

**UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA**

**Carrera de Ingeniería Electrónica**



**DESARROLLO DE UN SISTEMA DE  
EXPLORACIÓN Y MAPEO EN TRES  
DIMENSIONES PARA UN ROBOT MÓVIL**

Tesis para optar el título profesional de Ingeniero Electrónico

**José María Muñoz Huamán**

**Código 201210145**

**Asesor**

Oscar E. Ramos

Lima - Perú

Octubre 2019

*Dedicatoria:*

*A mi madre Flor de Luz.*

*Por haberme apoyado en todo momento, por sus consejos, sus valores, por la motivación constante que me ha permitido ser una persona de bien, pero más que nada, por su amor.*

*Agradecimientos:*

*A Oscar Ramos, por su amistad, paciencia y confianza depositada en mí.*

*A Victor Murray, por su amistad y paciencia, además por haber tomado su tiempo en leer este trabajo de tesis.*

*A mis amigos que me brindaron siempre su apoyo y por todos sus consejos. Juan Llanos, Luis Sanchez, Luis Montaña y Franco Lama.*

# TABLA DE CONTENIDO

## Resumen

<b>1. Introducción</b>	<b>1</b>
1.1. Problemática . . . . .	1
1.2. Justificación . . . . .	2
1.3. Alcances y limitaciones . . . . .	3
1.4. Organización de la tesis . . . . .	4
<b>2. Marco Teórico</b>	<b>5</b>
2.1. Robótica en Minería . . . . .	5
2.2. Control de Robots Móviles . . . . .	7
2.2.1. Control Cinemático PID . . . . .	8
2.2.2. Control Cinemático Polar . . . . .	9
2.3. Navegación del Robot Móvil . . . . .	10
2.3.1. Navegación Basada en Campos Potenciales . . . . .	11
Campo Potencial Atractivo. . . . .	12
2.3.1.1. Campo Potencial Repulsivo . . . . .	12
2.3.2. Movimiento Basados en Mapas Topológicos . . . . .	13
Mapa de Visibilidad Gráfica. . . . .	14
2.3.3. Movimiento Basado en Muestreo Aleatorio . . . . .	15
Mapa de Ruta Probabilístico. . . . .	15
Árbol Aleatorio de Exploración Rápida. . . . .	16
Movimiento basado en Mapa de Red. . . . .	17
2.4. Localización . . . . .	18
2.4.1. Modelo de movimiento y medición . . . . .	19
2.4.2. Filtros de Kalman . . . . .	20
2.4.3. Filtros de Partículas . . . . .	21
2.4.4. Filtros de Información . . . . .	22
2.5. Mapeo y Localización Simultánea ( <i>SLAM</i> ) . . . . .	22
2.5.1. <i>SLAM</i> basado en Filtro Extendido de Kalman . . . . .	23
2.5.2. <i>SLAM</i> basado en Filtros de Partículas . . . . .	24
2.5.3. GraphSLAM . . . . .	24

<b>3. Metodología</b>	<b>25</b>
3.1. Sistema Desarrollado . . . . .	25
3.1.1. Cinemática de Robot Móvil Diferencial . . . . .	26
3.1.2. Sistema de Percepción Bidimensional . . . . .	28
3.1.3. Sistema de Percepción Tridimensional . . . . .	30
3.1.4. Prototipo Final . . . . .	36
3.2. Generación de Trayectoria basada en Campos Potenciales . . . . .	37
3.3. SLAM basado en Filtros de Partículas . . . . .	39
3.3.1. Sistema Operativo del Robot (ROS) . . . . .	40
3.3.2. Estructura del Sistema de Autonomía del Robot Móvil . . . . .	42
3.4. Estructura del Sistema de Mapeo en Tres Dimensiones . . . . .	44
<b>4. Resultados</b>	<b>48</b>
4.1. Resultados del Controlador Polar . . . . .	48
4.2. Resultados del Sistema de Navegación . . . . .	50
4.3. Resultados de la Navegación Autónoma con el Lidar en dos dimensiones .	53
4.4. Resultado de la Navegación Autónoma usando SLAM en dos dimensiones	54
4.5. Resultados del mapeo en tres dimensiones con el sistema mecánico . . . .	58
4.5.1. Construcción tridimensional de una caja . . . . .	59
4.5.2. Pruebas del mapa tridimensional en el pasadizo dentro de un edificio	64
4.6. Resultado del mapa 3D mientras el robot se mueve . . . . .	67
<b>CONCLUSIONES</b>	<b>71</b>
<b>RECOMENDACIONES</b>	<b>74</b>

## ÍNDICE DE FIGURAS

2.1.	Representación gráfica de los sistemas holonómicos en la robótica móvil, donde en (a) se muestra el robot móvil omnidireccional y en (b) se muestra el robot móvil diferencial. . . . .	8
2.2.	Configuración para el control polar donde se muestran la posición del objetivo y los parámetros específicos(ver texto para más detalle). . . . .	9
2.3.	Representación gráfica de un espacio de configuración. En (a) se muestra el espacio de configuración con un punto de inicio y otro punto de llegada a la posición deseada y en (b) se puede ver el mismo espacio de configuración donde los sólidos geométricos representan los obstáculos. Las líneas continuas delimitan los bordes de la visibilidad gráfica para los tres obstáculos y las líneas entrecortadas representan el camino más corto entre el punto de inicio y el punto de meta. . . . .	15
3.1.	Representación gráfica de un robot móvil diferencial, donde $b$ es la distancia de su centro de masa hacia sus ruedas, el radio de cada rueda es representado por $r$ , y $\dot{\varphi}_l$ y $\dot{\varphi}_r$ son las velocidades de la rueda derecha y rueda izquierda respectivamente. . . . .	26
3.2.	Robot Kobuki, el cual es un robot móvil diferencial utilizado para la implementación del sistema de navegación autónoma. . . . .	27
3.3.	Sensor lidar RPLidar A2. . . . .	28
3.4.	En esta figura el punto verde representa el origen del sistema de referencia del sensor RPLidar A2, donde el eje $X$ va en el mismo sentido del cable de datos del sensor lidar. . . . .	29
3.5.	Diseño del sistema mecánico, desarrollado en un programa CAD. En (a) se muestra el diseño en CAD del sistema mecánico, en (b) se muestra la forma en que va ser colocado el sistema mecánico en el robot móvil kobuki y en (c) se muestra el diseño mecánico manufacturado en aluminio, junto al sensor lidar y el motor paso a paso. . . . .	31
3.6.	Representación gráfica de los sistemas de referencia del sistema mecánico, donde $M$ y $L$ son los sistemas de referencia del motor paso a paso y del sensor lidar, respectivamente. . . . .	32

3.7.	Representación gráfica de los movimientos de rotación del sistema mecánico. En (a) se muestra la posición inicial ( $0^\circ$ ) rotando hacia la izquierda, en (b) se muestra el sensor lidar rotado hacia el ángulo $-88^\circ$ y rotando hacia la derecha. En (c) se muestra que el sensor lidar regreso a su posición original ( $0^\circ$ ) y va rotar hacia la derecha. En (d) se muestra el sensor lidar en el ángulo $88^\circ$ rotando hacia la izquierda. . . . .	33
3.8.	En esta imagen se muestra las mediciones obtenidas del sensor RPLidar A2, dentro de un ambiente cerrado con forma rectangular. En (a) se muestra los datos del sensor lidar utilizando coordenadas polares en los ejes, y en (b) se muestra la conversión de los datos del sensor lidar a coordenadas cartesianas. . . . .	34
3.9.	Esta figura muestra a <i>muqi</i> , el prototipo final, que fue utilizado para realizar las pruebas de exploración autónoma y mapeo de un ambiente en tres dimensiones. . . . .	36
3.10.	En esta figura se muestra una simulación del algoritmo de campo potencial artificial. En (a) se muestra la prueba para las fuerzas de atracción, en (b) se muestra la prueba de las fuerzas de repulsión de cada uno de los obstáculos y en (c) se muestra la fuerza de navegación. . . . .	38
3.11.	Estructura del sistema para realizar la exploración del robot móvil diferencial, de forma autónoma. . . . .	42
3.12.	En esta figura se muestra de forma gráfica las etapas del sistema de autonomía del robot móvil. En (a) se muestra la etapa inicial que es el sensado de las distancias, en (b) se muestra la evasión de obstáculos del robot móvil, en (c) se muestra las posiciones aleatorias por donde el robot tiene que desplazarse para generar su trayectoria, y en (d) se muestra como el robot ya no sabe por donde avanzar y regresa a su posición inicial. . . . .	45
3.13.	Estructura del sistema para la construcción del mapa en tres dimensiones, mientras el robot móvil se va desplazando. . . . .	46
4.1.	Evolución temporal de las variables de estado usando el controlador polar para lograr una posición deseada dada por $x = 3m, y = 3m, \theta = 90$ . En (a) se muestra la evolución temporal en la posición del eje x, en (b) se muestra la evolución de la velocidad lineal, en (c) se muestra la evolución de la orientación y en (d) se muestra la evolución de la velocidad angular. . . . .	49
4.2.	Navegación autónoma implementada en el robot diferencial Kobuki. En (a) se muestra las fuerzas de atracción que llevan al Kobuki a la posición deseada, en (b) se muestra las fuerzas de repulsión que evitan que el robot diferencial se choque con los obstáculos y en (c) se muestra las fuerzas de navegación que genera la trayectoria del robot. . . . .	50
4.3.	Navegación autónoma del robot diferencial Kobuki, montado con un sensor lidar y dos cajas como obstáculos. . . . .	51

4.4.	Campos atractivos y repulsivos, y la trayectoria que el robot real sigue usando datos en línea provenientes del sensor lidar montado en la parte superior. En (a) se muestra las fuerzas de atracción, en (b) las fuerzas de repulsión y las posiciones de los obstáculos, finalmente en (c) se muestra las fuerzas de navegación generando la trayectoria. . . . .	52
4.5.	Robot móvil Kobuki en medio de los obstáculos colocados para probar la navegación autónoma utilizando el algoritmo SLAM. Esta prueba se realizó en el simulador Gazebo. . . . .	54
4.6.	Navegación autónoma del Kobuki utilizando SLAM. En (a) se muestra las fuerzas de atracción, en (b) se representa las fuerzas de repulsión y en (c) la suma de las fuerzas de atracción y repulsión. . . . .	55
4.7.	Mapa en dos dimensiones obtenido del algoritmo SLAM, el mapa es visualizado en la herramienta Rviz. . . . .	57
4.8.	Dimensiones de la caja donde se realizó las mediciones de forma interna, para generar la construcción en tres dimensiones. . . . .	58
4.9.	Construcción del sólido geométrico, a partir de las mediciones del sensor lidar. Desde las figuras (a) hasta (f) se nota el sólido tridimensional en torno a su eje en sentido horario para una mejor visualización. . . . .	60
4.10.	Histogramas de la cantidad de muestras con respecto a la posición en el eje $X$ del sólido geométrico. En (a) se muestra el histograma de la pared izquierda del sólido con una posición de $X = -0.44 m$ y en (b) se muestra el histograma de la pared derecha con una posición de $X = 0.44 m$ . . . . .	62
4.11.	Distribución de las mediciones con respecto a la posición ( $y$ ) de las paredes frontal y posterior de la caja real. . . . .	63
4.12.	Pasadizo dentro de un edificio, donde se colocó el sistema mecánico para realizar el mapa tridimensional del lugar. . . . .	65
4.13.	Mapa en tres dimensiones de un pasadizo. Desde las figuras (a) hasta (h) el mapa tridimensional rota en sentido horario con respecto a su eje vertical. . . . .	66
4.14.	Dimensiones del sótano donde se hizo las pruebas de exploración con el robot <i>muqi</i> . . . . .	67
4.15.	Exploración autónoma de <i>muqi</i> dentro de un ambiente cerrado. En (a) se muestra cuando <i>muqi</i> comienza a tomar mediciones, en (b) se observa cuando <i>muqi</i> comienza a explorar, en (c) el robot ingresa por el pasadizo a seguir explorando, en (d) <i>muqi</i> empieza a explorar el otro ambiente. En (e) se muestra que el robot esta moviéndose por el nuevo ambiente y en (f) se muestra que <i>muqi</i> ya no puede avanzar debido a las paredes y regresa a su posición inicial. . . . .	68
4.16.	Mapa generado por el algoritmo SLAM. Este mapa se construye mientras el <i>muqi</i> va explorando el sótano. . . . .	70

4.17. Resultado de las mediciones obtenidas, en el sótano, por el sensor lídar en la exploración del robot <i>muqi</i> . . . . .	71
4.18. Mapa en tres dimensiones del ambiente explorado con el robot <i>muqi</i> . En esta imagen se muestra diferentes vistas del sólido geométrico, en (a) se muestra la vista posterior; en (b) la vista superior; en (c) la vista frontal derecha y en (d) la vista frontal izquierda. . . . .	72

# Resumen

Actualmente, dentro de la minería subterránea, los topógrafos para realizar las mediciones de los túneles subterráneos y hacer una representación gráfica en tres dimensiones del ambiente se demoran entre 2 a 3 días. Esta demora genera ineficiencia dentro del proceso de extracción del mineral ya que no se tiene la información a tiempo real. Por ende, este trabajo de tesis se centra en el desarrollo e implementación de un sistema de localización y mapeo en tres dimensiones basado en sensado láser para la exploración autónoma de un robot móvil.

El robot móvil utiliza técnicas de SLAM (*Simultaneous Localization and Mapping*) para realizar un mapa en dos dimensiones dentro del túnel subterráneo y asimismo, estimar la posición del robot móvil dentro del ambiente mencionado. Se implementó un sistema de navegación basado en campos potenciales. Este algoritmo permite al robot generar su propia trayectoria, evitando colisionar con las paredes u obstáculos en el ambiente donde comenzará a explorar. Se toma la información del sensor lidar (*Light Detection and Ranging*) para generar el mapa en dos dimensiones.

Se elige el sensor lidar ya que emite impulsos rápidos de luz láser hasta 4000 impulsos por segundo, sobre una superficie. Para construir el mapa en tres dimensiones se diseñó un sistema mecánico que permite rotar el sensor láser haciendo que este pueda tomar medidas a diferentes alturas con su respectivo ángulo de inclinación. Se utiliza las transformaciones homogéneas, para convertir los datos obtenidos a valores en los tres ejes del plano cartesiano ( $X, Y, Z$ ). El mapa tridimensional obtenido se puede visualizar en un software de fácil interacción.

# Capítulo 1

## Introducción

En este primer capítulo se presenta una introducción a este trabajo de tesis. En la primera sección se comienza explicando la problemática existente en las empresas mineras, que trabajan con túneles subterráneos, y se termina explicando la motivación para desarrollar este trabajo. En las siguientes secciones se explican la propuesta de solución seguido de los alcances y limitaciones de esta tesis. Finalmente, se termina el capítulo con una descripción de la organización general de este documento.

### 1.1. Problemática

Actualmente, en la minería subterránea se utilizan diversos métodos de explotación para la extracción de minerales. Los más utilizados, en el Perú, son variaciones del método de corte y relleno ascendente como descendente. La presente tesis se enfoca en el método de corte y relleno ascendente, ya que es el más utilizado en el Perú [1]. En el método de corte y relleno ascendente, el mineral es arrancado por franjas horizontales y verticales empezando por la parte inferior de un tajo y avanzando verticalmente. Cuando se ha extraído la franja completa, se rellena el volumen correspondiente con material estéril, llamado “relleno”, que sirve de piso de trabajo a los obreros y al mismo tiempo permite sostener las paredes y el techo [2]. Asimismo, se utiliza grandes máquinas que construyen túneles subterráneos por donde se va a extraer los minerales.

La construcción de túneles subterráneos es de gran importancia para la minería subterránea, debido a que por medio de esta construcción se extrae los minerales de la veta mineralizada [3]. Estas construcciones son de diversos tamaños en las diferentes

unidades mineras del Perú, y dependen del tamaño de las máquinas que van a ingresar para la extracción del mineral. Durante el proceso de construcción pueden ocurrir diferentes accidentes (desprendimiento de rocas, emisión de gases, etc.) que ponen en peligro la vida de los trabajadores [4]. Para evitar los riesgos dentro de la minería subterránea, se necesita supervisar la zona de construcción y a su vez obtener una representación geométrica visual de toda la excavación [5]. Los especialistas encargados de la supervisión y la generación visual de un mapa topográfico del ambiente, dentro de la minería subterránea, son el topógrafo y el geomecánico. Este trabajo de tesis se inspira en las principales funciones del topógrafo dentro de la mina.

El topógrafo observa y mide las dimensiones del túnel subterráneo, para generar un mapa topográfico de éste. El túnel subterráneo es un ambiente bastante oscuro, el cual limita la visión del especialista generando imprecisión en sus mediciones y por ende en los mapas que generan. Esto es un problema en los mapas elaborados de estos ambientes. Para realizar la toma de mediciones dentro del túnel, el topógrafo se demora entre tres a cuatro horas y para generar un mapa en tres dimensiones, del lugar, se demoran entre dos a tres días. La demora en las funciones del topógrafo genera un retraso en el proceso de la extracción de minerales y a su vez proporciona poca información sobre la extracción de la veta mineralizada, por día. Las mineras ganan dinero por la cantidad de volumen extraído y asimismo paga a sus trabajadores por volumen trabajado. Por ende, la motivación de esta tesis es hacer más eficiente la labor del topógrafo apoyándose en el uso de la tecnología.

## **1.2. Justificación**

Debido al problema descrito en la sección anterior, este trabajo de tesis se centra en el desarrollo e implementación de un sistema de localización y mapeo en tres dimensiones basado en sensado láser para la exploración autónoma de un robot móvil.

Este sistema busca ofrecer una alternativa tecnológica de medición y la construcción tridimensional de un túnel subterráneo mediante el uso de un sensor láser. Además, la construcción se realiza mientras un robot móvil va explorando el ambiente. El desarrollo de este sistema beneficia a las personas que se encuentran involucradas dentro del proceso de extracción del mineral, debido a que el sistema reduce el tiempo que se demora un topógrafo en realizar el mapa tridimensional. Además, el sistema permite tener un mejor seguimiento a la producción de mineral al permitir conocer los volúmenes de mineral que se extrae cada período de tiempo durante un día. Asimismo, este sistema evita que el topógrafo se exponga a riesgos dentro de la minería subterránea. Al evitar estos tipos de accidentes se reduce costos y en caso de ocurrir algún imprevisto grave, el daño sería de forma directa hacia el robot y no hacia una persona.

### **1.3. Alcances y limitaciones**

La motivación, como se mencionó antes, es hacer más eficiente la labor del topógrafo dentro de un túnel subterráneo. Este proyecto de tesis es un prueba de concepto para demostrar que el sistema desarrollado funciona para explorar un ambiente desconocido y a su vez generar un mapa en tres dimensiones del lugar.

Los túneles subterráneos son ambientes con el terreno bastante accidentado, lleno de rocas y charcos de agua. Para este trabajo de tesis no se considera el desarrollo de un robot móvil para que se pueda desplazar en terrenos accidentados, como los túneles subterráneos de la minería. El objetivo de la tesis es el desarrollo de un sistema de exploración para un robot móvil.

Finalmente, el sistema de exploración se implementó en un robot móvil diferencial, el cual debe de desplazarse dentro de una superficie plana. Se utilizó un sensor lidar, con un rango máximo de medición de 6 metros. El prototipo en conjunto es capaz de realizar mediciones dentro de un ambiente que no exceda los 12 metros de diámetro.

#### **1.4. Organización de la tesis**

El documento está organizado en 4 capítulos, los cuales serán explicados en esta sección. En el capítulo 1 se presenta la problemática a resolver, la justificación y el alcance del presente trabajo de tesis. En el capítulo 2 se explica los algoritmos que se usan actualmente para el control de movimiento del robot móvil, la generación de trayectoria y los algoritmos SLAM (Simultaneous Localization And Mapping) que existen. En el capítulo 3 se describe la metodología que se hizo para el control de movimiento, la generación de trayectoria y la construcción del mapa tridimensional. En el capítulo 4 se muestra y explica con detalle los resultados obtenidos. Se realizaron pruebas en simulación y en un ambiente real asemejándose a la estructura de un túnel subterráneo. Finalmente, se describe las conclusiones de la presente tesis.

## Capítulo 2

### Marco Teórico

El presente capítulo muestra los trabajos de robótica que han sido aplicados en la minería subterránea, en los últimos diez años. Luego se presenta métodos que existen para la navegación autónoma de un robot móvil. Para lograr esto se muestra los controladores de robots móviles más utilizados en la actualidad. Asimismo, también se presenta algoritmos recientes desarrollados para la generación de trayectoria de robots móviles. Finalmente, se describe los tipos de algoritmos SLAM que existen para construir un mapa en dos dimensiones de un ambiente desconocido.

#### 2.1. Robótica en Minería

La minería actual ha llevado a que las compañías mineras busquen la eficiencia incorporando la robótica a sus procesos, generando, adicionalmente, una fuente fructífera de innovación local. Actualmente el país sudamericano con mayor inversión en minería es Chile, el cual ha iniciado la incorporación de la robótica en este rubro hace unos 15 años [6].

Los beneficios potenciales de los robots en la minería subterránea son abundantes, debido a que estas zonas son entornos de trabajo extenuantes y potencialmente peligrosos para los seres humanos. Por tal motivo, dándole un mayor grado de autonomía a los robots, estos se convierten en buenos ayudantes para los mineros o incluso pueden sustituir al personal que se despliega en los socavones [6]. Hace 20 años se empezó a realizar investigaciones sobre autonomía en la minería subterránea utilizando una máquina de carga con

sensores láser y de ultrasonido que transportaba los minerales por las minas subterráneas [7].

En los últimos años se ha ido desarrollando prototipos de robots móviles para realizar inspección dentro de los túneles subterráneos. En el año 2003 *Grounhog* fue el primer robot que exploró la mina de Florencia (*New Eagle*) [8], empleando sensores láser y una cámara de video. Asimismo en Alemania se creó un robot móvil llamado *Alexander* el cual exploró la mina de oro *Reiche Zeche* en Freiburg [9]. En el año 2014 se realizó la primera generación de modelos geométricos en tres dimensiones. Para este trabajo de investigación se utilizó una camioneta equipada de sensores láser y un microprocesador. Este prototipo fue implementado y probado dentro de una mina [10].

Para la construcción del mapa en dos dimensiones se utiliza el algoritmo SLAM, el cual es respaldado por trabajos anteriores. En Perú se desarrolló un robot móvil minero llamado *QHUBOT*. Este robot utiliza la técnica SLAM para la detección de puntos de referencia y para hacer que el robot sepa los lugares por donde puede desplazarse [11]. Sin embargo, este robot no hace exploración ya que trabaja con un mapa en dos dimensiones realizado anteriormente. Además, para realizar este robot se enfocaron más en la parte mecánica.

Recientemente, se ha comenzado a realizar trabajos de investigación con drones. Los drones son robots móviles aéreos que pueden ser adaptados a diferentes aplicaciones. Se utiliza el algoritmo SLAM para que los drones puedan navegar dentro de un ambiente interno usando cámaras, pero la navegación de estos robots han presentado dificultades al momento de implementar los algoritmos por la cantidad de sensores que se están utilizando y por las imágenes que se obtiene mediante la cámara. Por tal motivo, se utiliza algoritmos de seguimiento de líneas basados en visión [12]. El algoritmo basado en visión se emplea en diferentes ambientes, donde el dron es capaz de poder detectar puntos definidos por medio de la cámara. Esta metodología se utiliza para las zonas de rescate [13], donde la finalidad es enviar el dron a la zona de accidente y que pueda generar un mapa

tridimensional. El mapa es generado por la unión de varias imágenes dadas por una cámara RGBD (*RGB y profundidad de imagen*). También existe un trabajo donde combinan los métodos de procesamientos de imágenes con SLAM [14].

En el ámbito comercial, se ha empezado a desarrollar drones que exploran los túneles subterráneos y a su vez generan un mapa en tres dimensiones. Existe una empresa canadiense, llamada *Clickmox Solutions* que ha desarrollado un dron llamado *MineFly Drone*. Este dron utiliza como componentes un sensor láser, luz LED, sensores sonar y una cámara HD [15]. El dron no es autónomo, ya que es controlado por medio de un control remoto (radiofrecuencia).

