ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

MAPEO Y LOCALIZACIÓN SIMULTÁNEOS (SLAM) EN 3D MEDIANTE UN SENSOR LÁSER

PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN ELECTRÓNICA Y CONTROL

CISNEROS PINTO NELSON EDUARDO
nelsonneocrom@gmail.com

PERUGACHI FLORES GABRIEL ALEXANDER
alex.perugachi@gmail.com

DIRECTOR: DR. ANDRÉS ROSALES ACOSTA
andres.rosales@epn.edu.ec

Quito, Noviembre 2013

DECLARACIÓN

Nosotros, Nelson Eduardo Cisneros Pinto y Gabriel Alexander Perugachi Flores, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentado para ningún grado o calificación personal; y que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedemos nuestros derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

Nelson Eduardo Cisneros Pinto 171823685-2	Gabriel Alexander Perugachi Flores 171897603-6

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Nelson Eduardo Cisneros Pinto y Gabriel Alexander Perugachi Flores bajo mi supervisión.

_____

Dr. Andrés Rosales Acosta
DIRECTOR DEL PROYECTO

AGRADECIMIENTOS

A Dios, por concederme la salud y vida en todo momento para seguir en este largo camino.

A mis padres Gabriel y María de los Ángeles, por sus enseñanzas, sacrificios y dedicación para guiarme en los momentos buenos y malos de la vida

A mi hermana Wendy, que con su entusiasmo y paciencia es un pilar fundamental en mi vida.

A mi amigo y compañero de tesis Nelson que con su incondicional amistad y carisma supimos sacar a delante este proyecto con mucho esfuerzo

A mi novia Anita, que con su amor y amistad supo ayudarme en los fracasos y éxitos de este proyecto

A mis grandes e incondicionales amigos Daniel, Ximena y Dario.

A Ringo & Dingo que además de ser un equipo de fútbol es un grupo de amigos.

A el grupo SLAM que mutuamente nos colaboramos en nuestros proyectos

Un agradecimiento especial al Dr. Andrés Rosales que sin su valiosa ayuda no hubiese sido posible culminar con éxito este proyecto.

Por último un agradecimiento a todas las personas que nos colaboraron de una u otra forma en la realización de este trabajo.

ALEXANDER

AGRADECIMIENTOS

Quiero agradecer a mis padres a mi abuelito y a mi hermana por siempre acompañarme y ser un soporte fundamental durante toda mi vida, a mi tía que desde la distancia siempre estuvo pendiente de mí, y a toda mi familia.

A mis amigos que con lo que pasamos excelentes momentos y son la razón por la que voy a extrañar siempre los días en la universidad.

A mi compañero de tesis Alex que en las ocasiones en que por diferentes circunstancias no pude estar presente continuó con el trabajo y que con sus ideas se pudo formar un buen equipo y cumplir con el objetivo.

A mis profesores que compartieron su conocimiento y fueron un ejemplo a seguir.

A los compañeros que fueron pioneros en el tema de SLAM y que sin su ayuda habría sido muy complicada la realización de nuestro proyecto.

Finalmente a nuestro director de tesis Dr. Andrés Rosales y al Ing. David Pozo que supieron guiarnos para sortear los diferentes inconvenientes presentados durante este trayecto.

NELSON

DEDICATORIA

Dedico este trabajo a mis Padres, a mi hermana, a mi novia y en especial a mi abuelito que me está viendo desde el cielo.

ALEXANDER

DEDICATORIA

Dedico este trabajo a mi familia, por quienes me esfuerzo cada día.

NELSON

CONTENIDO

Resumen
Presentación

CAPÍTULO 1 1

MARCO TEÓRICO 1

1.1 INTRODUCCIÓN 1
1.2 LOCALIZACION Y MAPEO SIMULTANEO (SLAM) 1
1.2.1 DEFINICIÓN 1
1.2.2 PROBLEMAS Y APROXIMACIONES PARA RESOLVER SLAM 2
1.2.3 PROCEDIMIENTO GENERAL DE SLAM 3
1.2.4 ELEMENTOS DE SLAM 3

1.3 TIPOS DE SLAM 5
1.3.1 FULL SLAM 5
1.3.2 ONLINE SLAM 8

1.4 APLICACIONES DE SLAM 10
1.4.1 FÚTBOL DE ROBOTS 11
1.4.2 VEHÍCHULOS SUBMARINOS 11
1.4.3 MAPAS 3D 12
1.4.4 ASOCIACIÓN DE DATOS 12
1.4.5 MINERÍA 13

1.5 VENTAJAS DE SLAM 13

1.6 INCONVENIENTES DE SLAM 14

1.7 LOCALIZACION Y MAPEO SIMULTANEO EN 3D (SLAM 3D) 14
1.7.1 INTRODUCCIÓN 14
1.7.2 TÉCNICA DE SLAM 3D 15

1.8 PLATAFORMA ROBOTICA MÓVIL 17

1.9 HARDWARE DE ROBOTINO® 18
1.9.1 CHASÍS 18
1.9.2 BATERÍAS 19
1.9.3 UNIDAD DE ACCIONAMIENTO 19

1.9.4 UNIDAD DE CONTROL 23
1.9.5 SENSORES 26

1.10 ACCESORIOS 28
1.10.1 NORTHSTAR 28
1.10.2 LÁSER 30

1.11 SOFTWARE DE ROBOTINO® 32
1.11.1 MANEJO DE ROBOTINO® CON MATLAB 33
1.11.2 MATLAB PARA ADQUISICIÓN DE DATOS DEL ENTORNO 34

CAPÍTULO 2 36

LOCALIZACIÓN 36
2.1 INTRODUCCIÓN 36

2.2 MÉTODOS DE LOCALIZACIÓN 36
2.2.1 LOCALIZACIÓN POR FILTRO DE KALMAN 37
2.2.2 LOCALIZACIÓN MEDIANTE CADENAS DE MARKOV 38
2.2.3 MÉTODOS ALTERNATIVOS 38

2.3 MODELO OMNIDIRECCIONAL 39
2.3.1 MOVILIDAD OMNIDIRECCIONAL 39

2.4 FILTRO DE KALMAN 42
2.4.1 CARACTERÍSTICAS DEL FILTRO DE KALMAN 42
2.4.2 FILTRO RECURSIVO 43
2.4.3 ENFOQUE BAYESIANO 43
2.4.4 RESTRICCIONES DEL FILTRO DE KALMAN 44
2.4.5 PRESUNCIONES BÁSICAS 44
2.4.6 ALGORITMO PREDICCIÓN MÁS CORRECCIÓN 44
2.4.7 SELECCIÓN DE PARÁMETROS: 45

2.5 FILTRO EXTENDIDO DE KALMAN (EKF) 45
2.5.1 DEFINICIONES 45
2.5.2 FORMULACIÓN ALTERNATIVA DEL EKF 47

2.6 IMPLEMENTACIÓN DEL MODELO OMNIDIRECCIONAL 51

2.7 IMPLEMENTACIÓN DEL FILTRO EXTENDIDO DE KALMAN 52

CAPÍTULO 3 54

MAPEO DEL ENTORNO 54

3.1 INTRODUCCIÓN 54

3.2 CONSTRUCCIÓN DE MAPAS 54

3.2.1 MAPAS ABSOLUTOS 57

3.2.2 MAPAS RELATIVOS 58

3.3 SISTEMA DE NAVEGACIÓN 60

3.4 FUNDAMENTOS PARA EL POSICIONAMIENTO DE CARACTERÍSTICAS DEL ENTORNO 61

3.4.1 BARRIDO VERTICAL 61

3.5 IMPLEMENTACIÓN DE LA DETECCIÓN MEDIANTE LÁSER 68

3.6 IMPLEMENTACION DE DETECCIÓN DE LANDMARKS(MARCAS) 70

3.7 IMPLEMENTACION DE LA ANIMACIÓN DEL ROBOT EN 3D 75

CAPÍTULO 4 77

EXPERIMENTACIÓN Y RESULTADOS 77

4.1 ANÁLISIS DE FUNCIONAMIENTO DE NORTHSTAR (NS2) 77

4.2 ANÁLISIS DE DATOS DE POSICIÓN 79

4.3 ANÁLISIS DE LAS PRUEBAS REALIZADAS 83

4.3.1 MAPA 1 83

4.3.2 MAPA 2 88

4.3.3 DETECCIÓN DE MARCAS (LANDMARKS) EN EL ENTORNO 93

4.4 ANÁLISIS DE RESULTADOS DE LOCALIZACION Y MAPEO SIMULTANEOS EN 3D (SLAM) 96

4.5 ANÁLISIS DE RESULTADOS DE DETECCIÓN DE MARCAS (LANDMARKS) EN EL ENTORNO 105

4.6 RELACIÓN CON TRABAJOS ANTERIORES 107

4.6.1 TRABAJOS FUTUROS 108

CAPITULO 5 109

CONCLUSIONES Y RECOMENDACIONES 109

5.1 CONCLUSIONES 109

5.2 RECOMENDACIONES 111

REFERENCIAS BIBLIOGRÁFICAS 112

Anexo A
Manual de operación

Anexo B
Tablas de errores NS2

Anexo C
Diagramas de flujo

RESUMEN

El presente trabajo está enfocado a sistemas de localización y mapeo simultáneos (SLAM) en 3D, se ha desarrollado este tema debido a que es uno de lo más actuales con lo que respecta a sistemas de navegación autónomos. Los primeros trabajos de investigación en la base del SLAM, se realizaron a finales de la década de los 80 por Radall Smith, Matthew Self y Peter Cheeseman [6] llegando a la conclusión que el uso del filtro Extendido de Kalman (EKF) es la mejor alternativa para desarrollar el SLAM.

Se escogió realizar el SLAM en 3D con el Filtro Extendido de Kalman por la disposición de la herramienta computacional MATLAB, con la cual se trabaja y desarrolla el algoritmo principal, tanto para la navegación y localización de la plataforma móvil en un entorno controlado.

Anteriormente a este trabajo, en la Escuela Politécnica Nacional se realizó un proyecto de SLAM en dos dimensiones, arrojando como uno de sus resultados que en el estudio del SLAM es muy crítico el tema de la localización, ya que dependiendo del método puede o no tener mayor o menor error al momento de dibujar el mapa. En este proyecto se mejora este problema al usar simultáneamente dos métodos de localización que la plataforma móvil ofrecía.

La plataforma móvil usada es de la compañía alemana FESTO en su línea DIDACTIC, el nombre del robot es Robotino®. Los métodos de localización usados fueron el del propio del robot, que es un encoder, y un accesorio llamado NorthStar2 (NS2).

Los resultados que arrojó este trabajo fueron muy satisfactorios ya que se puede realizar mapas en 3D con cierta cantidad de detalles, los mismos consisten en:

puertas, ventanas, cajas, personas, paredes, lámparas y, en sí, cualquier objeto que se encuentre en el entorno.

Todos los objetivos planteados inicialmente para este proyecto fueron cumplidos a cabalidad, siendo los resultados obtenidos la base para proyectos futuros en el desarrollo de la robótica, especialmente en sistemas de navegación y localización.

PRESENTACIÓN

En este trabajo se realiza el estudio de localización y mapeo simultaneo en 3D con sensores láser. Para este tema se profundizó el conocimiento del Filtro Extendido de Kalman, además de técnicas de graficación para representar tanto el robot como el entorno estructurado.

En el Capítulo I se hace una breve introducción de la formulación y estructura de la problemática del SLAM como también de las técnicas desarrolladas en el tema, se tomará en consideración el hardware y software utilizados en el desarrollo de este Proyecto de Titulación para el manejo de la plataforma móvil Robotino®, y ciertos parámetros para su buen funcionamiento.

En el Capítulo II se describen los fundamentos básicos de los algoritmos utilizados, se tratará sobre el modelo matemático que corresponde al robot, en este caso es el modelo omnidireccional; se profundizará sobre el tema de localización y mapeo simultaneo explicando la construcción de mapas y localización de un robot móvil. Se describe también el manejo de las funciones creadas que permiten corregir la posición.

Se explicará sobre el filtro de Kalman y sobre su versión extendida.

En el Capítulo III se habla de la técnica usada para el mapeo del entorno en 3D como también del dibujo del boceto del robot y la detección de marcas (landmarks) del entorno.

En el Capítulo IV se analiza los resultados obtenidos de los mapas realizados.

En el Capítulo V se tiene las conclusiones y recomendaciones que surgieron en la elaboración de este proyecto.

Finalmente, como Anexos, se sugiere un manual de usuario para el manejo adecuado del robot, tablas que resumen las pruebas realizadas al NS2.

CAPÍTULO 1
MARCO TEÓRICO

1.1 INTRODUCCIÓN

El problema de localización y mapeo simultáneo (SLAM) consiste en responder la siguiente pregunta: ¿Es posible colocar a un robot móvil en un lugar y entorno desconocido y que éste construya incrementalmente un mapa consistente del entorno y que al mismo tiempo determine su ubicación en este mapa? [20].

La respuesta y solución a esta pregunta ha sido considerada por la comunidad robótica móvil como un “santo grial” ya que proporcionaría los medios para hacer un robot verdaderamente autónomo [20].

La solución a este problema ha sido uno de los más grandes avances de la robótica de la década anterior, por lo que el SLAM ha sido formulado y resuelto como un problema teórico en diferentes formas, como también ha sido implementado en varios dominios tales como: robots indoor y outdoor, robots acuáticos, robots voladores. En un nivel teórico y conceptual el SLAM es considerado totalmente resuelto, sin embargo se continúa realizando soluciones más generales con el fin de enriquecer la construcción y el uso de los mapas como parte de su algoritmo [20].

1.2 LOCALIZACION Y MAPEO SIMULTANEO (SLAM)

1.2.1 DEFINICIÓN

Se puede definir al SLAM como una técnica usada por robots y vehículos autónomos para construir un mapa de un entorno desconocido, mientras que al mismo tiempo obtiene su posición en ese entorno. Para ello utilizará los datos de control que gobiernan el vehículo y los datos proporcionados por sensores [20].

1.2.2 PROBLEMAS Y APROXIMACIONES PARA RESOLVER SLAM

El problema del SLAM consiste en colocar a un robot móvil en un lugar y entorno desconocido y que éste construya incrementalmente un mapa consistente del entorno y que al mismo tiempo determine su ubicación en este mapa, para resolverlo se debe efectuar ciertas consideraciones en los elementos que ayudarán en la adquisición de información del entorno [21].

Se debe modelar y tomar en cuenta las imprecisiones de los sensores.

Si se lo considera como un problema de carácter probabilístico se podrá usar métodos como la regla de Bayes, cadenas de Markov, Filtro de Kalman para sistemas lineales y Filtro Extendido de Kalman para sistemas no lineales, o si se lo considera como un sistema basado en representaciones de muestreo se usará un Filtro de partículas [21].

Se considerará el problema de asociación de datos entregada por los sensores y el problema de múltiples hipótesis generadas [20,21].

Para el desarrollo computacional se deberá tomar en cuenta la complejidad de los modelos matemáticos y de correlación de la información [20,21].

El considerar o no un entorno dinámico o estático conlleva a tener o no una complejidad computacional mayor, por lo que es recomendable empezar el desarrollo con un entorno estático [20,21].

El último problema sin ser el menos importante, es considerar si se usará un robot o múltiples robots y si éste constará con un solo sensor o múltiples sensores [21].

Los problemas más críticos al momento de realizar SLAM son el problema de las imprecisiones y el problema de la asociación de datos.

1.2.3 PROCEDIMIENTO GENERAL DE SLAM

A continuación se presenta el Diagrama 1.1 que describe el procedimiento general para la solución al problema del SLAM

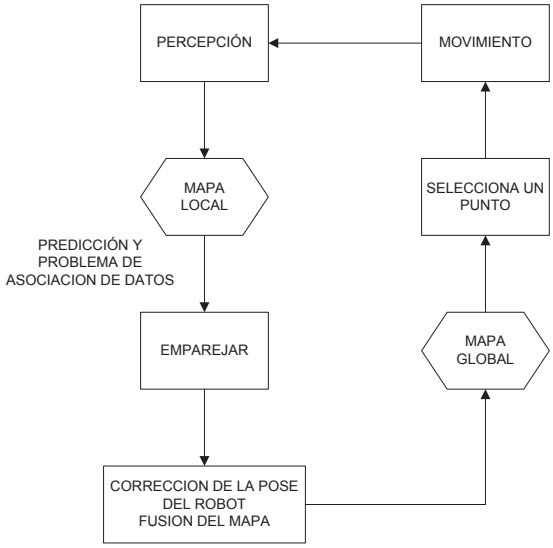


Diagrama 1.1 Procedimiento general de SLAM

1.2.4 ELEMENTOS DE SLAM

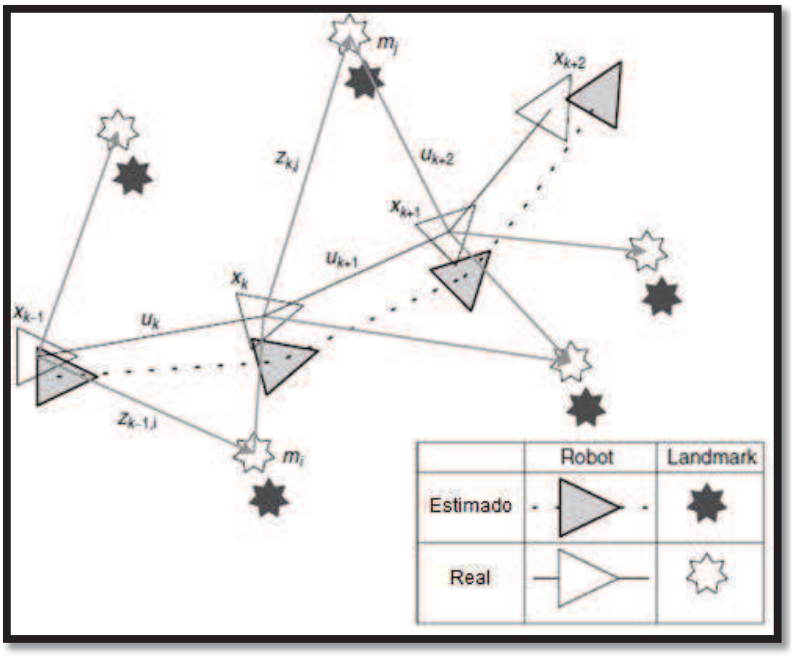


Fig. 1.1 Elementos que se usan en el SLAM, tomado de [20].

Dónde:

$\boldsymbol{k}$ es un periodo de muestreo de tiempo discreto. *k* = 1,2,…

$\boldsymbol{x_k}$: es la posición verdadera del vehículo en un periodo *k*.

$\boldsymbol{u_k}$**:** es el vector de control, se conoce, aplicado en un periodo *k-1* conduce al vehículo desde $\boldsymbol{x_{k-1}}$a $\boldsymbol{x_k}$en el periodo *k*.

$\boldsymbol{m_i}$**:** La verdadera localización o parametrización del *i*th obstáculo o landmarks.

$\boldsymbol{z_{ki}}$: Una observación (medida) de un obstáculo tomada desde la posición $\boldsymbol{x_k}$ en el tiempo *k*.

$\boldsymbol{z_k}$**:** La observación (genérica) (de uno o más obstáculos) tomada en el periodo *k*.

Historial de estados: $\boldsymbol{X_K = \{x_0, x_1, \ldots, x_k\} = \{X_{k-1}, x_k\}}$.

Historial de entradas de control: $\boldsymbol{U_K = \{u_1, u_2, \ldots, u_k\} = \{U_{k-1}, u_k\}}$.

Conjunto de todos los obstáculos: $\boldsymbol{m = \{m_0, m_1, \ldots, m_k\}}$.

Historial de observaciones $\boldsymbol{Z_K = \{z_0, z_1, \ldots, z_k\} = \{Z_{k-1}, z_k\}}$.

También se lo puede representar gráfica y matemáticamente como se muestra a continuación:

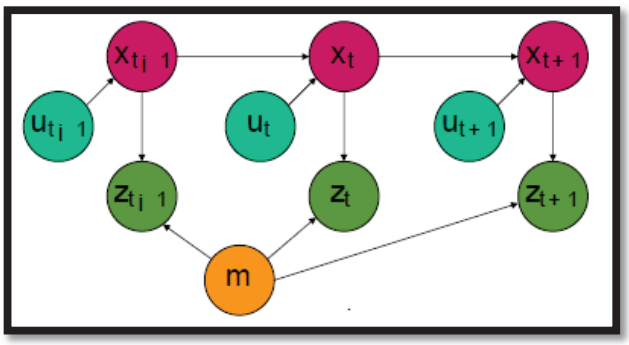


Fig. 1.2 SLAM, tomado de [21].

De la Fig. 1.2 se puede obtener matemáticamente (1.1) y (1.2)

$$Bel(x_t) = p(x_t|z_t,u_{t-1}, z_{t-1}, u_{t-2}, \dots, z_0) \tag{1.1}$$

$$Regla\ de\ Bayes\ \rightarrow Bel(x_t) = \frac{p(z_t|x_t, u_{t-1}, \dots, z_0)p(x_t|u_{t-1}, \dots, z_0)}{p(z_t|u_{t-1}, \dots, z_0)} \tag{1.2}$$

1.3 TIPOS DE SLAM

1.3.1 FULL SLAM

Se basa fundamentalmente en la estimación del camino y mapa completos, en la Fig. 1.3 se muestra un diagrama que representa al Full SLAM, además se muestra la (1.3) que es la función de probabilidad que rige en esta técnica.

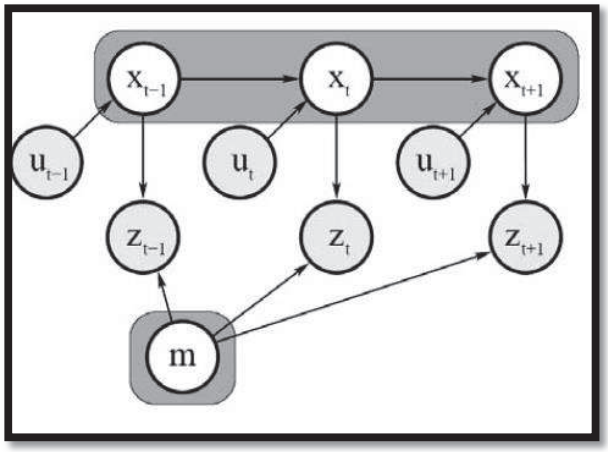


Fig. 1.3 SLAM, tomado de [21].

$$p(x_{1:t}, \mathrm{m}|z_{1:t,}, u_{1:t}) \quad (1.3)$$

El FastSLAM es un tipo de técnica Full SLAM la cual utiliza un modelo de distribución de filtro de partículas muestreado [21].

1.3.1.1 Características del FastSLAM

- Es una aproximación de los Filtros de Partículas.
- Complejidad crece exponencialmente al número de dimensiones.
- EKFs para cada partícula.
- Cada partícula representa una posición posible del robot.
- Para cada partícula:
    - Muestrea una nueva posición.
    - Actualiza su correspondiente EKF.
    - Obtiene una importancia.
- Re muestreo.
- Procesos no lineales y distribuciones no gaussianas [21].

1.3.1.2 Filtro de Partículas

Se puede definir como la representación de la distribución de probabilidad como un conjunto de partículas discretas las cuales ocupan el espacio de estados.

1.3.1.3 Características del Filtro de Partículas

- Representan la esperanza (belief) por medio de muestras aleatorias.
- Estimación de procesos no-gaussianos y no-lineales.
- Principio de Sampling Importance Resampling (SIR):
    - Obtiene la nueva generación de partículas.
    - Asigna un peso (importancia) a cada partícula dependiendo de cómo sus estimaciones se parezcan a las medidas.
    - Re muestrea.

