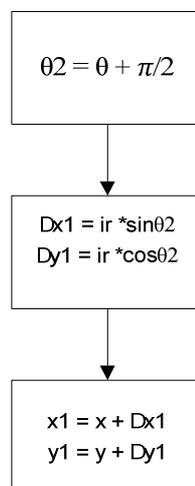


Ilustración 99: Operaciones para mapeo

En la ilustración 99 se observan las operaciones necesarias. Como el sensor se encuentra en un ángulo de 90 grados respecto a la dirección del robot, se deben sumar esos 90 grados al valor de la brújula. Así mismo se debe realizar la conversión de coordenadas polares a cartesianas, teniendo en cuenta este nuevo ángulo y el valor del sensor, que proporcionará el incremento de distancia desde la posición del robot. Por último se obtienen las coordenadas de traslación, pero en este caso las de los objetos u obstáculos que el robot ha ido encontrando a lo largo de su recorrido.

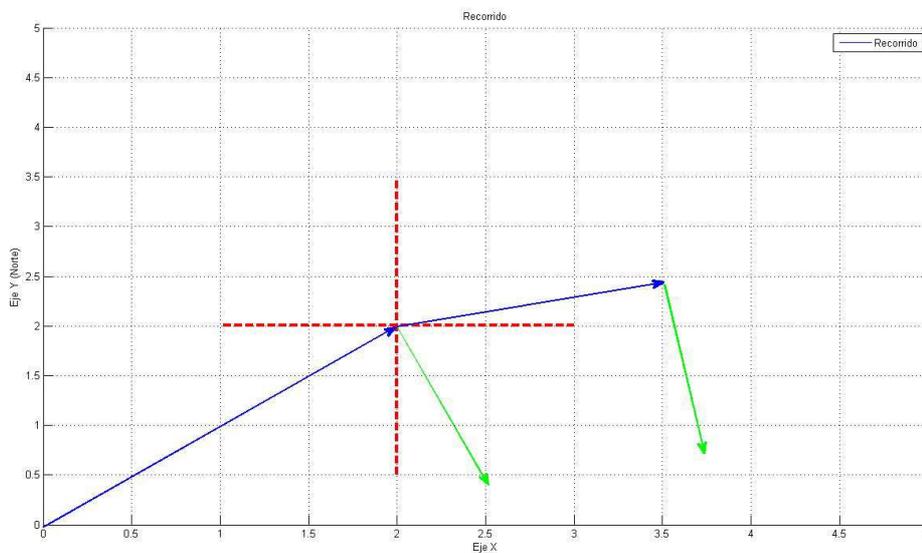


Ilustración 100: Recorrido, traslación y mapeo

Como este algoritmo depende de los valores que se utilizan para graficar el recorrido, el valor del sensor debe incorporarse dentro del mismo algoritmo para graficar el mapeo, quedando el flujo del algoritmo final como muestra la ilustración 101.