## **2.2. Control de Robots Móviles**

La generación de movimiento para un robot móvil necesita de un control cinemático adecuado. El objetivo de un controlador cinemático es seguir una trayectoria descrita por su posición o perfil de velocidad como una función del tiempo. Esto se hace a menudo dividiendo la trayectoria en segmentos de movimiento de forma claramente definida, por ejemplo, líneas rectas y segmentos de un círculo.

Un enfoque más apropiado en el control de movimiento de un robot móvil es usar un controlador de realimentación de estados. Con un controlador de este tipo, la tarea de planificación de ruta del robot se reduce a establecer posiciones intermedias situadas en la ruta solicitada. Con este fin, primero debe calcularse el modelo cinemático que relaciona la velocidad de las ruedas con la velocidad lineal y angular del robot en su propio marco de referencia. Luego, se debe utilizar las leyes de control que permiten un suave movimiento en la trayectoria del robot. Esta sección se enfoca en los controladores de movimiento de robots móviles que existen actualmente.

Para el control de movimiento de robots móviles, se debe tener en consideración la clasificación de estos a través de las restricciones a la locomoción que imponen sus



(a)



(b)

FIGURA 2.1: Representación gráfica de los sistemas holonómicos en la robótica móvil, donde en (a) se muestra el robot móvil omnidireccional [16] y en (b) se muestra el robot móvil diferencial [17].

ruedas. Esta clasificación divide a los robots en dos tipos: (i) robot holonómico y (ii) robot no-holonómico. El robot holonómico no tiene ninguna restricción en la posición y la orientación, con respecto a la velocidad. El robot se puede mover instantáneamente en cualquier dirección del espacio, sin necesidad de rotar previamente. Estos robots móviles son omnidireccionales, y algunos ejemplos se pueden ver en la Figura 2.1a. En cambio un robot no-holonómico tiene una restricción en su velocidad, y por tanto no existe una trayectoria que dependa solamente de la posición y orientación. Por ende, este robot no se puede mover instantáneamente en cada dirección del espacio. Los robots no-holonómicos no son omnidireccionales. Un ejemplo de este tipo de robots son los robots diferenciales y como el que se puede apreciar en la Figura 2.1b.

### 2.2.1. Control Cinemático PID

Un controlador Proporcional, Integral, Derivativo (PID) es un mecanismo de control por realimentación. Éste calcula la desviación o error entre un valor medido y un valor deseado y consta de tres parámetros distintos: el proporcional, el integral y el derivativo. El valor proporcional depende del error actual. El integral depende de los errores pasados

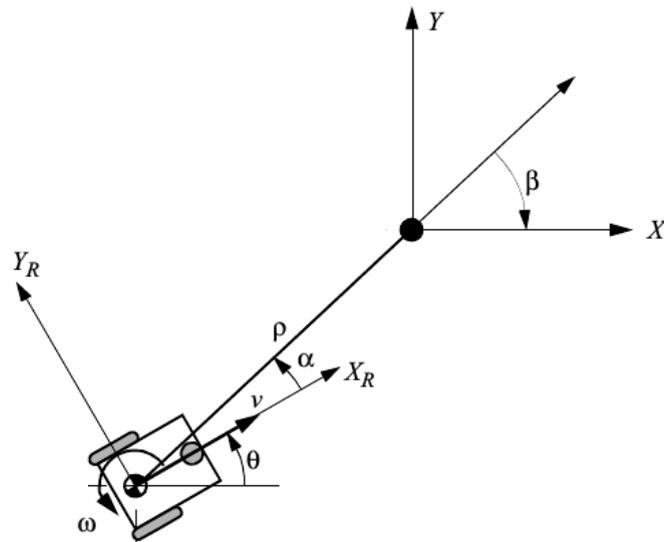


FIGURA 2.2: Configuración para el control polar donde se muestran la posición del objetivo y los parámetros específicos [18] (ver texto para más detalle).

y el derivativo es una predicción de los errores futuros. La suma de estas tres acciones es usada para ajustar al proceso por medio de un elemento de control. Este algoritmo es ampliamente difundido en el mundo industrial, y su forma más conocida es:

$$u(t) = K_c e(t) + \frac{K_c}{T_i} \int e(t) dt + K_c T_d \frac{de(t)}{dt} = P(t) + I(t) + D(t),$$

donde  $K_c$  es la ganancia proporcional,  $T_i$  es el tiempo integral,  $T_d$  es el tiempo derivativo y  $e(t)$  es el error entre la posición actual y la posición deseada del robot. Este algoritmo posee muchas variaciones, y dependiendo de la aplicación el controlador en cuestión puede trabajar como P, PI, PD o PID.

### 2.2.2. Control Cinemático Polar

Una forma alternativa de obtener un control suave de posición y orientación para un robot es a través de una ley lineal de control por realimentación de estados basada en coordenadas polares [19]. Como se muestra en la Figura 2.2, las coordenadas polares

asociadas con el robot son:

$$\begin{aligned}\rho &= \sqrt{(x_d - x)^2 + (y_d - y)^2}, \\ \alpha &= \text{atan2}(y_d - y, x_d - x) - \theta, \\ \beta &= -\text{atan2}(y_d - y, x_d - x) + \theta_d,\end{aligned}$$

donde  $\rho$  representa la distancia desde el robot a la posición deseada,  $\alpha$  y  $\beta$  son ángulos que se utilizan para el error de orientación y  $\text{atan2}$  es la función tangente inversa que retorna el ángulo en uno de los cuatro cuadrantes. La posición y orientación actual del robot es  $\mathbf{x} = (x, y, \theta)$ , y su posición y orientación deseada es  $(x_d, y_d, \theta_d)$ . En base a estas variables polares, la ley de control que define las velocidades lineales y angulares para lograr la posición y orientación deseada es

$$v = k_\rho \rho, \tag{2.1}$$

$$\omega = k_\alpha \alpha + k_\beta \beta, \tag{2.2}$$

donde  $k_\rho$ ,  $k_\alpha$  y  $k_\beta$  son las ganancias del controlador polar que deben ajustarse adecuadamente para una buena respuesta del sistema. Usando este marco de generación de movimiento logramos el objetivo tanto en posición como en orientación.

### 2.3. Navegación del Robot Móvil

Un robot es un sistema autónomo que puede percibir su entorno y actuar para poder alcanzar sus objetivos. Un robot móvil dentro de un entorno no se encuentra en una ubicación específica, ya que este tiene la capacidad de moverse. La principal característica que define a un robot autónomo es la capacidad de actuar sobre la base de sus propias decisiones, y no a través del control de una persona [20].

La navegación se define como el proceso o la actividad de determinar con exactitud la posición de uno mismo, planificar y seguir una ruta. En robótica, la navegación se refiere a la forma en que un robot encuentra su camino en el entorno, es una necesidad y requisito común para casi todos los robots móviles.

Un problema en la planificación de movimiento es producir un movimiento continuo que conecte un punto de inicio y un punto de meta, mientras se evita la colisión con obstáculos conocidos. La trayectoria que el robot describe se representa en un espacio de configuración. Una configuración describe la posición y orientación del robot, y el espacio de configuración ( $C$ ) es el conjunto de todas las posibles configuraciones. Por ejemplo, si el robot se considera como un único punto en un espacio de dos dimensiones, su espacio de trabajo, entonces el espacio de configuración es un plano y la configuración se representa como  $(x, y)$ .

Dentro de los métodos de planificación de movimiento, se utiliza con frecuencia el término espacio libre. El espacio libre ( $C_{free}$ ) es el conjunto de configuraciones que evita la colisión con obstáculos. Calcular la forma del espacio libre es difícil; sin embargo, es mucho más sencillo probar si una configuración determinada se encuentra dentro de  $C_{free}$ . En las siguientes secciones se explica los métodos de navegación basados en campos potenciales, movimientos basados en mapas topológicos y movimiento basado en muestreo aleatorio.

### **2.3.1. Navegación Basada en Campos Potenciales**

Los principales métodos de planificación de movimiento son descomposición de celdas, diagramas de voronoi y planificación basada en muestreo y campos potenciales. La función potencial más simple es el potencial atractivo y repulsivo [21]. La intuición detrás de los campos potenciales es directa: el objetivo atrae al robot mientras los obstáculos lo repelen. La suma de estos efectos atrae al robot hacia la meta mientras lo desvía de los

obstáculos. La función potencial se puede construir como la suma de potenciales atractivos y repulsivos

$$U(q) = U_{att}(q) + U_{rep}(q), \quad (2.3)$$

donde  $U_{att}(q)$  es el campo potencial de atracción y  $U_{rep}(q)$  es el campo potencial de repulsión. Los tipos de campo potenciales son atractivo y repulsivo.

**Campo Potencial Atractivo.** Este campo conduce al robot hacia la posición de la meta, suponiendo que existe una sección circular alrededor de este objetivo [22]. Es decir, cualquier posición dentro de esta sección circular se considerará que satisface el objetivo del algoritmo de planificación. Considere una energía potencial escalar que depende de la distancia entre la posición actual del robot  $\mathbf{q} = (x, y)$  y su posición objetivo  $\mathbf{q}_{goal} = (x_{goal}, y_{goal})$  dada por

$$U_{att}(\mathbf{q}) = \frac{1}{2}\zeta d^2(\mathbf{q}, \mathbf{q}_{goal}), \quad (2.4)$$

donde  $d(a, b)$  es una medida de la métrica entre dos puntos, y  $\zeta$  es un parámetro de ajuste que escala el efecto del potencial atractivo. En este caso, usamos la distancia euclidiana como una métrica ya que proporciona buenos resultados experimentales. El gradiente de este campo proporciona un campo vectorial que siempre apunta a la posición deseada como:

$$\nabla U_{att}(\mathbf{q}) = \zeta(\mathbf{q} - \mathbf{q}_{goal}). \quad (2.5)$$

### 2.3.1.1. Campo Potencial Repulsivo

Un potencial repulsivo mantiene al robot alejado de los obstáculos [22]. La intensidad de la fuerza de repulsión depende de la proximidad del robot al obstáculo cuanto

más cerca esté el robot de un obstáculo, más intensa será la fuerza de repulsión. Por lo tanto, el potencial repulsivo generalmente se define en términos de distancia al obstáculo más cercano  $D(\mathbf{q})$  como

$$U_{rep}(\mathbf{q}) = \begin{cases} \frac{1}{2}\eta\left(\frac{1}{D(\mathbf{q})} - \frac{1}{Q^*}\right)^2, & D(\mathbf{q}) \leq Q^* \\ 0, & D(\mathbf{q}) > Q^* \end{cases} \quad (2.6)$$

donde  $Q^* \in \mathbb{R}$  es una constante que le permite al robot ignorar los obstáculos que están lo suficientemente lejos de él, y  $\eta$  es una ganancia en el gradiente de repulsión. El gradiente es

$$\nabla U_{rep}(\mathbf{q}) = \begin{cases} \eta\left(\frac{1}{Q^*} - \frac{1}{D}\right)\frac{1}{D^2}\nabla D, & D \leq Q^* \\ 0, & D > Q^* \end{cases} \quad (2.7)$$

donde la dependencia de  $D$  en  $\mathbf{q}$  ha sido omitida para mayor claridad. Los escalares utilizados para este campo generalmente se determinan por ensayo y error [23].

### 2.3.2. Movimiento Basados en Mapas Topológicos

Otra forma de planificación de movimiento se basa en mapas topológicos o *roadmaps*. Un *roadmap* es una clase de mapa topológico que representa el espacio libre en entornos donde los nodos y bordes tienen un significado físico [24]. Un nodo de *roadmap* corresponde a una ubicación específica y un borde corresponde a una ruta entre ubicaciones vecinas. La forma en que el robot puede explorar un entorno desconocido es por medio de los datos del sensor que le permitirán construir un mapa de ruta.

Los robots usan *roadmaps* de la misma manera que las personas usan las carreteras, lo que les permite ir del inicio hasta la meta. El robot primero encuentra una ruta libre de colisiones en el mapa, atravesando éste hasta que se encuentra muy cerca de la meta. En este proceso, el robot va construyendo toda la trayectoria que debe seguir para llegar

al punto deseado. A continuación se realizará un breve resumen de un algoritmo que se basa en el método de *roadmap* llamado mapa de visibilidad gráfica.

**Mapa de Visibilidad Gráfica.** Para el planeamiento del camino del robot en situaciones prácticas, es preferible encontrar no solo un camino sino un buen camino. El buen camino depende del robot, cuanto más larga sea la ruta, más tiempo le llevará al robot alcanzar la posición deseada. Para un robot móvil en una fábrica, esto significa que puede transportar menos bienes por unidad de tiempo, lo que resulta en una pérdida de productividad. También existen problemas con respecto a cómo se mueven los robots. Por ejemplo, hay robots que se mueven en línea recta y para poder moverse a los lados necesitan rotar en su propio eje. En esta sección se obvia este aspecto, pero se enfoca en mostrar cómo se calcula la ruta más corta para un robot utilizando un mapa de visibilidad gráfica.

En la geometría computacional y la planificación del movimiento del robot, un mapa de visibilidad es un mapa de ubicaciones intervisibles, generalmente para un conjunto de puntos y obstáculos en un plano euclideo. Las características que definen un mapa de visibilidad son que sus nodos que compartan un borde si se encuentran a la vista el uno del otro, y que todos los puntos en el espacio libre del robot se encuentren dentro de la línea de visión de al menos un nodo en el mapa de visibilidad.

El gráfico de visibilidad estándar se define en un espacio de configuración poligonal bidimensional (ver Figura 2.3a). Los nodos del gráfico de visibilidad incluyen la ubicación de inicio, la ubicación del objetivo y todos los vértices de los obstáculos del espacio de configuración. Los bordes del gráfico son segmentos de línea recta que conectan dos nodos.

A partir de la Figura 2.3a se considera la inclusión de los nodos y bordes en el espacio libre y que los bordes de los obstáculos poligonales también sirven como bordes en el gráfico de visibilidad. Utilizando distancia euclidiana se puede buscar la ruta más corta como se aprecia en la Figura 2.3b.

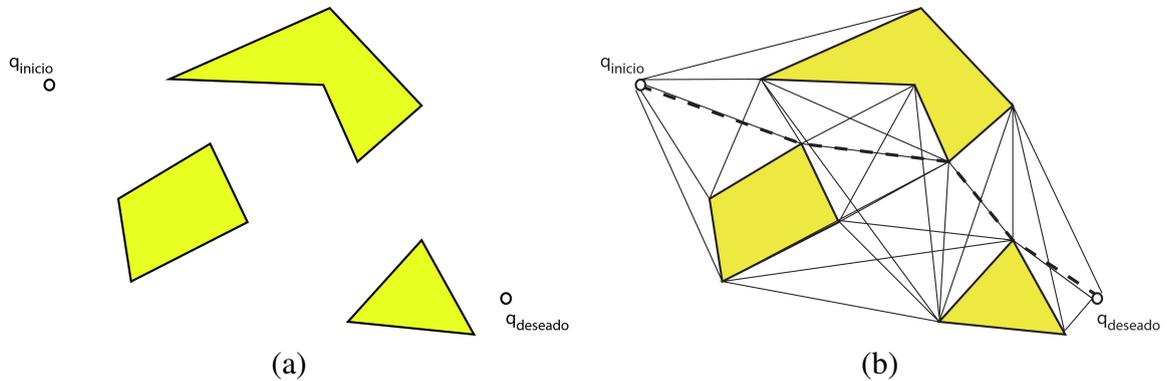


FIGURA 2.3: Representación gráfica de un espacio de configuración. En (a) se muestra el espacio de configuración con un punto de inicio y otro punto de llegada a la posición deseada y en (b) se puede ver el mismo espacio de configuración donde los sólidos geométricos representan los obstáculos. Las líneas continuas delimitan los bordes de la visibilidad gráfica para los tres obstáculos y las líneas entrecortadas representan el camino más corto entre el punto de inicio y el punto de meta.

### 2.3.3. Movimiento Basado en Muestreo Aleatorio

Las planificación de movimiento basada en muestreo, como los métodos de mapa de ruta probabilista (*Probabilistic Roadmap*) o aquellos basados en los árboles aleatorios de exploración rápida (*Rapidly Exploring Random Trees*) dan buenos resultados en la planificación de movimientos de robots con muchos grados de libertad. Con estos enfoques, se han propuesto varias estrategias para predisponer el muestreo hacia las regiones más prometedoras, mejorando con esto la eficiencia y permitiendo la solución de problemas difíciles de planificación de movimientos [25]. El éxito de estos planificadores en la solución de problemas desafiantes se puede explicar por el hecho de que no se requiere ninguna representación explícita del espacio de configuración libre. En esta sección se explica los métodos Mapa de Ruta Probabilístico, Árbol Aleatorio de Exploración Rápida y Movimiento basado en Mapa de Red.

**Mapa de Ruta Probabilístico.** La planificación del mapa de ruta probabilístico (*Probabilistic Roadmap* - PRM) es una de las alternativas para la planificación de las rutas frente a los obstáculos, donde las posiciones no se conocen con certeza. El PRM es una

técnica popular para la planificación de caminos en espacios de alta dimensión [26]. Este algoritmo es aplicable en muchas situaciones, incluida la planificación del movimiento del robot y la planificación de las rutas de las personas. Un PRM genera un gráfico aleatorio con vértices que representan las posiciones del robot y con bordes que representan el movimiento desde una posición hasta otra posición. Seguido a esto se busca, en este gráfico, una ruta desde el punto de inicio hacia el objetivo. El éxito de este algoritmo se debe a que reducen la búsqueda de gráficos. Sin embargo, la mayoría de los enfoques de PRM se basan en la suposición que el planificador conoce las ubicaciones de todos los obstáculos en el entorno antes de construir el gráfico del *roadmap* o trazar una ruta.

El algoritmo comienza con muestrear valores aleatorios de la configuración del robot móvil para que éste pueda crear posiciones dentro de un ambiente (espacio de configuración). El planificador de PRM es capaz de explorar todo el espacio de configuración. No todas las configuraciones en el espacio serán utilizables debido a los obstáculos que existen, por lo que el espacio dimensional se divide conceptualmente en dos áreas llamadas  $C_{free}$  y  $C_{obst}$  para configuraciones libres y obstruidas, respectivamente. En la fase de preprocesamiento se empieza a construir un gráfico PRM usando el algoritmo de muestreo que genera configuraciones para el robot y luego valida las posiciones generadas. Lo más simple y común que se hace es rechazar todas las posiciones que se cruzan con  $C_{obst}$ .

**Árbol Aleatorio de Exploración Rápida.** Un árbol aleatorio de exploración rápida (Rapidly Exploring Random Trees - RRT) es un algoritmo que está diseñado para buscar de manera eficiente espacios no convexos de gran dimensión [27]. Los RRT se construyen de forma incremental de una manera que reduce rápidamente la distancia esperada de un punto elegido al azar. Los RRT son particularmente adecuados para problemas de planificación de ruta que involucran obstáculos y restricciones diferenciales (no holonómicas).

Un RRT hace crecer un árbol enraizado en la configuración inicial utilizando muestras aleatorias del espacio de búsqueda. A medida que se extrae cada muestra, se

intenta establecer una conexión entre ella y el estado más cercano en el árbol. Si la conexión es factible (pasa completamente a través del espacio libre y obedece a cualquier restricción), esto resulta en la adición del nuevo estado al árbol. Con un muestreo uniforme del espacio de búsqueda, la probabilidad de expandir un estado existente es proporcional al tamaño de su región de Voronoi. Como las regiones más grandes de Voronoi pertenecen a los estados en la frontera de la búsqueda, esto significa que el árbol se expande preferentemente hacia grandes áreas no buscadas.

**Movimiento basado en Mapa de Red.** La navegación basada en mapas de red es bastante natural para las personas porque usar un mapa es la forma más conveniente de describir un entorno. Sin embargo, el uso de los mapas requiere de un buen proceso cognitivo para poder interpretar el mapa y establecer la correspondencia con el mundo real.

Para la navegación basada en mapas de red se pueden usar dos fuentes de información. El primero es la fuente idiotética, que proporciona información interna sobre el movimiento del robot. El segundo es la fuente alotética, que proporciona información externa sobre el ambiente. La información idiotética se refiere a la odometría del robot y la información alotética se refiere a algún sensor externo que permita al robot tener una perspectiva del entorno donde se encuentra [28].

Dadas las fuentes de información como la odometría y la de un sensor externo, hay muchas formas de poder integrarlas en una representación útil para la navegación de los robots. Los modelos correspondientes se separan en dos categorías recurriendo a mapas métricos, las posiciones de algunos objetos, principalmente los obstáculos que el robot puede encontrar, se almacenan en un marco de referencia común.

La exploración con estos mapas de red consiste en que el robot empieza en un punto preparado para poder tomar cualquier ruta y realizar varios viajes. Entonces el robot empieza a recibir tareas de un nodo central para que pueda desplazarse en diferentes

puntos del mapa. El mapa se categoriza en tres tipos diferentes: (i) un obstáculo (probabilidad de ocupación por encima de un umbral), (ii) limpio (probabilidad de ocupación nulo) y (iii) desconocido (nunca se detectó nada). De esta forma se empieza a construir el mapa y cuando el robot recibe este mapa que se actualiza constantemente, empieza a tener una mejor percepción del ambiente donde se encuentra.

## **2.4. Localización**

La navegación es una de las tareas con mayor exigencia para un robot móvil. Para realizar una buena navegación se necesita de cuatro componentes básicos, los cuales son: (i) percepción, el robot debe extraer e interpretar los datos de sus sensores para estimar su entorno; (ii) localización, el robot debe determinar su posición en el entorno; (iii) cognición, el robot debe tener la capacidad de decidir cómo actuar para lograr su objetivo; y (iv) control de movimiento, el robot debe modular sus velocidades para lograr una trayectoria deseada. De los cuatro componentes descritos, la localización ha recibido la mayor atención de investigación en la última década y, como resultado, se han logrado avances significativos.

El problema en la localización es estimar la posición del robot en relación con el mapa del entorno [29]. Cuando un robot se mueve en un entorno conocido y comienza en una ubicación conocida, se puede estimar su posición local por medio de la odometría. Pero esta estimación de la posición del robot con el tiempo comienza a volverse menos confiable debido a las incertidumbres, por ende el robot no tiene una posición absoluta establecida. Con el fin de reducir la incertidumbre, el robot tiene que recuperar su posición absoluta localizándose en relación al mapa. Para ello, el robot debe tener sensores que le permitan hacer observaciones del entorno y relacionar estas observaciones con un mapa del mundo real. En esta sección se explica sobre las metodologías de localización.

### 2.4.1. Modelo de movimiento y medición

Un modelo de movimiento probabilístico define la transición de la posición en dos etapas de tiempo consecutivas ( $t - 1$  y  $t$ ), después de que se ha llevado a cabo una acción de control ( $u_t$ ). Existen dos modelos de movimiento probabilístico complementarios que son el modelo de movimiento de velocidad y el modelo de movimiento de la odometría. El modelo de movimiento de velocidad asume que un robot es controlado a través de dos velocidades que son la velocidad lineal ( $v_t$ ) y la velocidad angular ( $\omega_t$ )

$$u_t = \begin{pmatrix} v_t \\ \omega_t \end{pmatrix}. \quad (2.8)$$

Un modelo probabilístico de medición describe la relación entre la posición del mundo real y la lectura del sensor (*observación*). Basándose en la estimación de la posición del mundo real, se genera un *belief* sobre las posibles mediciones. Un *belief* es una distribución probabilística que representa el conocimiento del robot sobre el entorno. El ruido en las mediciones del sensor se modela explícitamente, y es inherente a la incertidumbre de los sensores del robot. Para expresar el proceso de generación de mediciones, se requiere una especificación del entorno. Un mapa ( $m$ ) del entorno es una lista de objetos y ubicaciones:

$$m = \{m_1, m_2, \dots, m_N\}, \quad (2.9)$$

donde  $N$  es el total de números de objetos en el ambiente y  $m$  representa la posición del objeto  $i$ . Los mapas suelen ser mapas basados en características o mapas basados en la ubicación. Los mapas basados en ubicación son volumétricos y ofrecen una etiqueta para cualquier ubicación y los mapas basados en características son más populares debido a que la representación hace más fácil ajustar la posición de los objetos.

### 2.4.2. Filtros de Kalman

El filtro de Kalman (KF) se basa en la suposición que el sistema es lineal y tanto el modelo de movimiento como el modelo de medición se ven afectados por el ruido gaussiano [30]. El *belief* en el tiempo  $t$  está representado por una distribución gaussiana multivariable definida por su media ( $\mu_t$ ) y la covarianza ( $\Sigma_t$ ).