- Escenarios típicos de aplicación son: seguimiento, localización, etc.
- Problema: El número de partículas necesario para las representaciones crece exponencialmente con la dimensión del espacio de estados [21].

1.3.1.4 Funcionamiento del Filtro de Partículas

A continuación se muestra el Diagrama 1.2 que representa de manera simplificada el funcionamiento del Filtro de Partículas

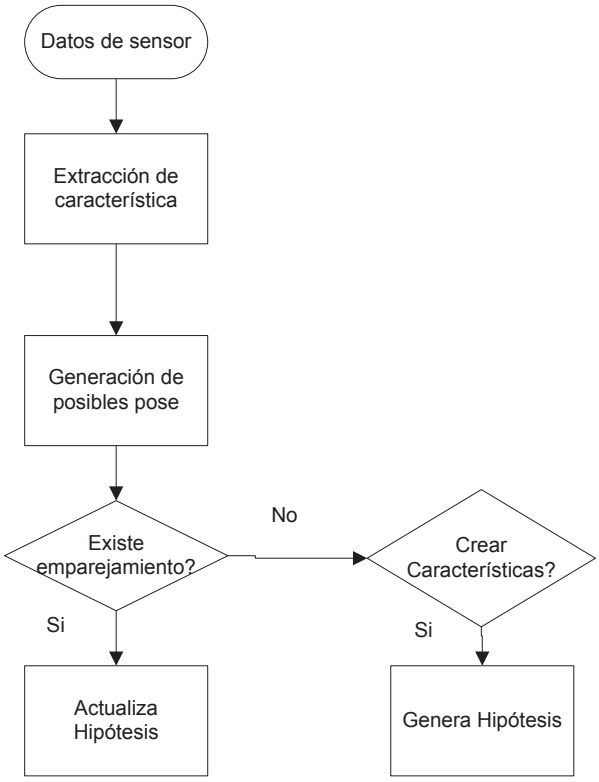


Diagrama 1.2 Filtro de partículas

Existe un procedimiento adicional para manejar un filtro de partículas, que consiste en que la dimensión del espacio de estados es reducida drásticamente factorizando el filtro de partículas, a este proceso se lo conoce como Rao-Blackwellization en honor a los matemáticos que lo desarrollaron [21].

1.3.1.5 Descripción de FastSLAM

- Está basado en una Rao-Blackwellization de un filtro de partículas.
- Cada *landmark* se representa por un EKF de 2x2
- Cada partícula tiene que almacenar M EKFs (M = Número de características del mapa) como se muestra en la Fig.1.4 [21].

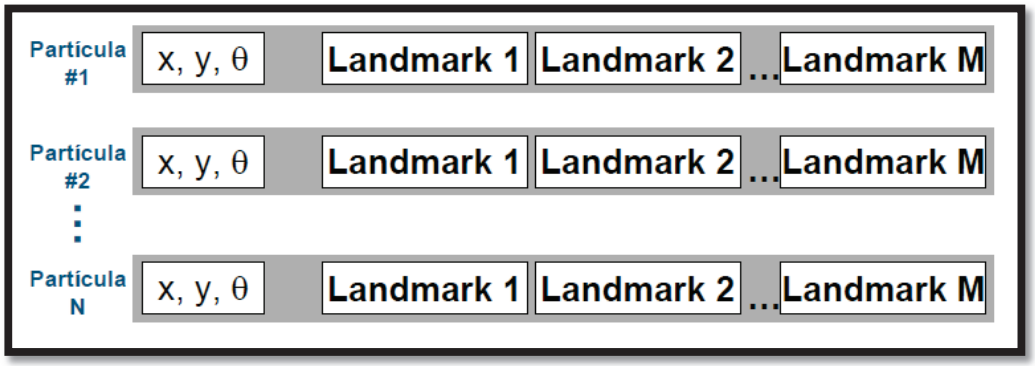


Fig. 1.4 Distribución de características en partículas, tomado de [21].

1.3.1.6 Inconvenientes de FastSLAM

- Cada mapa es demasiado grande en el caso de usar *grid maps.*
- Se debe calcular adecuadamente las distribuciones de partículas.
- El número de partículas crece exponencialmente con la dimensión del espacio de estados:
    - 1D – n partículas
    - 2D – n2 partículas
    - mD – nm partículas [21].

1.3.2 ONLINE SLAM

Se basa fundamentalmente en la estimación de la posición y del mapa más reciente, y los cálculos se los realiza cada periodo de muestreo. A continuación se presenta la Fig. 1.5 donde se muestra un diagrama que representa un OnlineSLAM [21].

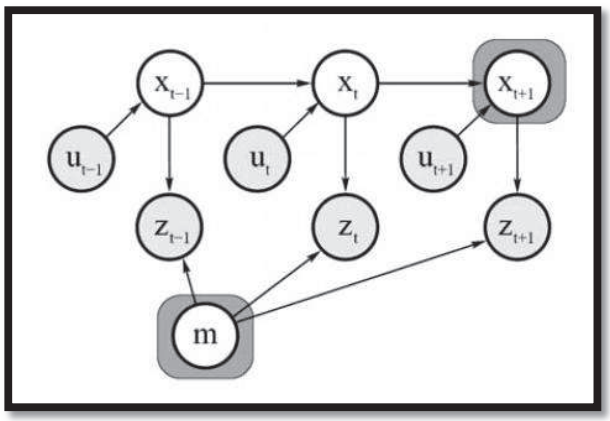


Fig. 1.5 OnlineSLAM, tomado de [21].

El Filtro Extendido de Kalman es un tipo de técnica OnlineSLAM la cual utiliza un modelo de distribución de probabilidad gaussiana [21].

1.3.2.1 Características del EKF SLAM

- Correspondencias son conocidas.
- La imprecisión crece con el tiempo:
    - Repetidas observaciones reducen la imprecisión.
- Imprecisión de *landmarks:*
    - Mayormente causadas por la imprecisión en la posición.
    - Correlación en la matriz de covarianza.
    - Información propagada de *landmarks* previas [21].

1.3.2.2 Descripción del EKF SLAM

A continuación en la Fig. 1.6 se muestra un diagrama de bloques que describe el funcionamiento del EKF SLAM.

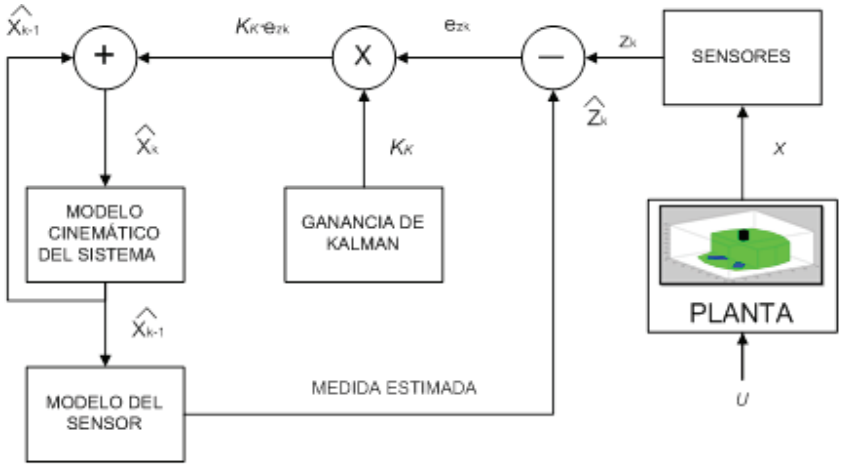


Fig. 1.6 Diagrama de bloques de EKF, tomado de [24].

En capítulos posteriores se revisará más a profundidad las ecuaciones que rigen al EKF.

1.3.2.3 Inconvenientes de EKF SLAM

- Sólo un conjunto de medidas para caracterizar las asociaciones consideradas.
    - Utiliza la asociación de probabilidad máxima.
- **Esfuerzo computacional**: el paso de actualización de las observaciones requiere que todas las *landmarks* y sus matrices de covarianza sean actualizadas cada periodo de muestreo, esto implica que el tiempo de procesamiento crece cuadráticamente con el número de *landmarks.*
- **No linealidad:** Se emplean modelos linealizados de modelos de movimiento y observación no lineales, esto implica que se producen errores en los modelos [21].

1.4 APLICACIONES DE SLAM

A continuación se mencionará las más recientes aplicaciones del SLAM en diferentes áreas de investigación en todo el mundo.

1.4.1 FÚTBOL DE ROBOTS

El terreno de juego contiene jugadores del propio equipo, jugadores del equipo contrario y un balón que hay que perseguir y encajar en la portería del adversario: el buen robot futbolista debe ser capaz de registrar los movimientos de todos estos elementos y usarlos para anticipar los propios, lo que no es nada fácil. Además, esos movimientos propios están guiados por la necesidad de colaborar con los compañeros de equipo con el objetivo de meter goles, lo cual requiere tener en mente una estrategia. Los avances en robótica futbolística se pueden aplicar a otras tareas que exijan movimientos coordinados, colaboración y estrategia en un universo complejo y cambiante, como en actividades de rescate, de construcción, de reparación e incluso en tareas domésticas [22].

Fig. 1.7 Robots AIBO en la “RoboCup”, tomado de [21].

1.4.2 VEHÍCHULOS SUBMARINOS

Una de las aplicaciones más recientes desarrollada por la Universidad de Oxford es MOOS: Software for Devoloping and Deploying Autonomus Oceanic Vehicle (Software para desarrollar y desplegar un vehículo submarino autónomo). A continuación se muestra un diagrama de bloques que describe esta aplicación

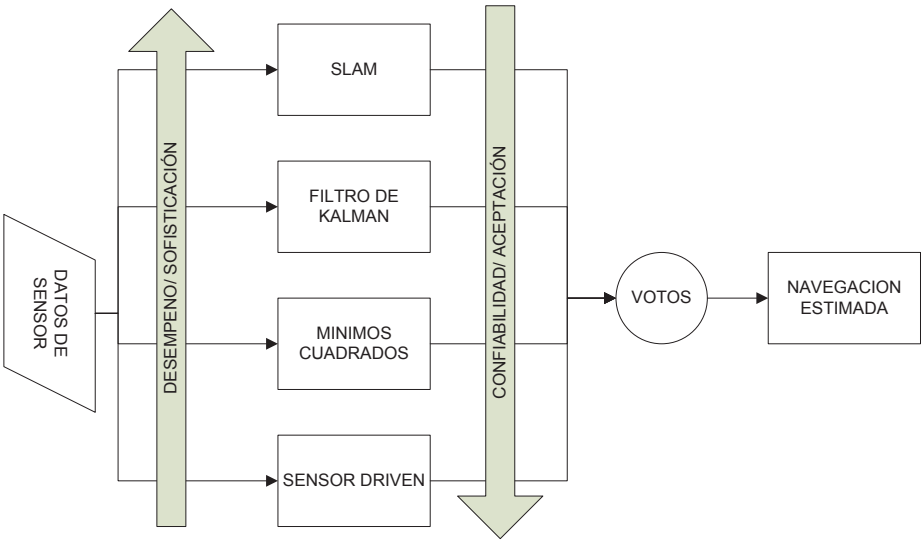


Fig. 1.8 Descripción de MOOS, tomado de [21].

1.4.3 MAPAS 3D

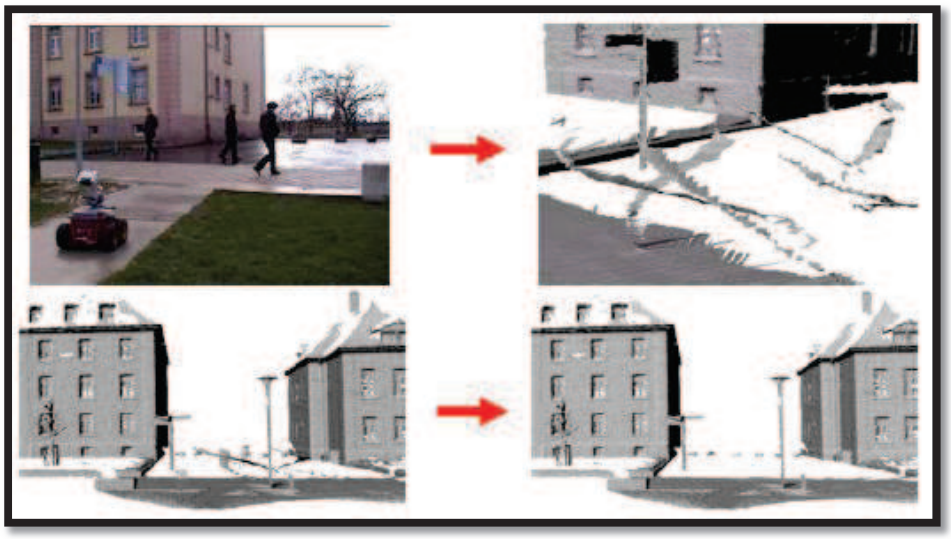


Fig. 1.9 Mapeo de ambientes dinámicos. tomado de [21].

En la Fig. 1.9 se puede apreciar varias tomas del mapa en 3D realizado por un robot móvil, en un ambiente dinámico en exteriores.

1.4.4 ASOCIACIÓN DE DATOS

Dado un mapa del entorno más un conjunto de observaciones de sensores se puede tener una asociación de datos a través del SLAM como se muestra en la Fig. 1.10 [21].

Fig. 1.10 Relocalización usando laser y visión. Tomado de [21].

1.4.5 MINERÍA

Para el transporte de materiales peligrosos a través de camiones no tripulados como se muestra en la Fig. 1.11 [21].

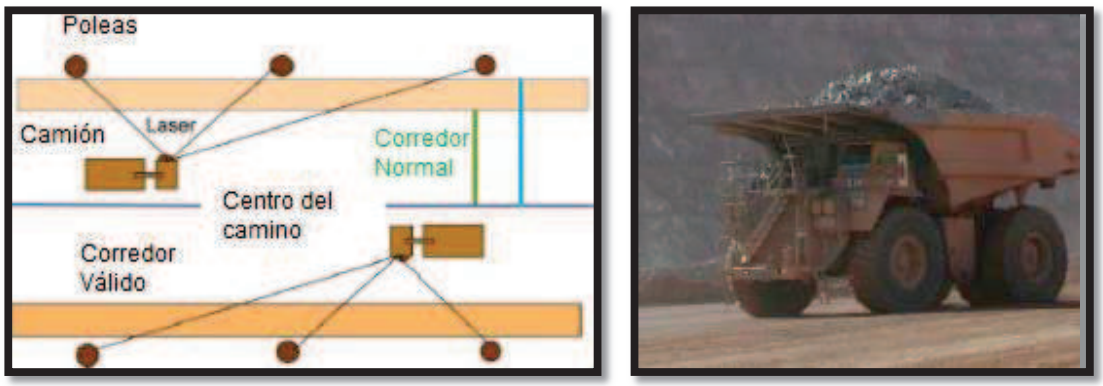


Fig. 1.11 Aplicaciones de SLAM, tomado de [21].

1.5 VENTAJAS DE SLAM

A continuación se mencionará las mayores ventajas que logra obtener al realizar un SLAM.

Para robots móviles, al realizar SLAM no se necesita conocer ni la posición ni el mapa ya que a medida que el robot navega en un entorno desconocido este va reconstruyendo el mapa y localizándose.

La autonomía en la navegación de robots móviles en lugares desconocidos se ve incrementada debido a que no se necesita conocer nada de información del mapa.

No se necesita usar sensores del tipo GPS para la localización, se usa sensores de menor envergadura como son del tipo láser, cámaras, etc. La robustez que presenta el SLAM ante la información de los sensores es amplia debido a la minimización del ruido a través de los filtros usados [21].

Al final se obtiene un mapa con la mayor cantidad de características del entorno y una navegación totalmente autónoma del robot [21].

1.6 INCONVENIENTES DE SLAM

El SLAM presenta inconvenientes en grandes entornos y sistemas no lineales debido al coste computacional que esto implica.

La complejidad del SLAM y coste computacional se ven incrementadas cuando se trata de entornos dinámicos.

Por último el SLAM no toma en cuenta ningún algoritmo ni arquitectura de navegación, esto es un tema que complementa al SLAM y que no forma parte [21].

1.7 LOCALIZACION Y MAPEO SIMULTANEO EN 3D (SLAM 3D)

1.7.1 INTRODUCCIÓN

En la actualidad se han desarrollado varios métodos para realizar SLAM 3D utilizando sensores Láser Rangers, Sonares o Cámaras mono o estéreo, en este Proyecto de Titulación se enfocará en un método en particular, el cual fue realizado con sensores Láser.

El método consiste en combinar la percepción en 3D (adquisición de datos) con la localización en 2D ya que el robot móvil se mueve sobre un solo plano, con esta consideración el costo computacional de la localización en 2D es menor al costo computacional necesario en una localización en tres dimensiones [25].

1.7.2 TÉCNICAS DE SLAM 3D

1.7.2.1 Escáner de rango láser 3D

Para la precepción de características de entorno en 3D es común utilizar un sensor láser comercial 2D que funciona con el principio de medición de tiempo de vuelo acoplado con un servomotor que permite el movimiento del láser y completa la adquisición de datos en 3D, en el presente Proyecto de Titulación se optó por usar un sensor láser comercial 2D pero el movimiento del robot móvil permite conseguir la detección en 3D, tema que será tratado con mayor amplitud en el Capítulo 3 [25].

El sensor láser 2D gira alrededor de su eje Z, registrándose filas de datos que corresponden a planos del entorno. Las filas de datos pueden ser descritos como nubes de puntos que al ser graficadas muestran la forma de los objetos correspondientes al entorno [25].

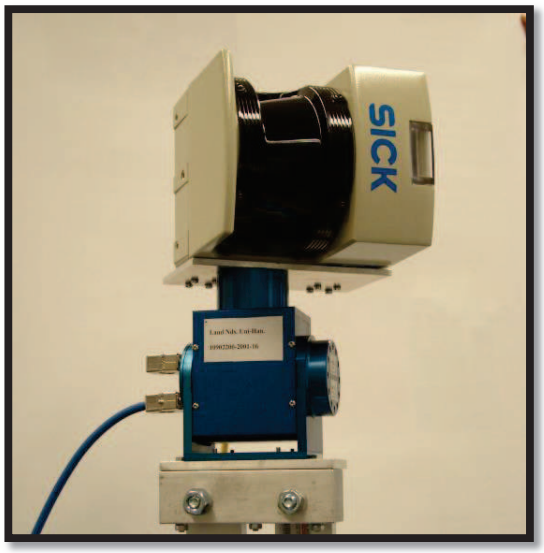


Fig. 1.12 Escáner de rango láser 3D tomado de [25].

La adquisición de datos con este tipo de sensor es de todos modos un reto, ya que se debe sincronizar el barrido del sensor láser con el barrido producido por el movimiento del servomotor acoplado al sensor, para esto se debe contar con un sistema operativo que pueda realizar las mediciones en el tiempo de muestreo más

adecuado y que las nubes de puntos sean ubicadas correctamente. En el presente Proyecto de Titulación no fue necesaria la sincronización con un servomotor extra, ya que se recibieron los datos por parte del sensor láser y se los ubicó en el espacio, este tema se explicará con mayor detalle en el Capítulo 3.

1.7.2.2 Escaneo en movimiento

Para poder ubicar los datos adquiridos al tener al robot móvil en movimiento, fue necesario tomar en cuenta su posición absoluta, de esta forma las nubes de puntos tienen un origen en común y ocupan su lugar en el espacio debido a que se trabaja en interiores y se considera que el robot se mueve en un solo plano.

Fig. 1.13 Ambiente real, tomado de [25].

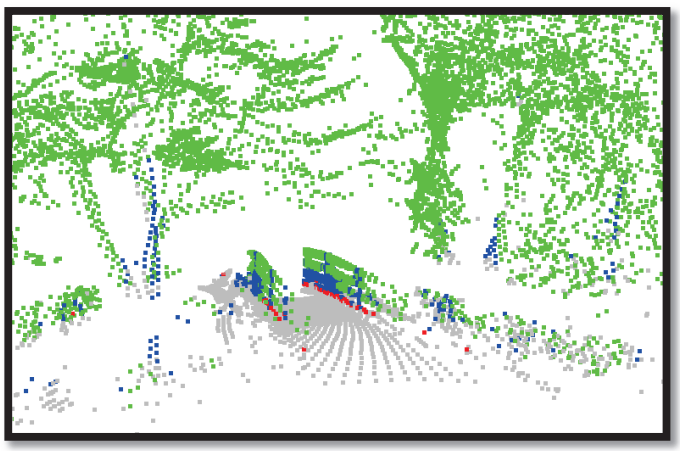


Fig. 1.14 Nube de puntos 3D, tomado de [25].

En las Fig. 1.13 y Fig. 1.14 Se observa el mapeo de un ambiente exterior, donde gracias a un sensor láser en posición vertical conjuntamente con un algoritmo, permiten detectar que el robot móvil se encuentra a niveles diferentes, el algoritmo consiste en detectar landmarks naturales, al realizar dos barridos se detecta que la landmarks ha cambiado de nivel con lo que los datos de los objetos en 3D deben ser corregidos en su coordenada en sobre el eje Z. Más detalles sobre SLAM 3D en ambientes desconocidos y exteriores se puede encontrar en [25].

1.8 PLATAFORMA ROBOTICA MÓVIL

Robotino® es una plataforma robótica construida por la empresa alemana FESTO S.A, su principal objetivo comercial es la automatización y control de procesos productivos, consolidando su desarrollo de productos con la innovación de tecnología para la obtención de una máxima producción [1].

Dentro de sus divisiones de productos y soluciones tecnológicas, se deriva la división FESTO Didactic, con 35 años de existencia dedicada a la formación y perfeccionamiento industrial, lo que a su vez también incorpora la utilización de equipamientos para la formación y calificación de personal en entornos productivos industriales [2].

Robotino® es un producto de la división Didactic, es un sistema robótico móvil con actuador omnidireccional, podría catalogarse como un sistema de aprendizaje para la formación y perfeccionamiento, e incluso una plataforma de investigación y desarrollo para escuelas técnicas en un mismo sistema [1].

Su sistema de navegación, permite un manejo omnidireccional con movimientos hacia delante, atrás y lateralmente, además permitiéndole girar sobre un punto determinado, la plataforma posee todo un sistema de sensores analógicos, digitales y una webcam para su interacción a través de visión artificial. Todo esto incorporado a su tarjeta de control que está provista de un sistema embebido con características de computadora de prestaciones industriales [1].

Gracias a sus motores, separados entre ellos por 120°, en conjunto con las ruedas omnidireccionales, permiten el movimiento hacia diferentes lugares en sistemas de coordenadas de 2 dimensiones, la estructura metálica donde se soporta todo el equipo está fabricada en acero inoxidable, cuyas junturas tienen una terminación de soldadura láser de precisión, lo que le entrega un diseño innovador para su utilización en tareas prácticas [1].

1.9 HARDWARE DE ROBOTINO®

Se proporcionará una breve descripción de cada uno de los componentes que posee el robot usados en este proyecto.

1.9.1 CHASÍS

La estructura base de Robotino® en acero inoxidable permite soportar todo el peso de los actuadores, tarjeta de control, y sistema de alimentación (Ver Fig. 1.15) [3].