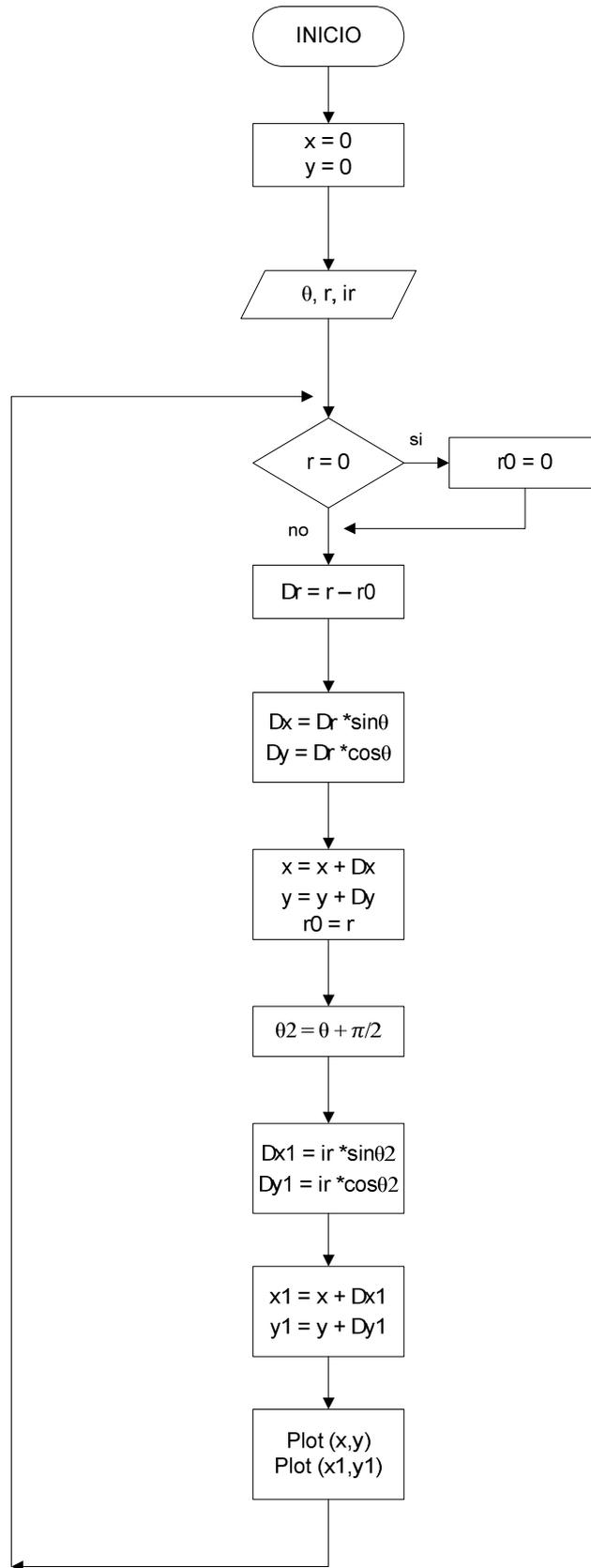


Ilustración 101: Diagrama de flujo. Recorrido y Mapeo

Los resultados obtenidos fueron positivos. La ilustración 102 muestra estos resultados, faltando por optimizar las esquinas en la realización del mapeo.

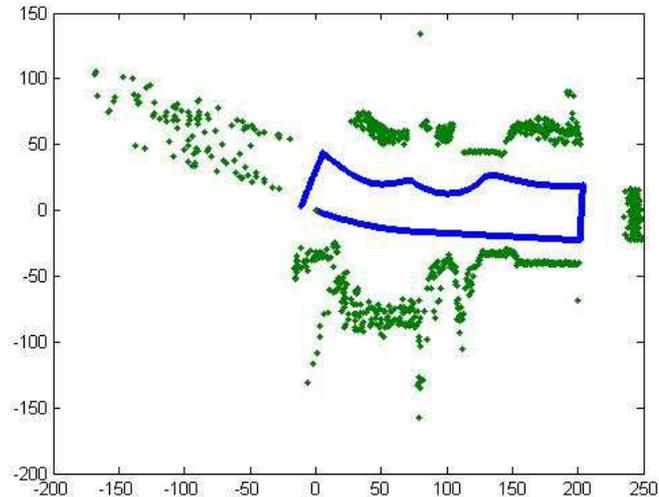


Ilustración 102: Mapeo 1

Para ello es necesario modificar el código de captación de medidas. Debido a que cuando el robot encontraba un obstáculo de frente, éste se paraba y realizaba un giro buscando de nuevo quedarse paralelo al obstáculo, el robot dejaba las esquinas sin mapear. Esto es así porque en ese preciso momento el robot dejaba de tomar mediciones.

Una vez modificado el código, se realizó una pequeña prueba en la que el robot tan solo realizaba el recorrido obteniendo medidas de dos paredes que formaban una esquina. Como se aprecia en la ilustración 103 el error fue solventado, proporcionando mediciones mientras realizaba el giro, que incorporadas a las tomadas mientras avanza su trayectoria, hacen posible la realización de un mapa más preciso y continuo.

diversas pruebas. En la ilustración 105 se visualiza el recorrido que realizará el robot alrededor del estrado donde los profesores imparten sus clases magistrales.