Un modelo de transición de estado, bajo la premisa que el sistema sigue el modelo de Markov, es definido como:

$$p(x_t \mid x_{0:t-1}, z_{1:t-1}, u_{1:t}) = p(x_t \mid x_{t-1}, u_t), \quad (2.10)$$

donde  $p(x_t \mid x_{t-1}, u_t)$  es llamada probabilidad de transición de estado y especifica cómo evoluciona el estado del entorno a lo largo del tiempo en función del control del robot  $u_t$ . Además  $x_{0:t-1}$  comprende todos los estados desde 0 hasta  $t_1$ .

Un modelo de medición del sensor usando la suposición de Markov, es definido como:

$$p(z_t \mid x_{0:t}, z_{1:t-1}, u_{1:t}) = p(z_t \mid x_t), \quad (2.11)$$

donde  $p(z_t \mid x_t)$  es llamado probabilidad de medición y especifica cómo se generan las mediciones a partir del estado del ambiente  $x_t$ .

El filtro de Kalman asume que el sistema es lineal: la transición de estado  $A$  (2.10), el modelo de movimiento  $B$  (2.8) y el modelo del sensor  $C$  (2.11) son funciones lineales únicamente dependiendo del estado  $x$  o del comando de control ( $u$ ), más un modelo de ruido gaussiano ( $Q$ ). El algoritmo del filtro de Kalman tiene dos etapas definidas, las

cuales son:

### Predicción

$$\bar{\mu}_t = A_t \mu_{t-1} + B_t u_t \quad (\text{a})$$

$$\bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t \quad (\text{b})$$

### Actualización

$$K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1} \quad (\text{c})$$

$$\mu_t = \bar{\mu}_t + K_t (z_t - C_t \bar{\mu}_t) \quad (\text{d})$$

$$\Sigma_t = (I - K_t C_t) \bar{\Sigma}_t \quad (\text{e}).$$

En estas etapas (a) es la estimación de la media a-priori, (b) estimación de la covarianza a priori, (c) Ganancia de Kalman, (d) actualización de la estimación de estado (a posteriori) y (e) actualización de la estimación de la covarianza (a posteriori).

El filtro de Kalman estándar requiere un sistema lineal, que no es suficiente para describir muchos problemas de la vida real. Por lo tanto, se han propuesto variaciones del algoritmo original que pueden hacer frente a diferentes niveles de no-linealidad: como filtro de Kalman extendido, filtro de Kalman unscented, entre otros. Estas variaciones se aproximan a los modelos de movimiento y medición de forma que tiendan a ser lineales.

### 2.4.3. Filtros de Partículas

El filtro de partículas es una implementación alternativa no paramétrica. Al igual que los filtros de histograma, los filtros de partículas se aproximan al estado posterior por un número finito de parámetros [31]. Sin embargo, difieren en la forma en que se generan estos parámetros, y en los que muestrean el espacio de estado. La idea del filtro de partículas es representar el *belief* predecido ( $x_t$ ), mediante un conjunto de muestras aleatorias de estado. En lugar de representar la distribución mediante una forma paramétrica (la función exponencial que define la densidad de una distribución normal), los filtros de

partículas representan una distribución por un conjunto de muestras extraídas de esta distribución. Tal representación es aproximada, pero no es paramétrica, por lo tanto se puede representar un espacio de distribuciones mucho más amplio que los gaussianos [32].

#### **2.4.4. Filtros de Información**

El filtro de información (IF) representa el *belief* por medio de un gaussiano. Por lo tanto, el filtro de información estándar está sujeto a las mismas suposiciones subyacentes al filtro de Kalman [33]. La diferencia entre el KF y el IF surge de la forma en que se representa el *belief* gaussiano. Mientras que en la familia de algoritmos de filtro de Kalman los gaussianos están representados por sus momentos (media, covarianza), en los filtros de información los gaussianos utilizan su representación canónica la cual está compuesta por una matriz de información y un vector de información. La diferencia en la representación conduce a diferentes ecuaciones de actualización. En particular, lo que es computacionalmente complejo en una representación resulta ser simple en la otra y viceversa. Las representaciones canónicas y de momentos a menudo se consideran duales entre sí, y así son el IF y el KF.

#### **2.5. Mapeo y Localización Simultánea (SLAM)**

Un problema mucho más difícil que la localización es cuando no se conoce el mapa del ambiente. En este caso, la información del sensor del robot se utiliza para recuperar la trayectoria del robot y construir un mapa del entorno. Este problema se llama mapeo y localización simultáneos (SLAM) [34]. Una solución a este problema brindaría capacidades para que un robot sea totalmente autónomo. SLAM es un problema difícil porque tanto el camino estimado como el mapa construido son afectados por el ruido. Ambos se vuelven cada vez más inexactos durante el viaje. Sin embargo, cuando se revisa un lugar que ha sido mapeado, se puede reducir la incertidumbre. Este proceso se denomina cierre de

bucle. Además, se puede optimizar el mapa después de que se detecta un evento de cierre de bucle. En esta sección se explica los métodos de *SLAM* más utilizados actualmente.

### 2.5.1. *SLAM* basado en Filtro Extendido de Kalman

El *SLAM* basado en el filtro extendido de kalman (EKF-*SLAM*) es el algoritmo más antiguo y el más influyente. El filtro extendido de kalman (EKF), es la versión no lineal del filtro de kalman aplicada a sistemas gaussianos no lineales [34]. El algoritmo *EKF-SLAM* aplica el filtro extendido de kalman al *SLAM* usando la asociación de datos de máxima verosimilitud. Al hacerlo, este algoritmo está sujeto a una serie de aproximaciones y suposiciones limitantes:

- **Mapas basados en funciones:** Los mapas, en el *EKF*, se componen de puntos de referencia. Por razones de computación, el número de puntos de referencia es generalmente pequeño (por ejemplo, más pequeño que 1000). Además, el enfoque de *EKF* tiende a funcionar bien cuanto menos ambiguos sean los puntos de referencia. Por esta razón, *EKF-SLAM* requiere una ingeniería significativa de los detectores de funciones, a veces utilizando balizas artificiales o puntos de referencia como características.
- **Ruido Gaussiano:** Como cualquier algoritmo *EKF*, *EKF-SLAM* hace una suposición de ruido Gaussiano para el movimiento del robot y la percepción. La cantidad de incertidumbre en la parte de la predicción debe ser relativamente pequeña, ya que de lo contrario la linealización en *EKF* tiende a introducir errores intolerables.
- **Mediciones positivas:** El algoritmo *EKF-SLAM*, al igual que el localizador *EKF*, solo puede procesar avistamientos positivos de puntos de referencia. No puede procesar información negativa que surge de la ausencia de puntos de referencia en las mediciones de un sensor. Esta es una consecuencia directa de la representación de *belief* gaussiano.

### 2.5.2. SLAM basado en Filtros de Partículas

El algoritmo SLAM basado en Filtros de Partículas es conocido como FastSLAM [35]. FastSLAM descompone el problema SLAM en un problema de localización de robot, y una colección de problemas de estimación de puntos de referencia (*landmarks*) que están condicionados a la estimación de la posición del robot. Como se señala en [36], esta representación factorizada es exacta, debido a las independencias condicionales naturales en el problema SLAM. FastSLAM usa un filtro de partículas modificado para estimar el camino predecido sobre el robot. Cada partícula posee filtros de Kalman que estiman las ubicaciones de los puntos de referencia condicionadas en la estimación de la ruta [37].

### 2.5.3. GraphSLAM

SLAM basado en gráficos construye un problema de estimación simplificado al extraer las mediciones del sensor sin ser procesadas. Estas medidas se reemplazan por los bordes en el gráfico que luego se pueden ver como "medidas virtuales". Más detalladamente, un borde entre dos nodos se etiqueta con una distribución de probabilidad sobre las ubicaciones relativas de las dos posturas, condicionadas a sus mediciones mutuas. En general, el modelo de observación  $p(z_t|x_t, m_t)$  es multimodal y, por lo tanto, el supuesto gaussiano no se cumple. Esto significa que una única observación  $z_t$  puede dar como resultado múltiples aristas potenciales que conectan diferentes posiciones en el gráfico y la conectividad del gráfico necesita ser descrita como una distribución de probabilidad. Dirigirse directamente a esta multimodalidad en el proceso de estimación llevaría a una explosión combinatoria de la complejidad. Como resultado de eso, la mayoría de los enfoques prácticos restringen la estimación a la topología más probable. Por lo tanto, se necesita determinar la restricción más probable resultante de una observación. Esta decisión depende de la distribución de probabilidad sobre las posiciones del robot [38].

## Capítulo 3

### Metodología

El capítulo anterior hace referencia a las teorías recientes que son utilizadas para la autonomía de un robot móvil y para la generación de un mapa bidimensional, dentro de un ambiente aún por conocer. En el presente trabajo de tesis se ha desarrollado un sistema de movimiento autónomo, del robot móvil, basado en campos potenciales, leyes de control polar y posiciones aleatorias dentro de un espacio de configuración. Este sistema permite lograr que el robot pueda explorar un ambiente desconocido de manera autónoma. Asimismo, también se ha desarrollado un sistema mecánico accionado por un motor paso a paso, el cual permite que el robot mientras se va desplazando dentro de un entorno pueda tomar mediciones en los tres ejes del plano cartesiano y así obtener una representación gráfica del lugar en tres dimensiones. Estos sistemas pueden ser implementados desde una computadora en tiempo real (*online*) o también pueden ser implementados por medio de un microcontrolador (*onboard*). En este capítulo se explicará, con detalles, los métodos que fueron implementados para realizar el presente trabajo.

#### 3.1. Sistema Desarrollado

En esta sección se describe los componentes que fueron utilizados para construir el prototipo funcional. Estos componentes se utilizan para la navegación autónoma del robot móvil, para construir el mapa en dos dimensiones y para tomar medidas en los tres ejes del plano cartesiano. Para la navegación se utilizó un robot móvil diferencial terrestre, y un sensor lidar para las mediciones del entorno.

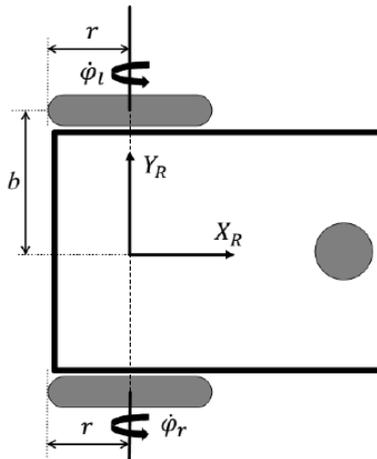


FIGURA 3.1: Representación gráfica de un robot móvil diferencial, donde  $b$  es la distancia de su centro de masa hacia sus ruedas, el radio de cada rueda es representado por  $r$ , y  $\dot{\phi}_l$  y  $\dot{\phi}_r$  son las velocidades de la rueda derecha y rueda izquierda respectivamente.

### 3.1.1. Cinemática de Robot Móvil Diferencial

La realización del modelo cinemático, del robot móvil diferencial, debe tener ciertas consideraciones con respecto a su comportamiento. Primero, se debe asumir que el robot se desplaza dentro de una superficie plana idealmente sin ningún tipo de rozamiento, y segundo, se debe tomar los ejes de las ruedas de forma perpendicular al suelo, existiendo un solo punto de contacto entre el suelo y la rueda.

Para obtener las ecuaciones matemáticas del modelo cinemático del robot móvil, se necesita conocer las dimensiones de éste. Las medidas requeridas son el radio de las ruedas y la distancia entre cada una de ellas. El radio de las ruedas es representado con la variable  $r$  y la distancia entre ruedas es representada como  $2b$ . Otro punto a considerar es que el robot tiene dos ruedas convencionales fijas y el sistema de referencia de este se encuentra en medio de las ruedas traseras, como se puede ver en la Figura 3.1. Además la Figura 3.1 muestra una rueda delantera. Esta rueda no presenta un mecanismo de control ya que su función es ser un apoyo del robot móvil, para su desplazamiento.

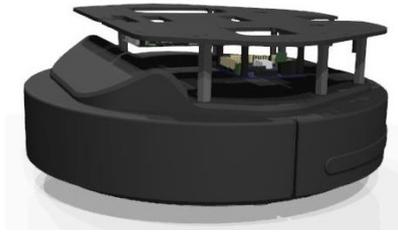


FIGURA 3.2: Robot Kobuki, el cual es un robot móvil diferencial utilizado para la implementación del sistema de navegación autónoma.

Teniendo en consideración las restricciones de las ruedas convencionales se halla la cinemática directa del robot móvil. En este caso, la cinemática directa consiste en:

$$v = \frac{r}{2}(\dot{\varphi}_r + \dot{\varphi}_l),$$
$$\omega = \frac{r}{2b}(\dot{\varphi}_r - \dot{\varphi}_l),$$

donde  $v$  es la velocidad lineal y  $\omega$  es la velocidad angular del robot. Estas dos variables de velocidad se pueden hallar a través de las velocidades de cada una de las ruedas y las dimensiones del robot. Por otro lado, la cinemática inversa, como su mismo nombre lo menciona, es todo lo contrario a la cinemática directa y se obtiene a partir de las expresiones anteriores como:

$$\dot{\varphi}_r = \frac{1}{r}(v + b\omega),$$
$$\dot{\varphi}_l = \frac{1}{r}(v - b\omega),$$

donde se halla las velocidades de las ruedas a partir de la velocidad lineal y de la velocidad angular del robot móvil.

Finalmente, para este trabajo de tesis se utilizó un robot móvil no-holonómico llamado Kobuki (ver Figura 3.2), el cual es fabricado por Yujin Robotics. Este robot trabaja con una velocidad máxima de translación de  $70 \text{ cm/s}$  y una velocidad máxima de rotación de  $180 \text{ grados/s}$ . Tiene una posibilidad de carga de hasta  $5 \text{ kg}$  y un tiempo de



FIGURA 3.3: Sensor lidar RPLIDAR A2 [40].

operación de 3 horas [39]. En este robot se implementó las ecuaciones de la cinemática directa teniendo en consideración sus propias dimensiones y asimismo, haciendo uso de las librerías que existen dentro de los paquetes de desarrollo del mismo robot Kobuki.

### 3.1.2. Sistema de Percepción Bidimensional

Para generar un mapa de dos dimensiones dentro de un entorno, se necesita realizar mediciones dentro del ambiente. En este trabajo de tesis se utiliza un sensor Lidar llamado RPLidar A2, el cual se basa en el principio de rango de triangulación láser [41]. Este sensor fue desarrollado por la empresa *SLAMTEC*.

El RPLidar A2, mostrado en la Figura 3.3, es un sensor que no necesita tener luz externa para poder obtener las medidas de las distancias, ya que tiene un láser infrarrojo de baja potencia que es controlado por medio de un pulso modulado. Este sensor puede girar en  $360^\circ$  y tiene un rango de alcance máximo de 6 metros. Trabaja a una frecuencia de 10Hz teniendo una tasa de medición de 4000 muestras por segundo. La transmisión de datos de este sensor se realiza por medio de un protocolo UART.

En la Tabla 3.1 se muestra el rendimiento de la medición del sensor RPLidar A2 [42]. En esta tabla se considera varias pruebas por cada una de las variables. Las variables

Item	Unidad	Mínimo	Típico	Máximo
Rango de distancia	m	0.15	-	6
Rango angular	grados	-	0 - 360	-
Resolución de la distancia	mm	-	menor a 0.5	-
Duración de muestra	ms	-	0.25	-
Resolución angular	grados	0.45	0.9	1.35
Frecuencia de muestreo	Hz	2000	4000	4100
Frecuencia de escaneo	Hz	5	10	15

TABLA 3.1: Especificaciones de Medición del sensor RPLidar A2.

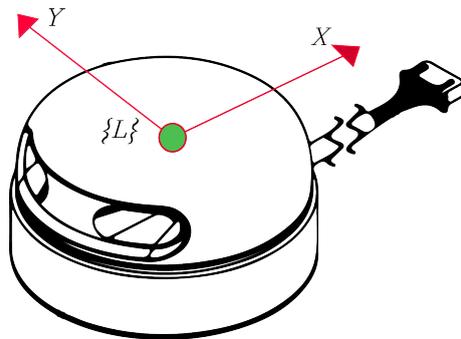


FIGURA 3.4: En esta figura el punto verde representa el origen del sistema de referencia del sensor RPLidar A2, donde el eje  $X$  va en el mismo sentido del cable de datos del sensor lidar.

más importantes son: (i) **Rango de distancia**, el sensor tiene un rango de medición entre  $0.15m$  y  $6m$ , los valores menores a  $0.15m$  y mayores a  $6m$  son considerados como valores erróneos. (ii) **Resolución de la distancia**, el sensor tiene un error menor a  $0.5\text{ mm}$  en sus mediciones, esto permite construir un mapa con las dimensiones reales del entorno. (iii) **Resolución angular**, este sensor tiene una resolución de  $1^\circ$  lo cual quiere decir que por cada grado que gira el sensor lidar, éste realiza una medición, por lo tanto el sensor toma 360 mediciones por cada rotación.

Para enviar los datos, a la computadora o al microcontrolador, el sensor trabaja con un protocolo de comunicación UART (*Universal Asynchronous Receiver-Transmitter*). Este protocolo se especifica en la Tabla 3.2, y a través del protocolo se obtiene las mediciones del sensor. Para la construcción del mapa en dos dimensiones se debe considerar el

sistema de referencia del sensor l dar. En la Figura 3.4 se muestra que el cable de comunicaci n indica la direcci n y posici n del eje  $X$ , y al lado derecho se muestra la direcci n y posici n del eje  $Y$ . Teniendo en cuenta la posici n del sistema de referencia del sensor l dar, se puede tomar los datos de las mediciones que se va realizando. El sensor l dar env a como datos la distancia de la medici n y el  ngulo donde se tom  la medici n. Estos datos obtenidos son tomados como coordenadas polares, las cuales son convertidas a coordenadas cartesianas. De esta forma estimamos las posiciones de los objetos que se encuentran dentro de un ambiente.

Color	Nombre de la se�al	Tipo	M�nimo	T�pico	M�ximo
Rojo	VCC	Potencia	4.9V	5V	5.5V
Amarillo	Tx	Salida	0V	3.3V	3.5V
Verde	Rx	Entrada	0V	3.3V	3.5V
Negro	GND	Potencia	0V	0V	0V
Azul	MOTOCTL	Entrada	0V	3.3V	5V

TABLA 3.2: Especificaciones el ctricas del sensor RPLidar A2.

### 3.1.3. Sistema de Percepci n Tridimensional

El sensor l dar es un sensor que permite construir mapas de dos dimensiones y, como ya fue mencionado, este sensor realiza mediciones mientras va rotando en  $360^\circ$ . Para este trabajo de tesis se necesita desarrollar un sistema mec nico para que el robot pueda realizar mediciones en los tres ejes del plano cartesiano mientras se va desplazando. Por tal motivo, se realiz  un dise o mec nico que es accionado por un motor paso a paso. Este dise o est  compuesto por una base que soporta al sensor l dar, la cual tiene un eje en la parte inferior. Un extremo del eje de la base se encuentra acoplado al eje del motor paso a paso, mientras que el otro extremo se encuentra unido a un rodamiento que permite que la rotaci n del motor se realice con el m nimo rozamiento. Este sistema mec nico fue dise ado con el software mec nico *Inventor*, y la Figura 3.5a muestra el dise o realizado

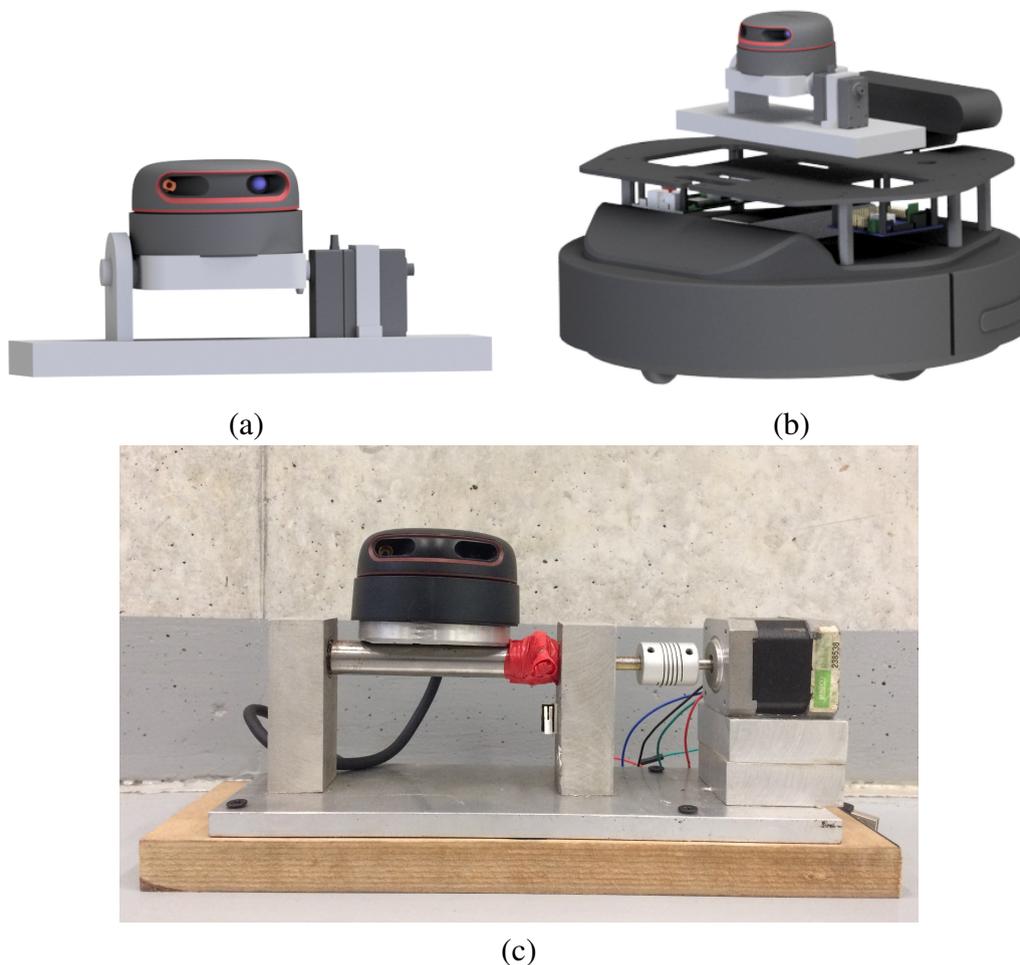


FIGURA 3.5: Diseño del sistema mecánico, desarrollado en un programa CAD. En (a) se muestra el diseño en CAD del sistema mecánico, en (b) se muestra la forma en que va ser colocado el sistema mecánico en el robot móvil kobuki y en (c) se muestra el diseño mecánico manufacturado en aluminio, junto al sensor lidar y el motor paso a paso.

en este programa. En la Figura 3.5b se puede ver la posición en la que estará colocado el sistema mecánico en el robot móvil.

El sistema mecánico final fue manufacturado, en aluminio, con las características mencionadas anteriormente agregando un switch de autocalibración. Como se puede ver en la Figura 3.5c, el switch se encuentra en la parte inferior del eje de la base. La idea principal de este sistema de autocalibración es indicarle al motor paso a paso en qué posición debe considerar el ángulo  $0^\circ$ . Una vez que este switch es accionado, el sensor

lídár se coloca en la posición que se muestra en la Figura 3.6. A partir de esta posición el sistema mecánico hace que el sensor lídár rote hacia atrás y hacia adelante teniendo un ángulo de apertura desde  $-88^\circ$  a  $+88^\circ$ . El motor paso a paso tiene una resolución de  $1^\circ$ . La rotación del motor en el rango de  $\pm 88^\circ$  hace que el sensor lídár pueda tomar medidas en los tres ejes del plano cartesiano  $(X, Y, Z)$ .