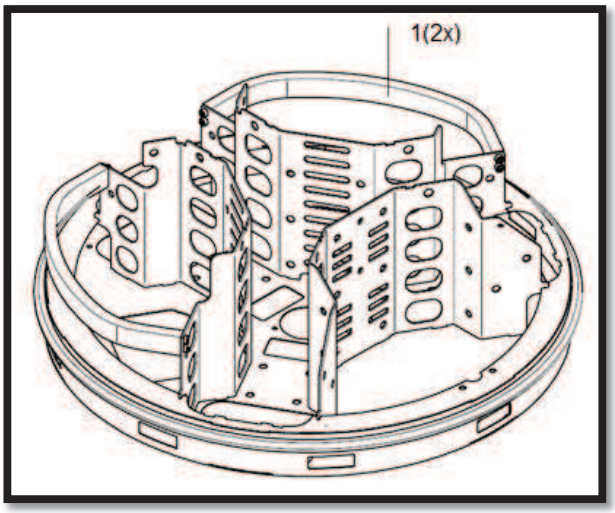


Fig. 1.151 Chasís (1 (2x) asas), tomado de [3].

Para un adecuado transporte del Robotino®, el chasís, posee 2 asas como se observa en la Fig.1.15. Es recomendable sujetar al robot única y exclusivamente de estas asas, ya que si es sujeto desde otro de sus componentes puede provocar daños severos al equipo (Ver Fig. 1.15) [3].

Es importante recalcar que el diseño del chasís del robot permite el acoplamiento de otros equipos como son: baterías, unidad de accionamiento, banda anticolisión, webcam y todos los sensores [3].

1.9.2 BATERÍAS

Para el funcionamiento autónomo del Robotino®, éste posee un sistema de alimentación de dos baterías recargables de 12VDC, las mismas que están sujetas al chasís por una correa [3].

Existe la posibilidad de cargar directamente las baterías que se encuentran en el chasís a través de un conector incorporado a éste. También es posible intercambiar las baterías cuando éstas ya cumplieron su vida útil (Ver Fig. 1.16) [3].

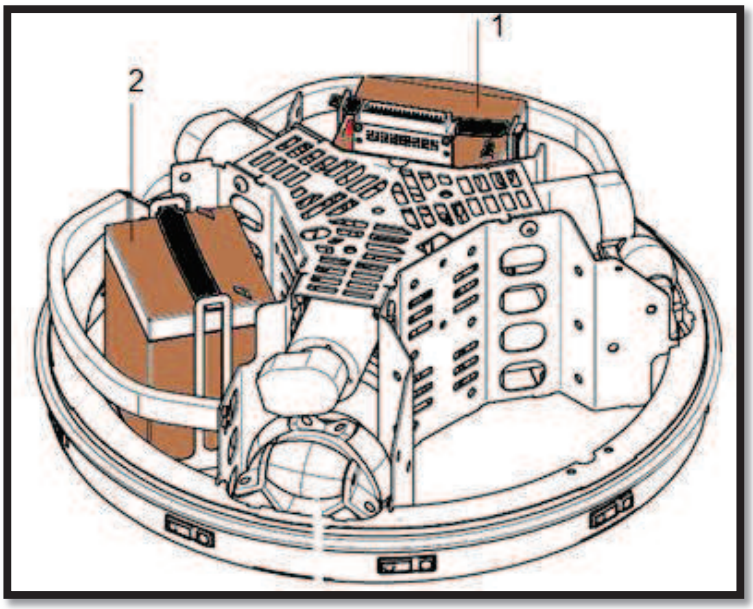


Fig. 1.162 Batería 12VDC (1) y (2), tomado de [3].

1.9.3 UNIDAD DE ACCIONAMIENTO

Para un movimiento en varias direcciones, el robot tiene incorporado tres actuadores omnidireccionales independientes, los mismos tienen una separación de 120° cada uno con respecto al otro (Ver Fig. 1.17) [3].

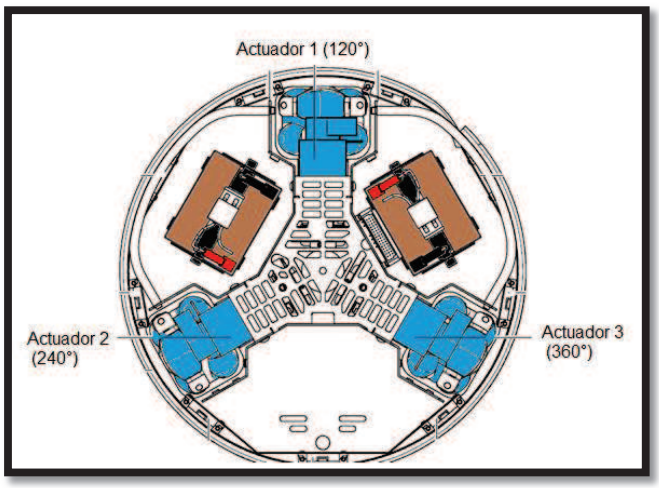


Fig. 1.17 Posición de los actuadores, tomado de [3].

Para un desplazamiento controlado cada uno de los actuadores poseen los siguientes elementos: motor DC, encoder incremental, rodillos omnidireccionales, reductor con una relación de reducción de 16:1 y correa dentada (Ver Fig. 1.18) [3].

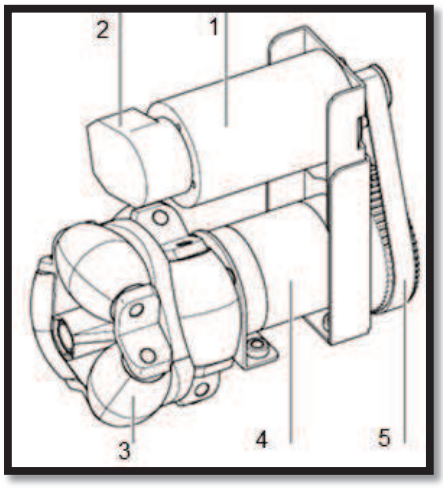


Fig. 1.18 Motor DC (1), encoder incremental (2), rodillos (3), reductor (4) y correa dentada (5), tomado de [3].

1.9.3.1 Motor DC

El motor DC de escobillas tiene un sistema electrónico de regulación integrado y sus características se presentan en la Tabla 1.1 [3].

Tabla 1.1 Especificaciones técnicas de motor DC

Motor DC (42x25)	Unidad de medida	Valor
Tensión nominal	VDC	24
Velocidad nominal	RPM	3600
Par nominal	Ncm	3,8
Intensidad nominal	A	0,9
Par de arranque	Ncm	20
Intensidad de arranque	A	4
Velocidad sin carga	RPM	4200
Intensidad sin carga	A	0,17
Intensidad de desmagnetización	A	71
Momento de inercia de la masa	Gcm	71
Peso del motor	Gr	390

1.9.3.2 Encoder incremental

Los encoder son sensores que generan señales digitales en respuesta al movimiento sea éste lineal, rotacional o una combinación de ambos. En el mercado existen dos tipos de encoder, uno de ellos es un sensor que responde a la rotación y el otro al movimiento lineal. Existe aplicaciones en las cuales se usan estos dos tipos de sensores conjuntamente con dispositivos mecánicos, con esta fusión se logra medir movimientos lineales, velocidad y posición [4].

En el caso particular de Robotino® usa un encoder de tipo incremental, caracterizado porque determina su posición contando el número de impulsos que se generan cuando un rayo de luz es atravesado por marcas opacas en la superficie de un disco unido al eje, el número de marcas opacas del disco es 512 (Ver Fig. 1.19) [4,19].

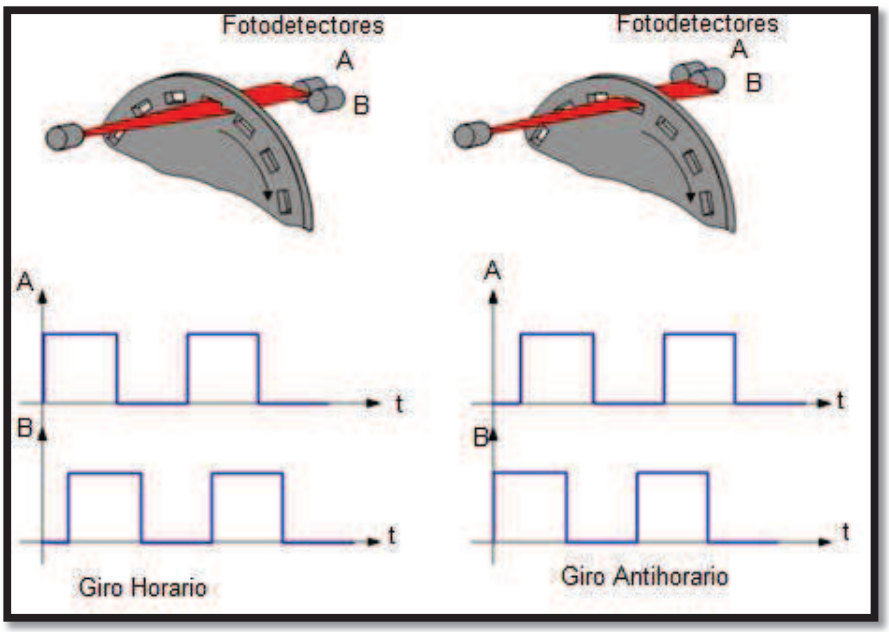


Fig. 1.19 Encoder incremental.

En el estator hay como mínimo dos pares de foto receptores ópticos, escalados un número entero de pasos más ¼ de paso. Al girar el rotor genera una señal cuadrada, el escalado hace que las señales tengan un desfase de ¼ de periodo si el rotor gira en un sentido y de ¾ si gira en el sentido contrario, lo que se utiliza para discriminar el sentido de giro y la resolución del encoder depende del número de impulsos por revolución (Ver Fig. 1.19) [4].

Este tipo de sensores son muy usados en la industria, ya que pueden ser utilizados como transductores de retroalimentación para el control de velocidad de motores, dispositivos de control de puertas, robótica, máquinas de ensamblaje, máquinas taladradoras, equipo médico, entre otros [4].

1.9.3.3 Rodillos omnidireccionales

Las ruedas omnidireccionales que posee Robotino® son de 80 mm de diámetro, éstas nos brindan dos ventajas muy importantes: toda la tracción está dada únicamente por tres ruedas y la facilidad de movimiento en el entorno, sin cambiar el ángulo de orientación del robot (Ver Fig.1.20) [4].

Fig. 1.20 Rodillo omnidireccional, tomado de [4].

1.9.3.4 Reductor de relación (16:1)

La caja reductora que posee Robotino® es un reductor planetario de alta eficiencia, sus partes internas están fabricadas de acero. A continuación se presenta la Tabla 1.2 con las características técnicas de la caja reductora [4].

Tabla 1.2. Especificaciones de reductor

Reproductor planetario (PLG 42S)	Valor
De una sola etapa, Nm	3.5
De una sola etapa, i	4.1-8.1
2 etapas, Nm	6
2 etapas, i	16.1-64.1
3 etapas, Nm	14
3 etapas, i	100.1- 512.1

1.9.3.5 Correa dentada

Correa de caucho de dientes que evita que ésta deslice durante la transmisión del movimiento, asegurando así una relación de transmisión constante [4].

1.9.4 UNIDAD DE CONTROL

Toda la plataforma Robotino® es comandada por la unidad de control, ésta está encargada de manejar y distribuir las instrucciones del algoritmo de control a los

diferentes componentes del robot. Otra de sus funciones es transmitir y receptar información concerniente a los sensores, actuadores y accesorios a la PC [3].

La unidad de control está compuesta por los siguientes elementos: punto de acceso WiFi, tarjeta compact flash, interfaces, display, interface de entrada y salida (E/S) [3].

En la Fig. 1.21 se puede observar la posición física de la unidad de control en el robot.

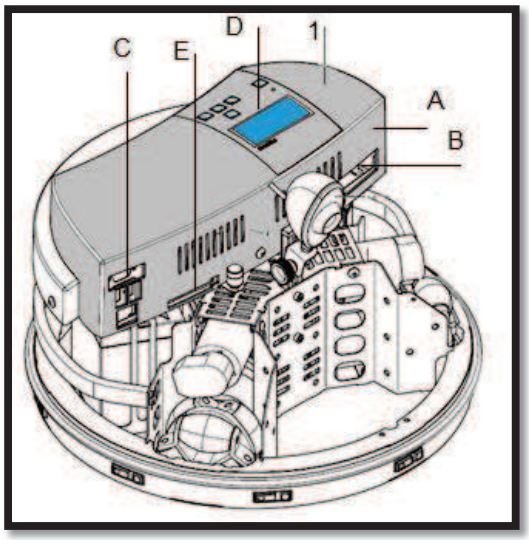


Fig. 1.21 Unidad de control (1), punto de acceso WiFi (A), interface de E/S (B), interfaces (C), display (D), tarjeta compact flash (E), tomado de [3].

1.9.4.1 Compact Flash.

El sistema operativo que comanda al equipo, las librerías de funciones y los algoritmos de control, se encuentra en la tarjeta compact flash. La ventaja del Robotino® es que esta tarjeta compact flash puede ser actualizada a una versión superior de sistema operativo, el cual nos permita manejar nuevos accesorios, mejorar y aumentar librerías [4].

El Robotino® de fábrica trae consigo una tarjeta compact flash de 1GB de capacidad con un sistema operativo 2.0, para este Proyecto de Titulación se usó una tarjeta de 4GB de capacidad con un sistema operativo 2.4 api2.

1.9.4.2 Interfaces.

Esta unidad trae una serie de interfaces para su configuración y conectividad, éstas son: conexión VGA, entradas USB, conexión Ethernet (Ver Fig. 1.22) [3].

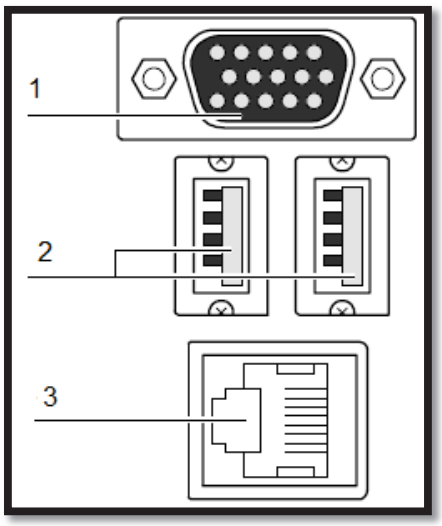


Fig. 1.22 Conector VGA (1), puertos USB 1 y 2 (2) y conector Ethernet (3), tomado de [3].

La función principal de estas interfaces es permitir la comunicación con un monitor, mouse y un teclado, para de esta forma tener acceso total al sistema operativo del robot, también puede tener un acceso vía Ethernet en caso de fallos en la comunicación WiFi [4].

1.9.4.3 Teclado membrana y display.

En la unidad de control existe un teclado con un display, en éste se puede observar información del estado operativo del Robotino®, selección de parámetros básicos de configuración, programas demostrativos y funciones de encendido del equipo [4].

En el display se muestra: el estado de carga de baterías, dirección IP del robot y el sistema operativo con el cual está funcionando la tarjeta compact flash. A continuación se muestra los componentes físicos de este teclado (Ver Fig. 1.23) [4].

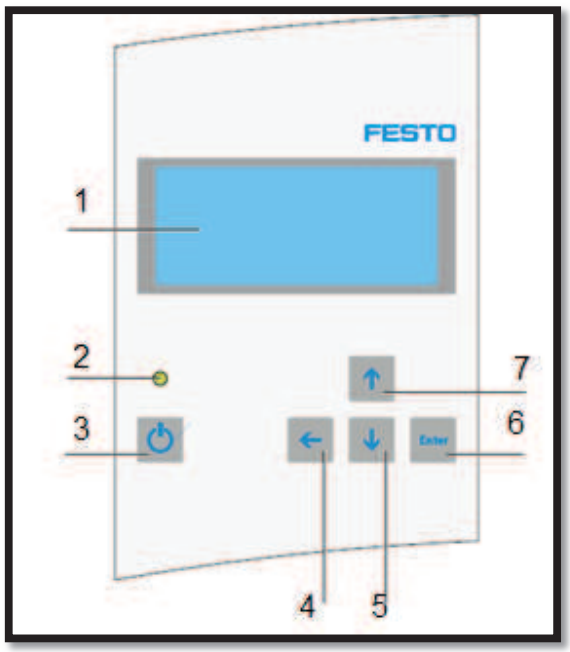


Fig. 1.23 Display (1), luz de encendido (2), botón de encendido/apagado (3), botones de navegación (4) (5) (7) y botón enter (6), tomado de [3].

1.9.4.4 Interface E/S.

Esta interface permite la conexión de sensores y actuadores externos al robot, ésta cuenta con:

- 8 entradas analógicas (0 a 10 V) (AIN0 hasta AIN7).
- 8 entradas digitales (DI0 hasta DI7).
- 8 salidas digitales (DO0 hasta DO7).
- Relés para actuadores adicionales (REL0 y REL1). Los contactos de los relés pueden utilizarse como NA, NC o conmutados [4].

1.9.5 SENSORES

Se hará una descripción de los sensores incorporados al robot, los mismos que ayudan a medir las distancias hacia a otros objetos u obstáculos al equipo [4].

1.9.5.1 Sensor anticolisión.

El sensor anticolisión está compuesto por dos bandas metálicas que circundan al chasís, en condiciones normales de funcionamiento dichas bandas mantienen una distancia entre sí, lo cual permite que hagan contacto cuando existe una presión en la banda. Todo este proceso hace posible que el robot tenga la capacidad de detenerse a la mínima colisión con algún objeto (Ver Fig. 1.24) [1].

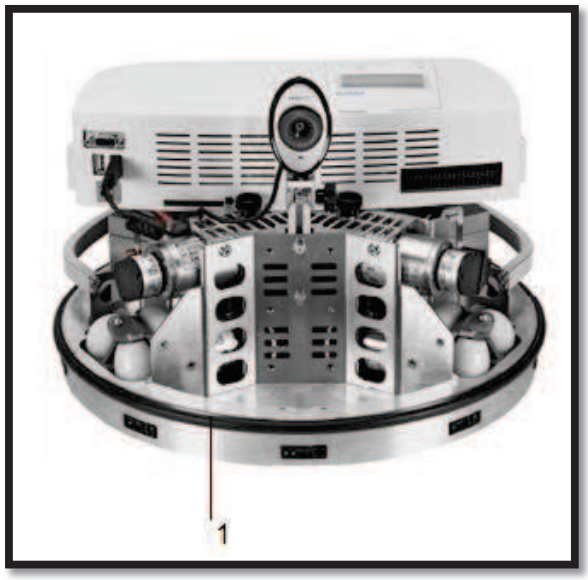


Fig. 1.24 Sensor anticolisión (1), tomado de [3].

1.9.5.2 Sensores infrarrojos

El robot está provisto de 9 sensores infrarrojos, los cuales permiten la medición de la distancia entre el robot y elementos en su zona circundante en un rango entre 4 y 30 cm. Los sensores están colocados en el chasís en un ángulo de 40° entre sí (Ver Fig. 1.25) [4].

Los sensores son llamados IR1-9 pero en el software son direccionados como:

“IR1 en Distance 1” de la misma forma de IR2 a “Distance 2” hasta llegar a “Distance 9” [4].

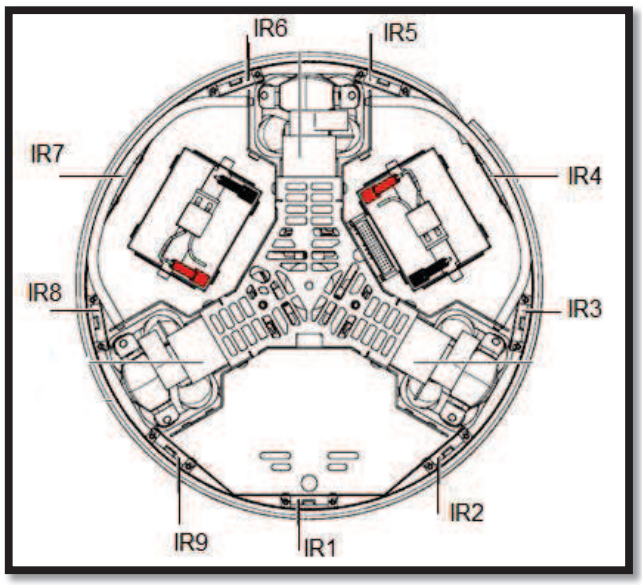


Fig. 1.25 Ubicación de sensores infrarrojos, tomado de [3].

1.10 ACCESORIOS

EL robot trae consigo únicamente los elementos ya antes descritos, pero la compañía FESTO ofrece ciertos dispositivos que nos permite un mejor desempeño del robot, para propósitos de este trabajo se adquirió un sensor láser HOKUYO URG-04LX-UG01 y un sistema de localización NorthStar.

1.10.1 NORTHSTAR

El sistema de localización NorthStar 2, también conocido como NS2, está conformado por 2 accesorios, un emisor y un detector [12].

El detector es un instrumento de medida, el cual nos permite determinar la posición del Robotino®, entregándonos tanto las coordenadas en X y Y como también el ángulo de orientación respecto al eje de referencia global absoluto [12].

El proceso de localización que realiza el NorthStar 2 (NS2), se basa en la emisión de dos regiones de luz infrarrojas sobre un plano superior, en este caso es el techo del entorno, con lo cual se puede determinar el punto de referencia global; las luces son proporcionadas por el proyector, que es el punto de referencia [12].

1.10.1.1 Proyector NorthStar 2 (NS2)

El proyector NorthStar 2 es el accesorio complementario del NorthStar 2 detector, éste emite luces infrarrojas invisibles, o también conocidas como spots al techo del entorno, para que de esta forma el detector NS2 pueda calcular la posición y orientación del robot [12].

El diseño de NS2 proyector es la versión actualizada de los accesorios de localización, posee un diseño esférico mejorado, el cual permite localizar los spots en el techo del entorno. Posee un trípode como soporte, el cual está unido mediante un tornillo de regulación [12].

Para una calibración inicial, el NS2 proyector permite observar momentáneamente los spots cuando el operador toca el anillo metálico que rodea al proyector. (Ver Fig. 1.26) [12].

Fig. 1.26 Proyector NorthStar 2 (NS2), tomado de [12].

1.10.1.2 Emisor NorthStar 2 (NS2)

A diferencia de la versión anterior de NorthStar, el NS2 emisor presenta una gran facilidad e montaje sobre el Robotino®. El dispositivo puede ser montano en la ranura de ventilación de la unidad de control mediante un tornillo hexagonal de 2.5 mm (Ver Fig. 1.27) [12].

La comunicación del NS2 emisor con el Robotino® es mediante un cable USB [12].

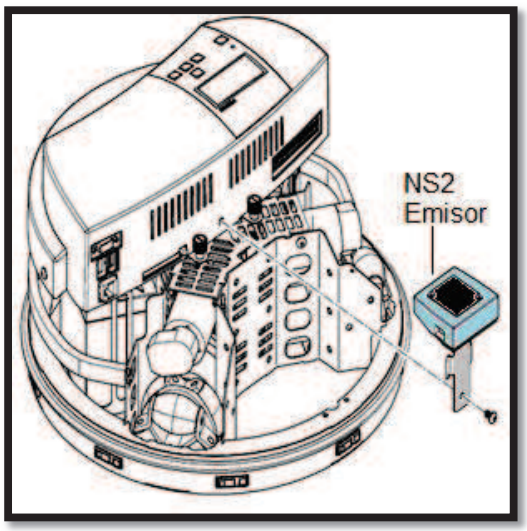


Fig. 1.27 Emisor NorthStar 2 (NS2), tomado de [12].

1.10.2 LÁSER

El láser utilizado en este Proyecto de Titulación es de la marca Hokuyo el modelo URG-04LX-UG01. Éste es un sensor láser para escaneo de área [9].