Ilustración 105: Estrado de una clase de la ETSE

En él se han incorporado un par de cajas de cartón para que el recorrido fuese un poco más interesante, haciendo que el robot modifique su trayectoria para no chocar con ellos. En la ilustración 106 se obtiene el resultado del recorrido mostrado en la ilustración 105, donde disponemos del recorrido (dibujo continuo) y su correspondiente mapeo (dibujo punteado). En la parte de la esquina, siguiendo el recorrido del robot, se aprecia claramente como el robot se para en un punto al detectar obstáculo y realiza un giro mientras sigue mapeando, a su correspondiente distancia, la forma del hueco entre la puerta y el estrado de la clase. Siguiendo la grafica en sentido ascendente, realiza un giro hacia la derecha bordeando la esquina del estrado y donde encuentra la primera caja, la sorteas y corrige su trayectoria para volver a conseguir la distancia programada a la que seguir mapeando el entorno. Finalmente, encuentra la segunda caja donde el procedimiento se repite.

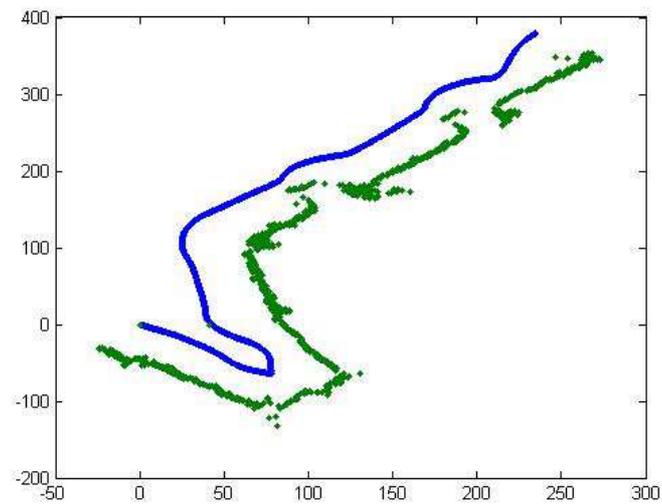


Ilustración 106: Recorrido y mapeo del estrado de una clase

Continuando las pruebas, se decidió improvisar en el mismo escenario, un entorno cerrado (ilustración 107) e intentar conseguir que el robot comenzara y finalizara su recorrido en el mismo punto.

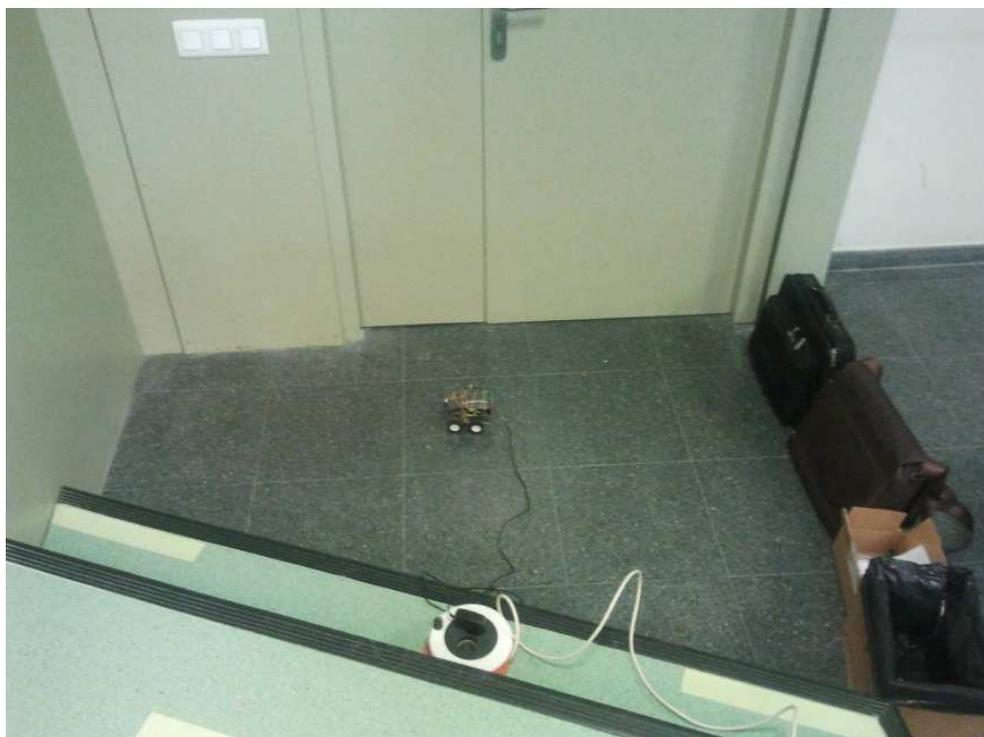


Ilustración 107: Improvisación de un entorno cerrado

En el resultado mostrado en la ilustración 108, se pueden destacar algunos puntos:

- Las medidas aportadas por los sensores y encóders no acaban de estar bien tratadas, puesto que la zona correspondiente al estrado (completamente recta) describe una parábola que no corresponde correctamente con la realidad.
- Se aprecia que el robot comienza en un punto pero no consigue acabar en él, o por lo menos dentro de la misma trayectoria.

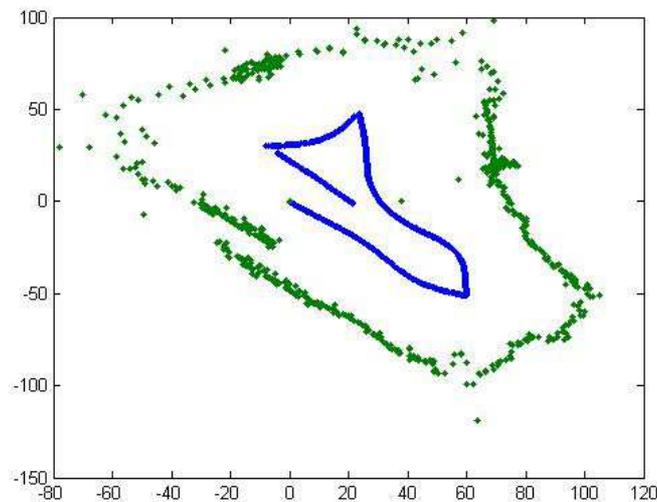


Ilustración 108: Recorrido y mapeo (1) en entorno cerrado

Para intentar mejorar estos aspectos, se realizaron modificaciones en el código interno del microprocesador. En él se debía tener en cuenta que cuando el robot intenta corregir su trayectoria, una vez que se encuentra fuera de la distancia programada, una de las ruedas se acelera más que la otra con el fin de hacer girar al robot.

En los resultados obtenidos, mostrados en la ilustración 109, se obtiene un dibujo en el que se mejora el trazado del recorrido (ósea del robot) y del mapeo, ya no aparece la nombrada parábola, que ha sido sustituida por una recta que corresponde con el obstáculo encontrado.

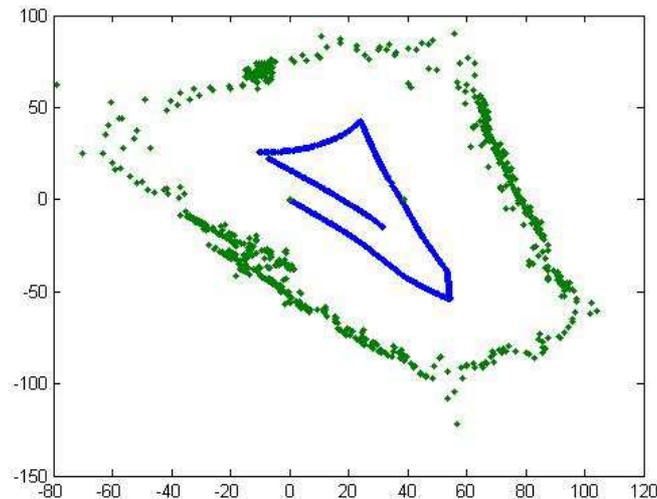


Ilustración 109: Recorrido y mapeo (2) en entorno cerrado

De igual manera, las medidas para el mapeo, del principio y final del movimiento se han aproximado casi llegando a coincidir.