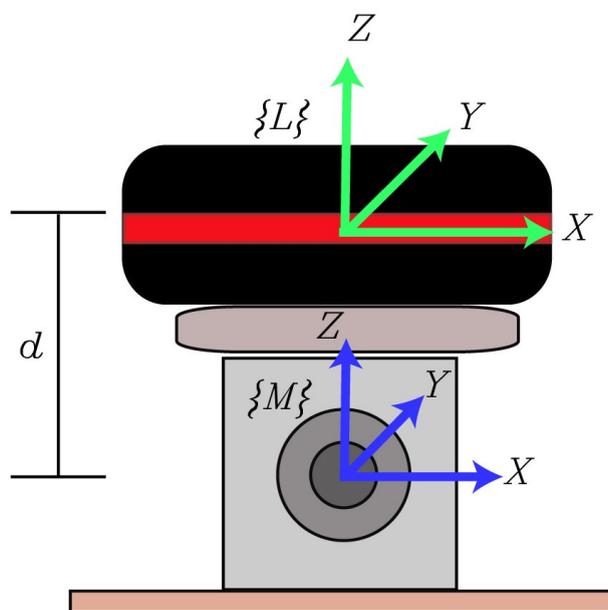


FIGURA 3.6: Representación gráfica de los sistemas de referencia del sistema mecánico, donde  $M$  y  $L$  son los sistemas de referencia del motor paso a paso y del sensor lídár, respectivamente.

Para generar el mapa en tres dimensiones se debe tener en consideración los sistemas de referencia de todos los componentes del sistema mecánico, en este caso el sensor lídár y el motor paso a paso como se muestra en la Figura 3.6. Este sistema genera un movimiento de rotación que es considerado dentro de las ecuaciones matemáticas para obtener los valores en los tres ejes del plano cartesiano. Una vez que el sistema mecánico es autocalibrado accionado por el switch, el sensor lídár es posicionado como muestra la Figura 3.7a; en esa posición se considera que el ángulo del motor paso a paso es  $0^\circ$ . Luego el sensor lídár comienza a rotar hacia el lado izquierdo hasta que llega al ángulo  $-88^\circ$  (ver Figura 3.7b). Una vez que el motor paso a paso llega al ángulo  $-88^\circ$ , éste comienza a girar hacia la derecha hasta que llega al ángulo  $88^\circ$ , como se muestran en las Figuras 3.7c y

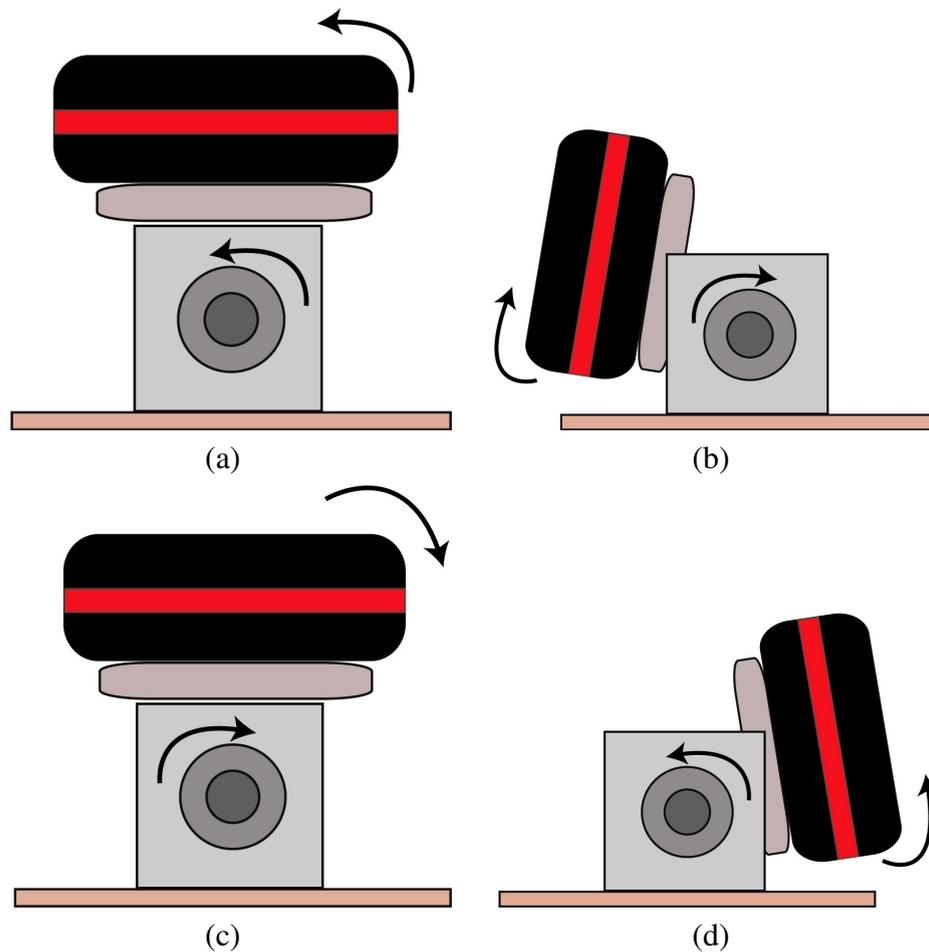
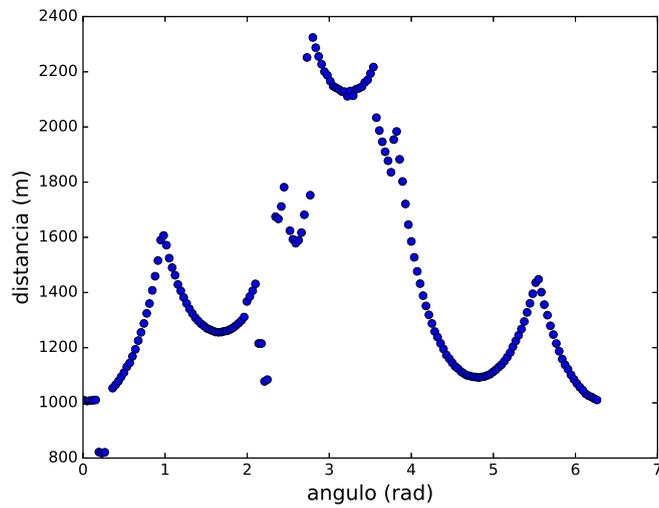


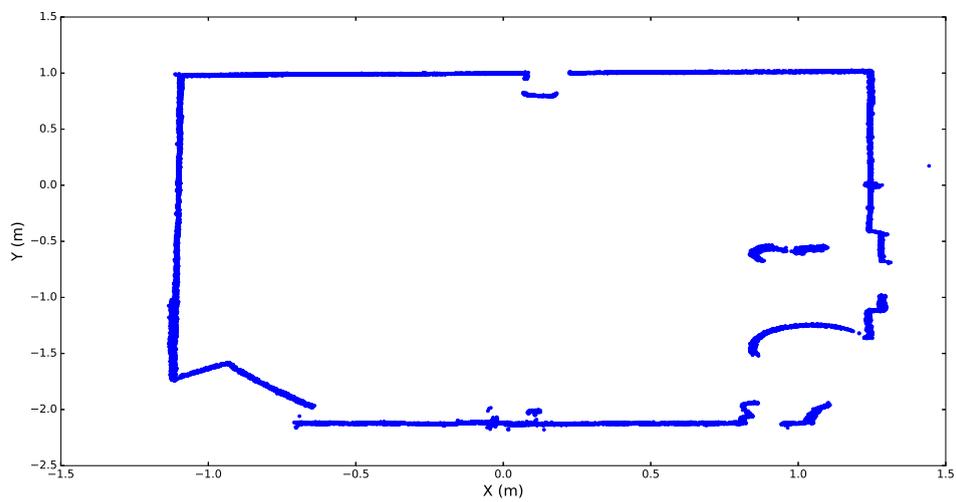
FIGURA 3.7: Representación gráfica de los movimientos de rotación del sistema mecánico. En (a) se muestra la posición inicial ( $0^\circ$ ) rotando hacia la izquierda, en (b) se muestra el sensor LIDAR rotado hacia el ángulo  $-88^\circ$  y rotando hacia la derecha. En (c) se muestra que el sensor LIDAR regresa a su posición original ( $0^\circ$ ) y va rotar hacia la derecha. En (d) se muestra el sensor LIDAR en el ángulo  $88^\circ$  rotando hacia la izquierda.

3.7d. El ángulo de apertura que tiene el motor paso a paso hace que el sensor LIDAR pueda tener un panorama más amplio para poder tomar mediciones dentro de un ambiente.

El mapa tridimensional se construye con las mediciones del sensor LIDAR y los ángulos del motor paso a paso. Para esto, primero se convierte los datos del sensor LIDAR; es decir, se hace una conversión de coordenadas polares a coordenadas cartesianas. En la Figura 3.8a se muestra un ejemplo de coordenadas polares obtenidas del sensor RPLidar A2 y en la Figura 3.8b se puede ver la conversión a coordenadas cartesianas. En esta



(a)



(b)

FIGURA 3.8: En esta imagen se muestra las mediciones obtenidas del sensor RPLidar A2, dentro de un ambiente cerrado con forma rectangular. En (a) se muestra los datos del sensor l dar utilizando coordenadas polares en los ejes, y en (b) se muestra la conversi n de los datos del sensor l dar a coordenadas cartesianas.

figura se puede ver que la parte inferior izquierda y la parte inferior derecha tienen esa forma debido a la presencia de obstáculos dentro del ambiente. Las mediciones mostradas en la Figura 3.8 fueron realizadas dentro de un ambiente cerrado con forma rectangular; esto sirve para explicar la diferencia entre las coordenadas. Las ecuaciones utilizadas para convertir las coordenadas son:

$$\begin{aligned}x &= r \cos(\theta_L), \\y &= r \sin(\theta_L),\end{aligned}$$

donde  $r$  es la distancia medida por el sensor y  $\theta_L$  es el ángulo de rotación del sensor lidar por cada medición que realiza.

Los valores del sensor lidar convertidos deben ser llevados al sistema de referencia del motor paso a paso, para lo cual se utiliza una matriz homogénea que contiene los componentes de rotación y traslación de todo el sistema mecánico. La matriz de transformación homogénea es representada como:

$$T_L^M = R_y(\phi_M)T_z(d),$$

donde  $\phi_M$  representa al ángulo de rotación del motor paso a paso y  $d$  la distancia entre el sistema de referencia del sensor lidar y el sistema de referencia del motor paso a paso, como se puede ver en la Figura 3.6. La matriz de transformación homogénea tiene una matriz de rotación, denotada por  $R_y(\phi_M)$ , originada por la acción del motor paso a paso en el eje  $Y$ , y una matriz de traslación, denotada por  $T_z(d)$ , en el eje  $Z$ , originada a partir de la pequeña distancia que existe entre los sistemas de referencia del sistema mecánico. La posición y orientación de los valores en los tres ejes del plano cartesiano (nube de

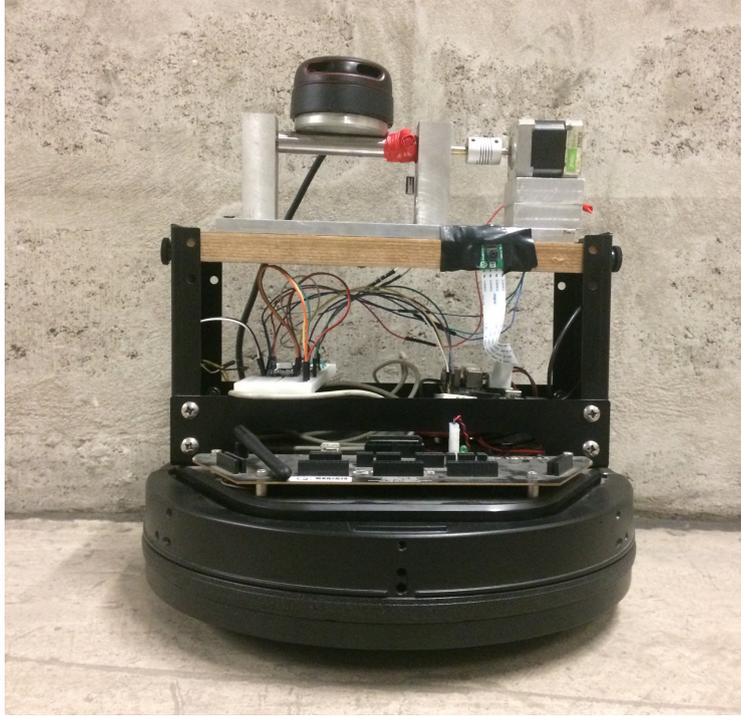


FIGURA 3.9: Esta figura muestra a *muqi*, el prototipo final, que fue utilizado para realizar las pruebas de exploración autónoma y mapeo de un ambiente en tres dimensiones.

puntos) es representada como:

$$\tilde{p}^L = T_L^M \tilde{p}^M \quad (3.1)$$

donde  $\tilde{p}^L$  representa la coordenada homogénea con respecto al sistema de referencia del sensor lidar,  $T_L^M$  es la matriz homogénea descrita anteriormente y  $\tilde{p}^M$  es la coordenada homogénea con respecto al sistema de referencia del motor paso a paso. La multiplicación de estas matrices ayuda a encontrar las coordenadas en los ejes  $(X, Y, Z)$  del ambiente que se está mapeando.

### 3.1.4. Prototipo Final

El prototipo final del presente trabajo de tesis se muestra en la Figura 3.9. Este robot puede desplazarse de forma autónoma en un ambiente desconocido y a su vez puede

generar un mapa en tres dimensiones del lugar. El prototipo está compuesto por un robot móvil diferencial Kobuki, un sensor lidar y un motor paso a paso, los cuales se encuentran conectados a un microcontrolador Raspberry Pi 3. Este microcontrolador es usado para poder controlar todos los componentes mencionados y poder almacenar la información de la nube de puntos que genera el sistema mecánico. El prototipo final es llamado *muqi*, debido al duende minero contado en las leyendas de la sierra sur.

### 3.2. Generación de Trayectoria basada en Campos Potenciales

Para hacer que el robot se mueva a su objetivo final sin toparse con obstáculos, se hace uso de la teoría presentada en la sección 2.3.1. Primero, un objetivo final deseado  $q_g = (x_g, y_g, \theta_g)$  que describe la posición y la orientación se especifica en términos del marco inercial. Entonces, los datos locales actuales del entorno se obtienen por medio de un sensor como un lidar, que fue usado en los experimentos, pero se puede usar cualquier otro sensor de profundidad. Estos datos proporcionan las posiciones de los obstáculos que se encuentran en el campo de visión del robot, y se utilizan como posiciones de obstáculos que definen el campo de repulsión.

Usando el campo potencial definido por la posición deseada y los datos detectados, se toma un paso en la posición actual. Este paso da un pequeño incremento en la dirección y define la posición deseada que debe seguir el controlador polar. A medida que el robot se mueve, esta posición deseada también cambia según el campo potencial, guiando el movimiento hacia la meta y evitando los obstáculos. A medida que el robot se mueve, continúa escaneando y actualizando su mapa de obstáculos y, por lo tanto, actualizando todo el campo potencial compuesto por las partes atractivas y repulsivas.

Las fuerzas proporcionadas por el campo de potencial artificial se descomponen en magnitud y dirección que conducen continuamente al robot. Para mostrar cómo funciona el enfoque, se realiza una simulación, que utiliza obstáculos colocados artificialmente

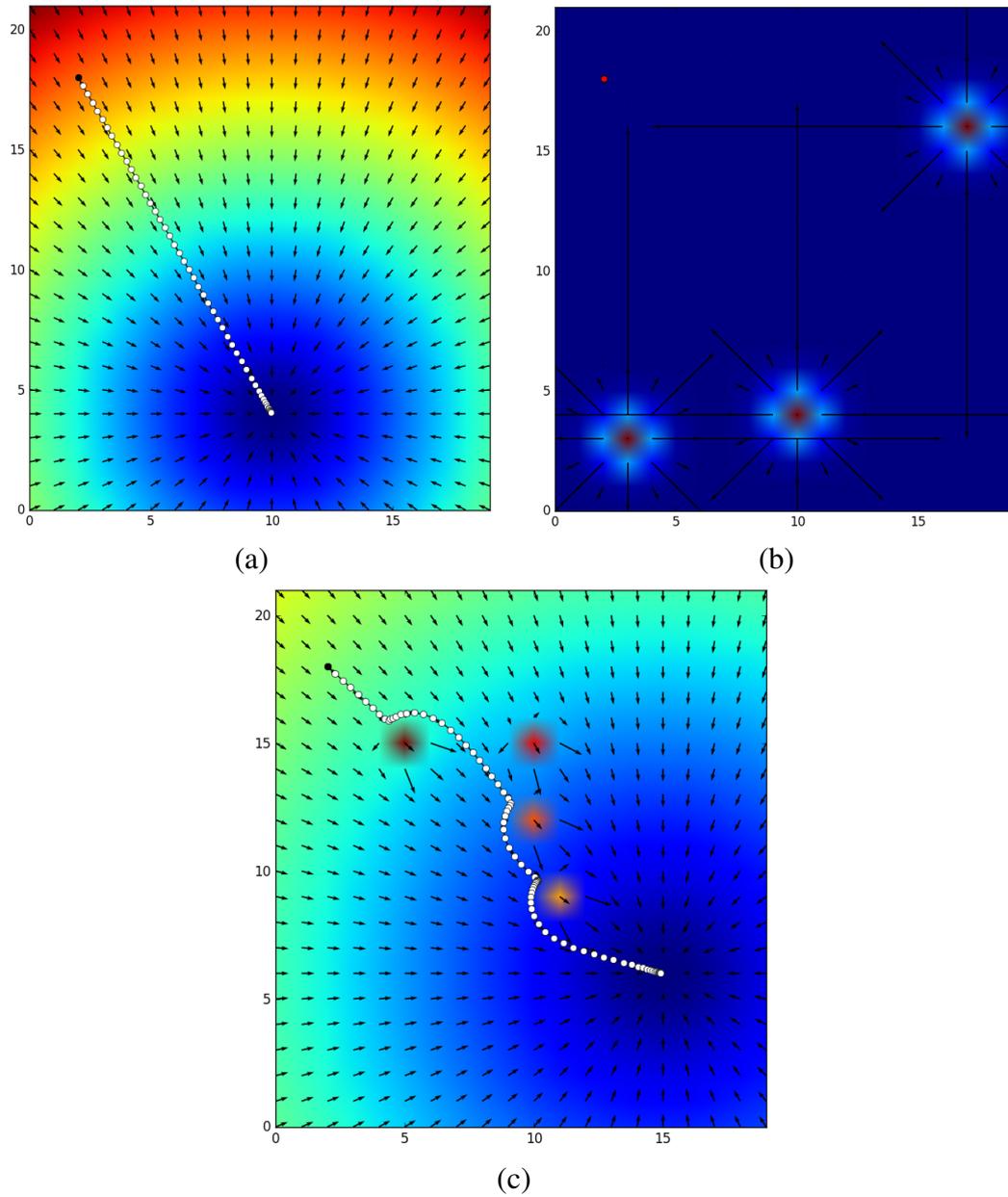


FIGURA 3.10: En esta figura se muestra una simulación del algoritmo de campo potencial artificial. En (a) se muestra la prueba para las fuerzas de atracción, en (b) se muestra la prueba de las fuerzas de repulsión de cada uno de los obstáculos y en (c) se muestra la fuerza de navegación.

y una posición deseada. La Figura 3.10a muestra el campo de atracción y las fuerzas que guían hacia la posición deseada. El robot simulado se encuentra inicialmente en la esquina superior izquierda  $q_0 = (2, 18)$  y se mueve hacia la meta  $q_{goal} = (10, 4)$  guiado por el campo potencial (no se considera orientación). Los puntos blancos muestran la ruta que debe seguir el robot y que luego guiará al controlador. Un campo repulsivo puro se muestra en la Figura 3.10b, donde se agregaron tres obstáculos en diferentes posiciones del espacio de trabajo. Cada obstáculo se proyecta como un círculo rojo debido a su gran magnitud dentro de su campo potencial, donde las fuerzas apuntan hacia afuera en cada obstáculo. La fuerza de navegación total compuesta por la superposición de las fuerzas de atracción y repulsión se muestra en la Figura 3.10c.

### 3.3. SLAM basado en Filtros de Partículas

Los filtros de partículas son un método que fue introducido en años recientes como un medio eficaz para resolver el problema de la localización y mapeo simultáneo [43]. Para este trabajo de tesis se utilizó un paquete de código abierto llamado *gmapping*. Este paquete de *ROS (Robot Operating System)* emplea el algoritmo SLAM basado en filtros de partículas [44]; la teoría de este algoritmo fue presentada en la sección 2.5.2.

El SLAM basado en filtros de partículas consiste en que cada partícula lleva la información de un mapa del entorno. Este mapa se enfoca en calcular la probabilidad del mapa no solo utilizando la odometría del robot, sino también incluye las mediciones que hace un sensor, en este caso un sensor lidar que se encuentra encima del robot. Estas informaciones enviadas al algoritmo hacen que la incertidumbre sobre la posición y orientación del robot disminuya.

Uno de los principales problemas de la técnica SLAM basada en filtros de partículas es la cantidad de partículas requeridas para construir un mapa preciso. Este efecto

es conocido como el problema del agotamiento de partículas. Por tal motivo el paquete *gmapping* propone dos pasos para solucionar este problema. El primero consiste en calcular la probabilidad alrededor de la posición y orientación del robot dependiendo de la cantidad de partículas obtenidas mediante los datos del sensor láser. Como fue mencionado anteriormente al tener la información del sensor láser y la odometría se puede obtener un mapa corregido y más preciso. Además, el error de estimación de la posición del robot disminuye con el tiempo y se requiere menos partículas para representar las futuras posiciones del robot dentro del entorno. El segundo paso consiste en un remuestreo adaptativo, el cual permite un remuestreo cada vez que sea necesario, manteniendo una cantidad razonable de partículas. Con este paso el paquete *gmapping* reduce el problema del agotamiento de partículas.

Este algoritmo muestra el mapa, en dos dimensiones, del entorno a mapear. El mapa obtenido tiene colores establecidos que identifican cada zona del lugar que se está mapeando. Como se mencionó en la sección 2.3.3, el mapa originado por el SLAM tiene tres zonas, cada una con color característico. Estas zonas son: (i) zona desconocida, la cual está representada por un área de color verde oscuro, y muestra al robot qué lugares aún le falta explorar; (ii) zona explorada, la cual está representada por una área de color gris y muestra qué lugares del entorno ya ha explorado; (iii) zona de obstáculos, representada por un área de color negro, y muestra al robot la posición de los obstáculos que se encuentran dentro del ambiente que se está explorando.

### **3.3.1. Sistema Operativo del Robot (ROS)**

ROS es una colección de entornos de trabajo para el desarrollo de software de robots. Este sistema tiene una amplia variedad de herramientas, librerías y paquetes que buscan la creación de software complejo para tener robots robustos con un comportamiento variado. La finalidad de ROS es crear un compendio universal de software para que los

nuevos desarrolladores no tengan que volver a hacer algoritmos que ya se encuentran desarrollados, además de brindar modularidad al sistema de un robot.

El conjunto de herramientas y librerías que son proporcionadas por ROS, permiten lograr una comunicación entre programas. Esto quiere decir que se puede comunicar entre sí diferentes programas de un mismo sistema, ya sea en el mismo computador o en varios computadores, todos enfocados en conseguir un objetivo común. Asimismo, los paquetes de ROS son algoritmos que son implementados con frecuencia en robótica. Por la facilidad de comunicación y gestión de mensajes ROS se ha convertido en la plataforma utilizada por muchas empresas, fabricantes y laboratorios de investigación dedicados a la robótica. Finalmente, debido al diseño de comunicación para los mensajes, ROS permite simular sensores y actuadores de forma independiente y a su vez de forma sistemática de manera iterativa, usando por defecto el simulador dinámico llamado Gazebo.

La estructura de ROS está conformada por: (i) **tópicos**, que son canales de comunicación a través de los cuales diversos nodos pueden comunicarse mediante publicación o suscripción; (ii) **paquetes**, que puede contener un nodo, una librería o un conjunto de datos; (iii) **nodos**, que son procesos que intercambian información para crear tareas complejas; (iv) **stack**, que es un conjunto de paquetes con alguna funcionalidad específica. Además ROS tiene un visualizador llamado Rviz.

Rviz es un herramienta de visualización en tres dimensiones, que es parte nativa de ROS. Esta herramienta posibilita que cualquier plataforma robótica pueda ser representada en un modelo 3D, respondiendo en tiempo real a los experimentos que se hace en el mundo real. Además, esta herramienta se puede usar para mostrar lecturas de sensores, datos recibidos por la visión estereoscópica o visualizar el algoritmo SLAM evitando obstáculos, etc. Esta herramienta dispone de muchas opciones de configuración.