Fig. 1.28 Láser Hokuyo URG-04LX-UG0, tomado de [9].

En la Tabla 1.3 se muestra los datos técnicos del láser mencionado.

Tabla 1.3 Datos técnicos de Láser Hokuyo URG-04LX-UG0, tomado de [9].

Especificaciones	Valores
Longitud de onda de láser	785 nm
Potencia del láser	Menor a 0.8mW
Voltaje de entrada	5v ± 5%
Consumo de corriente	500mA o menos (800mA al iniciarse)
Clase de seguridad	1 (IEC60825-1)
Precisión	20~1000 mm ± 10mm 1000~4000 mm ±1% de la medida
Velocidad de escaneo	100 ms/escaneo
Área de escaneo	240°
Resolución angular	0.351°
Datos entregados	683
Interface de comunicación	USB-mini (5pin)

Al usar este láser mediante MATLAB el número de puntos se reduce a 513, lo que corresponde a un área de 180° (Ver Fig.1.29) [9].

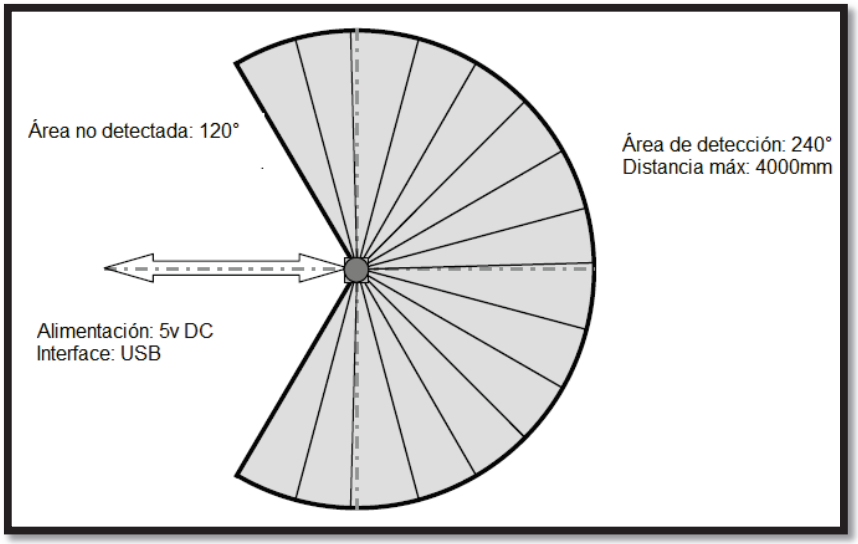


Fig. 1.29 Rango de funcionamiento del láser, tomado de [9].

1.10.2.1 Principio de funcionamiento de los sensores láser

El principio de funcionamiento del proceso de medida de distancia entre dos puntos mediante láser consiste en el cálculo de la diferencia de fase de dos ondas de igual frecuencia, una directa y otra reflejada, a este proceso se lo conoce como interferometría láser [10].

La interferometría láser consiste en generar un haz de luz de una longitud de onda conocida, a esta onda se la divide en dos partes ortogonales mediante un separador. Una de las ondas divididas se la aplica sobre un espejo plano fijo mientras que la otra se la aplica sobre el objeto que se desea conocer la distancia. Una vez que son reflejadas las ondas, se las une nuevamente en el separador, en este proceso se generan máximos y mínimos a cada múltiplo de la longitud de onda del haz inicial. Con esta información se puede determinar la distancia entre dos puntos contando dichas oscilaciones, se obtiene una medida estable con mínima influencia del color y reflectancia del objeto [10].

1.11 SOFTWARE DE ROBOTINO®

El sistema operativo del Robotino® permite ser programado y controlado a través del programa llamado RobotinoView®, el cual es propio del fabricante, pero gracias a la versatilidad de compatibilidad que brinda el sistema operativo, éste también puede ser manejado a través de otros software como son: C, C++, Java, .NET, MATLAB, Simulink, Labview y Microsoft Robotics Developer Studio [4]. Se escogió usar MATLAB para el desarrollo de todos los algoritmos de control y de manejo de información del Robotino®.

La mayor prestación que brinda MATLAB es la manipulación de matrices, adicionalmente ofrece representaciones de datos, funciones, interfaces de usuarios y comunicación con otros programas en otros lenguajes [4].

Además MATLAB ofrece el manejo de herramientas adicionales como son Simulink y GUI que son programas adicionales que complementan el gran desarrollo de MATLAB [4].

1.11.1 MANEJO DE ROBOTINO® CON MATLAB

Dada la importancia que tiene MATLAB, se han creado los controladores que permiten establecer la comunicación y operar la unidad por medio del mismo. Para lo cual se debe hacer los siguientes pasos:

1) Sobre la plataforma Windows 7 instalar MATLAB R2011b. Se debe tomar en cuenta la versión ya sea ésta de 32 o 64 bits.

2) MATLAB necesita un compilador para usar los drivers, normalmente se usa el Microsoft Visual Studio C/C++, para este trabajo se ha usado Microsoft Visual Studio® 2010 Ultimate.

3) Después de instalar el compilador, habrá que dirigirse a la web: openrobotino.org para descargar los archivos que se necesitarán, para lo cual se debe visitar la página web (http://wiki.openrobotino.org/index.php?title=Downloads#Matlab), donde se puede elegir:

“OpenRobotino API” según sea la versión del compilador, y el paquete de versión más actualizada de MATLAB para Robotino® que se encuentra en la misma página. [4].

4) Se instalará primero “OpenRobotino API” seguido de los drivers Robotino® para MATLAB, siguiendo las instrucciones de instalación de los mismos [4].

Después de terminados estos pasos se abre MATLAB y se ejecutan los comandos:

```
>>addpath(strcat( getenv('ROBOTINOMATLAB_DIR'), '/blockset' ) );
>>addpath(strcat( getenv('ROBOTINOMATLAB_DIR'), '/toolbox' ) );
>>startup;
```

Cabe mencionar que cada vez que se cierre MATLAB se tiene que ejecutar una sola vez los comandos mencionados.

1.11.2 MATLAB PARA ADQUISICIÓN DE DATOS DEL ENTORNO

Los datos del entorno serán los que provengan desde el láser Hokuyo URG-04LX-UG01, para esto se tiene los correspondientes toolboxes.

Para configurar el láser se debe crear el objeto, para esto se colocan los siguientes comandos:

```
LaserRangeFinderId = LaserRangeFinder_construct;
LaserRangeFinder_setComId(LaserRangeFinderId, ComId);
```

De esta forma se configura el láser. Para la obtención de datos se usa el siguiente comando:

```
[success, seq, stamp, angle_min, angle_max, angle_increment, time_increment, scan_time, range_min, range_max, ranges, numRanges, intensities, numIntensities ] = LaserRangeFinder_getReadings(LaserRangeFinderId );
```

Con lo que se adquiere todos los parámetros provenientes del láser. De éstos el parámetro que se utilizará para realizar la gráfica de los puntos detectados es el denominado ranges.

Es recomendable destruir el objeto del láser una vez que se termine el programa, para esto se usa el comando:

```
LaserRangeFinder_destroy(LaserRangeFinderId);
```

1.11.2.1 MATLAB para localización con odometría

Para obtener datos de odometría se debe configurar el comando:

```
OdometryId = Odometry_construct;
```

Para leer los datos de la posición a partir del sistema de odometría se utiliza:

```
[ a, b, phi ] = Odometry_get(OdometryId );
```

Donde el valor de la variable a corresponde a la posición del Robotino® en la posición X y la variable b corresponde a la posición sobre el eje Y y la variable *phi* al ángulo en el que se encuentra el Robotino® respecto al eje de coordenadas globales. Para la destrucción del objeto creado se debe utilizar el comando:

```
Odometry_destroy(OdometryId );
```

1.11.2.2 MATLAB para localización con NorthStar (NS2)

La configuración de NS2, mediante comandos de MATLAB son los siguientes:

```
roomId=6;
ceilingCal=1700;
NorthStar_setRoomId(NorthStarId, roomId );
NorthStar_setCeilingCal(NorthStarId, ceilingCal );
NorthStar_roomId(NorthStarId );
NorthStar_sequenceNo(NorthStarId );
posTheta=0;posX=0; posY=0;
```

El valor de *roomId* corresponde a un frecuencia de 60 Hz del proyector con lo que el switch debe colocarse en 1 para unas frecuencias de SpotA 2070 y SpotB 3150.

El valor de *ceilingCal* se colocó en 1700 debido a que en las pruebas realizadas.

Se toman los datos del detector mediante los comandos:

```
posTheta = NorthStar_posTheta( NorthStarId )*180/pi;
posX   = NorthStar_posX( NorthStarId );
posY = NorthStar_posY( NorthStarId );
```

Al finalizar el programa se debe destruir los objetos creados en el inicio, para esto se utiliza el siguiente comando.

```
NorthStar_destroy(NorthStarId );
```

CAPÍTULO 2
LOCALIZACIÓN

2.1 INTRODUCCIÓN

En esta sección del proyecto se tiene como objetivo aclarar de una manera óptima los conceptos necesarios para el desarrollo del mismo, se debe tener en cuenta que los conceptos son extensos porque tienen varios campos de aplicación, concretamente todos los sistemas lineales y no lineales, para esto se hará un resumen de los conceptos más importantes.

Todos los sistemas relacionados a robots móviles se pueden expresar matemáticamente en base a sus aspectos físicos. No es la excepción el Robotino®, el cual obedece a un modelo omnidireccional, el mismo que se describirá a detalle.

La base de este proyecto es la aplicación del filtro de Kalman en su modo extendido (EKF), el cual fue desarrollado por Rudolf E. Kalman en 1960 y se le denomina un filtro óptimo recursivo. A lo largo del capítulo se analizará lo concerniente al EKF. Para el desarrollo de SLAM se debe considerar qué tipo de mapa se desea realizar con la información de los sensores, además qué ecuaciones gobiernan el movimiento del robot móvil.

2.2 MÉTODOS DE LOCALIZACIÓN

En esta sección se discutirán algunos de los métodos de localización más representativos propuestos en la literatura. Actualmente es posible decir que, si bien muchos problemas asociados a la localización han sido eficientemente solucionados, aún quedan caminos de estudios no transitados, tales como la localización en 3D en ambientes dinámicos o la localización de sistemas multi-agentes sin formación.

A continuación, se hará una breve introducción a los dos métodos más comunes usados en localización: Localización mediante el Filtro de Kalman y la Localización basada en cadenas de Markov [6].

2.2.1 LOCALIZACIÓN POR FILTRO DE KALMAN

La localización se puede ver como un problema de filtrado, en el cual el mapa consiste en $\{m1, m2,..., mk\}$ ubicaciones de objetos del entorno (características). El vector de estado es el vector de la ubicación del robot (también conocido como *pose* del robot, donde la pose indica la posición del robot en el mapa y su orientación respecto al eje de las abscisas globales). El vector de observación consiste en las ubicaciones de las características en el marco de coordenadas locales al robot. Si las características observadas pueden ser identificadas inequívocamente respecto a las almacenadas en el mapa, dada la pose actual del robot, entonces la solución al problema de localización del robot móvil en el marco de coordenadas global, es directa. Pero si la asociación de datos es incorrecta se hace necesario adoptar técnicas de seguimiento de hipótesis múltiples para vencer la mono-modalidad del Filtro de Kalman. La Fig. 2.1 muestra un esquema de la arquitectura de localización basada en el Filtro de Kalman [6].

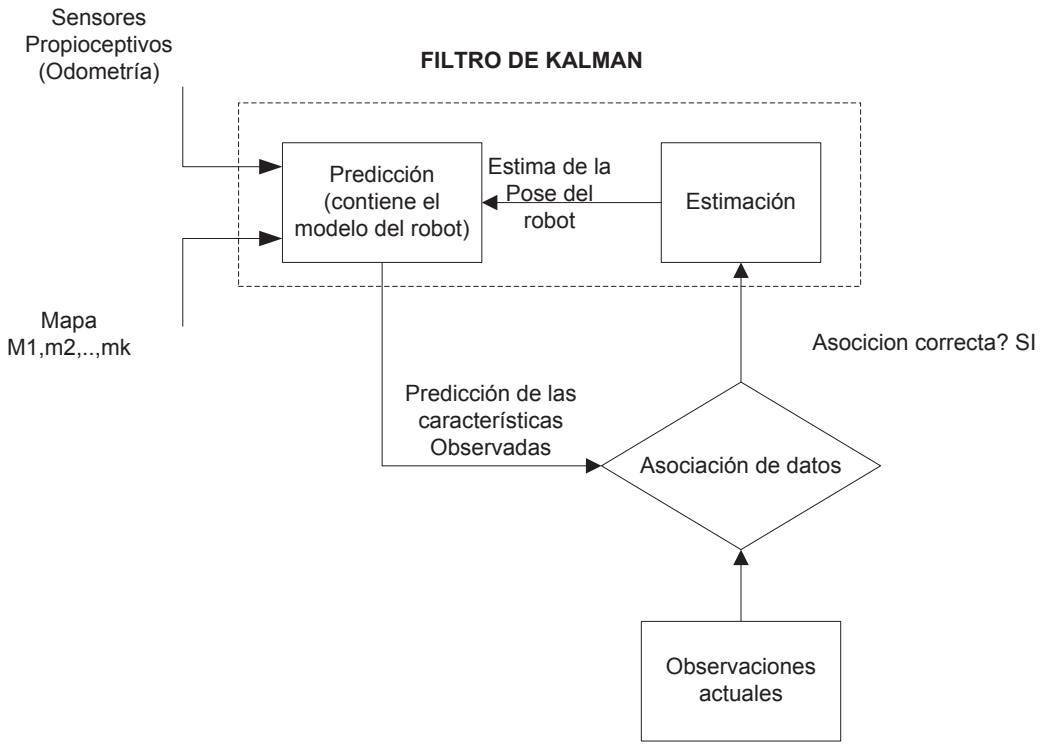


Fig. 2.1 Arquitectura del Filtro de Kalman usado en la localización, tomado de [6].

2.2.2 LOCALIZACIÓN MEDIANTE CADENAS DE MARKOV

La localización basada en Markov (Cadenas de Markov), es uno de los primeros métodos propuestos para solucionar el problema de localización global del robot [16]. Una cadena de Markov es una sucesión de ensayos similares u observaciones en la cual cada ensayo tiene el mismo número finito de resultados posibles y en donde la probabilidad de cada resultado para un ensayo dado depende sólo del resultado del ensayo inmediatamente precedente y no de cualquier resultado previo [23].

Las Cadenas de Markov funcionan en dos etapas, tal como lo hace un filtro recursivo la etapa de predicción y la etapa de corrección basada en la observación.

A pesar de la simplicidad del método de Cadenas de Markov para la localización, éste no deja de ser uno de los métodos más robustos para solucionar el problema de localizar un robot dentro de un entorno global. Entre los principales inconvenientes que presenta este algoritmo se pueden citar: el elevado coste computacional y su poca confiabilidad para operar en entornos dinámicos [16].

2.2.3 MÉTODOS ALTERNATIVOS

Es necesario resaltar que existen otras técnicas aplicadas a la localización de robots móviles, tales como la localización basada en el Método de Monte Carlo y en Filtros de Partículas. La localización basada en el Método de Monte Carlo tiende a ser menos robusta que la localización basada en Markov pero con menos coste computacional, lo que la hace más factible de ser implementada en sistemas de tiempo real.

El Filtro de Partículas de Rao-Blackweillzed, a su vez, presenta una mejora respecto al Filtro de Kalman, pues, a diferencia del Filtro de Kalman, no es necesario que las perturbaciones del sistema sean estrictamente gaussianas. Por el contrario, las perturbaciones de todo el sistema pueden tener cualquier distribución de probabilidad. El Filtro de Partículas es el algoritmo más general a ser implementado en la localización de un robot móvil [6].

2.3 MODELO OMNIDIRECCIONAL

De los robots móviles una clase especial son los omnidireccionales. Este tipo de robots son diseñados para moverse en dos dimensiones, siendo capaces de trasladarse en direcciones x y y, además puede girar sobre su centro de gravedad en un ángulo θ. Este tipo de robots tiene tres grados de libertad. El robot omnidireccional es capaz de controlar los tres grados de libertad que posee de manera independiente [11].

Para que el robot móvil cumpla con una operación efectiva debe poseer un seguimiento de su posición actual, para cumplir con este objetivo se vuelve primordial la implementación de un algoritmo, el cual requerirá el modelo del robot [11].

2.3.1 MOVILIDAD OMNIDIRECCIONAL

Las restricciones no holonómicas consisten en no poder realizar movimientos de forma independiente. Un robot móvil al no tener restricciones no holonómicas tiene la capacidad de poder dirigirse a cualquier dirección en cualquier orientación, teniendo una gran ventaja para poder desplazarse en espacios reducidos, la complejidad en el control disminuye así como la precisión de los movimientos es mayor [11].

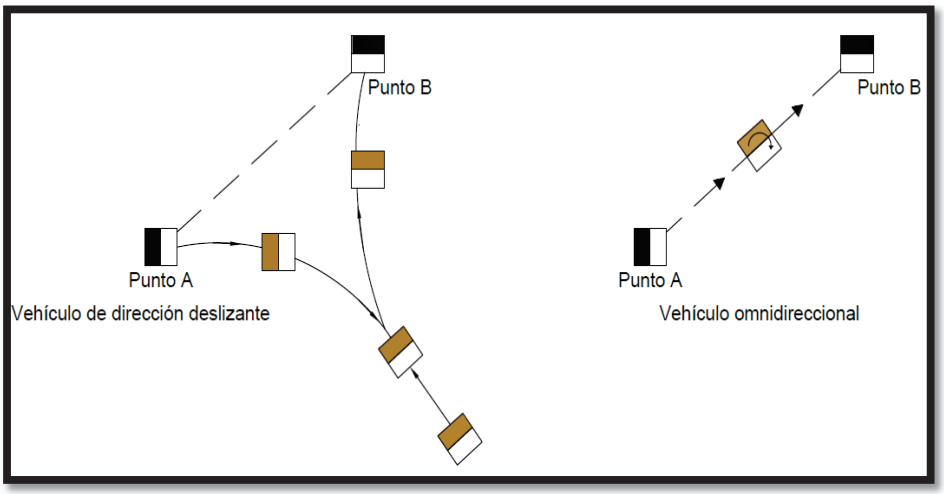


Fig. 2.2 Cinemática del robot omnidireccional, tomado de [11].

Es necesario encontrar una relación entre las variables que pueden ser controladas, como son las posiciones y velocidades angulares de las ruedas, con las variables que describen el movimiento del robot en el entorno, para cumplir con esta meta es necesario obtener el modelo del robot omnidireccional [11].

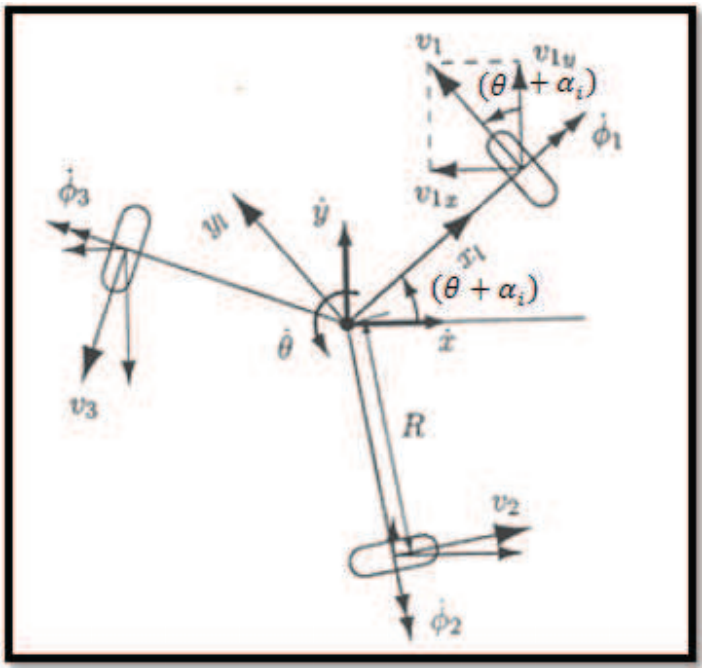


Fig. 2.3 Diagrama de la cinemática del robot, tomado de [11].

El primer paso a seguir para encontrar el modelo del robot omnidireccional es definir el eje de referencia global (x, y), el cual representa el ambiente del robot móvil. Se representa como (x, y, θ) a la posición y orientación global del robot y como $(\dot{x}, \dot{y}, \dot{\theta})$ a la velocidad del robot [11].

Se define como (x_L, y_L) al marco o eje de referencia local o relativa, que coincide con el centro de gravedad del robot móvil. Se colocan de forma simétrica y separadas por un ángulo de 120° entre cada una, relativo al eje de referencia local.

La velocidad global $(\dot{x}, \dot{y}, \dot{\theta})$ del robot móvil en el ambiente es determinada por las velocidades de traslación de las ruedas v_i. Se puede dividir en dos partes a la velocidad traslacional de la rueda v_i, una traslacional pura y una rotacional pura [11].

$$v_i = v_{trans,i} + v_{rot} \tag{2.1}$$

Se analizará primero a la velocidad traslacional pura [11].

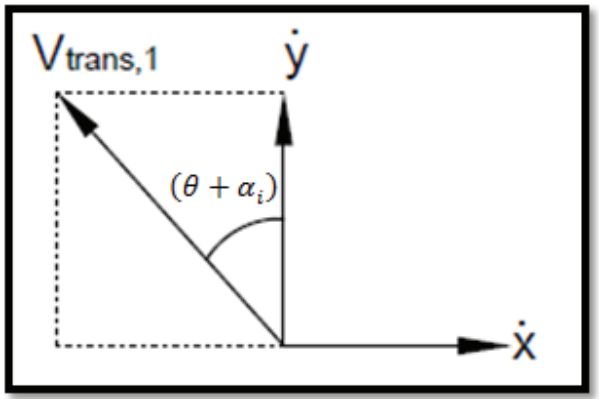


Fig. 2.4 Modelo de velocidad, tomado de [11].

La velocidad de traslación pura se define en función de las velocidades $\dot{x}, \dot{y}$. [11].

$$v_{trans,1} = -sen(\theta) * \dot{x} + \cos(\theta) * \dot{y} \tag{2.2}$$

Se generaliza:

$$v_{trans,1} = -sen(\theta + \alpha_i) * \dot{x} + \cos(\theta + \alpha_i) * \dot{y} \tag{2.3}$$

Ecuación para rotación pura:

$$v_{rot} = R * \dot{\theta} \tag{2.4}$$

Donde, R es la distancia desde el centro de gravedad del robot al punto donde se encuentran las ruedas a lo largo de una trayectoria radial. El valor de R es 12.5 cm [11].

Substituyendo (2.3) y (2.4) en (2.1) se obtiene:

$$v_i = -sen(\theta + \alpha_i) * \dot{x} + \cos(\theta + \alpha_i) * \dot{y} + R * \dot{\theta} \tag{2.5}$$

Al haber unido las velocidades de rotación y de traslación del robot móvil se proceden a relacionar la velocidad traslacional del eje con la velocidad angular de las ruedas de la forma siguiente:

$$v_i = r * \dot{\phi}_i \tag{2.6}$$

Donde r es el radio de las ruedas omnidireccionales: $r = 4$ cm.