Con respecto al recorrido, la parábola que aparece en la parte superior de la figura se debe a que el robot efectivamente ha realizado esa trayectoria. Esto es así ya que los obstáculos encontrados (mochilas de los portátiles) son muy oscuras. Por las características que presenta el sensor IR, le cuesta llegar a detectar los colores oscuros y por lo tanto el robot se fue aproximando a ellos. Otro aspecto del recorrido a tener en cuenta, es que este, al moverse el robot de forma autónoma y entre un rango de valores de distancia al obstáculo de entre 38 y 42 cm, se puede encontrar dentro de este rango y por lo tanto no modifica su trayectoria para acabar exactamente donde empezó.

La última prueba realizada, para comprobar sus resultados respecto a los obtenidos, fue implementar el algoritmo tal y como se presentó en el apartado teórico, en el que no se usa el valor de la brújula y se obtienen los valores de la variable θ mediante ecuaciones matemáticas, usando los valores de los encoders y la distancia comprendida entre una rueda y otra. Los resultados fueron peores que los obtenidos usando la brújula y por lo tanto se decidió desestimar esta opción.

Capítulo 8. CONCLUSIONES

El objetivo principal de este proyecto, realizar mapas con un robot móvil, se ha logrado alcanzar satisfactoriamente. Para conseguirlo, se ha tenido que trabajar, realizando un estudio previo, con algoritmos de caracterización de datos procedentes de los sensores del robot. Al trabajar con estos algoritmos se ha conseguido procesar sus resultados para utilizarlos en otros algoritmos. Estos últimos, son utilizados hoy en día en investigaciones de mapeo y localización de robots móviles.

La localización de un robot en un entorno conocido puede parecer una operación trivial, pero como se ha podido comprobar a lo largo del trabajo realizado, se hace necesario realizar un exhaustivo estudio de los componentes incorporados. El mal funcionamiento de tan solo uno de ellos repercutirá enormemente en el resultado final.

Aun sabiendo que los componentes funcionan correctamente, se encuentran errores procedentes de las mediciones realizadas. Se pueden obtener errores sistemáticos (deterministas) que en un principio se deberían poder solucionar mediante una calibración, pero que también dependerá su solución de la fiabilidad de las herramientas con las que se trabaje. Entre los errores sistemáticos que podemos

encontrar hay que destacar dos, los producidos por una desalineación de las ruedas y los producidos por la incertidumbre de las mismas, que presenten diferentes diámetros, deformaciones, etc.

Por otro lado también se encuentran errores no sistemáticos, más difíciles o casi imposibles de corregir. Las mediciones pueden dar resultados erróneos debidos a que las ruedas patinen, o a que el entorno donde trabaje puede presentar un suelo desigual.

De igual manera, en las ecuaciones que conforman los algoritmos, ha aparecido la resolución de los encóders, contra mayor sea esta menor será el error introducido. Haciendo ver que a mayor presupuesto, mejores componentes y por lo tanto mayor precisión.

Al ir introduciendo todos estos errores se puede concluir que el sistema utilizado dará unos resultados estimados, en los que se deberá tener en cuenta que a mayores distancias recorridas, los errores se irán acumulando y produciendo un peor resultado final.

Teniendo todo esto en cuenta, y siendo un proyecto educativo que no disponía de presupuesto ilimitado como el que puede tener una gran empresa, es gratificante comprobar que los resultados obtenidos han cumplido con creces los objetivos planteados. Se ha conseguido realizar un robot completamente autónomo y con las características necesarias para realizar un correcto mapeo y que este sea identificable con el entorno en el que se encontraba. Para ello se han empleado gran parte de los conocimientos adquiridos durante los años de estudio en la universidad, pero se han tenido que ampliar al tratar con componentes concretos, de fabricantes concretos, que aportan sus propias herramientas de desarrollo y que en algunas cuestiones requieren de una previa formación para ir familiarizándose con ellas.

Finalmente cabe comentar que se han conseguido otros objetivos alternativos al proyecto en sí. Entre ellos destacar la sincronización de trabajo formando un equipo, en este caso reducido, del personal involucrado, y por otro lado aumentar la capacidad

de investigación encontrando solución a problemas que se han ido planteando y aprendiendo de ellos.

Gracias al exhaustivo trabajo realizado por mi compañero, incluyendo tanto el desarrollo como la optimización del robot, se ha podido realizar un gran estudio y en consecuencia llegar a los resultados obtenidos.