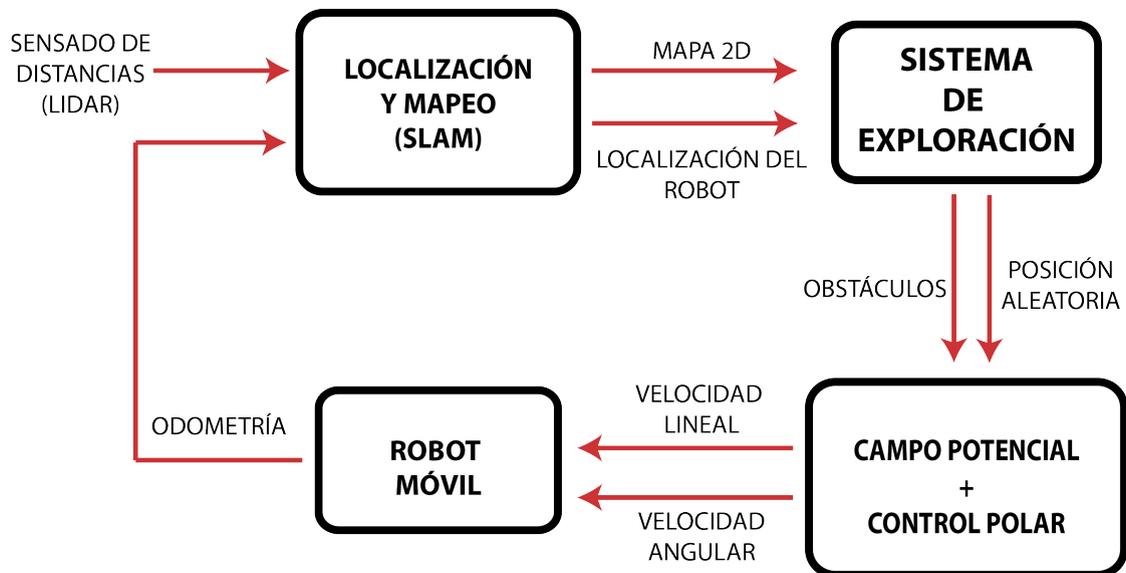


FIGURA 3.11: Estructura del sistema para realizar la exploración del robot móvil diferencial, de forma autónoma.

### 3.3.2. Estructura del Sistema de Autonomía del Robot Móvil

La estructura del sistema de navegación autónoma del robot móvil diferencial se muestra en la Figura 3.11. Para que el robot móvil pueda localizarse dentro de un entorno desconocido, se utiliza el primer bloque que es el SLAM. Este algoritmo tiene como entradas los datos del sensor lidar y la odometría del robot  $(x_{medido}, y_{medido}, \theta_{medido})$ . El algoritmo SLAM, como se mencionó anteriormente, trabaja con filtros de partículas que ayudan a corregir la posición del robot dentro del ambiente que se va a desplazar. El SLAM tiene dos salidas, una es el mapa en dos dimensiones del lugar y la segunda salida es la localización corregida del robot en el ambiente  $(x_{corregido}, y_{corregido}, \theta_{corregido})$ . El mapa generado por el SLAM se da en tiempo real, lo cual permite que el robot pueda saber qué lugares se encuentra explorando.

El segundo bloque de la estructura consiste en la etapa de exploración. Este algoritmo tiene como entradas la localización del robot y el mapa en dos dimensiones dadas por el algoritmo SLAM. Con la información de la localización del robot y el mapa en dos

dimensiones la etapa comienza a generar posiciones deseadas de forma aleatoria dentro del ambiente, teniendo en consideración los obstáculos que se encuentran dentro de ella.

El tercer bloque consiste en la navegación del robot móvil. Esta navegación es realizada junto a los campos potenciales, considerando que el control de posición y orientación del robot móvil se realiza utilizando el controlador polar. Estos conceptos fueron explicados en la sección 2.3.1 y la sección 2.2.2 respectivamente. Los métodos mencionados son empleados para que el robot pueda generar su propia trayectoria evitando los obstáculos, y a su vez pueda tener un movimiento suave controlando su posición y orientación dentro del plano cartesiano. Este bloque necesita de la posición de los obstáculos en el ambiente y de la posición deseada aleatoria, para que el robot, junto a los campos potenciales pueda generar su propia trayectoria gracias a las fuerzas de atracción y de repulsión. Asimismo, el controlador polar permite que el robot móvil pueda llegar exactamente a la posición deseada dada de forma aleatoria. Finalmente, esta etapa tiene como salidas a la velocidad lineal y a la velocidad angular. Estas velocidades ingresan al robot móvil diferencial haciendo que éste se desplace dentro del ambiente donde se está explorando. El robot móvil como salida tiene los valores de la odometría, los cuales ingresan a la etapa del SLAM haciendo que esta estructura se repita de forma iterativa.

La interacción autónoma del robot móvil dentro de un ambiente desconocido se puede ver en la Figura 3.12. En esta Figura se muestra la etapa inicial de la exploración del robot móvil, donde el robot móvil está representado por el círculo de color negro y la figura geométrica de color rojo representa las luces láser que emite el sensor líder para realizar las mediciones. En esta etapa el sensor líder comienza a tomar las primeras mediciones del ambiente desconocido, y con esta información se comienza a construir el mapa en dos dimensiones del lugar. La Figura 3.12(b) muestra el mapa originado por el SLAM, el cual tiene los colores característicos que representan las zonas mencionadas en la sección 3.3. En esta figura se puede ver cómo el robot móvil se desplaza dentro del ambiente. Cuando el robot recibe la información del mapa en dos dimensiones del

algoritmo SLAM sabe que debe desplazarse por las zonas de color gris. En ese instante comienza a generar posiciones de forma aleatoria desplazándose hacia la izquierda. Este desplazamiento es originado por los campos potenciales explicado en la sección 2.3.1. Cuando el robot móvil se encuentra muy próximo a un obstáculo, en este caso una pared, los campos potenciales de repulsión evitan que el robot pueda chocarse contra la pared u objeto. Estas fuerzas de repulsión están representadas por flechas de color azul y la trayectoria del robot está representada por líneas de color verde, en la Figura 3.12(b). En la Figura 3.12(c) se muestra el desplazamiento que va realizando el robot móvil mientras va explorando el ambiente. Como se observa, el desplazamiento del robot es hacia la derecha, luego hacia la izquierda; esto se hace forma iterativa y el cambio sucede cuando el robot se encuentra con un obstáculo. Cuando el robot cuando se encuentra con un obstáculo, rota en su propio eje y comienza a desplazarse alejándose del mismo. El robot sigue explorando el ambiente, ya que el mapa SLAM sigue mostrando zonas desconocidas. Finalmente, en la Figura 3.12(d) se muestra la situación cuando el robot ya no puede seguir explorando debido a que a su alrededor se encuentran obstáculos que impiden el paso hacia las zonas que aún falta por explorar. En este caso el robot rota en su propio eje y regresa a la posición donde comenzó a explorar ( $X = 0, Y = 0$ ).

### **3.4. Estructura del Sistema de Mapeo en Tres Dimensiones**

En esta sección se presenta de forma detallada las etapas que conforman la estructura del sistema para generar el mapa en tres dimensiones del ambiente. En la Figura 3.13 se muestra la estructura que genera los valores dentro de los tres ejes del plano cartesiano, mientras el robot va explorando un ambiente.

El proceso comienza con la toma de mediciones de parte del sensor lidar, las mediciones que son obtenidas en tiempo real se encuentran en el formato de coordenadas

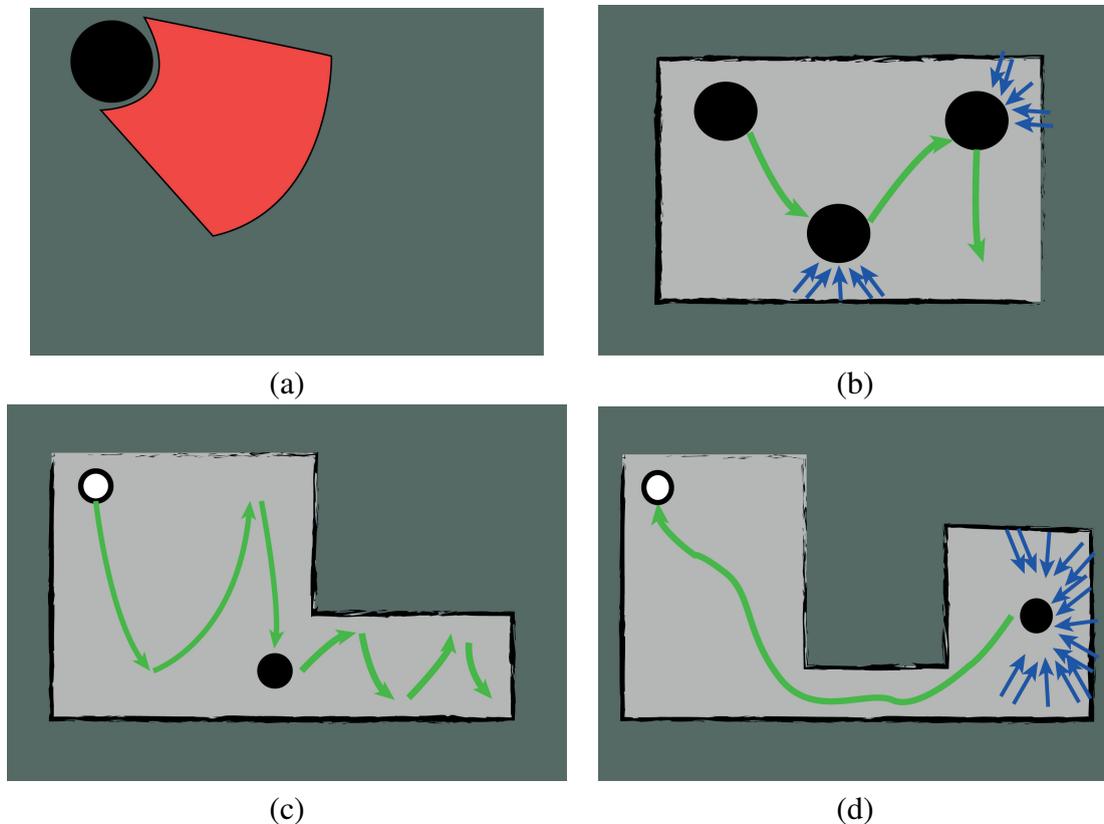


FIGURA 3.12: En esta figura se muestra de forma gráfica las etapas del sistema de autonomía del robot móvil. En (a) se muestra la etapa inicial que es el sensado de las distancias, en (b) se muestra la evasión de obstáculos del robot móvil, en (c) se muestra las posiciones aleatorias por donde el robot tiene que desplazarse para generar su trayectoria, y en (d) se muestra como el robot ya no sabe por donde avanzar y regresa a su posición inicial.

polares  $(r, \theta)$ . Estas medidas deben ser convertidas a coordenadas cartesianas para encontrar las posiciones de los objetos dentro del plano cartesiano bidimensional  $(X_L, Y_L)$  y además conocer las dimensiones que tiene el ambiente que se está explorando.

La segunda etapa de este proceso consiste en hallar los valores de los ejes del plano cartesiano tridimensional, a partir de transformaciones homogéneas. Para generar estos valores se necesita obtener los ángulos de rotación del motor paso a paso. Las matrices de transformación homogénea y el principio de funcionamiento del sistema mecánico fueron explicados en la sección 3.1.3. Este sistema mecánico cuenta con dos sistemas de referencia, uno es el sistema de referencia del sensor lidar y el segundo es el sistema de

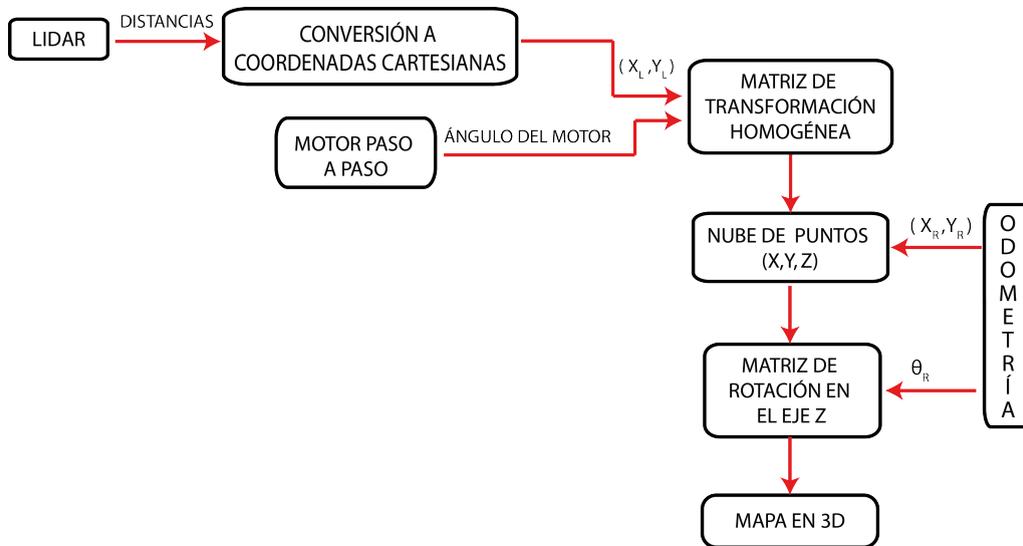


FIGURA 3.13: Estructura del sistema para la construcción del mapa en tres dimensiones, mientras el robot móvil se va desplazando.

referencia del motor paso a paso. Asimismo, el sistema mecánico genera movimiento de rotación y un efecto de traslación; por ende, se utiliza transformaciones homogéneas ya que esto está compuesto por una matriz de rotación en el eje  $Y_M$  del motor paso a paso y una matriz de traslación con un desplazamiento en el eje  $Z_M$  de dicho motor. La potencia y la resolución angular del motor paso a paso permiten que el sensor lidar tenga un ángulo de apertura de  $176^\circ$ , el cual permite al sensor tener una amplia proyección en el eje  $Z$  para tomar mediciones. El resultado de las transformaciones homogéneas es una nube de puntos del entorno explorado  $(X, Y, Z)$ .

Finalmente, la última etapa del sistema para generar el mapa en tres dimensiones consiste en hacer que el mapa tenga correlación con el desplazamiento del robot dentro del ambiente. Para encontrar dicha correlación se debe obtener los valores de la odometría del robot y estos valores deben ser sumados a las posiciones del plano cartesiano tridimensional obtenido de las transformaciones homogéneas. La odometría del robot móvil diferencial es la estimación de la posición y orientación de éste mientras se encuentra explorando el ambiente. La odometría del robot esta compuesta por tres valores  $(X_R, Y_R, \theta_R)$ , los cuales son sumados a la nube de puntos  $(X, Y, Z)$ . Cuando el robot realiza rotaciones en su

propio eje ( $Z_R$ ) para el cambio de dirección en su trayectoria, esto debe ser considerado en la generación de la nube de puntos. Para este caso se utiliza el valor del ángulo de rotación del robot móvil, obtenido de la odometría ( $\theta_R$ ), dentro de una matriz de rotación del eje  $Z$ . Esta matriz de rotación se multiplica con la nube de puntos obtenida anteriormente y el resultado es el mapa local en tres dimensiones del ambiente real explorado en ese instante.

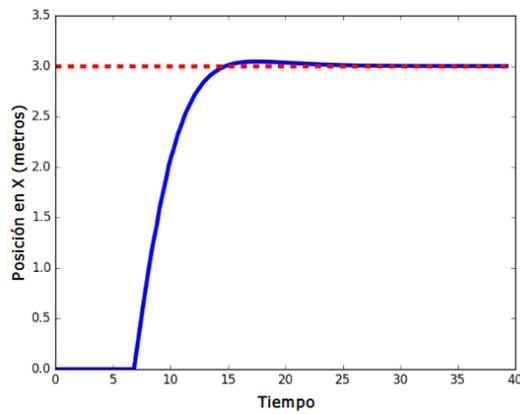
## Capítulo 4

### Resultados

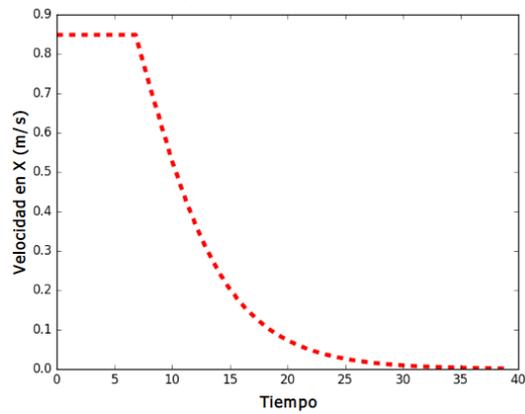
En el presente capítulo se muestra los resultados que fueron obtenidos al realizar las diferentes pruebas con el sensor RPLidar A2 y el robot móvil Kobuki. Las pruebas fueron realizadas de manera independiente para cada componente y también se hizo pruebas con el sistema en conjunto del robot explorador *muqi*. Estas pruebas fueron realizadas utilizando el simulador Gazebo y también dentro de ambientes reales. Este simulador trabaja en conjunto con ROS y permite poder diseñar robots y probar algoritmos de forma rápida en diferentes escenarios. Finalmente, se explicará con detalle cada una de las pruebas y resultados que se realizaron.

#### 4.1. Resultados del Controlador Polar

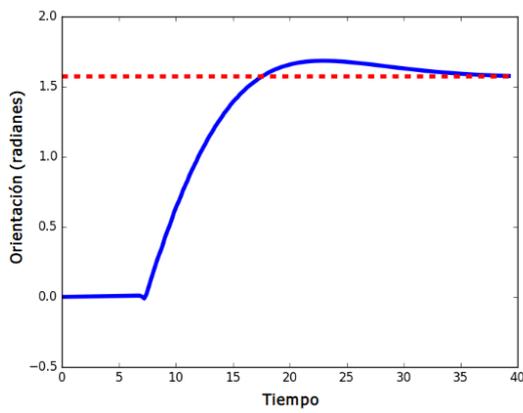
Para probar el controlador polar, se usó una simulación dinámica en Gazebo con el robot Kobuki sin obstáculos, y las variables de estado que incluyen velocidad, posición y orientación se obtuvieron en línea a partir de la odometría simulada. Usando la información de esta odometría, el controlador se aplicó en línea. Para estas pruebas, la posición deseada es  $(x = 3m, y = 3m)$  y la orientación deseada  $\theta = 90^\circ$ . La Figura 4.1a muestra la evolución temporal de la posición  $(x)$  donde se logra una convergencia a la posición deseada en menos de 20 segundos. Esto se debe a la distancia hacia el obstáculo. Diferentes distancias conducen a diferentes tiempos de convergencia, y la tasa de convergencia también se puede modificar cambiando las ganancias en 2.2 para la velocidad angular, y en 2.1 para la velocidad lineal. Las otras subfiguras en la Figura 4.1 muestran la evolución



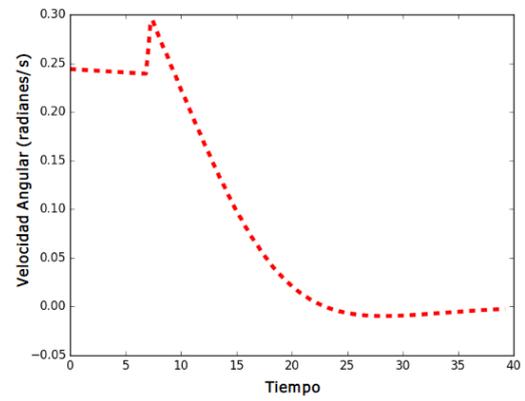
(a)



(b)



(c)



(d)

FIGURA 4.1: Evolución temporal de las variables de estado usando el controlador polar para lograr una posición deseada dada por  $x = 3m, y = 3m, \theta = 90$ . En (a) se muestra la evolución temporal en la posición del eje  $x$ , en (b) se muestra la evolución de la velocidad lineal, en (c) se muestra la evolución de la orientación y en (d) se muestra la evolución de la velocidad angular.

temporal de la velocidad lineal en  $x$ , la orientación y la velocidad angular. Para la orientación, hay un sobreimpulso que se debe a la excesiva dependencia del controlador en la posición en lugar de la orientación (ver Figura 4.1c).

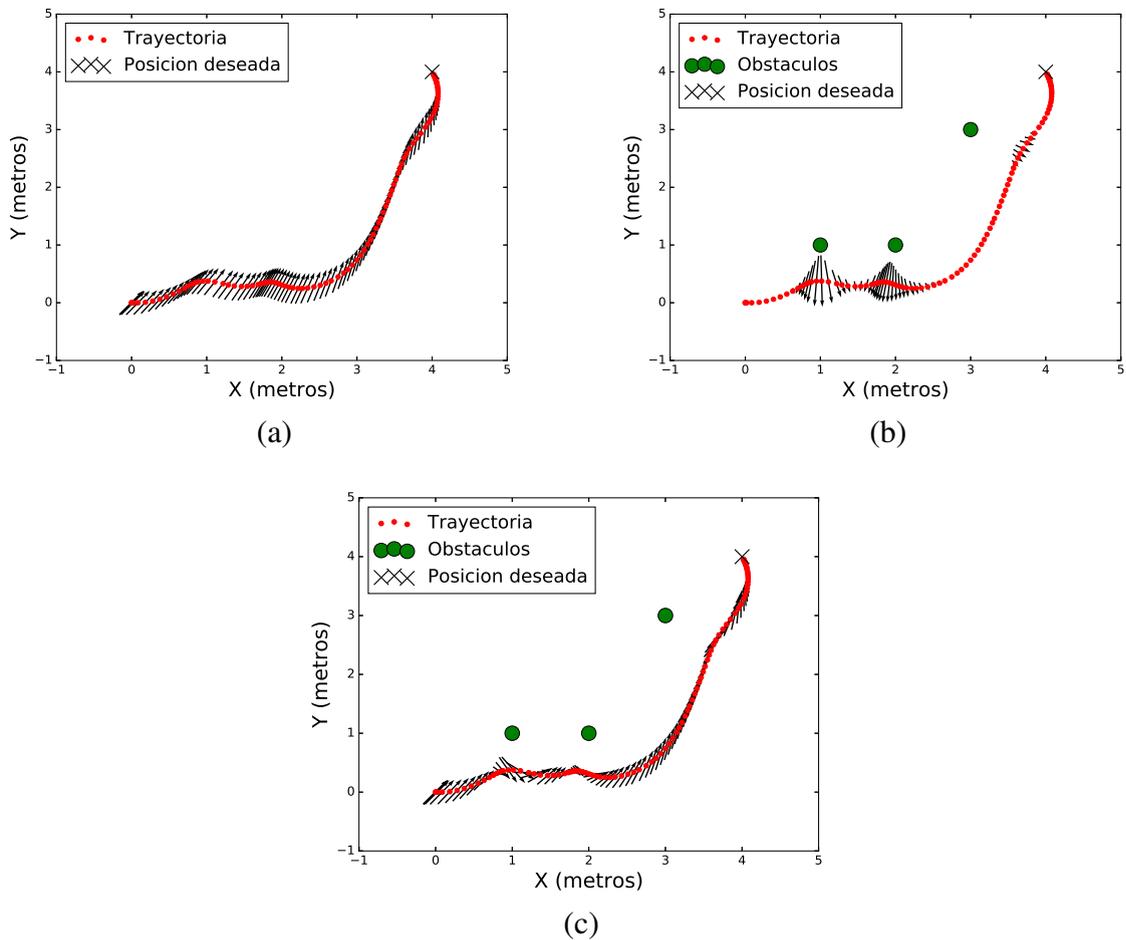


FIGURA 4.2: Navegación autónoma implementada en el robot diferencial Kobuki. En (a) se muestra las fuerzas de atracción que llevan al Kobuki a la posición deseada, en (b) se muestra las fuerzas de repulsión que evitan que el robot diferencial se choque con los obstáculos y en (c) se muestra las fuerzas de navegación que genera la trayectoria del robot.