Reemplazando (2.5) y (2.6) y reordenando se obtiene:

$$\dot{\emptyset}_i = -sen(\theta + \alpha_i) * \dot{x} + \cos(\dot{\theta} + \alpha_\iota) * \dot{y} + R * \dot{\theta} \quad (2.7)$$

Se puede transformar (2.7) a (2.8) y como matriz a (2.9)

$$\dot{\emptyset}_i = J_{inv} * \dot{u} \quad (2.8)$$

En la relación (2.8) J_{inv} es el jacobiano inverso que permite tener una relación directa entre las velocidades angulares de las ruedas $\dot{\emptyset}_i$y la velocidad global $\dot{u}$[11].

$$\begin{bmatrix} \dot{\emptyset}_1 \\ \dot{\emptyset}_2 \\ \dot{\emptyset}_3 \end{bmatrix} = \frac{1}{r} \begin{bmatrix} -sen(\theta + \alpha_1) & \cos(\theta + \alpha_1) & R \\ -sen(\theta + \alpha_2) & \cos(\theta + \alpha_2) & R \\ -sen(\theta + \alpha_3) & \cos(\theta + \alpha_3) & R \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} \quad (2.9)$$

Donde $\alpha_1 = \frac{\pi}{3}$; $\alpha_2 = 0$; $\alpha_3 = -\frac{\pi}{3}$, de acuerdo a la disposición de las ruedas en la plataforma móvil.

2.4 FILTRO DE KALMAN

A partir del filtro de Kalman se puede obtener un buen marco para la estimación de una variable de la que se tiene datos a lo largo del tiempo. Consiste en una técnica de estimación Bayesiana usada para seguir sistemas estocásticos dinámicos adquiridos por intermedio de sensores ruidosos [14].

El filtro de Kalman tiene como objetivo obtener un estimador óptimo de las variables de estado de un sistema dinámico, que está basado en observaciones ruidosas dentro de un modelo de incertidumbre de la dinámica del sistema [14].

2.4.1 Características del filtro de Kalman

Consiste en un algoritmo recursivo óptimo de procesamiento de datos:

- El filtro de Kalman considera toda la información existente.

- Para poder estimar el valor de las variables de interés procesa todas las mediciones sin importarle la precisión que éstas tengan [15].
- Utiliza la información existente acerca de las condiciones iniciales de las variables de interés [15].

2.4.2 Filtro recursivo

- Es recursivo porque no necesita almacenar todos los datos anteriores y volverlos a procesar cuando se tenga nueva información.
- Es un programa que no toma muestras en tiempo continuo sino en tiempo discreto, en este sentido se le considera un filtro [15].

¿Por qué se necesita al Filtro de Kalman?

- Debido a que en los sistemas existen más variables que las variables de control.
- Las variables de estado y las variables medidas se conocen con un grado de incertidumbre.
- Las mediciones son afectadas por ruido, desviaciones y la imprecisión de los instrumentos.
- Gracias al filtro de Kalman se produce una estimación óptima estadísticamente.
- Se considera todas las mediciones además de la información a priori que se tenga del sistema y de los instrumentos de medida [15].

2.4.3 Enfoque Bayesiano

- El filtro de Kalman debe propagar la densidad de probabilidad condicional de las cantidades deseadas, dependiendo del conocimiento de los datos obtenidos a partir de los instrumentos de medida.

$$f_{x(i)|z(1),z(2),z(2),\dots,z(i)}(x|z_1, z_2, \dots, z_i) \tag{2.10}$$

2.4.4 Restricciones del filtro de Kalman

- El filtro de Kalman puede realizar la propagación de la densidad de probabilidad para problemas donde el sistema es descrito a través de un modelo lineal y los ruidos en la medición son del tipo gaussiano [15].

2.4.5 Presunciones básicas

- En el caso de que existan no linealidades el modelo se puede aproximar a un modelo lineal.
- Es más conveniente trabajar con modelos lineales ya que existe teoría más completa.
- El ruido blanco consiste en que no se tiene una relación con el tiempo, ya que el conocimiento de éste no ayuda a predecirlo [15].

El método estima el estado x perteneciente a $\Re^n$de un proceso controlado en tiempo discreto que es gobernado por la ecuación diferencial del tipo:

$$X_k = Ax_{k-1} + Bu_k + w_{k-1} \tag{2.11}$$

con una medida z correspondiente a la observación y perteneciente a $\Re^m$que es:

$$z_k = Hx_k + v_k \tag{2.12}$$

Las variables aleatorias w_ky $v_k$representan el ruido del proceso y de la medida, respectivamente y se asume que son independientes y blancos [16].

2.4.6 Algoritmo predicción más corrección

Ecuaciones de predicción:

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_k \tag{2.13}$$

$$P_k^- = AP_{k-1}A^T + Q \tag{2.14}$$

Ecuaciones de corrección:

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1} \tag{2.15}$$

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) \quad (2.16)$$

$$P_k = (I - K_k H)P_k^- \quad (2.17)$$

2.4.7 Selección de parámetros:

- Estimas iniciales $\hat{x}_0$ y P_0: no demasiado críticas.
- Matrices de covarianzas de ruido R y Q.
- R se puede estimar a través de las medidas.
- Q es difícil de estimar (no hay acceso al estado).
- Habitualmente se realiza un ajuste (tunning) de dichos parámetros.

2.5 FILTRO EXTENDIDO DE KALMAN (EKF)

2.5.1 Definiciones

Para resumir los términos usados en el EKF se presenta la Tabla 2.1 con las variables que están involucradas en el desarrollo del algoritmo de este proyecto:

Tabla 2.1: Variables de EKF, tomado de [6].

Variable	Dimensión	Descripción
$\hat{x}_k^-$	$n\ x\ 1$	Estado predicho.
$f()$		Modelo del movimiento.
A	nxn	Matriz jacobiana de derivadas de f con respecto a x.
u_k	$lx1$	Vector de señales de control.
P_k^-	nxn	Matriz de covarianzas del estado predicho.
Q	nxn	Matriz de covarianzas del ruido del proceso.
W	nxn	Matriz jacobiana de derivadas de f respecto al ruido.
K_k	nxm	Matriz de ganancia de Kalman.
$h()$		Modelo de medición.
R	mxm	Matriz de covarianzas del ruido en la medida.
V	mxm	Matriz jacobiana de derivadas de h con respecto al ruido.
$\hat{x}_k$	$nx1$	Vector de estados estimado.
z_k	$mx1$	Vector de medida en estado k.
H	mxn	Matriz jacobiana de derivadas de h con respecto a x.
P_k	nxn	Matriz de covarianzas del vector de estado en el instante k.
I	nxn	Matriz identidad.
T_m		Tiempo de muestreo

Las ecuaciones del proceso y/o de medida son no lineales [16].

$$x_k = a(x_{k-1}, u_k) + w_{k-1} \tag{2.18}$$

$$z_k = h(x_k) + v_k, \tag{2.19}$$

El Filtro Extendido Kalman linealiza el sistema.

Ecuación de proceso:

$$A_k = \frac{\partial a}{\partial x}(\hat{x}_{k-1}, u_k) \tag{2.20}$$

Ecuación de medida:

$$H_k = \frac{\partial h}{\partial x}(\hat{x}_k^-) \tag{2.21}$$

Ecuación de predicción:

$$\hat{x}_k^- = \alpha(\hat{x}_{k-1}, u_k) \tag{2.22}$$

$$P_k = A_k P_{k-1} A_k^T + Q \tag{2.23}$$

Ecuaciones de corrección:

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R)^{-1} \tag{2.24}$$

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - h(\hat{x}_k^-)) \tag{2.25}$$

$$P_k = (I - K_k H_k) P_k^- \tag{2.26}$$

2.5.2 Formulación alternativa del EKF

Las ecuaciones del proceso y/o de medida son no lineales [16].

$$\hat{x}_k^- = \alpha(\hat{x}_{k-1}, u_k, 0) \tag{2.27}$$

$$z_k = h(x_k, v_k) \tag{2.28}$$

El filtro de Kalman extendido linealiza el sistema.
Ecuación del proceso:

$$A_{x,k} = \frac{\partial a}{\partial x}(\hat{x}_{k-1}, u_k, 0), \tag{2.29}$$

$$A_{u,k} = \frac{\partial a}{\partial u}(\hat{x}_{k-1}, u_k, 0), \tag{2.30}$$

Para poder aplicar el EKF es necesario linealizar el modelo, para lo que se realiza las derivadas parciales respecto a las variables de posición y a las variables de velocidad angular, con lo que se obtiene dos matrices correspondientes a los jacobianos $A_{x,k}$ y $A_{u,k}$.

$$A_{x,k} = \begin{bmatrix} \frac{\partial a_1}{\partial x_1} & \frac{\partial a_1}{\partial x_2} & \frac{\partial a_1}{\partial x_3} \\ \frac{\partial a_2}{\partial x_1} & \frac{\partial a_2}{\partial x_2} & \frac{\partial a_3}{\partial x_3} \\ \frac{\partial a_3}{\partial x_1} & \frac{\partial a_3}{\partial x_2} & \frac{\partial a_3}{\partial x_3} \end{bmatrix} \tag{2.31}$$

$$A_{u,k} = \begin{bmatrix} \frac{\partial a_1}{\partial u_1} & \frac{\partial a_1}{\partial u_2} & \frac{\partial a_1}{\partial u_3} \\ \frac{\partial a_2}{\partial u_1} & \frac{\partial a_2}{\partial u_2} & \frac{\partial a_3}{\partial u_3} \\ \frac{\partial a_3}{\partial u_1} & \frac{\partial a_3}{\partial u_2} & \frac{\partial a_3}{\partial u_3} \end{bmatrix} \tag{2.32}$$

$$A_{x,k} = \begin{bmatrix} 1 & 0 & \frac{1}{225} * \left(2 * \sqrt{3} * T_m * w_2 * \left(\cos\left(\frac{\pi}{6}+\theta\right) + sen\left(\frac{\pi}{3}+\theta\right)\right)\right) - \frac{1}{225} * \left(2 * \sqrt{3} * T_m * w_1 * \left(\cos\left(\frac{\pi}{6}+\theta\right) - sen(\theta)\right)\right) - \frac{1}{225} * \left(2 * \sqrt{3} * T_m * w_3 * \left(sen(\theta) + sen\left(\frac{\pi}{3}+\theta\right)\right)\right) \\ 0 & 1 & \frac{1}{225} * \left(2 * \sqrt{3} * T_m * w_3 * \left(\cos(\theta) + cos\left(\frac{\pi}{3}+\theta\right)\right)\right) - \frac{1}{225} * \left(2 * \sqrt{3} * T_m * w_2 * \left(\cos\left(\frac{\pi}{3}+\theta\right) - sen\left(\frac{\pi}{6}+\theta\right)\right)\right) - \frac{1}{225} * \left(2 * \sqrt{3} * T_m * w_1 * \left(cos(\theta) + sen\left(\frac{\pi}{6}+\theta\right)\right)\right) \\ 0 & 0 & 1 \end{bmatrix} \tag{2.33}$$

$$A_{u,k} = \begin{bmatrix} -\frac{1}{225} * \left(2 * \sqrt{3} * T_m * \left(\cos(\theta) + sen\left(\frac{\pi}{6}+\theta\right)\right)\right) & -\frac{1}{225} * \left(s * \sqrt{3} * T_m * \left(\cos\left(\frac{\pi}{3}+\theta\right) - sen\left(\frac{\pi}{6}+\theta\right)\right)\right) & \frac{1}{225} * \left(2 * \sqrt{3} * T_m * \left(\cos(\theta) + \cos\left(\frac{\pi}{3}+\theta\right)\right)\right) \\ \frac{1}{225} * \left(2 * \sqrt{3} * T_m * \left(\cos\left(\frac{\pi}{6}+\theta\right) - sen(\theta)\right)\right) & -\frac{1}{225} * \left(2 * \sqrt{3} * T_m b * \cos\left(\frac{\pi}{6}+\theta\right) + sen\left(\frac{\pi}{3}+\theta\right)\right) & \frac{1}{225} * \left(2 * \sqrt{3} * T_m * \left(sen(\theta) + sen\left(\frac{\pi}{3}+\theta\right)\right)\right) \\ 0 & 0 & 1 \end{bmatrix} \tag{2.34}$$

Con las matrices de jacobianos $A_{x,k}$ y $A_{u,k}$, se aplica (2.36) con el objetivo de hallar la matriz de covarianzas del estado predicho. Es totalmente válido el uso de (2.36) ya que se toma en cuenta las perturbaciones en la entrada de control.

Ecuaciones de predicción:

$$\hat{x}_k^- = a(\hat{x}_{k-1}, u_k, 0) \tag{2.35}$$

$$P_k^- = A_{x,k} P_{x-1} A_{x,k}^T + A_{u,k} Q A_{u,k}^T \tag{2.36}$$

De acuerdo a (2.35) se halla la matriz del estado predicho como se muestra a continuación:

A partir de la matriz J_{inv} se halla su inversa para poder obtener los valores de posición predicha en función de las velocidades angulares.

$$J_{inv} = \begin{bmatrix} -sen\left(\theta + \frac{\pi}{3}\right) & \cos\left(\theta + \frac{\pi}{3}\right) & R \\ -sen(\theta) & \cos(\theta) & R \\ -sen\left(\theta - \frac{\pi}{3}\right) & \cos\left(\theta - \frac{\pi}{3}\right) & R \end{bmatrix} \tag{2.37}$$

Se halla la matriz inversa:

$$J = \begin{bmatrix} -sen\left(\theta + \frac{\pi}{3}\right) & \cos\left(\theta + \frac{\pi}{3}\right) & 12.5 \\ -sen(\theta) & \cos(\theta) & 12.5 \\ -sen\left(\theta - \frac{\pi}{3}\right) & \cos\left(\theta - \frac{\pi}{3}\right) & 12.5 \end{bmatrix}^{-1} \tag{2.38}$$

$$J = \begin{bmatrix} -\frac{1}{9} * \left(2 * \sqrt{3} * \left(\cos(\theta) + sen\left(\frac{\pi}{6} + \theta\right)\right)\right) & -\frac{1}{9} * \left(2 * \sqrt{3} * \left(\cos\left(\frac{\pi}{3} + \theta\right) - sen\left(\frac{\pi}{6} + \theta\right)\right)\right) & \frac{1}{9} * \left(2 * \sqrt{3} * (\cos(\theta) + sen\left(\frac{\pi}{6} + \theta\right))\right) \\ \frac{1}{9} * \left(2 * \sqrt{3} * \left(\cos\left(\frac{\pi}{6} + \theta\right) - sen(\theta)\right)\right) & -\frac{1}{9} * \left(2 * \sqrt{3} * \left(\cos\left(\frac{\pi}{6} + \theta\right) + sen\left(\frac{\pi}{3} + \theta\right)\right)\right) & \frac{1}{9} * \left(2 * \sqrt{3} * \left(sen(\theta) + \text{sen}\left(\frac{\pi}{3} + \theta\right)\right)\right) \\ \frac{2}{75} & \frac{2}{75} & \frac{2}{75} \end{bmatrix} \tag{2.39}$$

Una vez que se obtiene la matriz inversa de J_{inv} es necesario multiplicar por el vector de las variables de control, es decir las velocidades angulares de cada rueda del robot móvil, también se multiplica por el intervalo de tiempo recorrido y por el radio de cada rueda:

$$u_k = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} \tag{2.40}$$

$$\hat{x}\_p_k^- = r * T_m * J * u_k \tag{2.41}$$

$$\hat{x}p_k^- = \begin{bmatrix} \frac{1}{225} * \left(\left(2 * \sqrt{3} * T_m * w_3 * \left(\cos(\theta) + \cos\left(\frac{\pi}{3} + \theta\right)\right)\right) - 2 * \sqrt{3} * T_m * w_2 * \left(\cos\left(\frac{\pi}{3} + \theta\right) - sen\left(\frac{\pi}{6} + \theta\right)\right) - \left(2 * \sqrt{3} * T_m * w_1 \left(\cos(\theta) + sen\left(\frac{\pi}{6} + \theta\right)\right)\right)\right) \\ \frac{1}{225} * \left(\left(2 * \sqrt{3} * T_m * w_1 * \left(\cos\left(\frac{\pi}{6} + \theta\right) - \text{sen}(\theta)\right)\right) - 2 * \sqrt{3} * T_m * w_2 * \left(\cos\left(\frac{\pi}{6} + \theta\right) + sen\left(\frac{\pi}{3} + \theta\right)\right) + \left(2 * \sqrt{3} * T_m * w_3 \left(sen(\theta) + sen\left(\frac{\pi}{3} + \theta\right)\right)\right)\right) \\ \frac{1}{1875} * 2 * T_m * (w_1 + w_2 + w_3) \end{bmatrix} \tag{2.42}$$

Una vez que se tiene la matriz $\hat{x}p_k^-$ se debe tomar en cuenta el valor del estado anterior de modo que se tenga una posición absoluta.

$$\hat{x}_k^- = \hat{x}_{k-1}^- + \hat{x}p_k^- \tag{2.43}$$

$$\hat{x}_k^- = \begin{bmatrix} x_{K-1} + \frac{1}{225} * \left(\left(2 * \sqrt{3} * T_m * w_3 * \left(\cos(\theta) + \cos\left(\frac{\pi}{3} + \theta\right) \right) \right) - 2 * \sqrt{3} * T_m * w_2 * \left(\cos\left(\frac{\pi}{3} + \theta\right) - sen\left(\frac{\pi}{6} + \theta\right) \right) - \left(2 * \sqrt{3} * T_m * w_1 \left(\cos(\theta) + sen\left(\frac{\pi}{6} + \theta\right) \right) \right) \right) \\ y + \frac{1}{225} * \left(\left(2 * \sqrt{3} * T_m * w_1 * \left(\cos\left(\frac{\pi}{6} + \theta\right) - \text{sen}(\theta) \right) \right) - 2 * \sqrt{3} * T_m * w_2 * \left(\cos\left(\frac{\pi}{6} + \theta\right) + sen\left(\frac{\pi}{3} + \theta\right) \right) + \left(2 * \sqrt{3} * T_m * w_3 \left(sen(\theta) + sen\left(\frac{\pi}{3} + \theta\right) \right) \right) \right) \\ \theta + \frac{1}{1875} * 2 * T_m * (w_1 + w_2 + w_3) \end{bmatrix} \tag{2.44}$$

De acuerdo a (2.44) se tiene el vector de posición predicha mediante el uso del modelo del robot omnidireccional.

Ecuaciones de corrección:

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R)^{-1} \tag{2.45}$$

$$\hat{x}_k = \hat{x}_k^- + K_k (z_k - \hat{x}_k^-) \tag{2.46}$$

$$P_k = (I - K_k H_k) P_k^- \tag{2.47}$$

Donde K_k es la matriz de ganancia de Kalman, esta matriz es fundamental para darle un peso a la diferencia entre el valor predicho y el valor medido por los sensores del robot móvil.

Finalmente se calcula la matriz de covarianza del vector de estado en función de la matriz de ganancia de Kalman de la matriz de covarianza del vector de estado, y de la matriz del jacobiano de la matriz de modelo de medición con respecto a x.

Los valores de odometría y NorthStar que en este caso conforman el vector de medición z_k se consiguen de forma directa, por lo que la matriz H está conformada por una matriz identidad. Las matrices Q y R se hallaron mediante prueba y error. Donde Q es la matriz de covarianzas asociadas al proceso y R es la matriz de covarianzas asociadas al observador (odometría y NS2).

$$z_k = \begin{bmatrix} x_{od} \\ y_{od} \\ \theta_{ns2} \end{bmatrix}; \quad H_k = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.48}$$

$$Q_k = \begin{bmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_\theta^2 \end{bmatrix}; \quad R_k = \begin{bmatrix} \sigma_{xob}^2 & 0 & 0 \\ 0 & \sigma_{yob}^2 & 0 \\ 0 & 0 & \sigma_{\theta ob}^2 \end{bmatrix} \tag{2.49}$$

Dónde:

$\sigma_x^2, \sigma_y^2, \sigma_\theta^2$: varianzas del proceso sobre X, Y, θ

σ_{xob}^2, $\sigma_{yob}^2, \sigma_{\theta ob}^2$:varianzas del observador, Odometría y NS2.

2.6 IMPLEMENTACIÓN DEL MODELO OMNIDIRECCIONAL

La función Modelo permite obtener los datos de posición predicha, para esto relaciona las velocidades angulares de las ruedas del robot móvil omnidireccional con sus velocidades lineales, otros datos que se deben conocer son la distancias desde el centro de gravedad del robot hasta las ruedas y el radio de cada rueda. Una vez que se tiene la velocidad lineal se le multiplica por el intervalo de tiempo para obtener el desplazamiento.

A continuación se muestra el Diagrama 2.1 que describe la función del modelo implementado.

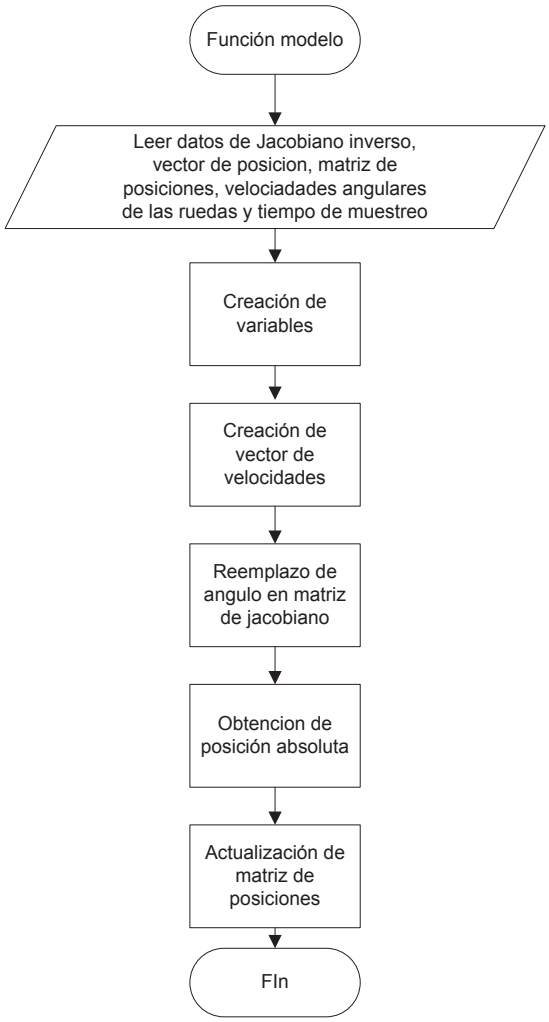


Diagrama 2.1 Implementación del modelo

2.7 IMPLEMENTACIÓN DEL FILTRO EXTENDIDO DE KALMAN

En la función Filtro a partir de los datos de posición que provienen del modelo y los datos obtenidos de la odometría generan mediante las ecuaciones de predicción y de corrección una posición corregida, que será usada para el posicionamiento del robot móvil, graficar su trayectoria y los puntos que provienen del sensor láser en el mapa del entorno.

A continuación se muestra el Diagrama 2.2 que representa la implementación del filtro de Kalman

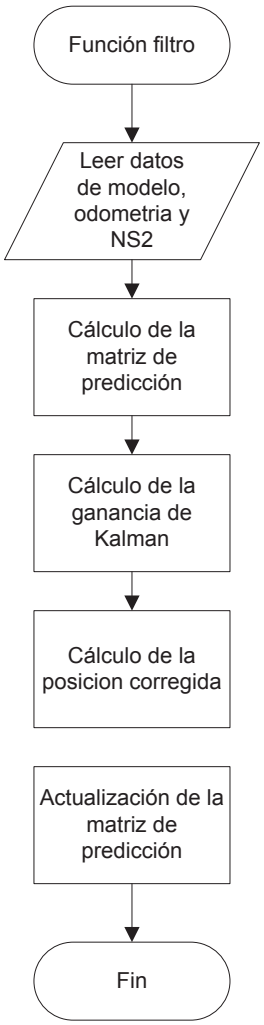


Diagrama 2.2 Implementación del Filtro de Kalman

CAPÍTULO 3
MAPEO DEL ENTORNO

3.1 INTRODUCCIÓN

La localización de un robot móvil es crítica, ya que, por ejemplo si no se conoce con certeza en qué posición se encuentra el robot nunca se podrá saber a qué distancia exacta se encuentra una característica del entorno respecto a éste, lo que puede producir errores graves en el mapa y pérdida total del robot.