Capítulo 9. BIBLIOGRAFÍA:

[1].*Fundamentos de robótica. Mc Graw Hill*

Autores: Barrientos, Antonio. Peñin, Luis Felipe. Balaguer, Carlos. Aracil, Rafael.

[2].*Leyenda del “Hombre de palo”*

<http://www.toledo-turismo.com/turismo/contenido/mas-toledo/culturarte/ciudad-leyendas/hombre-palo.aspx>

Patronato Municipal de turismo. Ayuntamiento de Toledo.

[3].*Robótica: manipuladores y robots móviles. Ed. Marcombo. Boixareu editores.*

Autor: Aníbal Ollero Baturone.

[4].*Robots sociales: el juego de interaccion*

Autor: chus

<http://www.tucamon.es/contenido/robots-sociales-el-juego-de-la-interaccion>

[5].*Robonaut2. http://www.nasa.gov/mission_pages/station/main/robonaut.html*

[6].*Probabilistics Robotics. (Intelligent Robotics and Autonomous Agents series)*

Autor: Sebastian Thrun.

[7].Artículo: estimación de posición por odometría.

Autor: Cuentos cuánticos.

[8].Modelo cinemático de un robot móvil tipo diferencial y navegación a partir de la estimación odométrica.

Autores: Valencia v., Johnny a.; Montoya o., Alejandro; ríos, Luis Hernando.

[9].Modelado cinemático y dinámico de un robot móvil omni-direccional.

Autores: V. F. Muñoz Martínez, G. Gil-Gómez y A. García Cerezo.

[10].Algoritmos de construcción de mapas para la navegación de robots, basada en información procedente de Sensores de ultrasonidos"

Autor: Milagros Martínez Díaz

[11].Construcción del estímulo Obstáculo sobre un mapa de rejilla para navegación de un robot en exteriores

Autores: Lía García-Pérez, José María Cañas, María C. García-Alegre

[12].Robot Motion Planning

Autor: Ganesh Swaminathan

[13].Merging occupancy grid maps from multiple robots

Autores: Andreas Birk Member, IEEE, Stefano Carpin Member, IEEE

[14].Using occupancy grids for mobile robot perception and navigation

Autor: Alberto Elfes

[15].Mobile Robot Localization and Mapping with Uncertainty using Scale-Invariant Visual Landmarks

Autores: David Lowe, Jim Little

[16].Intelligent control of nonholonomic mobile robots with fuzzy perception

Autores: F. Cuesta, A. Ollero, B.C. Arrue, R. Braunstingl

[17]. "Simultaneous localization and map building of skid-steered robots," Robotics Automation Magazine, IEEE, vol. 14, no. 1, pp. 79-89, 2007.

Autores: G.C. Anousaki and K.J. Kyriakopoulos.

[18]. Generación automática de mapas en espacios cerrados mediante robots móviles

Autor: David González Arjona

Resum:

El projecte tracta de la fabricació d'un robot mòbil que sigui capaç de realitzar el mapeig del seu entorn, evitant els obstacles que es pugui trobar al transcurs del seu recorregut. Per tal d'obtenir els resultats desitjats, s'ha realitzat la caracterització dels seus components emprant algorismes probabilístics, amb els que s'ha pogut determinar la seva eficiència. Finalment s'han realitzat els algorismes corresponents per graficar el mapeig del recorregut del robot i del seu entorn, aconseguint el objectiu plantejat.

Resumen:

El proyecto trata de la fabricación de un robot móvil que sea capaz de realizar un mapeo de su entorno, evitando los obstáculos que se pueda encontrar en el transcurso de su recorrido. Para obtener los resultados deseados, se ha realizado la caracterización de sus componentes utilizando algoritmos probabilísticos, con los que poder determinar su eficiencia. Finalmente se han realizado los algoritmos correspondientes para graficar el mapeo de su recorrido y entorno, consiguiendo el objetivo planteado.

Summary:

The project deals with the manufacture of a mobile robot that is able to plot a map its environment, avoiding obstacles you may encounter during its journey. To get the desired results, was performed the characterization of its components using probabilistic algorithms with which to determine its efficiency. Finally, the corresponding algorithms have been made to plot the route and mapping its environment, achieving the objective.