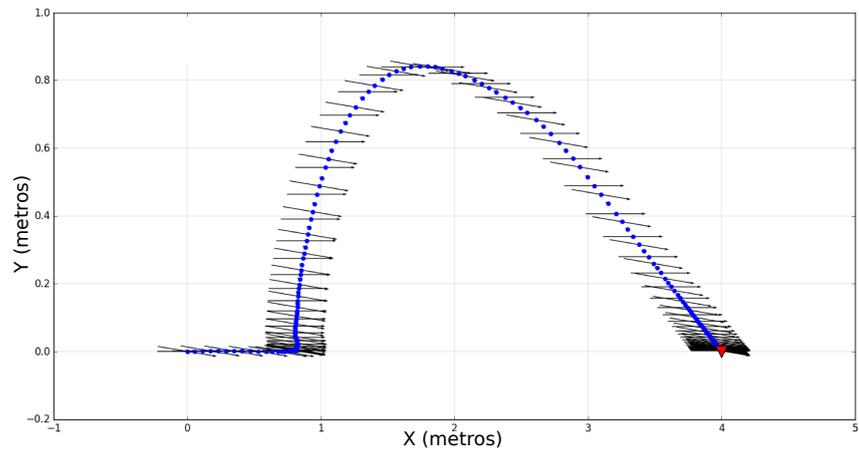
## 4.2. Resultados del Sistema de Navegación

La metodología propuesta se implementó como un algoritmo iterativo en el robot Kobuki. Como se describe en la sección 3.2, el algoritmo toma iterativamente posiciones deseadas donde a través del campo potencial artificial y el controlador polar impulsan al robot a través de las posiciones requeridas. Se expuso al robot a diferentes obstáculos

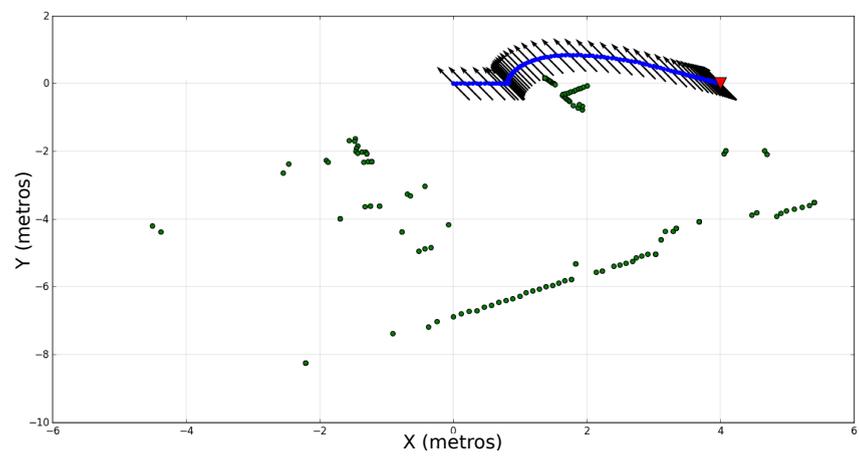


FIGURA 4.3: Navegación autónoma del robot diferencial Kobuki, montado con un sensor lidar y dos cajas como obstáculos.

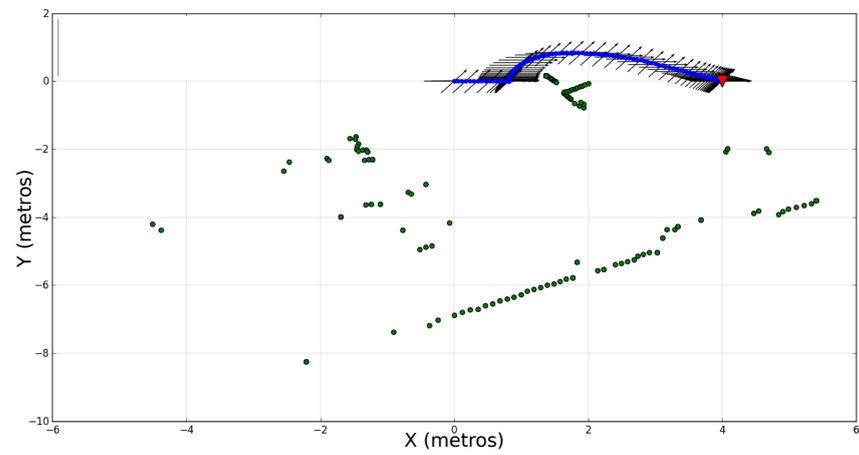
cuyas posiciones fueron conocidas a priori. La Figura 4.2 muestra un ejemplo de aplicación del sistema de navegación. La Figura 4.2a muestra las fuerzas de atracción para cada posición (puntos rojos) en las que el algoritmo se itera en el entorno de dos dimensiones dentro del movimiento del robot. Para cada posición, el campo de potencial atractivo se puede ver con la dirección de las flechas. La Figura 4.2b se compone de las fuerzas de repulsión basadas en los obstáculos que se muestran como marcadores verdes. Para cada obstáculo, la magnitud de las fuerzas aumenta cuando el robot está más cerca y su dirección señala los obstáculos, permitiendo que el robot los evite. La Figura 4.2c muestra la superposición de ambas fuerzas con los obstáculos reales. Cada fuerza proporciona una posición deseada intermedia para el robot, que constituye la posición deseada continuamente actualizada para el controlador polar. Aunque las fuerzas cercanas a los obstáculos tienen un alto índice de cambio, la trayectoria es suave. Esto demuestra la efectividad del sistema de navegación a pesar de las características no holonómicas del robot.



(a)



(b)



(c)

FIGURA 4.4: Campos atractivos y repulsivos, y la trayectoria que el robot real sigue usando datos en línea provenientes del sensor lidar montado en la parte superior. En (a) se muestra las fuerzas de atracción, en (b) las fuerzas de repulsión y las posiciones de los obstáculos, finalmente en (c) se muestra las fuerzas de navegación generando la trayectoria.

### 4.3. Resultados de la Navegación Autónoma con el Lidar en dos dimensiones

Para probar la autonomía del robot móvil en un entorno real, se usa un sensor lidar (RPLidar A2) que se colocó sobre el robot Kobuki. Se usa el sensor lidar para poder hacer una actualización continua de las posiciones de los obstáculos a medida que el robot se mueve. El robot usa tópicos creados en ROS para obtener la información del sensor, la cual se compone en rango y orientación por cada punto medido, a medida que el sensor lidar gira. La información del lidar se tuvo que convertir a coordenadas cartesianas para conocer las posiciones de cada obstáculo dentro del espacio de trabajo. Estas posiciones van como entrada hacia el algoritmo principal. El robot Kobuki puede desplazarse dentro del mapa generado por el lidar, para que esto ocurra se tuvo que considerar el marco de referencia del robot como el marco de referencia del sensor lidar. Entonces, el algoritmo de navegación autónomo usa las coordenadas del entorno detectados mediante el sensor lidar y así genera su propia trayectoria evitando los obstáculos del entorno. Para una prueba, se colocaron dos cajas en el espacio de trabajo, como se ve en la Figura 4.3. La posición del objetivo era  $(x = 4m, y = 0m)$  y la posición donde inició el robot fue  $(x = 0m, y = 0m)$ . La Figura 4.4a muestra la trayectoria del robot en azul y las flechas indican la fuerza de atracción que llevan al robot hacia la posición deseada. La Figura 4.4b muestra el mapa completo, donde los puntos verdes representan el entorno y los obstáculos del entorno, detectados por el lidar, donde se mueve el robot. En este caso los obstáculos del entorno son las dos cajas mostradas en la Figura 4.3 y que se muestran muy cercanas a la trayectoria formada por el robot en la Figura 4.4b. Como se observa, estas flechas apuntan hacia afuera del obstáculo, proporcionando autonomía al robot mientras intenta alcanzar la posición deseada. Estas flechas representan a las fuerzas de repulsión, que evitan la colisión entre el robot y los obstáculos del entorno. Finalmente en la Figura 4.4c se muestra toda la fuerza de navegación que impulsa el movimiento del robot móvil, asimismo esta fuerza genera la trayectoria representada por los puntos de color azul. La forma de la trayectoria es debido a las dos cajas que se encuentran entre la posición inicial y la posición final.

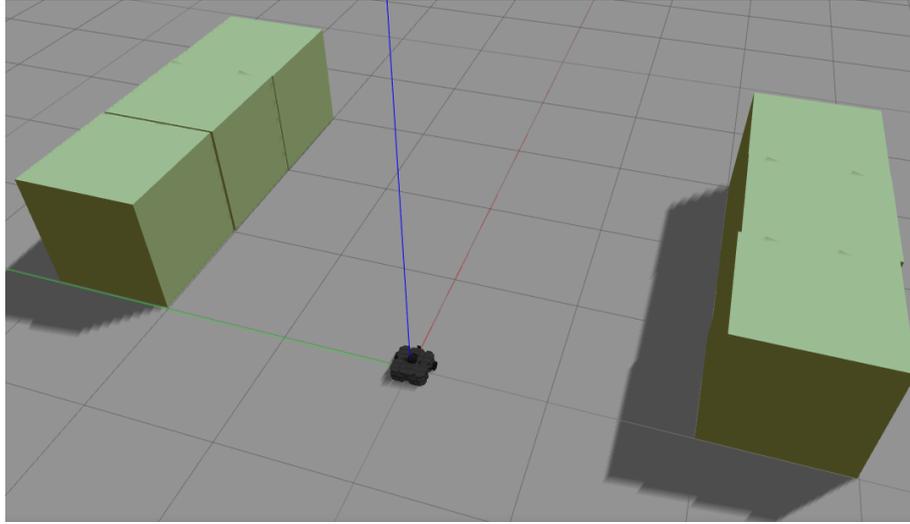
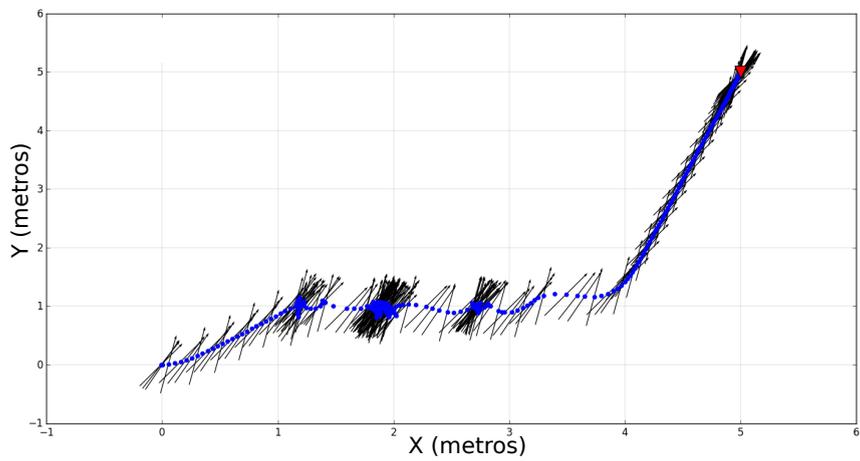


FIGURA 4.5: Robot móvil Kobuki en medio de los obstáculos colocados para probar la navegación autónoma utilizando el algoritmo SLAM. Esta prueba se realizó en el simulador Gazebo.

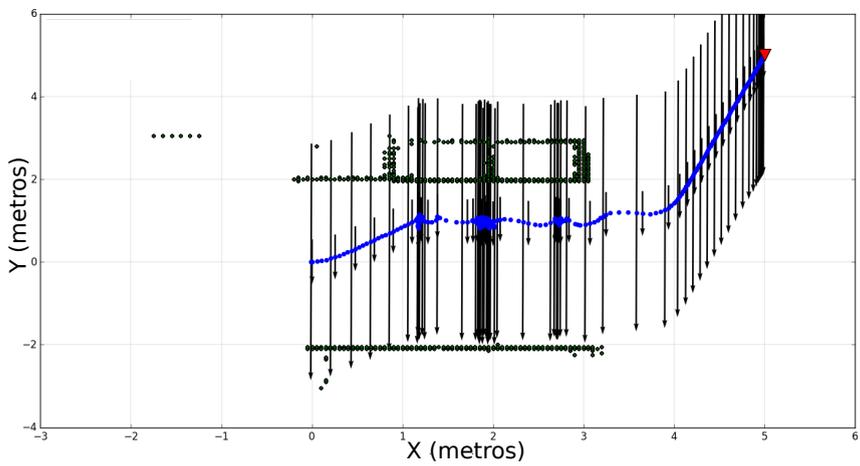
#### 4.4. Resultado de la Navegación Autónoma usando SLAM en dos dimensiones

Para probar la navegación autónoma del robot Kobuki usando el algoritmo SLAM, se utilizó el simulador dinámico Gazebo. En este simulador se creó un escenario donde se colocó al robot Kobuki en medio de varios obstáculos. Los obstáculos fueron colocados de tal forma que pueda representar la entrada de un túnel. En la Figura 4.5 se puede ver el escenario que fue creado. Un conjunto de obstáculos tiene una longitud de 3 metros y el ancho entre los conjuntos de obstáculos es de 4 metros. El robot tiene que desplazarse por ese espacio hasta llegar a la posición deseada, estimar la posición de los obstáculos y construir el mapa en dos dimensiones.

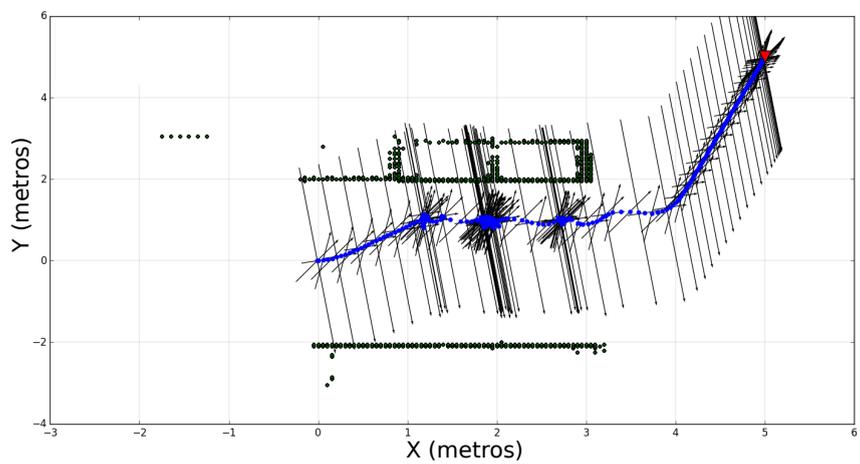
Como se mencionó en la sección 3.3, para realizar esta prueba se utilizó el paquete de SLAM llamado *gmapping*, dado por ROS. Este paquete permite crear un mapa en dos dimensiones a partir de los datos del sensor lidar y la odometría del robot móvil. Cuando se ejecuta el paquete de SLAM, empieza a crear tópicos por donde se envían los datos de la odometría del robot y los datos del lidar. Con la información recibida por el nodo del *gmapping*, éste publica las dimensiones del entorno, dentro de un tópico, hacia el robot



(a)



(b)



(c)

FIGURA 4.6: Navegación autónoma del Kobuki utilizando SLAM. En (a) se muestra las fuerzas de atracción, en (b) se representa las fuerzas de repulsión y en (c) la suma de las fuerzas de atracción y repulsión.

Kobuki. Para que el robot móvil se pueda desplazar con respecto a las dimensiones del mapa recibido por el SLAM, su marco de referencia debe ser remplazado por el marco de referencia del sensor lidar. La precisión del mapa originado por el SLAM es buena, debido a que el algoritmo reduce la cantidad de mediciones que envía el sensor lidar haciendo que el mapa sea uniforme y los obstáculos se encuentren en la posición correcta dentro del mapa.

Para el desplazamiento autónomo del robot, dentro del escenario creado, se utilizó las etapas de localización y mapeo (SLAM), y campo potencial más controlador polar que fueron explicados en la sección 3.3.2. Como primer paso el sensor lidar permite localizar las posiciones de todos los obstáculos dentro del ambiente. Estas posiciones son enviadas al algoritmo de campos potenciales para así generar la trayectoria. Para realizar esta prueba se elige como posición deseada el punto  $(x = 5m, y = 5m)$ . En la Figura 4.6a se puede denotar los puntos de color azul que representan la trayectoria por la cual el robot Kobuki se ha desplazado. La trayectoria va desde la posición  $(x = 0m, y = 0m)$  donde comenzó el robot hasta la posición deseada que está representada por un triángulo invertido de color rojo. También se puede ver que hay flechas de color negro que tienen diferentes tamaños y diferentes direcciones. El tamaño de la flecha indica la velocidad lineal a la que el robot debe desplazarse y la dirección indica la orientación necesaria para que el robot pueda llegar a la posición deseada. El tamaño de las flechas depende de la posición de los obstáculos y de la posición que se quiere llegar con respecto a la posición del robot que se va desplazando cada intervalo de tiempo.

En la Figura 4.6b los puntos verdes representan a los obstáculos dentro del escenario; como se puede observar los puntos son uniformes y alineados. El algoritmo SLAM del paquete de *gmapping* se encarga de reducir la cantidad de mediciones del sensor lidar, luego añade los datos de la odometría haciendo que la estimación de las posiciones de los objetos dentro del ambiente sea de forma clara y precisa. En esta figura también se puede observar las flechas del campo de repulsión, donde cada flecha tiene una dirección hacia

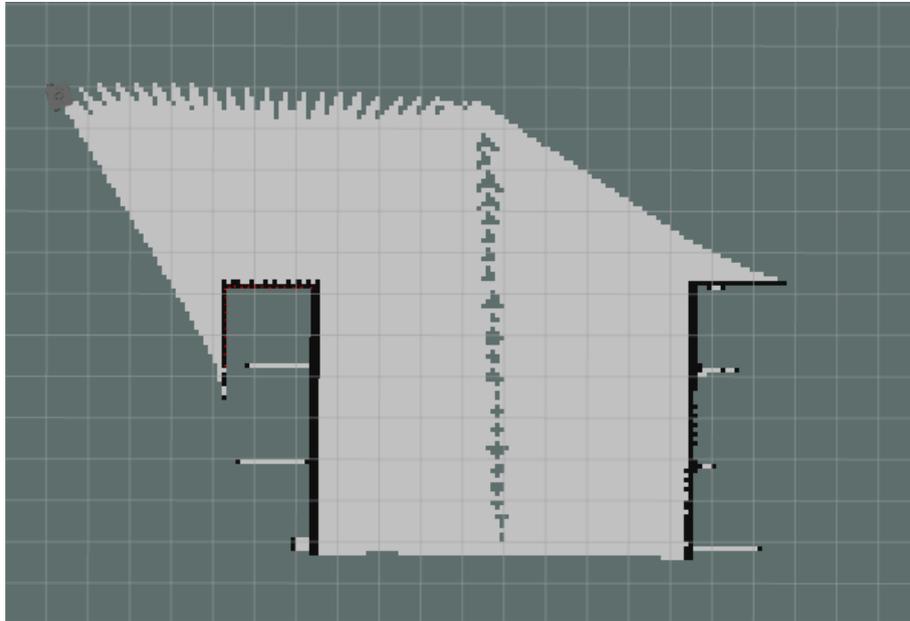


FIGURA 4.7: Mapa en dos dimensiones obtenido del algoritmo SLAM, el mapa es visualizado en la herramienta Rviz.

afuera en cada obstáculo. El tamaño de las flechas va acorde con la trayectoria del robot Kobuki, esto quiere decir que mientras el robot se encuentre más cerca al obstáculo el tamaño de la flecha será más grande y por ende la fuerza de repulsión será mayor. Éstas fuerzas ayudan a que el robot pueda evadir los obstáculos y evitar las colisiones.

Finalmente, en la Figura 4.6c se muestra las sumas de fuerzas de atracción y repulsión. La suma de estas fuerzas generan la trayectoria que debe realizar el robot para llegar a la posición deseada, evitando los obstáculos. En esta figura se puede observar que en algunas regiones de la trayectoria se nota una concentración de fuerzas de navegación. Esto se debe a la interacción de su movimiento autónomo, ya que el robot va retrocediendo para evitar el choque originado por el campo de repulsión y a su vez se encuentra atraído hacia la posición deseada originado por el campo de atracción.

La Figura 4.7 muestra el mapa generado por el paquete *gmapping*. Como se puede ver en la imagen, el mapa está compuesto por tres colores diferentes. Estos colores quieren dar a mostrar el espacio libre por donde el robot puede desplazarse, también muestra los

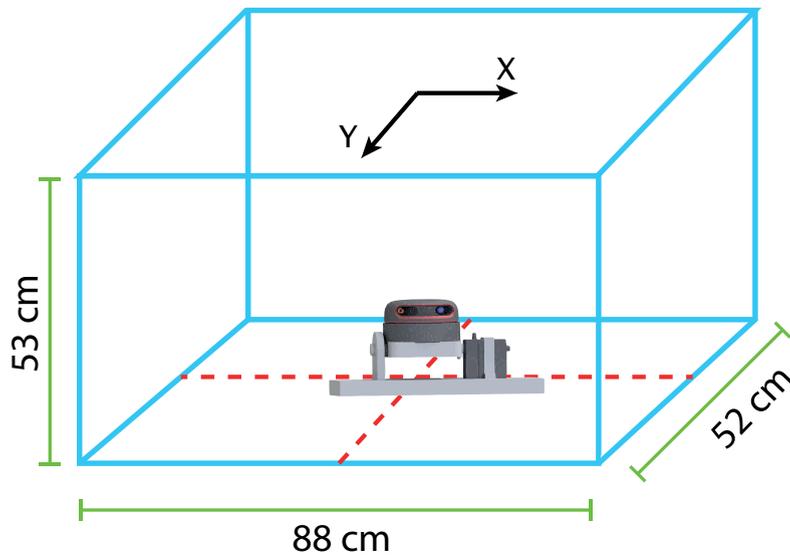


FIGURA 4.8: Dimensiones de la caja donde se realizó las mediciones de forma interna, para generar la construcción en tres dimensiones.

obstáculos dentro del entorno y el espacio que falta explorar y fueron descritos en la sección 3.3. Este mapa ayuda al robot a tener una información precisa del lugar donde se encuentra navegando. El robot pudo generar su propia trayectoria con la información brindada por el algoritmo SLAM haciendo que pueda llegar a la posición deseada mientras evitaba los obstáculos en su camino.

#### 4.5. Resultados del mapeo en tres dimensiones con el sistema mecánico

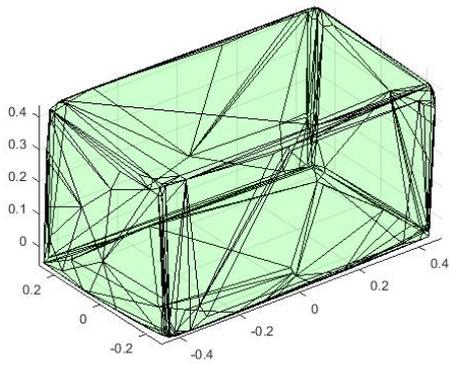
En esta sección se muestra las pruebas que fueron realizadas para la construcción tridimensional de un ambiente real. Se utilizó el sistema mecánico que fue construido (ver Figura 3.5c) y explicado en la sección 3.1.3. Las pruebas realizadas fueron únicamente con el sistema mecánico de forma estática y no montado al robot diferencial. Estas pruebas sirvieron para validar el algoritmo de construcción en tres dimensiones.

#### 4.5.1. Construcción tridimensional de una caja

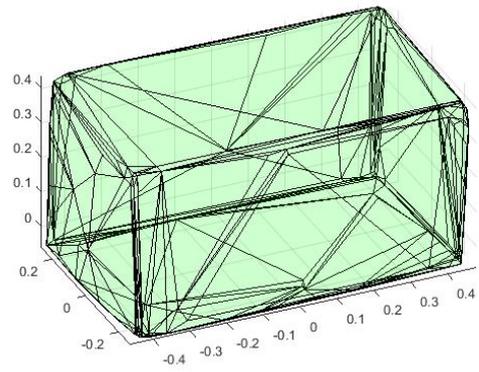
En esta prueba se utilizó una caja real con las dimensiones mostradas en la Figura 4.8. Esta prueba se realizó con la finalidad de probar el algoritmo de construcción tridimensional, ya que se tiene un espacio vacío por dentro y se conoce la forma del objeto. Se eligió una caja ya que se puede obtener sus dimensiones utilizando una simple cinta métrica y además es bastante factible poder estimar el punto medio de la caja, para colocar el sistema mecánico. Esta prueba es importante ya que nos brinda una perspectiva del funcionamiento del algoritmo y asimismo permite obtener el error que existe en las mediciones de las caras laterales de la caja. Para poder realizar esta prueba se tuvo que colocar el sistema mecánico dentro de la caja, haciendo que el sensor lidar se posicione en el centro de la caja como se muestra en la Figura 4.8. Además en esta figura se muestra un sistema de referencia por encima del sensor lidar, el cual hace referencia a la posición ( $x = 0m, y = 0m$ ) de la caja. Este sistema de referencia ayuda a conocer las posiciones de las paredes laterales de esta. Una vez colocado el sistema mecánico dentro de la caja, se tomó mediciones durante 40 segundos. El sistema mecánico es controlado por medio de un Raspberry Pi3, donde el microcontrolador recibe la información de los componentes del sistema mecánico y envía estos datos hacia una computadora de forma remota. Los datos recibidos son almacenados para el procesamiento posterior.