Una vez superado el tema de localización, el mapeo cobra vital importancia porque es la representación del entorno tomando en cuenta las características más importantes del mismo para poder establecer que se trate del mismo entorno. Con este objetivo se utiliza un láser, el cual fue descrito en los capítulos anteriores.

En este capítulo se consolidará la parte de la localización con la gran cantidad de datos que entrega el láser para de esta forma reconstruir lo más aproximadamente posible al entorno en el que se está moviendo el robot.

El mapeo es tan importante como la localización, ya que va a depender de la cantidad de datos que se obtenga del entorno para así poder brindar ciertos detalles importantes al mapa final.

3.2 CONSTRUCCIÓN DE MAPAS

La misión de los robots móviles es realizar actividades difíciles en ambientes conocidos parcialmente, o desconocidos con escasa información previa. El nivel de autonomía debe ser elevado con el fin de que el robot pueda navegar evadiendo los obstáculos que se le presenten y poder localizarse por sí mismo. Para ser independiente el robot debe interpretar las señales que provienen de sus sensores, para obtener el modelo del espacio que lo rodea mediante la elaboración de un mapa [6].

Construir un mapa es un proceso dinámico que usa información entregada por sensores externos que corresponden a las características presentes en el entorno.

La información es recibida desde diferentes posiciones en la trayectoria descrita por el robot móvil. Una vez elaborado el mapa este podría ser usado para mejorar los caminos, ubicar objetos de navegación o para planear trayectorias de navegación. El mapa podría ser usado para alertar al robot móvil de cierto cambio en el mapa elaborado previamente, con lo que se podría determinar la nueva característica que se haya presentado y que representa un cambio en el ambiente que rodea al robot móvil [6].

Una persona puede reconocer si ha existido un cambio en el ambiente que lo rodea, por lo que un robot móvil autómata debería comportarse de una manera similar, para esto debe contar con un mecanismo de actualización de características que lo permita, sobre todo si requiere desplazarse en un lugar donde existen otros agentes móviles que no necesariamente son robóticos, por lo que se debe elegir el tipo de mapa más conveniente [6].

Los cuatro tipos de mapa más comunes son los siguientes:

- Reconocimiento de lugares: El mapa está conformado por un conjunto de lugares, los cuales pueden ser reconocidos por el robot móvil. Los lugares no requieren relaciones geométricas entre ellos [6].
- Mapas Topológicos: este tipo de mapa está constituido por grafos, los cuales están conectados entre sí con el fin de representar los lugares pertenecientes al entorno. No se detalla información geométrica entre cada grafo [6].
- Mapas Métricos: las características son representadas sobre ejes de coordenadas fijos del entorno, como de la posición del robot móvil, entre los objetos y características del mapa existe información métrica. Dentro del espacio métrico, el ambiente es representado desde un conjunto de objetos definidos en este ambiente [6].

- Mapas Híbridos: introducen dentro de mapas topológicos información geométrica, para poder tener una idea de la posición de cada lugar conectado dentro el mapa topológico [6].

En la Fig.3.1 se puede observar la clasificación de los mapas explicada anteriormente. Los mapas métricos tiene una subdivisión que son: mapas basados en características y los mapas creados en base a grillas [6].

- Mapas basados en características: este tipo de mapa almacena de manera explícita cualquier tipo de característica ya sea geométrica o no, como puede ser un conjunto de puntos o líneas pertenecientes al entorno, siendo detectadas por el robot al igual que la posición que le corresponde en el ambiente. Estas características pueden ser esquinas paredes, puertas, etc. [6].
- Mapa de Grillas: en lugar de utilizar conjuntos de puntos o líneas con el fin de representar los objetos correspondientes al entorno, es posible crear grillas de ocupación. Para lograrlo se debe discretizar al mapa en una grilla de alta resolución, cada una de las celdas de la grilla es ocupada por un valor de probabilidad de estar o no ocupada por un objeto, el valor es actualizado a través de la regla de Bayes [6].

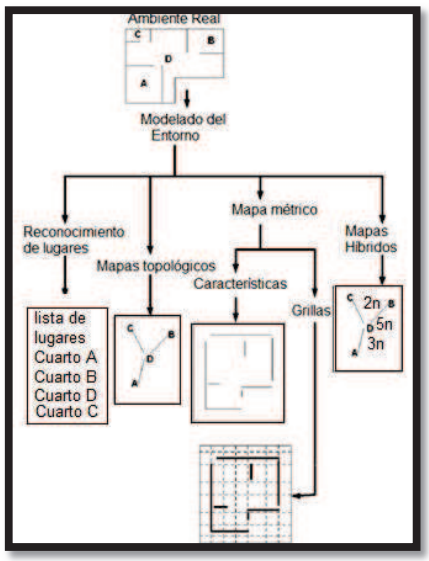


Fig. 3.1 Tipos de mapas, tomado de [6].

3.2.1 Mapas Absolutos

Los mapas absolutos se los puede representar como se muestra en la Fig. 3.2, la posición de las características está referida a un eje de referencia global y fijo [6].

El vector de estados del mapa absoluto es constituido por n características en un instante k. (3.1) representa el vector de estado del mapa absoluto.

$$p_m = \begin{bmatrix} p_1 \\ \vdots \\ p_n \end{bmatrix} \tag{3.1}$$

Un mapa absoluto puede ser local o global. Un mapa global tiene sus ubicaciones y mediciones referidas a un sistema de referencia global. En el mapa se puede encontrar todas las ubicaciones posibles correspondientes a las características del entorno. En cambio en un mapa local se obtiene en un instante de tiempo k refiriéndose al sensor que lo adquirió, siendo éste el origen del centro de coordenadas, un mapa local puede estar dentro de un mapa global, no permanece fijo el origen del sistema de coordenadas a medida que avanza el tiempo [6].

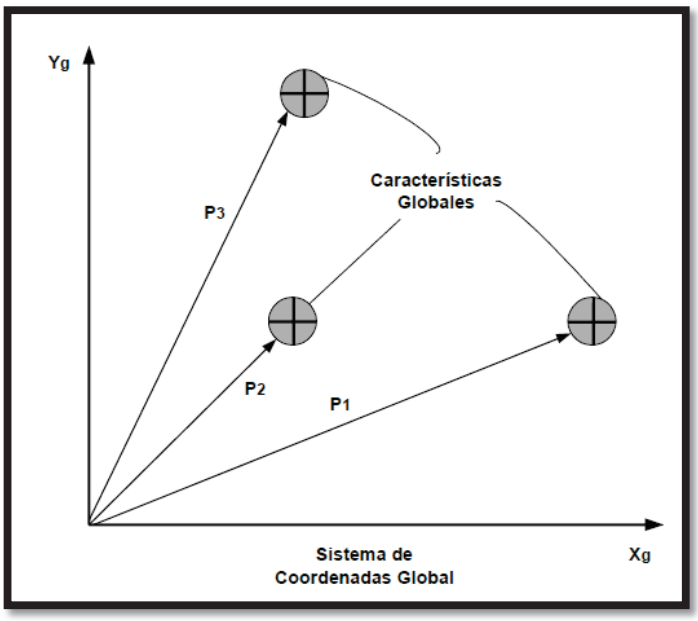


Fig. 3.2 Ejemplo de mapa absoluto, tomado de [6].

3.2.2 Mapas Relativos

El mapa relativo guarda la relación entre cada una de las características que se hayan detectado. Las relaciones conforman lo que es llamado como mapa relativo de estados. El estado relativo entre dos características cualesquiera de un mapa global, *pi* y *pj*, se escribe como:

$$p_{ij} = p_j - p_i \tag{3.2}$$

Si se tratara de una mapa métrico en (3.2), p_iy p_j, corresponden únicamente a sus coordenadas en un sistema de coordenadas común a ambos puntos. En la Fig. 3.3 es posible observar una representación general de estados relativos los cuales poseen un sistema de coordenadas común a las características adquiridas.

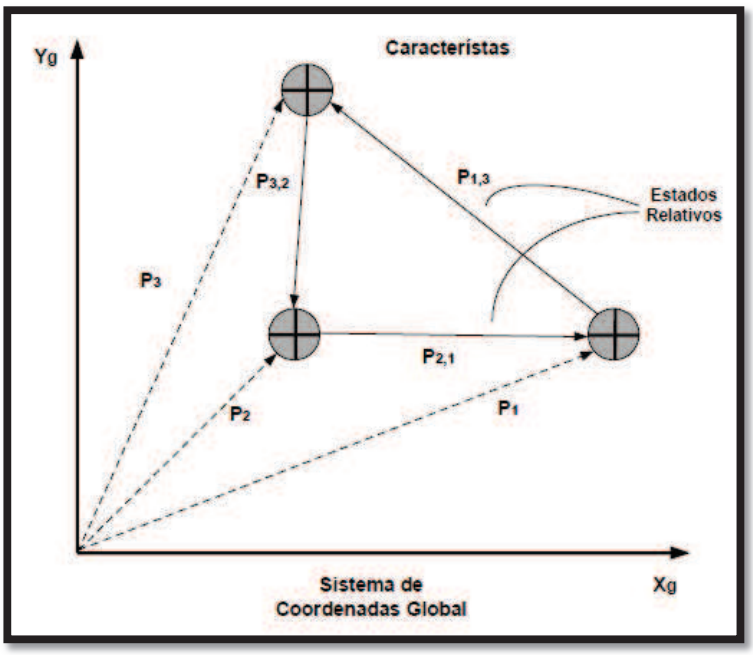


Fig. 3.3 Ejemplo de mapa relativo, tomado de [6].

Para realizar la transformación de un mapa absoluto a un mapa relativo se utilizan matrices de transformación [6].

Mediante la expresión (3.3) es posible describir la transformación de un vector de estados absolutos a un vector de estados relativos [6].

$$p_{relativo} = \Omega p_{absoluto} \tag{3.3}$$

Donde, Ω es la matriz de transformación entre los vectores absolutos y relativos. De la misma forma, es posible transformar de un mapa relativo a absoluto.

Los mapas relativos y absolutos describen la manera en que se consideran y ubican los estados, pero no explican la forma en que los datos son procesados, por lo que se realiza una nueva clasificación en los mapas a ser creados, que depende que estrategia se elija. Los nuevos mapas son:

- Mapas dinámicos y estáticos: están referidos a la posibilidad de que la localización de los objetos o características del mapeo sean variables en el

tiempo. Si se da el caso de que la localización varíe, los mapas varían con el tiempo por lo que son dinámicos, si esto no se da el mapa es estático. Mapas dinámicos y estáticos pueden ser métricos (globales, basados en características, relativos, etc.) así como topológicos [6].

- Mapas probabilísticos: es una clase de mapa métrico, ya que usa celdas de ocupación. Las celdas de ocupación están medidas en función de la probabilidad de que éstas representen un objeto u obstáculo en el mapa. La probabilidad que tendrá una celda de ocupación si se encuentra en el lugar, donde se halle el objeto será más cercana a uno que a cero. Este tipo de mapas tienen como ventaja que la probabilidad no es fija ya que varía en función las medidas adquiridas del mismo objeto. A partir de las celdas con mayor probabilidad es posible extraer alguna primitiva geométrica asociada a las mismas [6].

3.3 SISTEMA DE NAVEGACIÓN

El sistema de navegación del robot se basa esencialmente en un evasor de obstáculos, esto quiere decir que el robot viaja siguiendo una trayectoria definida hasta que los sensores infrarrojos detecten algún objeto a una distancia menor, por ejemplo, a 30 cm. Al ocurrir esto el robot realiza un giro sobre su propio eje a una velocidad angular de 10 rad/s, sentido antihorario, hasta que la señal dada por los sensores aumente su valor al rango mínimo de los 30 cm para continuar en la trayectoria establecida anteriormente.

La disposición de sensores se encuentra en el apartado 1.9.5.2. Se escogió usar los 3 sensores frontales para la evasión.

A continuación se presenta el Diagrama 3.1 que representa la navegación del robot

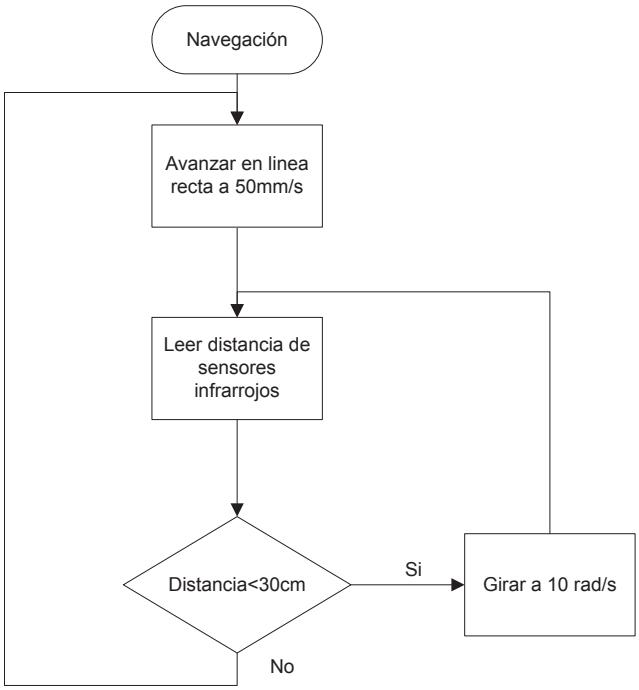


Diagrama 3.1 Navegación

3.4 FUNDAMENTOS PARA EL POSICIONAMIENTO DE CARACTERÍSTICAS DEL ENTORNO

3.4.1 Barrido vertical

Una vez que se obtiene los valores de distancia que provienen del sensor láser se le asocia a su respectivo ángulo, mediante (3.4):

$$\begin{bmatrix} \alpha_{v1} \\ \alpha_{v2} \\ \vdots \\ \alpha_{vn} \end{bmatrix} = 0.351 * \begin{bmatrix} 1 \\ 2 \\ \vdots \\ n \end{bmatrix} \tag{3.4}$$

Dónde:

α_{vi}: Es el ángulo correspondiente a cada punto entregado por el sensor láser usado para la detección vertical.

n: Es el número total medidas realizadas por el láser. Para detección vertical n tiene un valor de 513.

El vector índice es multiplicado por 0.351 que representa la diferencia angular entre medidas, este valor es dado por el fabricante del láser y corresponde al relación entre el ángulo máximo de barrido (180°) y la cantidad de datos que nos entrega el láser (513); este valor de 0,351 tiene como unidad los grados y debe transformarse a radianes con (3.5).

$$\begin{bmatrix} \beta_{v1} \\ \beta_{v2} \\ \vdots \\ \beta_{vn} \end{bmatrix} = \frac{\pi}{180°} * \begin{bmatrix} \alpha_{v1} \\ \alpha_{v2} \\ \vdots \\ \alpha_{vn} \end{bmatrix} \tag{3.5}$$

3.4.1.1 Transformación de coordenadas polares a coordenadas rectangulares.

En la Fig. 3.4 se puede apreciar la disposición de los ejes sobre la plataforma móvil con los que se realiza todo el desarrollo matemático para localizar los puntos detectados y ubicarlos en el mapa del entorno.

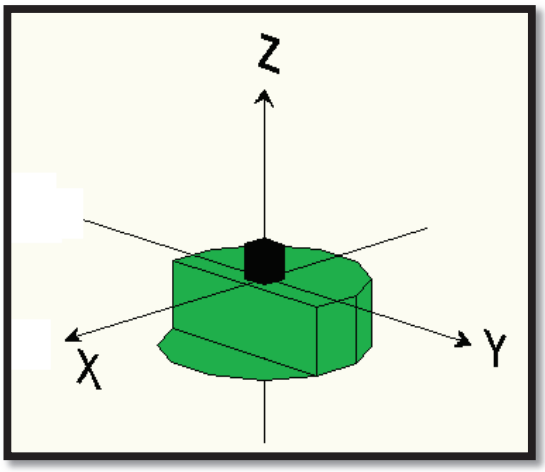


Fig. 3.4 Ejes sobre el robot móvil.

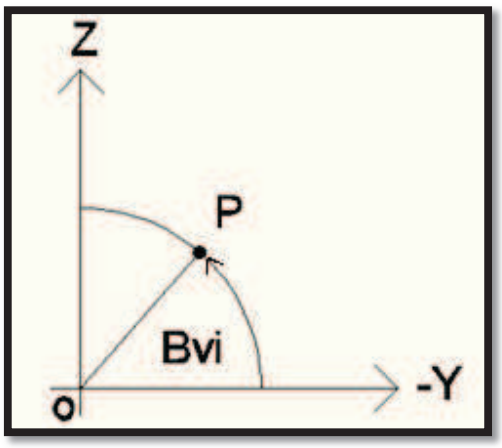


Fig. 3.5 Ejes para barrido vertical

Como se puede apreciar en la Fig. 3.5 el barrido se realiza desde el eje Y negativo debido a la disposición de los ejes de referencia sobre la plataforma y la manera en que el láser realiza el barrido.

$$y_{v1i} = -r_{vi} * cos(\beta_{vi});\ con\ i \quad desde\ 1\ hasta\ \ n \tag{3.6}$$

$$z_i = -r_{vi} * sen(\beta_{vi});\ \ con\ \ i \quad desde\ 1\ hasta\ \ n \tag{3.7}$$

Dónde:

z_i: Coordenada del punto detectado en el eje Z.

y_{v1i}: Coordenada del punto detectado en el eje Y.

r_{vi}: Distancia desde el láser hasta el punto detectado.

β_{vi}: Ángulo de barrido en radianes.

3.4.1.2 Rotación y traslación para barrido vertical

Rotación

Debido a que el barrido se realiza en el plano Y Z el valor de la coordenada Y calculada mediante (3.6), corresponde en el plano XY a la distancia ρ desde el origen relativo hacia el punto detectado.

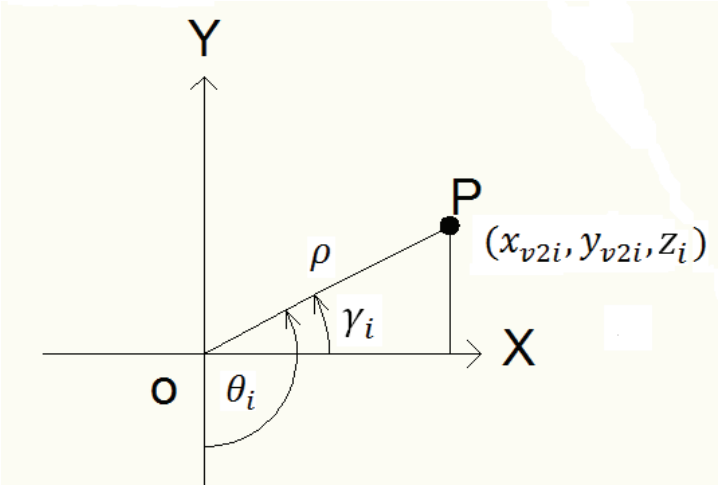


Fig. 3.6 Coordenadas del punto detectado P

De acuerdo a la Fig.3.6 Para calcular las componentes en X y Y de las coordenadas del punto detectado se usa (3.8) y (3.9), ya que debido a la disposición de los ejes en el robot móvil se debe rotarlos de acuerdo a (3.10)

$$x_{v2i} = \rho * \cos(\gamma_i) \tag{3.8}$$

$$y_{v2i} = \rho * sen(\gamma_i) \tag{3.9}$$

$$\gamma_i = \theta_i - \frac{\pi}{2} \tag{3.10}$$

Reemplazando (3.10) en (3.8) y (3.9):

$$x_{v2i} = \rho * \cos\left(\theta_i - \frac{\pi}{2}\right) \tag{3.11}$$

$$y_{v2i} = \rho * sen\left(\theta_i - \frac{\pi}{2}\right) \tag{3.12}$$

Se puede representar (3.11) y (3.12) como:

$$x_{v2i} = -\rho * sen(\theta_i) \tag{3.13}$$

$$y_{v2i} = \rho * \cos(\theta_i) \tag{3.14}$$

La distancia del origen al punto detectado corresponde a:

$$\rho = y_{v1i} \tag{3.15}$$

Reemplazando (3.15) en (3.13) y (3.14)

$$x_{v2i} = -y_{v1i} * sen(\theta_i) \tag{3.16}$$

$$y_{v2i} = y_{v1i} * \cos(\theta_i) \tag{3.17}$$

Traslación

A partir de las ecuaciones (3.16) y (3.17) se realiza la traslación que depende de los valores corregidos de posición provenientes del EKF como se puede observar en Fig. 3.7.

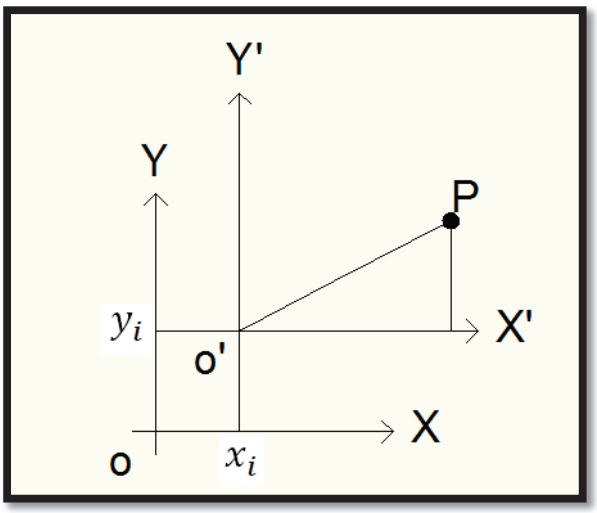


Fig. 3.7 Ejes del robot móvil en relación a los ejes absolutos.

Para trasladar la posición de los datos obtenidos por el láser a partir de (3.16) y (3.17) se le adiciona las coordenadas absolutas del robot móvil obtenidas del EKF.

$$x_{v3i} = -y_{v1i} * sen(\theta_i) + x_i \tag{3.18}$$

$$y_{v3i} = y_{v1i} * \cos(\theta_i) + y_i \tag{3.19}$$

Dónde:

x_{v2i}: Coordenada en X del punto detectado, rotado.

y_{v2i}: Coordenada en Y del punto detectado, rotado.

x_{v3i}: Coordenada en X del punto detectado, rotado y trasladado.

y_{v3i}: Coordenada en Y del punto detectado, rotado y trasladado.

θ_i: Posición angular del robot móvil.

γ_i: Ángulo relativo del robot móvil.

y_{v1i}: Coordenada del punto detectado en el eje y.

x_i: Coordenada en X del robot móvil.

y_i: Coordenada en Y del robot móvil.

ρ: Distancia desde el origen hacia el punto de coordenadas en el eje X y en el eje Y.

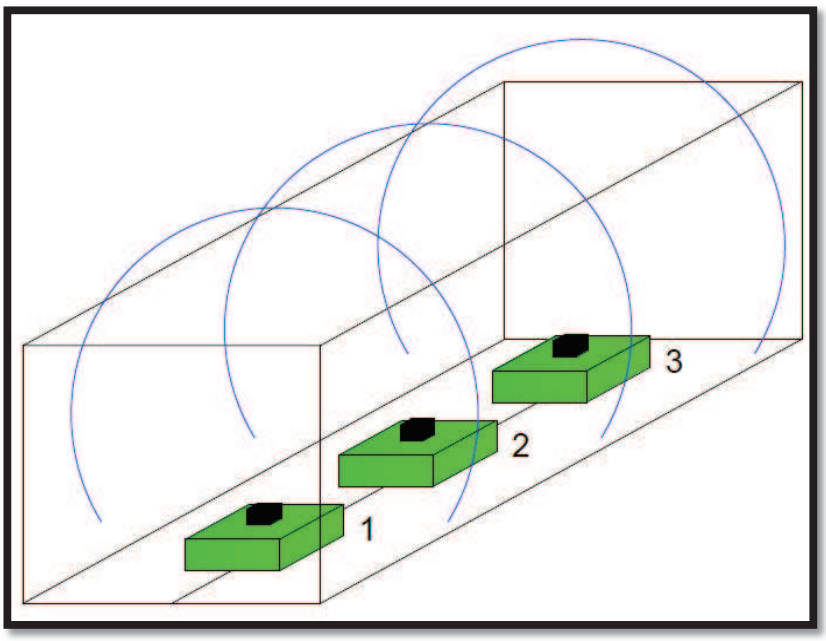


Fig. 3.8 Barrido Vertical con un solo láser.

Se colocó el láser en una disposición tal que éste emite su haz de luz con dirección perpendicular al plano de movimiento del Robotino®, como se muestra en color azul en la Fig. 3.8, además que se aprovechó el movimiento traslacional del robot a lo largo y ancho del plano X Y, se logró adquirir la mayor cantidad de información correspondiente al entorno.

El radio de adquisición de datos del láser, aproximadamente 4 metros, juega un papel fundamental al momento de tomar la información de los puntos de las características del entorno, ya que si alguna característica está a una distancia mayor a los 4 metros se considera como un espacio vacío.

La información de los puntos obtenidos por el láser se encuentran, como ya se mencionó anteriormente, en coordenadas polares, se realizó la conversión a coordenadas rectangulares por facilidad de posicionamiento de éstos en el mapa.

Como se trató anteriormente, se maneja un mapa de características métricas relativas, por lo tanto todos los datos obtenidos se trasladaron y rotaron al eje de referencia global.

Una vez que se tienen los puntos rotados y trasladados al eje de referencia global, se procede a graficarlos mediante el comando plot3, para lo cual se debe indicar las coordenadas x*, y, z*, éstos se deben ingresar por vectores.

3.5 IMPLEMENTACIÓN DE LA DETECCIÓN MEDIANTE LÁSER

La función Láser se encarga de recoger los datos provenientes desde el láser, éstos datos llegan referenciados al láser que se encuentra sobre el robot, la función se encarga de rotarlos y trasladarlos hacia el punto de referencia absoluto que es el punto de partida del robot, ya con los datos rotados y trasladados la función los grafica en 3D en forma de puntos y a un solo color para tener el mayor detalle posible.

Los datos que ingresan a la función láser son:

- Las distancias medidas por el láser.
- El ángulo correspondiente a cada distancia medida por el láser.
- La posición del robot móvil.

Una vez que la función recibe los datos provenientes del sensor láser, se los discrimina de acuerdo a la longitud que poseen, ya que los datos mayores a 4 metros no son tomados en cuenta, debido a que se debe considerar el alcance del láser. Con los datos discriminados se procede a transformar de coordenadas polares a rectangulares, a través de (3.6) y (3.7), con lo que se obtienen las coordenadas (y, z) correspondientes al sistema de referencia relativo que tiene como origen al sensor láser sobre el robot.

Con las coordenadas rectangulares del vector de datos provenientes del sensor láser, se procede a aplicar (3.18) y (3.19) para obtener las coordenadas (x, y, z) que están rotadas y trasladadas hacia el marco de referencia global con origen el punto de partida del robot, para esto es necesario tomar en cuenta la posición actual del robot móvil, ya que son necesarias sus coordenadas $(x,\ y)$ y su ángulo de orientación.

Una vez que se tienen los puntos rotados y trasladados al eje de referencia global, se procede a graficarlos mediante el comando plot3, para lo cual se debe indicar las coordenadas (x, y, z) en forma de vectores, es recomendable realizar la gráfica con puntos, debido a que se puede obtener mayor detalle de las características presentes en el ambiente. Es mejor realizar la gráfica a un solo color ya que al pintar los puntos de acuerdo a niveles o alguna otra lógica ocasiona que el mapa sea más complicado de rotar al momento de observarlo una vez terminada la adquisición de datos.

A continuación se presenta el Diagrama 3.2 que representa la implementación de la detección mediante láser

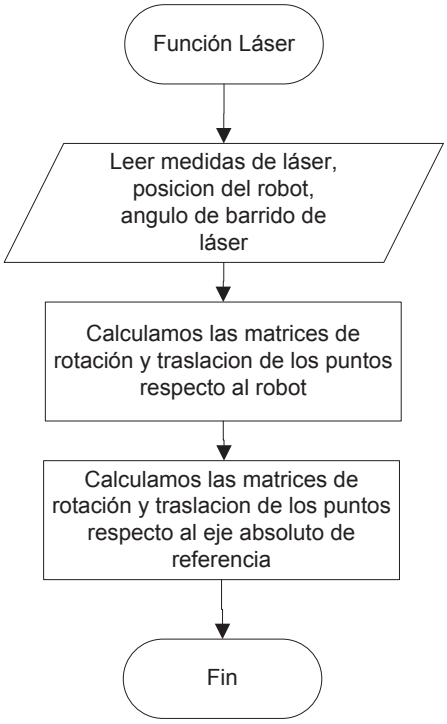


Diagrama 3.2 Implementación de la detección mediante láser

3.6 IMPLEMENTACION DE DETECCIÓN DE LANDMARKS(MARCAS)

Para una correcta detección de líneas y esquinas es necesario conocer los conceptos como las ecuaciones que les corresponden.

Distancia entre dos puntos: es necesario conocer la distancia que existe entre dos puntos consecutivos para de esta manera saber si la recta continúa o si se trata de una recta diferente.

$$D_1 = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \tag{3.20}$$

Dónde:

D_1: Distancia entre las coordenadas de los dos puntos.

x_1, y_1: Coordenadas del primer punto.

x_2, y_2: Coordenadas del segundo punto.

Distancia de punto a recta: a partir del valor de distancia entre el punto y la recta se puede determinar que los siguientes puntos corresponden a una recta diferente.

$$D_2 = \left| \frac{a * x_1 + b * y_1 + c}{\sqrt{a^2 + b^2}} \right| \tag{3.21}$$

Dónde:

D_2: Distancia entre las coordenadas de los dos puntos.

x_1, y_1: Coordenadas del punto.

a, b, c: Coeficientes de la recta.

Intersección de dos rectas: una vez que se tienen las rectas, para detectar las esquinas se debe hallar su intersección, en este caso se usó el método por determinantes.

$$x_3 = \frac{\begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix}}{\begin{vmatrix} c_1 & b_1 \\ c_2 & b_2 \end{vmatrix}} \tag{3.22}$$

$$y_3 = \frac{\begin{vmatrix} a_1 & c_1 \\ a_2 & c_2 \end{vmatrix}}{\begin{vmatrix} c_1 & b_1 \\ c_2 & b_2 \end{vmatrix}} \tag{3.23}$$

Dónde:

x_3: Coordenada sobre el eje X correspondiente a la intersección de las rectas.

y_3: Coordenada sobre el eje Y correspondiente a la intersección de las rectas.

a_1, b_1, c_1: Coeficientes de la primera recta.

a_2, b_2, c_2: Coeficientes de la segunda recta.

Compuerta: Es una distancia establecida como valor de comparación para decidir si un punto corresponde a la línea más cercana. Se establecen dos tipos de compuerta, punto a punto y línea a punto. Una característica fundamental de una compuerta es su flexibilidad, es decir ésta debe ser capaz de adaptarse según la distancia que se tenga desde el robot hacia el punto detectado, ya que dependiendo de la distancia del robot hacia una pared la distancia entre los puntos detectados no será siempre la misma, por esta razón se le llama compuerta dinámica.

Las compuertas deben calibrarse mediante la multiplicación de factores hallados experimentalmente por la distancia proporcionada directamente por el sensor láser del robot móvil hacia el punto detectado. La calibración de las compuertas se la realiza una vez que se ha terminado el mapeo, se varía los factores hasta que se grafiquen las líneas y las esquinas correctamente.

Una vez que se tiene la matriz conformada por los primeros puntos de los barridos verticales y su distancia, se procede a comprobar que la separación entre los puntos esté dentro de un rango que corresponda a la misma línea, con este fin se utiliza (3.20) y (3.21). Para obtener las distancias punto a punto y punto a línea, se necesita comparar las distancias con compuertas dinámicas para poder determinar la pertenencia del punto a la línea, una compuerta fija no serviría porque el sensor láser no siempre se encuentra a la misma distancia de los objetos.

Cada vez que un punto no esté dentro de las compuertas de distancia punto a punto o de punto a recta aumenta un contador que indica el número de fallas. En un principio se toma los dos primeros puntos y se realiza una regresión lineal que entrega los coeficientes de la recta que se les aproxima, se realiza la comparación con el siguiente punto, se verifica las distancias para determinar si se producen fallas, si esto ocurre se incrementa el contador de fallas, si no es el caso se reinicia dicho contador, se toma los tres puntos y se procede a realizar la regresión lineal, el proceso se repite hasta que se cumplan cinco fallas consecutivas. Cuando esto ocurre se descarta los puntos que provocaron las fallas, se reinicia el contador de

fallas y se empieza de nuevo con los puntos que se tiene a continuación, repitiéndose el proceso descrito anteriormente.

Una vez que se tiene las ecuaciones de las rectas se realiza una combinación entre éstas para hallar todas las intersecciones posibles mediante (3.22) y (3.23). Las esquinas halladas no serán las definitivas, ya que es posible que existan intersecciones que no correspondan al mapa, por eso deben ser filtradas mediante la comparación de las distancias que existen entre los extremos de las líneas mediante (3.20), cuando las distancias no estén dentro del rango se eliminan las intersecciones generadas por los extremos de las líneas evaluadas. Se debe tomar en cuenta que al tener rectas paralelas sus coeficientes serán iguales o combinaciones lineales por lo que es importante crear una seguridad en este caso, ya que MATLAB entrega una respuesta NaN que significa que no se tiene un número, por lo que se usa la función isnan que permite identificar este caso y evitar el error.

A continuación se presenta el Diagrama 3.3 que representa implementación de la detección de marcas del entorno

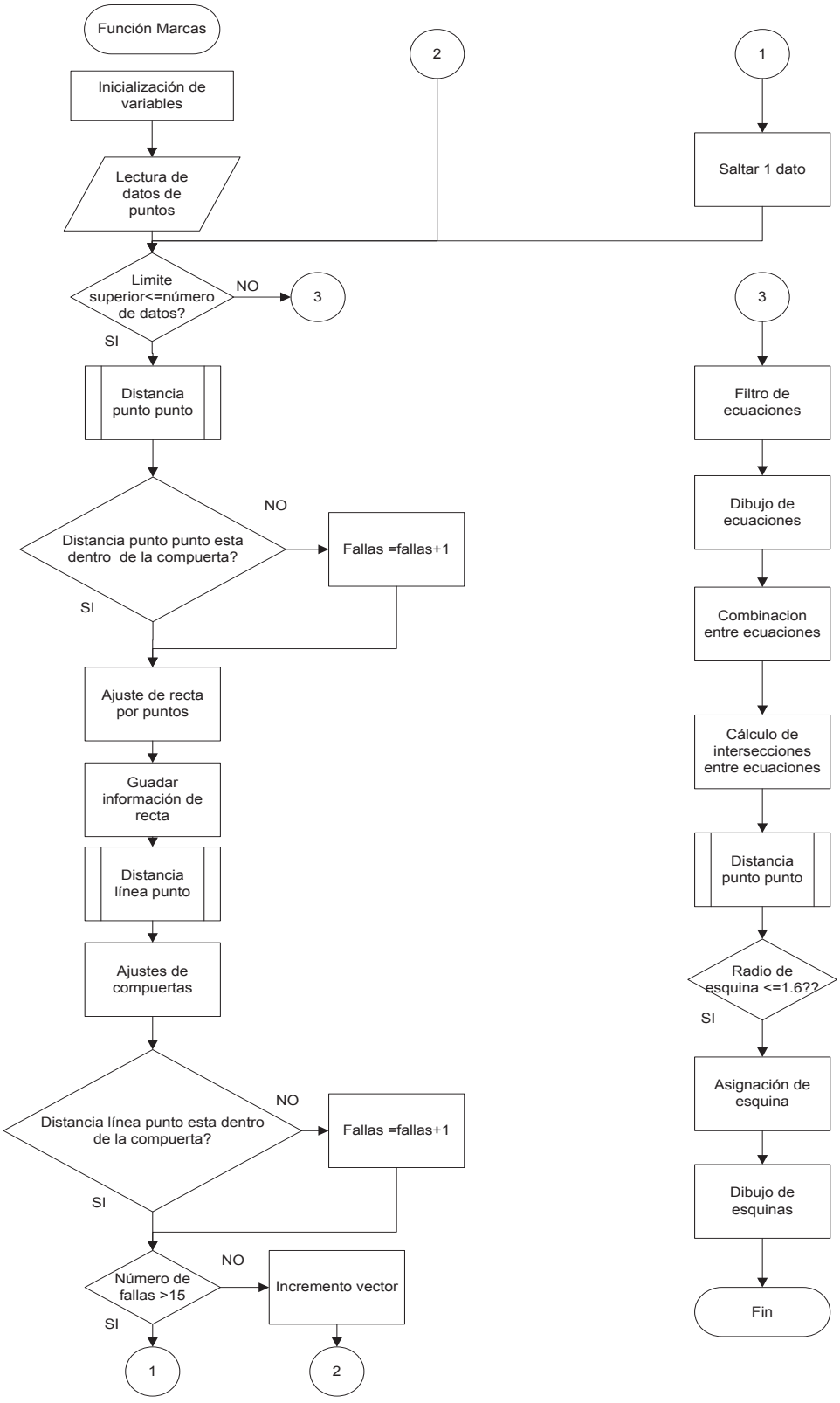


Diagrama 3.3 Implementación de la detección de marcas

3.7 IMPLEMENTACION DE LA ANIMACIÓN DEL ROBOT EN 3D

En esta parte del proyecto, siendo el detalle final, se trató de elaborar un boceto aproximado del Robotino® para verlo en forma de animación en 3D en tiempo real.

En primera instancia se desarrolló la silueta del robot en un plano de dos dimensiones para posteriormente obtener el gráfico en 3D.

Toda esta serie de puntos que se obtuvo del robot fue ingresada a una función, la cual se encarga de unir todos los puntos en una secuencia en línea recta, obteniendo así el boceto del Robotino® en 3D.

Posteriormente, este boceto se pasa por otra función donde se pinta con un color cualquiera cada plano del boceto del Robotino®, en este caso se utilizó un color verde para distinguirlo de los puntos que se forman con los datos del entorno.

Para efectos de animación todos estos puntos deben ser trasladados y rotados cada cierto tiempo para dar la impresión de movimiento en tiempo real.

Una sola función se encarga de envolver todo este proceso de dibujo del Robotino®, la misma está trabajando a la par con la función que permite graficar el mapa del entorno, dando una apariencia de navegación tridimensional.

A continuación se presenta el Diagrama 3.4 que representa la implementación de la animación del robot en 3D y su boceto en Fig.3.9

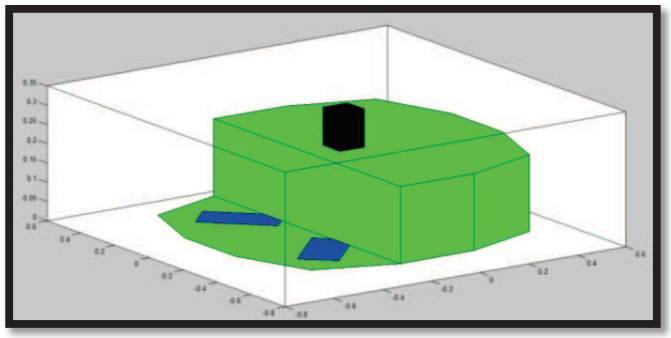


Fig. 3.9 Boceto del robot.

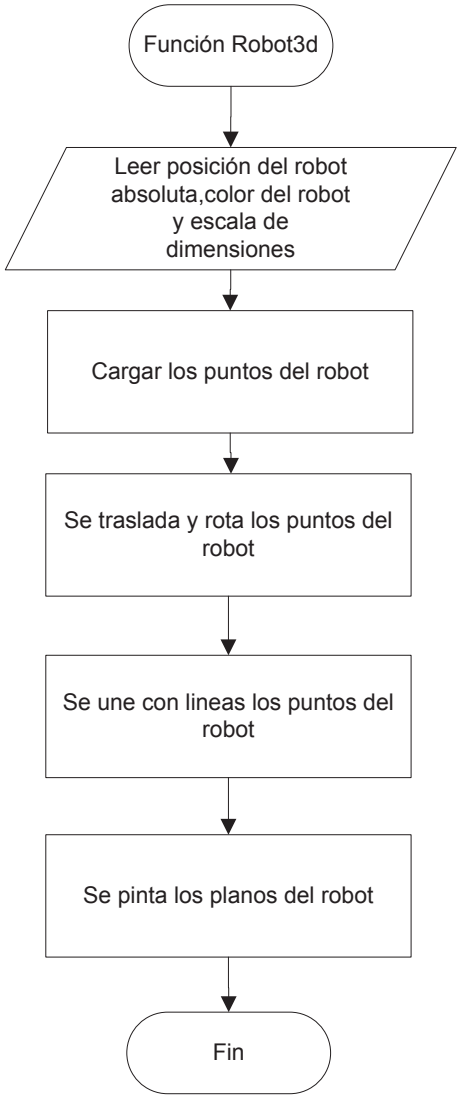


Diagrama 3.4 Implementación de la animación del robot en 3D

CAPÍTULO 4
EXPERIMENTACIÓN Y RESULTADOS

4.1 ANÁLISIS DE FUNCIONAMIENTO DE NORTHSTAR (NS2)

El equipo de localización NorthStar 2 o NS2 de la empresa Festo fue adquirido por la Escuela Politécnica Nacional con el fin de facilitar el trabajo de localización del robot, pero este equipo no presentó la ayuda esperada. Se realizó diferentes pruebas de calibración de tal forma que se pueda confiar en los datos de posición que entrega este equipo.

Las pruebas consistieron en realizar una especie de plano cartesiano sobre el piso del entorno, siendo el origen el emisor. Cada punto del plano se colocó a una separación de 30 cm. (Ver Fig. 4.1).

Fig. 4.1 Mediciones realizadas con NS2.

De esta prueba se tomó datos de posición en $x, y, Theta$ para diferentes alturas de techo desde una altura mínima de 1.7 m hasta 2.8 m.

Se presenta cada una de las tablas con su respectiva altura de techo para diferentes posiciones y orientación en el Anexo B. En base a los errores obtenidos y a las relaciones encontradas, se determina que la posición que entrega el equipo NS2 es totalmente incoherente para casi todas las alturas de techo.

A continuación se presenta la Tabla 4.1 con datos tomados en los cuatro cuadrantes alrededor del emisor en un alcance de 1200 mm a una altura de techo de 1700 mm.

Tabla 4.1 Datos de posición y orientación proporcionados por NS2 para una altura de techo de 1700 mm con su respectiva relación con datos reales.

x real [mm]	y real [mm]	theta real [deg]	xns2 [mm]	yns2 [mm]	theta ns2 [deg]	relación x real/ns2	relación y real/ns2
-300	300	0	-190	178	-0,6	1,58	1,69
-300	600	90	-108	414	93,5	2,78	1,45
-300	900	90	-218	702	94,82	1,38	1,28
-600	300	-90	-432	207	-89,4	1,39	1,45
-600	600	-90	-446	323	-83,8	1,35	1,86
-600	900	90	-368	421	102,3	1,63	2,14
-300	-300	0	-177	-276	-1,8	1,69	1,09
-300	-600	0	-184	-513	-2,6	1,63	1,17
-300	-900	-90	-239	-626	-96,7	1,26	1,44
-600	-300	-90	-433	-131	-94,1	1,39	2,29
-600	-600	-90	-437	-260	-98,7	1,37	2,31
-600	-900	180	-310	-350	170,12	1,94	2,57
300	0	180	214	35	177,01	1,40	0,00
600	0	180	361	43	177,6	1,66	0,00
300	-300	180	195	-209	178,8	1,54	1,44
300	-600	-90	157	-484	-89,7	1,91	1,24
300	-900	-90	197	-757	-85,86	1,52	1,19
600	-300	0	356	-221	2,7	1,69	1,36
600	-600	0	333	-444	1,6	1,80	1,35
600	-900	0	371	-607	4,6	1,62	1,48
300	300	0	173	227	-0,009	1,73	1,32
300	600	0	201	425	-6,5	1,49	1,41
300	900	90	269	727	86,7	1,12	1,24
600	300	180	367	247	176,3	1,63	1,21
600	600	180	327	442	175,6	1,83	1,36
600	900	90	422	503	80,1	1,42	1,79

Con respecto al ángulo de orientación que entrega el NS2, éste está dentro de un límite aceptable, por lo que se adopta con certeza la altura de techo de 1700mm para manejo del NS2.

4.2 ANÁLISIS DE DATOS DE POSICIÓN

Para este análisis se ha tomado los valores correspondientes a los diferentes elementos con que se cuenta para poder realizar la localización del robot móvil. Se tiene el sensor NS2 que según las pruebas realizadas presenta un error bastante alto para las coordenadas (x, y), pero en cuanto al ángulo de orientación arroja buenos resultados.

Se toma en cuenta la odometría y los datos obtenidos del modelo, que para trayectorias cortas tienen una precisión aceptable, pero tienen el inconveniente de la acumulación del error en el tiempo.

Finalmente, mediante el EKF se obtiene los valores de posición sobre los ejes X, Y, con los que se realiza la rotación y traslación tanto de los puntos obtenidos por parte del láser, como el gráfico en 3D del robot móvil, para poder observar la trayectoria seguida.