En la sección 3.4 se explicó de forma detallada los pasos que se debe seguir para la construcción de un sólido tridimensional dentro de un ambiente real, usando el sensor lidar y el sistema mecánico. Las mediciones realizadas fueron dibujadas usando *Python*, y se obtuvo el sólido mostrado en la Figura 4.9. La figura muestra un prisma rectangular, donde las caras del sólido son planas y las líneas de las aristas son continuas. Asimismo, se puede observar que existen líneas que se entrelazan entre algunas aristas del sólido, debido al algoritmo usado para graficar la nube de puntos de un sólido geométrico.

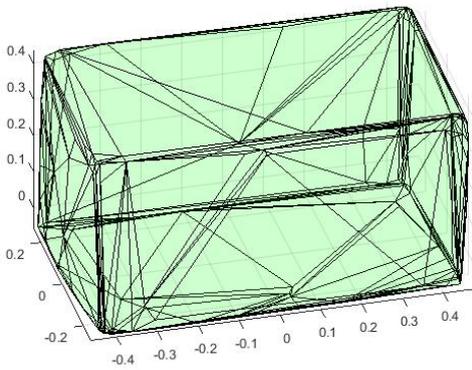
El diseño mecánico del sistema permite al sensor lidar realizar mediciones en cada



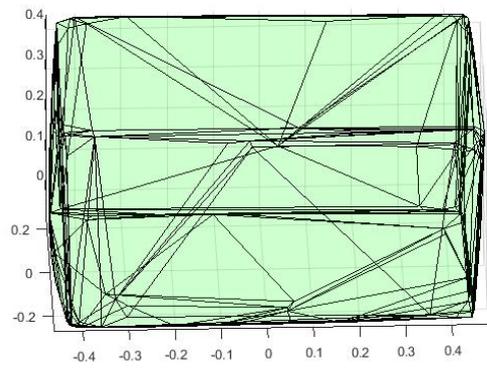
(a)



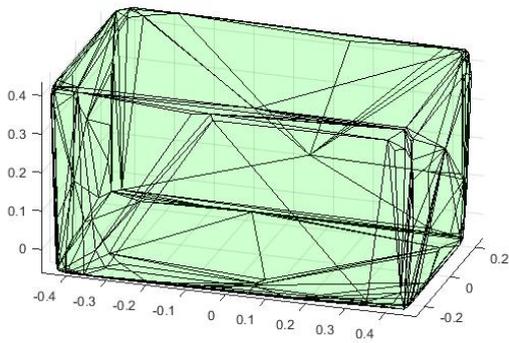
(b)



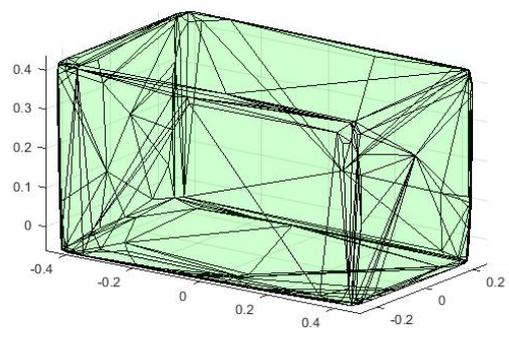
(c)



(d)



(e)



(f)

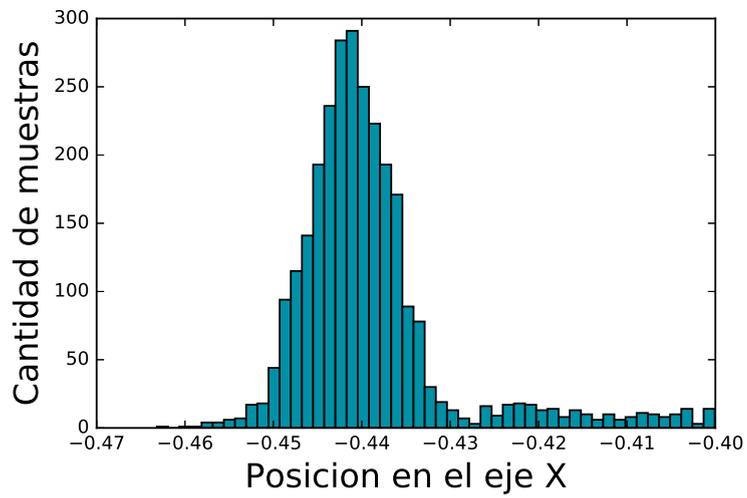
FIGURA 4.9: Construcción del sólido geométrico, a partir de las mediciones del sensor LIDAR. Desde las figuras (a) hasta (f) se nota el sólido tridimensional en torno a su eje en sentido horario para una mejor visualización.

rincón de la caja, por tal motivo se muestra pocas imperfecciones en el sólido generado y a simple vista tiene las mismas dimensiones que la caja real. Para estimar el error de nuestro sólido geométrico generado, se colocó el sistema al centro y dentro de la caja. Esa posición se considera como la posición ( $x = 0 \text{ m}, y = 0 \text{ m}$ ), y así poder corroborar que tan cerca se encuentran las paredes de la caja virtual a la posición de las paredes de la caja real.

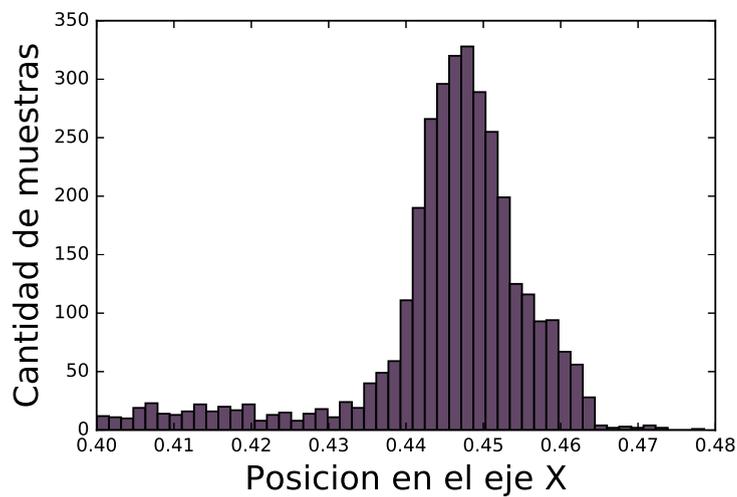
Teniendo el sensor lidar colocado en el centro de la caja real, las posiciones de cada pared son las siguientes: en el eje  $X$  las posiciones son  $X = -0.44 \text{ m}$  y  $X = 0.44 \text{ m}$ ; en el eje  $Y$  las posiciones son  $Y = -0.255 \text{ m}$  y  $Y = 0.255 \text{ m}$ . Se obtuvo las posiciones de cada pared utilizando las medidas de longitud de cada pared. En la Figura 4.10 se muestran los histogramas de la cantidad de datos con respecto a las posiciones en el eje  $X$ .

Primero se analiza los resultados de la región izquierda del eje  $X$ , donde la posición de la pared de la caja real es  $X = -0.44 \text{ m}$ . En la Figura 4.10a se observa que la mayor cantidad de muestras se acumulan en la posición correcta de la pared real. Además se tiene un promedio de  $\bar{x} = -0.439 \text{ m}$ , el cual indica que la mayor cantidad de mediciones se encuentran acumuladas dentro de esa posición. La desviación estándar es de  $\sigma = 0.008 \text{ m}$ , eso muestra una dispersión pequeña de las mediciones con respecto al promedio de éstas. Asimismo, estos datos tienen un error absoluto de  $E_{absoluto} = 0.184 \text{ m}$  y un error relativo de  $E_{relativo} = 0.215 \%$ . El valor de los errores relativos es bastante pequeño y aceptable para la aplicación propuesta.

Los resultados de la región derecha del eje  $X$ , donde la posición de la pared de la caja real es  $X = 0.44 \text{ m}$ , que fueron obtenidos serán analizados a continuación. En la Figura 4.10b se observa el histograma de esta pared, el cual tiene un promedio de  $\bar{x} = 0.445 \text{ m}$  y una desviación estándar de  $\sigma = 0.011 \text{ m}$ . A partir de estos resultados se puede inferir que la concentración de mediciones tiene una ligera desviación con respecto a la posición de la pared de la caja real. Asimismo, el error absoluto es  $E_{absoluto} = 0.190 \text{ m}$



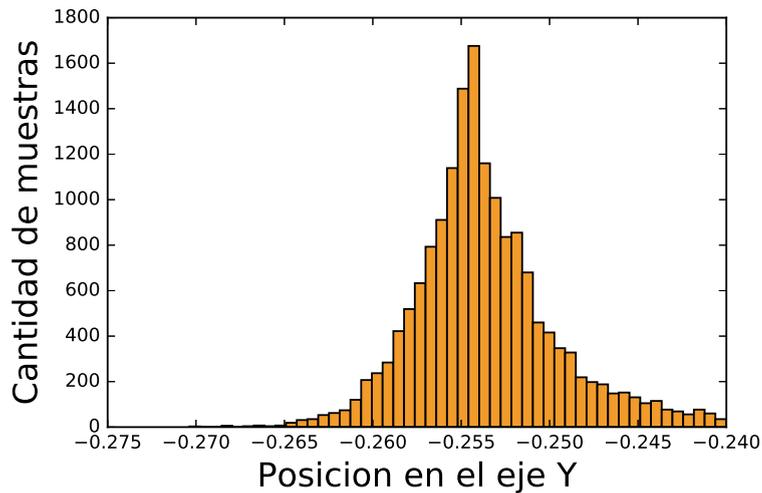
(a)



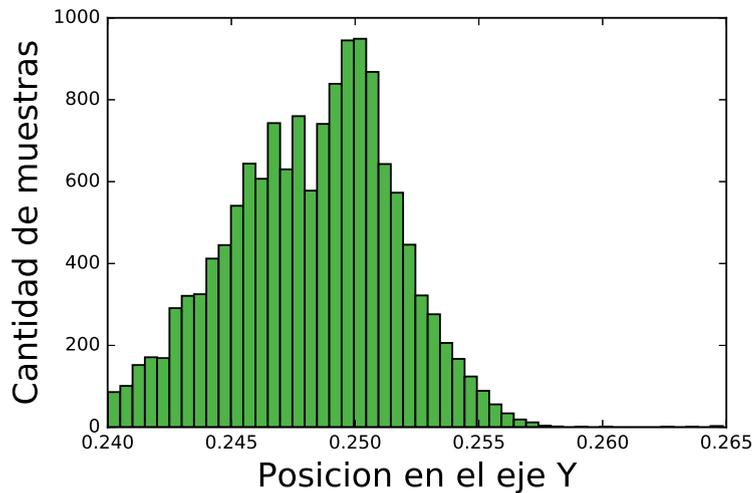
(b)

FIGURA 4.10: Histogramas de la cantidad de muestras con respecto a la posición en el eje  $X$  del sólido geométrico. En (a) se muestra el histograma de la pared izquierda del sólido con una posición de  $X = -0.44$  m y en (b) se muestra el histograma de la pared derecha con una posición de  $X = 0.44$  m.

y el error relativo es  $E_{relativo} = 1.221\%$ . En este caso el valor de los errores aumentó un poco comparado con los valores anteriores, pero los valores son aún bastante pequeños.



(a)



(b)

FIGURA 4.11: Distribución de las mediciones con respecto a la posición ( $y$ ) de las paredes frontal y posterior de la caja real.

Los resultados para la sección izquierda del eje  $Y$ , donde la pared de la caja real tiene como posición  $Y = -0.255\text{ m}$ , son analizados a continuación. En la Figura 4.11a se muestra el histograma para esta cara de la caja a lo largo del eje  $Y$ , el cual tiene un promedio de  $\bar{y} = -0.253\text{ m}$  y una desviación estándar de  $\sigma = 0.003\text{ m}$ . Además el

error absoluto es de  $E_{absoluto} = 0.001 \text{ m}$  y el error relativo es de  $E_{relativo} = 0.52 \%$ . Esta pared del sólido geométrico se encuentra bastante cerca a la posición real y tiene un error bastante pequeño. Para la cara derecha de la caja en el eje  $Y$ , posicionada en  $Y = 0.255 \text{ m}$  el histograma obtenido se muestra en la Figura 4.11b, donde se tiene un promedio de  $\bar{y} = 0.248 \text{ m}$ , una desviación estándar de  $\sigma = 0.003 \text{ m}$ , un error absoluto de  $E_{absoluto} = 0.006 \text{ m}$  y un error relativo de  $E_{relativo} = 2.638 \%$ . Los valores obtenidos muestran que esta sección del sólido geométrico tiene un error bastante aceptable con respecto a las dimensiones de la caja, aún teniendo un mayor error relativo con respecto a los resultados de las paredes anteriores.

Los histogramas mostrados en la Figura 4.10 y Figura 4.11 tienden a tener la forma de una distribución normal, por su forma acampanada y tiende a ser simétricos con respecto a la media, que viene a ser la posición correcta de las paredes de la caja. La forma que tiene es debido al teorema de límite central, ya que el tamaño de sus muestras es lo suficientemente grande y esto hace que la distribución de las medias sigan aproximadamente a una distribución normal.

#### **4.5.2. Pruebas del mapa tridimensional en el pasadizo dentro de un edificio**

Se colocó el sistema mecánico en el pasadizo de un edificio, para obtener un mapa tridimensional del ambiente. El sistema mecánico se colocó en la posición que se muestra en la Figura 4.12. Se trató de colocar todo el sistema, de forma estática, al centro del marco de color rojo que se observa en la figura. Además, en esta figura se puede ver las dimensiones que tiene esa zona a mapear. El experimento duró 70 segundos, donde los datos del sensor lidar y del motor fueron enviados de forma remota hacia la computadora por medio del Raspberry Pi3.

El mapa tridimensional es mostrado en la Figura 4.13. En esta figura se muestra el mismo mapa en diferentes rotaciones en sentido horario, donde se puede observar las



FIGURA 4.12: Pasadizo dentro de un edificio, donde se colocó el sistema mecánico para realizar el mapa tridimensional del lugar.

paredes y las columnas del pasadizo. La estructura del ambiente es simétrica, ya que en sus paredes y columnas existe una proporcionalidad a lo largo de todo el pasadizo. Esta prueba se realizó con un sistema mecánico anterior, el cual estaba compuesto por un sensor lidar y un servomotor. Este sistema permitía al sensor lidar tener un ángulo de apertura que iba de  $-15^\circ$  a  $15^\circ$ . En la Figura 4.13 se puede observar que no se muestra el techo del pasadizo, lo cual se debe al alcance en la toma de medidas en el eje  $Z$ . Si esta experiencia se hubiese realizado junto al robot móvil, el mapa en tres dimensiones del pasadizo se hubiese mostrado con más detalles ya que el desplazamiento del sistema mecánico hace que el sensor lidar tome mayores medidas y se tenga una mayor cantidad de información.

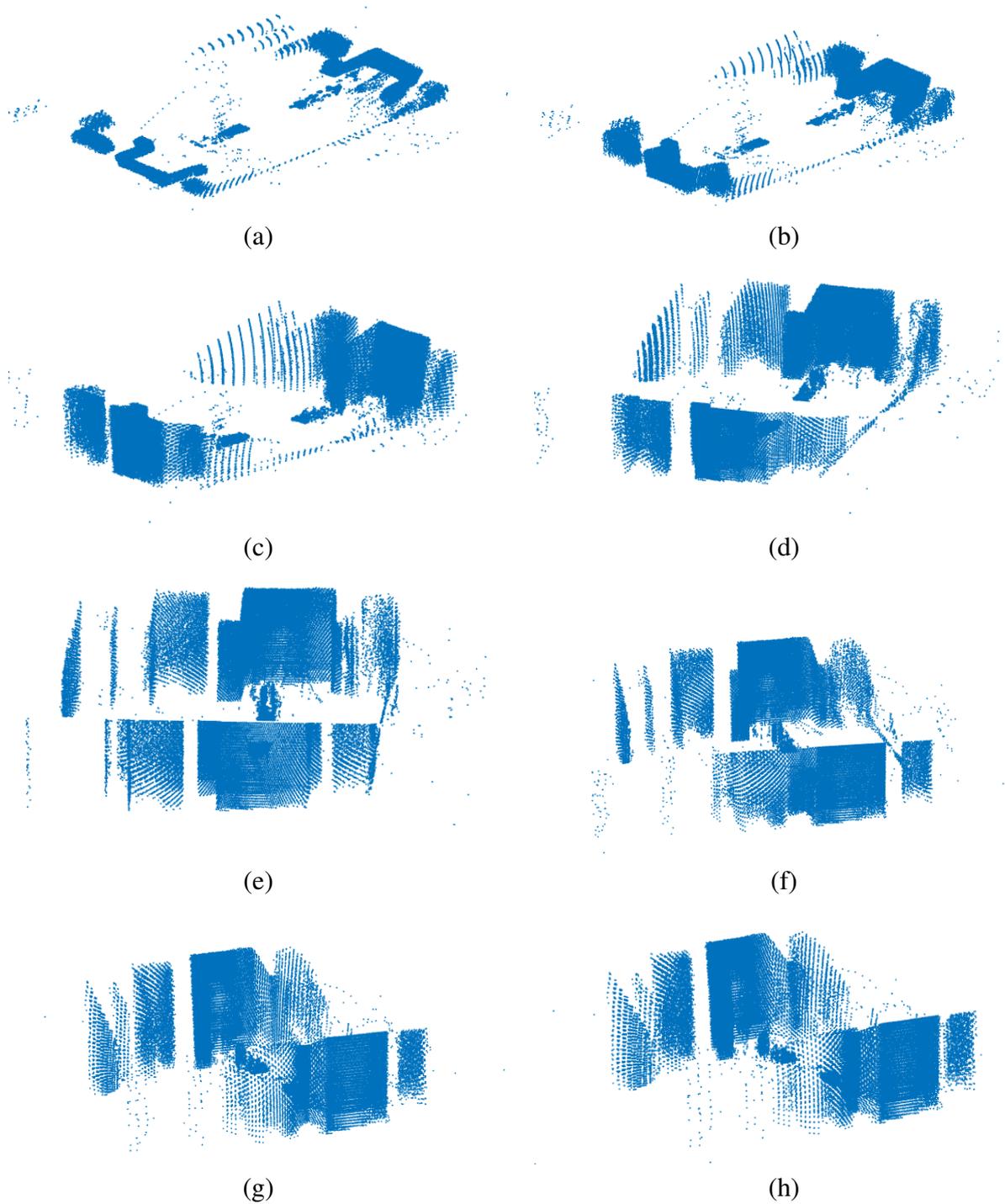


FIGURA 4.13: Mapa en tres dimensiones de un pasadizo. Desde las figuras (a) hasta (h) el mapa tridimensional rota en sentido horario con respecto a su eje vertical.

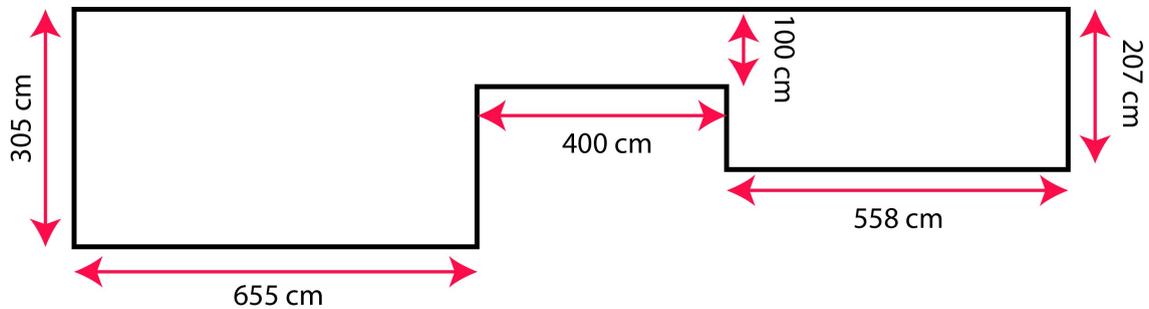


FIGURA 4.14: Dimensiones del sótano donde se hizo las pruebas de exploración con el robot *muqi*.

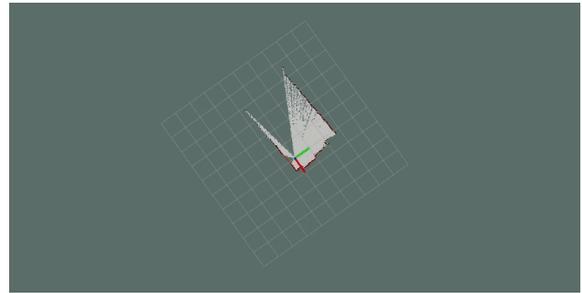
#### 4.6. Resultado del mapa 3D mientras el robot se mueve

En esta sección se explicará los resultados que fueron obtenidos en la exploración del robot *muqi*, como un sistema total. Estas pruebas fueron realizadas en el sótano de la Universidad de Ingeniería y Tecnología (UTEC). El sótano está compuesto por dos ambientes de diferentes tamaños unido por un pasadizo, como se muestra en la Figura 4.14. El reto de *muqi* es poder explorar un ambiente y luego ingresar por el pasadizo hacia el segundo ambiente; una vez que llegue a una posición donde no pueda seguir explorando, el robot regresa a la posición donde comenzó.

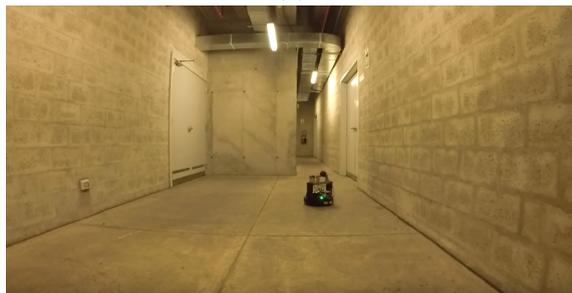
Los ambientes que *muqi* comenzó a explorar se muestran en la Figura 4.15. El robot comienza a explorar en la Figura 4.15a; una vez que empieza a tomar mediciones con el sensor lidar, éste empieza a generar el mapa en dos dimensiones. Con la información dada por el algoritmo SLAM, *muqi* comienza a generar posiciones aleatorias dentro del ambiente y así comienza a explorar. Como se muestra en la Figura 4.15b. En la Figura 4.15c se muestra el punto exacto cuando *muqi* llega a la entrada del pasadizo para explorar y llegar hacia el otro ambiente. En la Figura 4.15d se muestra el instante en que el robot comienza a navegar dentro del pasadizo explorando el ambiente. En la Figura 4.15e se puede ver cómo el robot ingresó hacia el otro ambiente y sigue explorando y mapeando en dos y tres dimensiones al mismo tiempo. El movimiento del robot va de izquierda a derecha en movimientos de zigzag. Cuando encuentra un obstáculo por delante, *muqi* se



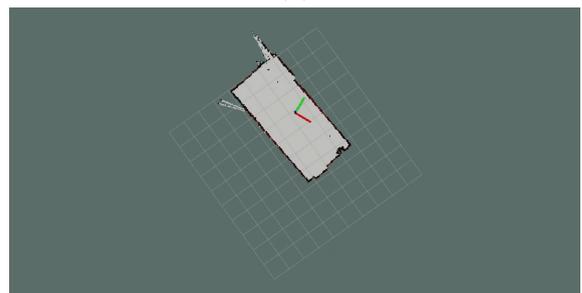
(a)



(b)



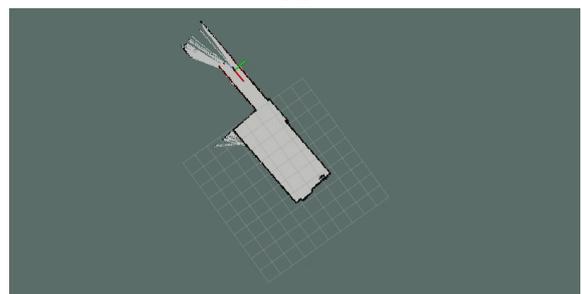
(c)



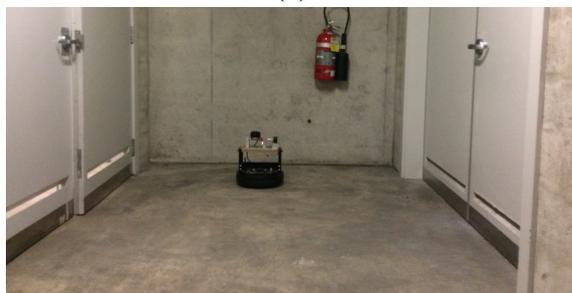
(d)



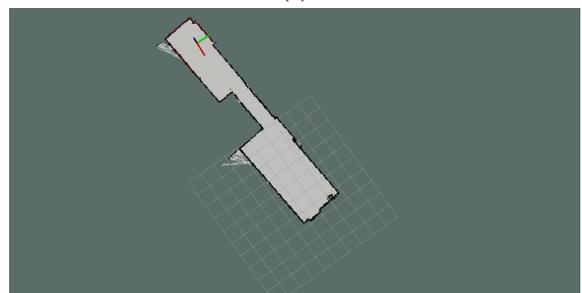
(e)



(f)



(g)



(h)

FIGURA 4.15: Exploración autónoma de *muqi* dentro de un ambiente cerrado. En (a) se muestra cuando *muqi* comienza a tomar mediciones, en (b) se observa cuando *muqi* comienza a explorar, en (c) el robot ingresa por el pasadizo a seguir explorando, en (d) *muqi* empieza a explorar el otro ambiente. En (e) se muestra que el robot esta moviéndose por el nuevo ambiente y en (f) se muestra que *muqi* ya no puede avanzar debido a las paredes y regresa a su posición inicial.

detiene gira y comienza a explorar hacia el otro extremo. En la Figura 4.15f se puede observar que *muqi* ya no puede seguir explorando debido a que en su camino se encuentran las tres paredes que impiden la exploración, en este instante el robot comienza a regresar a su posición inicial ( $x = 0m, y = 0m$ ).

El robot *muqi* fue controlado por medio del Raspberry Pi3, el cual recibe órdenes de una computadora vía wifi. El envío de la información fue en tiempo real de forma remota. El robot debe realizar la construcción de los mapas en dos dimensiones y tres dimensiones del ambiente, para hacer esto *muqi* debe controlar los ángulos del motor paso a paso. Cuando el motor paso a paso marca el ángulo  $0^\circ$ , el sensor lidar se encuentra de forma horizontal (ver Figura 3.7a) y en esa posición envía las mediciones dentro del tópico *scan* hacia el algoritmo de *gmapping* (SLAM). Este envío de información es en tiempo real, ya que el robot necesita conocer el ambiente de forma detallada y de la forma más precisa posible. Los datos de las mediciones para la construcción del sólido tridimensional no, son enviados en tiempo real debido a la cantidad de mediciones que se esta tomando por cada vez que gira el sensor lidar. Estos datos son almacenados en la memoria micro SD del microcontrolador Raspberry Pi3, para luego procesarlos y hacerlos visible.

La exploración de *muqi* duró 5 minutos, donde se desplazó por todo el sótano a diferentes velocidades. Las velocidades del robot fue cambiante de acuerdo a las situaciones que el robot encontraba; es decir, si el robot navega por un espacio sin obstáculos, se comienza a mover a una velocidad de  $0.3m/s$  y cuando el robot se encuentra con un obstáculo, comienza a disminuir su velocidad a  $0.05m/s$ . Para obtener las velocidades lineal y angular se tuvo que multiplicar una constante proporcional a cada velocidad para así disminuir sus valores. Mientras el robot navega a bajas velocidades el sistema mecánico puede realizar una mayor cantidad de mediciones del ambiente, obteniendo una mejor resolución del sólido geométrico en tres dimensiones. El robot *muqi*, para que pueda explorar el ambiente, utiliza la información del mapa que genera el algoritmo SLAM. El mapa bidimensional que se obtiene es mostrado en la Figura 4.16. La construcción del

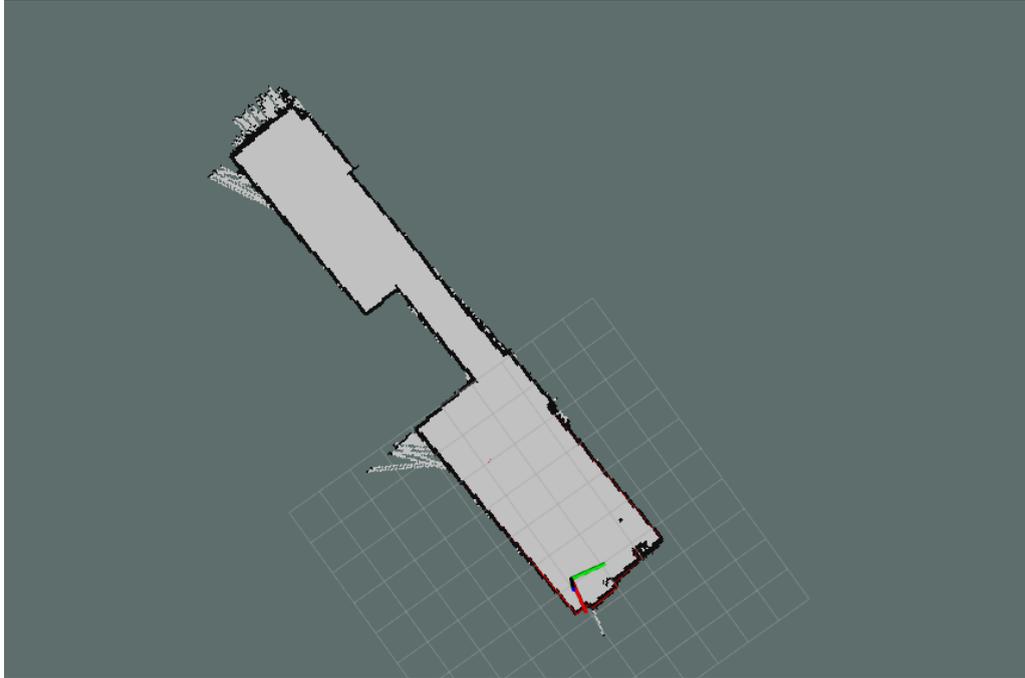


FIGURA 4.16: Mapa generado por el algoritmo SLAM. Este mapa se construye mientras el *muqi* va explorando el sótano.

mapa del SLAM es en tiempo real y esto se puede visualizar usando la herramienta Rviz. El mapa tiene la forma y estructura del ambiente mostrado en la Figura 4.14. Además, se observa en el mapa dos ambientes unidos por un pasadizo donde las líneas negras representa las paredes del lugar y la zona gris representa el área explorada por el robot *muqi*. Además, se puede ver que existe zonas desconocidas (color verde oscuro) pero que no pueden ser exploradas por los obstáculos que impiden el paso.

Los resultados del sólido tridimensional obtenido luego de la exploración de *muqi*, se muestra en la Figura 4.18, donde se observa dos ambientes unidos por medio de un pasadizo. Además, se puede ver que las paredes son totalmente planas sin ninguna deformación. Se puede apreciar las puertas que tiene el ambiente y además el techo del sótano. El techo no es uniforme debido a que en ese lugar existen unas vigas transversales, canaletas de cables y tuberías de agua. La resolución de este sólido geométrico se logró disminuyendo la velocidad del sensor lidar de 600 RPM a 200 RPM y la velocidad del robot kobuki, ya que con esta reducción se pudo reducir la cantidad de vibración de todo el

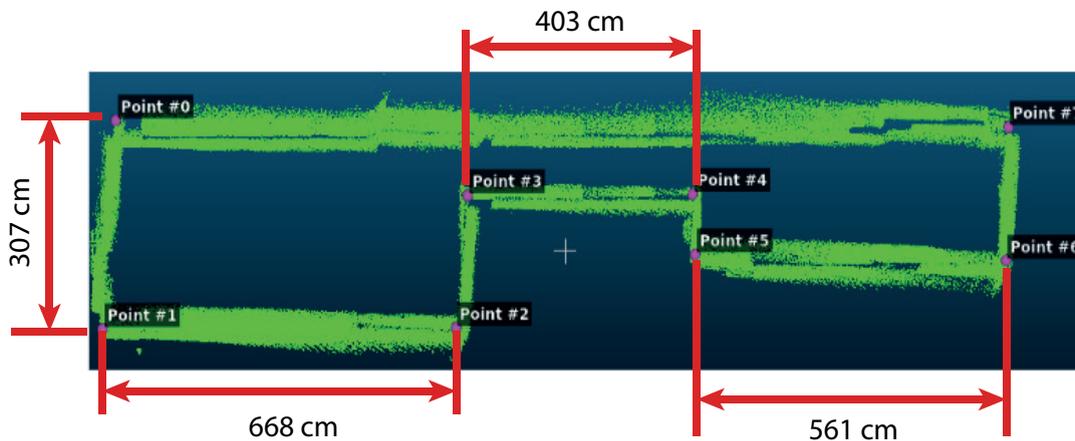
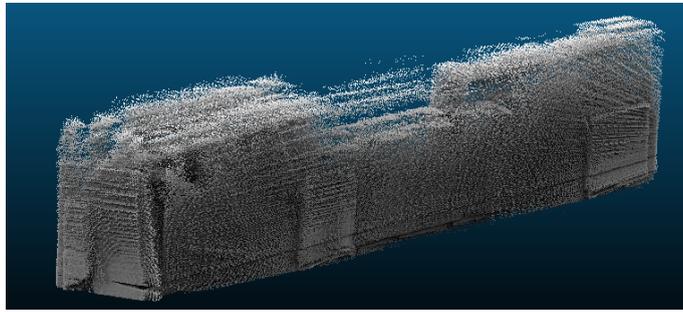


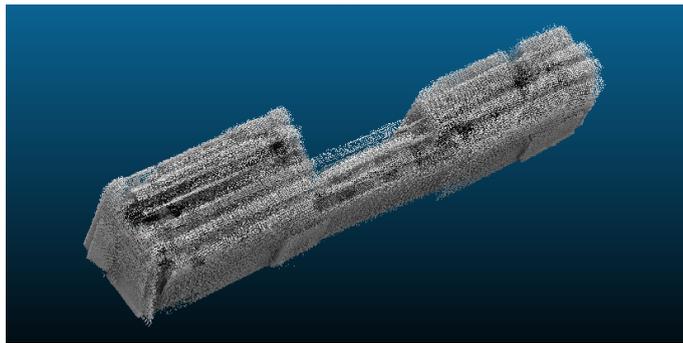
FIGURA 4.17: Resultado de las mediciones obtenidas, en el sótano, por el sensor lidar en la exploración del robot *muqi*.

sistema. Los datos obtenidos por el sistema de sensado son procesados en *CloudCompare*. *CloudCompare* es un software de código abierto, que permite realizar procesamiento a la nube de puntos utilizando diferentes tipos de algoritmos.

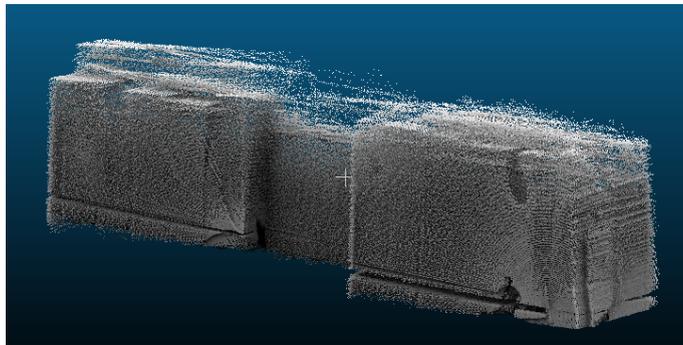
Para analizar el error entre las dimensiones reales del sótano con respecto al mapa obtenido, se extrajo un plano en dos dimensiones del sólido geométrico y se midió sus lados. Esto se muestra en la Figura 4.17. En la figura se puede observar que tiene la misma forma de la Figura 4.14, este mapa tiene un error absoluto de  $5\text{cm}$  y un error relativo de  $0.455\%$ . Estos errores son bastante pequeños y se originan debido a las vibraciones que genera el sensor lidar al momento de rotar, además de las vibraciones que genera el motor paso a paso y también las vibraciones que origina el robot Kobuki. La suma de todas estas vibraciones, hace que el sensor lidar tenga una mayor dispersión de puntos en sus medidas con respecto a una posición dentro de un ambiente.



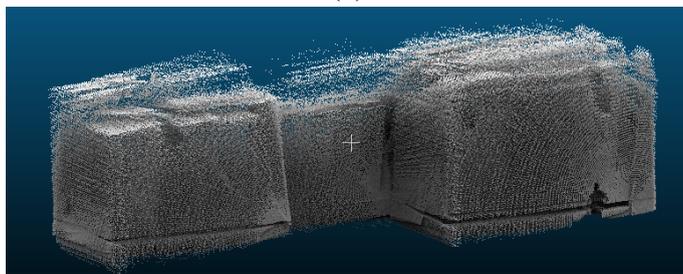
(a)



(b)



(c)



(d)

FIGURA 4.18: Mapa en tres dimensiones del ambiente explorado con el robot *muqi*. En esta imagen se muestra diferentes vistas del sólido geométrico, en (a) se muestra la vista posterior; en (b) la vista superior; en (c) la vista frontal derecha y en (d) la vista frontal izquierda.

## CONCLUSIONES

El presente trabajo de tesis es esfuerzo inicial en la solución al problema planteado el cual presenta un algoritmo para el movimiento autónomo de un robot móvil basado en campos potenciales y en un control polar realimentado. Esto se aplicó para la navegación de un robot móvil de accionamiento diferencial. El controlador de movimiento se basa en coordenadas polares. Los resultados obtenidos para el controlador de movimiento demuestran que es una buena solución para las restricciones no holonómicas que presenta el robot. De la misma forma, el rendimiento del robot en simulación muestra su rápida convergencia a través de trayectorias suaves. Las pruebas en un robot Kobuki con accionamiento diferencial muestran resultados bastante exitosos. También demuestran la capacidad del robot de movimiento autónomo a pesar del entorno que es desconocido. El lidar proporciona la información del entorno para que el robot pueda decidir su movimiento hacia la meta evitando obstáculos en tiempo real.

Se realizó un sistema mecánico para la construcción de un mapa en tres dimensiones basado en el sensor lidar, un motor paso a paso y la odometría del robot móvil. Esto se aplicó para el mapeo tridimensional en una caja, dentro de un sótano y el pasadizo dentro de un edificio. Se demuestra que la mayor concentración de mediciones se encuentran en las posiciones reales de las paredes dentro del ambiente. El sistema mecánico puede generar un mapa en tres dimensiones y dos dimensiones al mismo tiempo. El mapa en dos dimensiones es generado por el algoritmo SLAM. Con este mapa se demuestra que el robot móvil tiene la información suficiente para explorar dentro de un ambiente desconocido. Se demuestra que el sistema mecánico construido genera mapas tridimensionales con errores menores a los 5 cm.

## RECOMENDACIONES

Los nuevos métodos de generación de trayectoria pueden ser utilizados para obtener un rango amplio de aplicaciones como rescate de personas, inspección de entornos, delivery, etc. Además, las pruebas pueden ser reemplazadas con ambientes de mayor dificultad de acceso para la exploración del robot móvil. Se puede emplear un sensor lidar con mayor cantidad de mediciones por vuelta, ya que esto va a permitir tener una mejor resolución y precisión en la nube de puntos obtenida, beneficiando la exploración autónoma y construcción del mapa tridimensional.

El robot móvil diferencial, utilizado para las pruebas experimentales, no tiene las características físicas adecuadas para la exploración de socavones subterráneos dentro de las minas subterráneas. El sistema puede ser implementado en otro tipo de robots móviles (terrestre o aéreo). El robot móvil terrestre necesita tener las características adecuadas para moverse en superficies bastante accidentadas y húmedas como son los túneles dentro de la mina subterránea. Para un robot móvil aéreo (drone) se necesita que el robot pueda desplazarse a una altura constante y tenga un diseño mecánico adecuado para la humedad del entorno.

## REFERENCIAS BIBLIOGRÁFICAS

- [1] E. de la Cruz and A. Ángel, “Métodos de explotación en la mediana minería del Perú,” *Rev.Instituto de Investigación (RIIGEO)*, pp. 125–134, Junio 2012.
- [2] E. Muñoz Del Pino, “Riesgos en la minería subterránea,” *Gobierno de Chile Servicio Nacional de Geología y Minería*, Abril 2012.
- [3] (2018, Noviembre) Etapas del proceso: Construcción de túneles mineros. [En línea]. Disponible en: <http://www.construccionminera.cl/etapas-del-proceso-construccion-de-tuneles-mineros>
- [4] (2018, Noviembre) ¿cómo se generan los gases tóxicos en minería? [En línea]. Disponible en: <http://www.revistaseguridadminera.com/operaciones-mineras/como-se-generan-los-gases-toxicos/>
- [5] (2018, Noviembre) Medidas de seguridad en perforación subterránea y superficial. [En línea]. Disponible en: <http://www.revistaseguridadminera.com/operaciones-mineras/medidas-de-seguridad-en-la-perforacion-minera-subterranea-y-superficial/>
- [6] C. Carmona, “Ingenieros del cobre y minería,” *Industria Minera*, pp. 81–84, Julio 2014.
- [7] S. Scheduling, G. Dissanayake, E. M. Nebot, and H. Durrant-Whyte, “An experiment in autonomous navigation of an underground mining vehicle,” *IEEE Transactions on robotics and Automation*, vol. 15, no. 1, pp. 85–95, 1999.

- [8] S. Thrun, S. Thayer, W. Whittaker, C. Baker, W. Burgard, D. Ferguson, D. Hahnel, D. Montemerlo, A. Morris, Z. Omohundro *et al.*, “Autonomous exploration and mapping of abandoned mines,” *IEEE Robotics & Automation Magazine*, vol. 11, no. 4, pp. 79–91, 2004.
- [9] S. Grehl, M. Donner, M. Ferber, A. Dietze, H. Mischo, and B. Jung, “Mining-rox–mobile robots in underground mining,” in *Proceedings Third International Future Mining Conference*, 2015, pp. 57–64.
- [10] R. Zlot and M. Bosse, “Efficient large-scale three-dimensional mobile mapping for underground mines,” *Journal of Field Robotics*, vol. 31, no. 5, pp. 758–779, 2014.
- [11] A. Mauricio, A. Nieves, Y. Castillo, K. Hilasaca, C. Fonseca, J. Gallardo, R. Rodríguez, and G. Rodríguez, “Multi-robot exploration and mapping strategy in underground mines by behavior control,” in *Multibody Mechatronic Systems*. Springer, 2015, pp. 101–110.
- [12] C. R. Verschoor and A. Visser, “Integrating disparity and edge detection algorithms to autonomously follow linear-shaped structures at low altitude,” in *AI & Robotics and 5th RoboCup Iran Open International Symposium (RIOS), 2013 3rd Joint Conference of*. IEEE, 2013, pp. 1–7.
- [13] J. Škoda and R. Barták, “Camera-based localization and stabilization of a flying drone,” 2015.
- [14] F. Heukels, “Simultaneous localization and mapping (slam): towards an autonomous search and rescue aiding drone,” Master’s thesis, University of Twente, 2015.
- [15] C. Solutions. (2017, Octubre) Minefly. [En línea]. Disponible en: <http://www.clickmox.com/minefly-drone.html>
- [16] A. Robots. (2018, Octubre) 3wd 48mm omni wheel robot kit. [En línea]. Disponible en: <https://www.active-robots.com/3wd-48mm-omni-wheel-robot-kit.html>

- [17] ECV. (2018, Octubre) Videovigilancia móvil. [En línea]. Disponible en: <http://videovigilancia.website/videovigilancia-movil/>
- [18] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to autonomous mobile robots*. MIT press, 2011.
- [19] D. Chwa, “Sliding-mode tracking control of nonholonomic wheeled mobile robots in polar coordinates,” *IEEE transactions on control systems technology*, vol. 12, no. 4, pp. 637–644, 2004.
- [20] M. J. Matarić, *The robotics primer*. Mit Press, 2007.
- [21] O. Khatib, “Real-time obstacle avoidance for manipulators and mobile robots,” in *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, vol. 2. IEEE, 1985, pp. 500–505.
- [22] Y. Koren and J. Borenstein, “Potential field methods and their inherent limitations for mobile robot navigation,” in *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*. IEEE, 1991, pp. 1398–1404.
- [23] H. M. Choset, S. Hutchinson, K. M. Lynch, G. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun, *Principles of robot motion: theory, algorithms, and implementation*. MIT press, 2005.
- [24] ———, *Principles of robot motion: theory, algorithms, and implementation*. MIT press, 2005.
- [25] M. Elbanhawi and M. Simic, “Sampling-based robot motion planning: A review,” *Ieee access*, vol. 2, pp. 56–77, 2014.
- [26] L. J. Guibas, C. Holleman, and L. E. Kavraki, “A probabilistic roadmap planner for flexible objects with a workspace medial-axis-based sampling approach,” in *Intelligent Robots and Systems, 1999. IROS’99. Proceedings. 1999 IEEE/RSJ International Conference on*, vol. 1. IEEE, 1999, pp. 254–259.

- [27] A. Bry and N. Roy, “Rapidly-exploring random belief trees for motion planning under uncertainty,” in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 723–730.
- [28] S. Thrun, “Learning occupancy grid maps with forward sensor models,” *Autonomous robots*, vol. 15, no. 2, pp. 111–127, 2003.
- [29] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, “Monte carlo localization for mobile robots,” in *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, vol. 2. IEEE, 1999, pp. 1322–1328.
- [30] L. Jetto, S. Longhi, and G. Venturini, “Development and experimental validation of an adaptive extended kalman filter for the localization of mobile robots,” *IEEE Transactions on Robotics and Automation*, vol. 15, no. 2, pp. 219–229, 1999.
- [31] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, “A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking,” *IEEE Transactions on signal processing*, vol. 50, no. 2, pp. 174–188, 2002.
- [32] I. A. Calle Flores, “Navegación autónoma de un robot móvil usando técnicas probabilísticas de localización y mapeo basadas en métodos monte carlo secuenciales,” 2014.
- [33] S. Thrun and Y. Liu, “Multi-robot slam with sparse extended information filers,” in *Robotics Research. The Eleventh International Symposium*. Springer, 2005, pp. 254–266.
- [34] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. MIT press, 2005.
- [35] D. Hahnel, W. Burgard, D. Fox, and S. Thrun, “An efficient fastslam algorithm for generating maps of large-scale cyclic environments from raw laser range measurements,” in *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, vol. 1. IEEE, 2003, pp. 206–211.

- [36] K. P. Murphy, “Bayesian map learning in dynamic environments,” in *Advances in Neural Information Processing Systems*, 2000, pp. 1015–1021.
- [37] M. Montemerlo and S. Thrun, *FastSLAM: A scalable method for the simultaneous localization and mapping problem in robotics*. Springer, 2007, vol. 27.
- [38] G. Grisetti, R. Kummerle, C. Stachniss, and W. Burgard, “A tutorial on graph-based slam,” *IEEE Intelligent Transportation Systems Magazine*, vol. 2, no. 4, pp. 31–43, 2010.
- [39] (2018, Junio) About kobuki. [En línea]. Disponible en: <http://kobuki.yujinrobot.com/about2/>
- [40] friv2hub. (2018, Octubre) Rplidar datasheet best of build kernel and modules – nvidia jetson tx2. [En línea]. Disponible en: <http://www.friv2hub.com/rplidar-datasheet>
- [41] M.-C. Amann, T. Bosch, M. Lescure, R. Myllyla, and M. Rioux, “Laser ranging: a critical review of usual techniques for distance measurement,” *Optical engineering*, vol. 40, no. 1, pp. 10–19, 2001.
- [42] SLAMTEC, *Introduction and Datasheet*, RPLIDAR A2. [En línea]. Disponible en: <http://bit.ly/2h0CYIK>
- [43] K. Nummiaro, E. Koller-Meier, and L. Van Gool, “An adaptive color-based particle filter,” *Image and vision computing*, vol. 21, no. 1, pp. 99–110, 2003.
- [44] G. Grisetti, C. Stachniss, and W. Burgard, “Improved techniques for grid mapping with rao-blackwellized particle filters,” *IEEE transactions on Robotics*, vol. 23, no. 1, pp. 34–46, 2007.