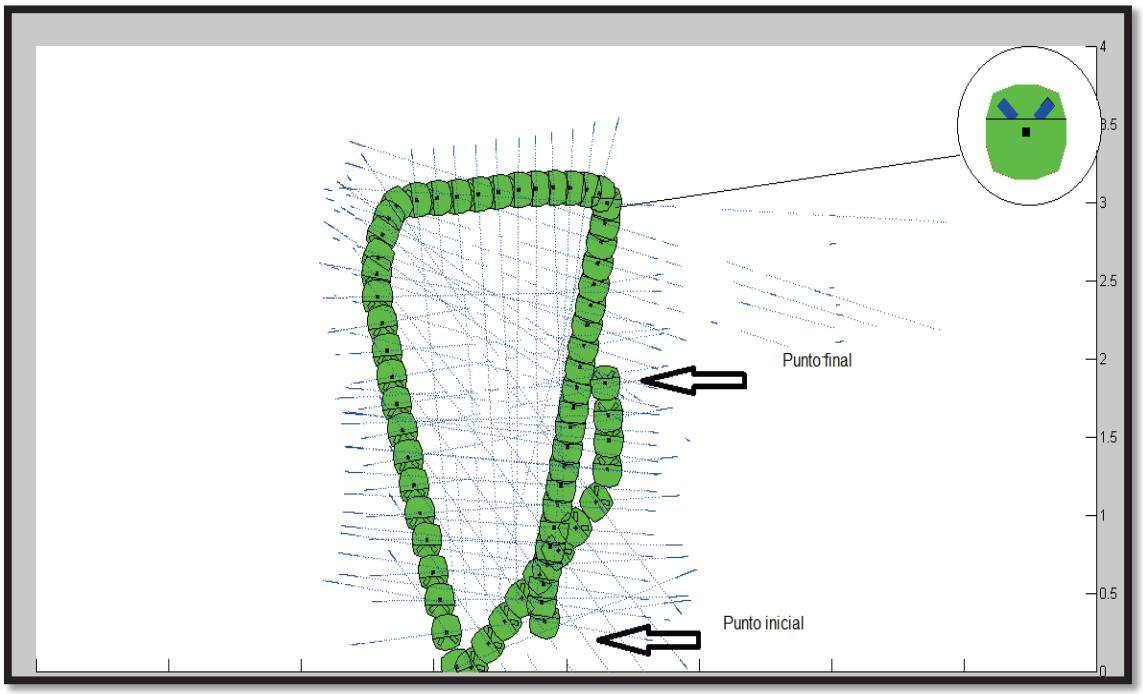


Fig. 4.2 Trayectoria seguida por el robot móvil. Se indica los puntos de inicio y fin de la trayectoria.

En la Fig. 4.2 se puede observar la vista superior del mapa realizado por el robot móvil. En color azul se puede ver los puntos obtenidos del entorno y en color verde la trayectoria seguida por el Robotino®. En la ampliación se aprecia la vista superior de la figura que representa al robot. Se indica el punto inicial y el punto final de la trayectoria, pues el robot avanza por el entorno hasta encontrarse con los objetos que provocan que cambie su orientación.

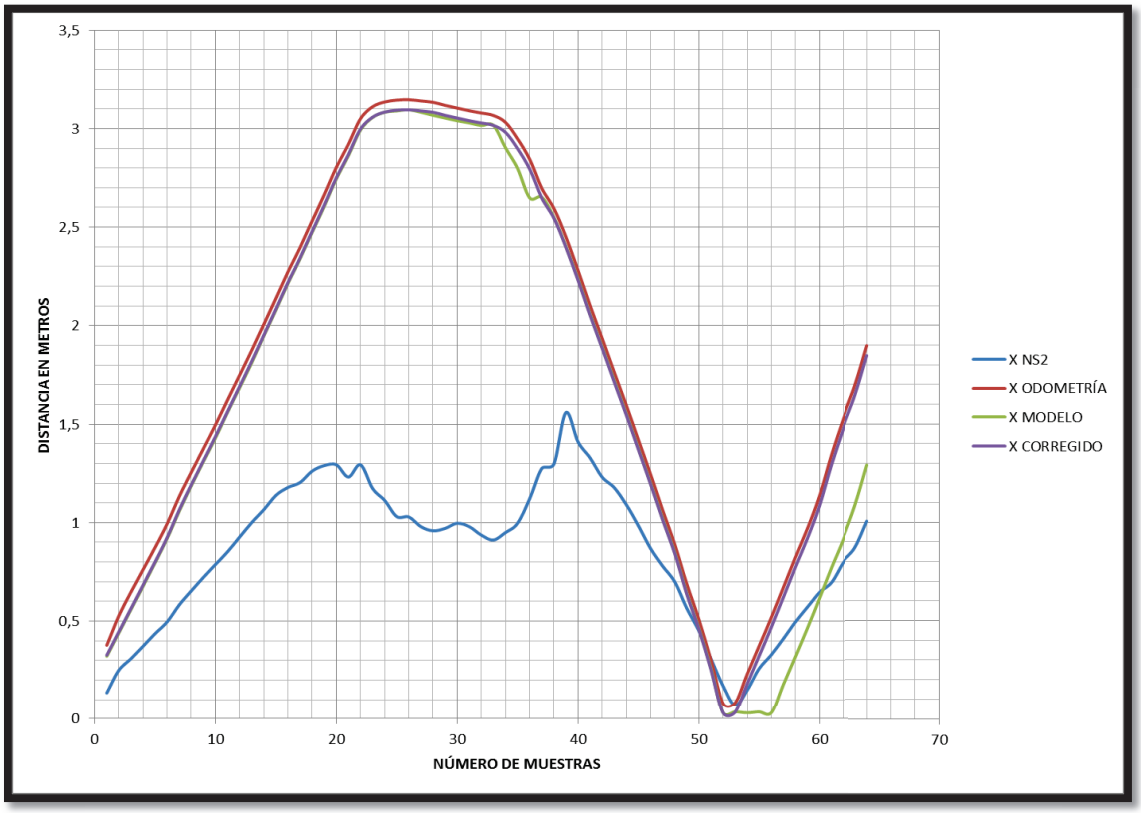


Fig. 4.3 Datos de posición correspondientes a las coordenadas sobre el eje X.

En la Fig. 4.3 se puede observar la evolución de la coordenada sobre el eje X. En color celeste los datos que se obtienen del sensor NS2, en color rojo los valores de la odometría, en color verde la posición obtenida del modelo y en color lila los valores de x corregidos para la trayectoria descrita en la Fig. 4.2.

Se puede observar que en un principio el robot avanza en la coordenada x. A partir de la muestra 20 existe un giro, los valores de x llegan a su pico para posteriormente

decrecer, pues el robot móvil se encuentra en sentido contrario. Se realiza un nuevo giro a partir de la muestra 50 y x vuelve a incrementar su valor. Las coordenadas de posición fueron obtenidas a medida que se realizaba el mapeo.

Los datos proporcionados tanto por el modelo como por la odometría son bastante similares, por lo que la corrección se encuentra en este rango de valores. A partir de la muestra 50, se puede observar que la posición entregada por el modelo dista de la posición de odometría, aumentando así su varianza, lo que provoca que la corrección se aproxime al valor de odometría.

El valor entregado por el sensor NS2 difiere de la odometría y del valor calculado por el modelo, razón por la cual en el EKF se usa la coordenada x proveniente de la odometría mientras que los valores del modelo son usados para la predicción.

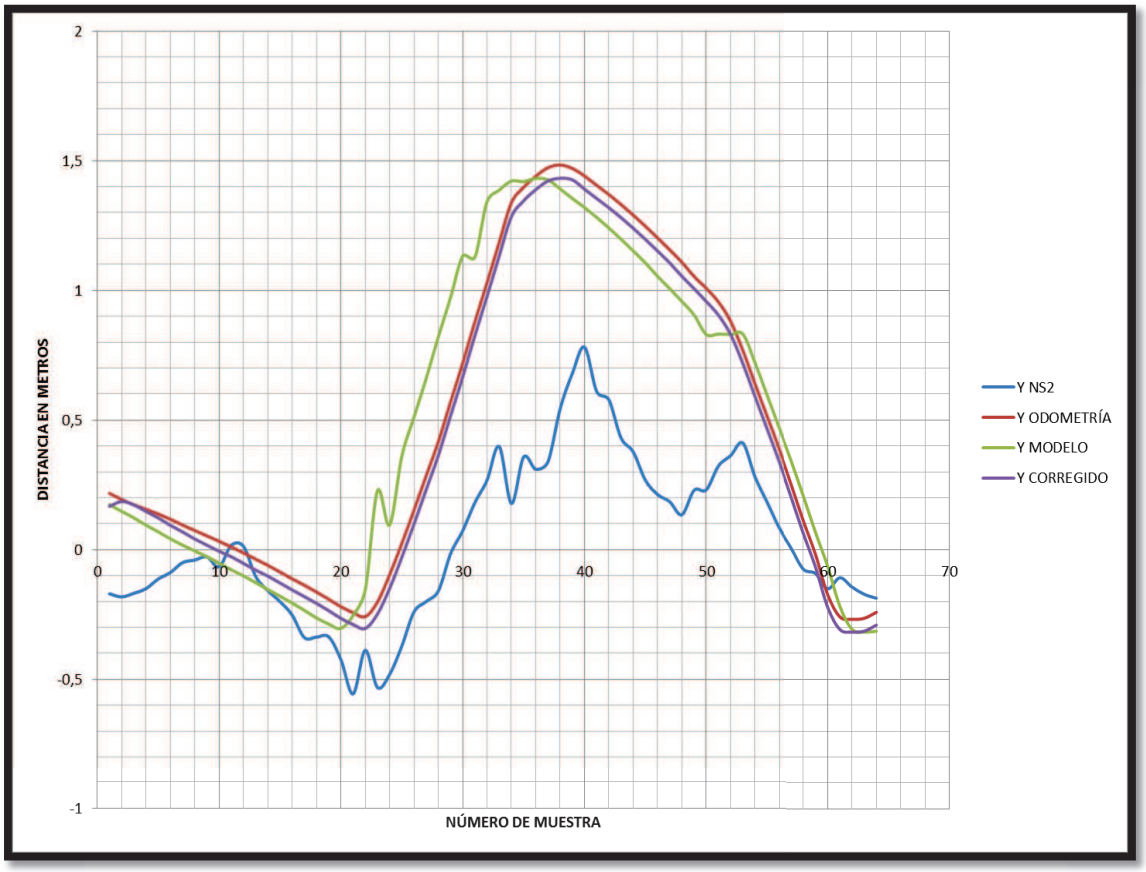


Fig. 4.4 Datos de posición correspondientes a las coordenadas sobre el eje Y.

En la Fig. 4.4 se muestra la evolución de la coordenada sobre el eje Y para la trayectoria mostrada en la Fig. 4.2. De color celeste se observa los valores de y de NS2, en color rojo los valores de y proporcionados por la odometría, en color verde los valores del modelo y en color lila los valores de y corregidos.

El valor de la coordenada y disminuye al iniciar la trayectoria del robot móvil, después de la muestra 20 éste realiza un giro, por lo que la coordenada y empieza a aumentar hasta el siguiente giro, donde disminuye de nuevo su valor.

La corrección de la coordenada y se encuentra entre el modelo y la odometría en el intervalo de 1 a 20 muestras. En el intervalo entre las muestras 20 y 50 la varianza de los datos obtenidos del modelo aumenta, haciendo que el valor de la coordenada y tienda al valor de odometría.

El valor entregado por el sensor NS2 dista mucho del valor obtenido de la odometría y del calculado por el modelo, por lo que es preferible no tomarlo en consideración y optar por la odometría conjuntamente con la predicción del modelo, para que éstos a su vez ingresen al EKF y así obtener la posición corregida.

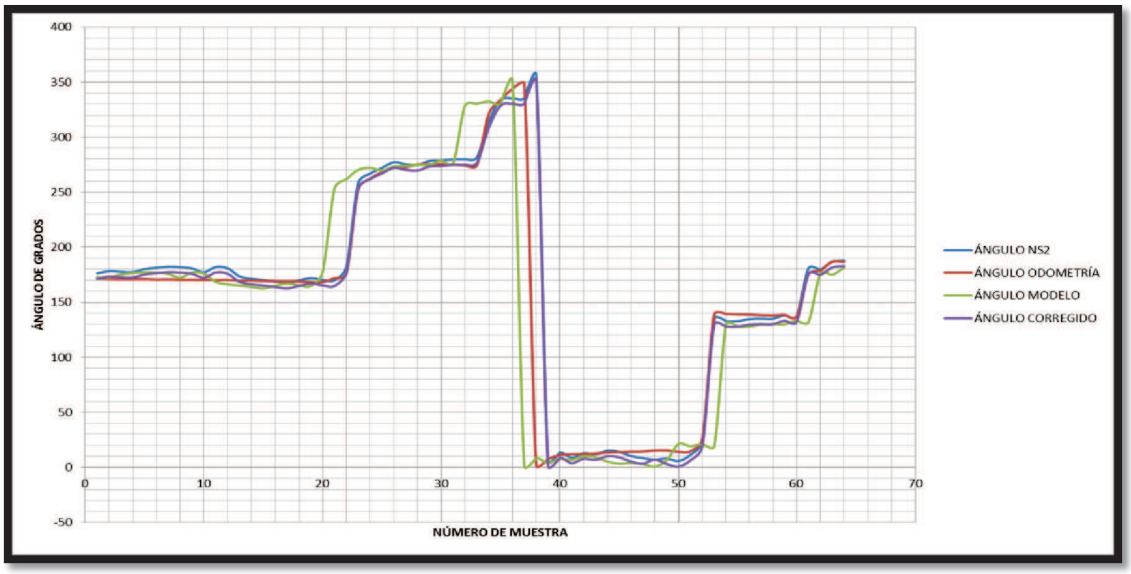


Fig. 4.5 Datos de posición correspondientes al ángulo de orientación expresado en grados.

En la Fig. 4.5 se observa el comportamiento del ángulo proporcionado por el sensor NS2 en color celeste, el ángulo de la odometría en color rojo, el ángulo que es resultado del cálculo del modelo en color verde y finalmente, el ángulo de la corrección realizada en color lila.

Se observa que las curvas son bastante similares entre sí. Desde la muestra 1 el ángulo se aproxima a 180°, a partir de la muestra 20 se produce un giro y el ángulo tiende a 270°.

Cerca de la muestra 40 se produce un giro y se llega a 0°, a partir de la muestra 50 gira nuevamente hacia 120°. Finalmente, a partir de la muestra 60 el ángulo tiende a 180°. Las curvas del modelo, odometría y del NorthStar 2 tienen una forma parecida entre sí, por lo que la corrección se encuentra dentro del mismo rango de valores.

La ventaja del valor proporcionado por el sensor NS2 es que a lo largo del tiempo éste tiene un valor constante de error, lo que no ocurre con la odometría, ya que ésta presenta el inconveniente de acumular su error conforme aumenta el tiempo de funcionamiento.

4.3 ANÁLISIS DE LAS PRUEBAS REALIZADAS

Para el análisis de los resultados obtenidos se consideró dos mapas del mismo lugar donde el robot móvil transitó en dos ocasiones.

Cada recorrido es considerado como un conjunto de datos. Se compara las coordenadas de puntos pertenecientes a una misma pared o techo dentro del mismo mapa.

4.3.1 Mapa 1

En la Fig. 4.6 se presenta el mapa realizado por el Robotino®, en color verde se representa la figura del robot describiendo su trayectoria, mientras que en color azul se muestra los puntos obtenidos por parte del sensor láser.

Se puede observar la reconstrucción de un pasillo de la Facultad de Ingeniería Eléctrica y Electrónica mostrada en la Fig.4.7, se nota el techo y la puerta de una oficina cercana al pasillo.

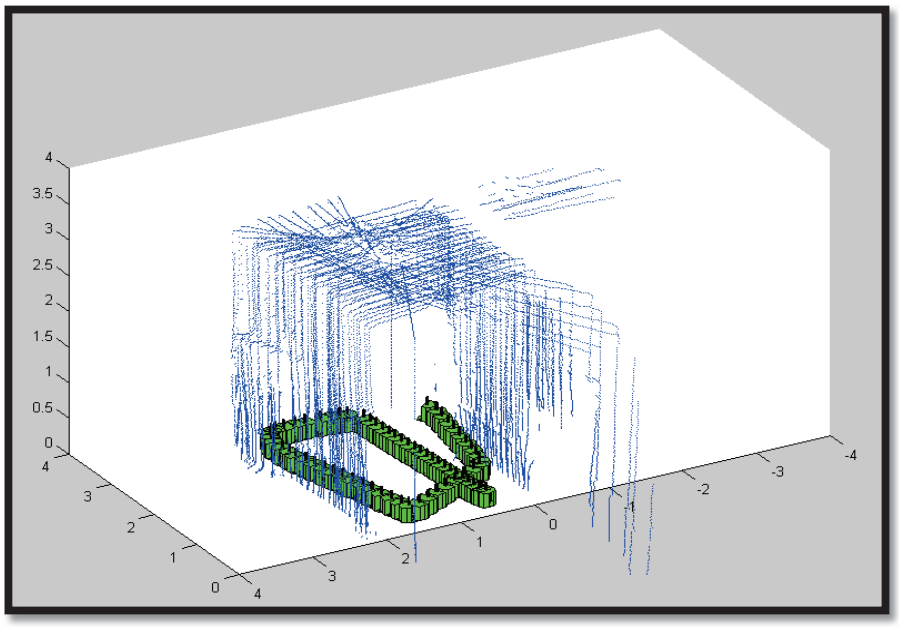


Fig. 4.6 Mapa1.

Fig. 4.7 Pasillo de la Facultad de Ingeniería Eléctrica y Electrónica.

En las siguientes tablas se muestra la diferencia entre coordenadas de puntos pertenecientes al Mapa 1, realizado por el robot móvil al recorrer el mismo lugar por dos ocasiones.

4.3.1.1 Comparación 1

Tabla 4.2 Cálculo de la diferencia entre 2 grupos de datos correspondientes a un mismo punto, pared izquierda en la parte superior del pasillo del Mapa 1.

Coordenada	Grupo 1[m]	Grupo 2[m]	Error[%]
X	0.588	0.577	1.87
Y	-0.3293	-0.1008	25.3
Z	2.952	2.979	0.9

En la Tabla 4.2 se nota que el error entre dos puntos mapeados correspondientes a las coordenadas en el eje Y es de 25.3 %. Mientras que el error para las coordenadas x y z son de 1.87 % y de 1% respectivamente.

La Fig. 4.8 muestra las coordenadas de los puntos del primer y segundo grupo de datos del Mapa 1, correspondiente a una arista formada entre el plano del techo y el plano de la pared del pasillo mapeado.

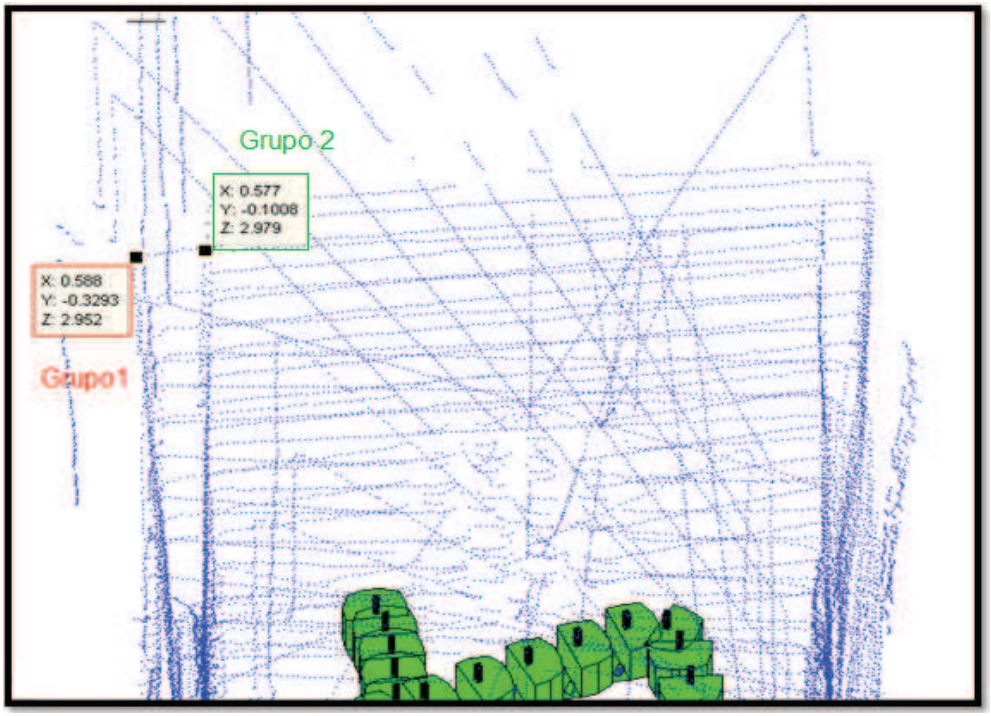


Fig. 4.8 Posición de los puntos del primer y segundo grupo de datos del Mapa 1 correspondientes a la Comparación 1.

4.3.1.2 Comparación 2

Tabla 4.3 Cálculo de la diferencia entre 2 grupos de datos correspondientes a un mismo punto, base de la pared izquierda del pasillo del Mapa 1.

Coordenada	Grupo 1[m]	Grupo 2[m]	Error[%]
X	2.662	2.672	0.37
Y	-1.427	-1.707	19.62
Z	0.2193	0.2252	2.69

Los puntos analizados y mostrados en la Tabla 4.3 corresponden a la parte baja de la pared, donde se nota que el error entre coordenadas en el eje Y es de 19.6 %, mientras que las coordenadas sobre el eje X y Z tienen errores menores al 3%.

La Fig. 4.9 muestra las coordenadas de los puntos del primer y segundo grupo de datos del Mapa 1, correspondiente a la parte baja de la pared del pasillo ya antes mencionado.

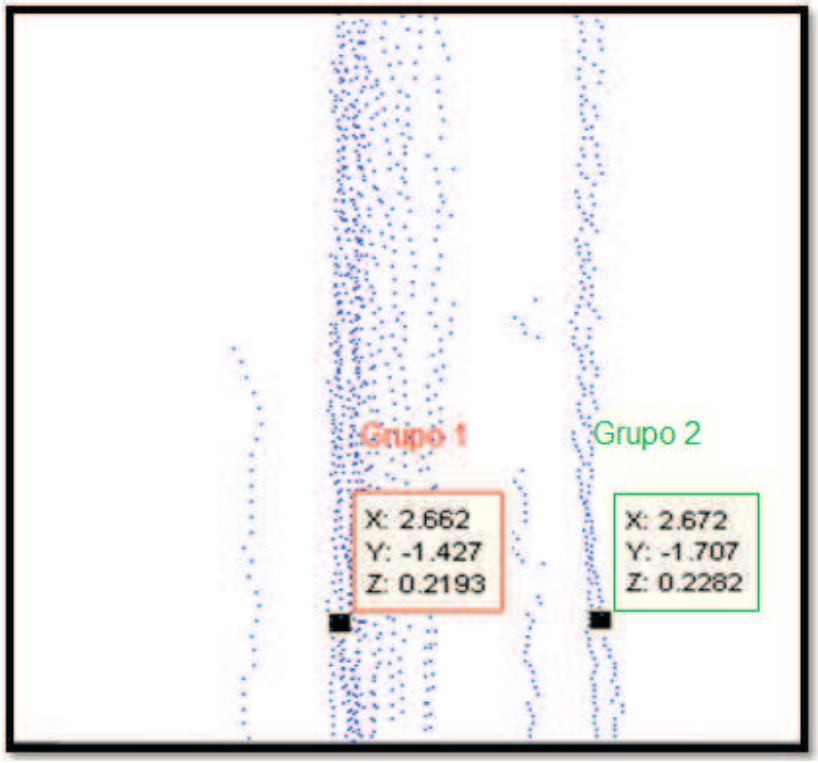


Fig. 4.9 Posición de los puntos del primer y segundo grupo de datos del Mapa 1 correspondientes a la Comparación 2.

4.3.1.3 Comparación 3

Tabla 4.4 Cálculo de la diferencia entre 2 grupos de datos correspondientes a un mismo punto, pared del fondo del pasillo del Mapa 1.

Coordenada	Grupo 1[m]	Grupo 2[m]	Error[%]
X	2.433	2.447	0.57
Y	2.281	2.316	1.53
Z	0.7192	0.7067	1.73

Al comparar los datos en la Tabla 4.4 se obtiene que el error entre coordenadas en el eje Y es de 1.5 %, mientras que para las coordenadas sobre los ejes X y Z el error es de 0.57 % y 1.73%.

La Fig. 4.10 muestra las coordenadas del punto del primer y segundo grupo de datos del Mapa 1, correspondientes a la pared del fondo del pasillo ya mencionado.

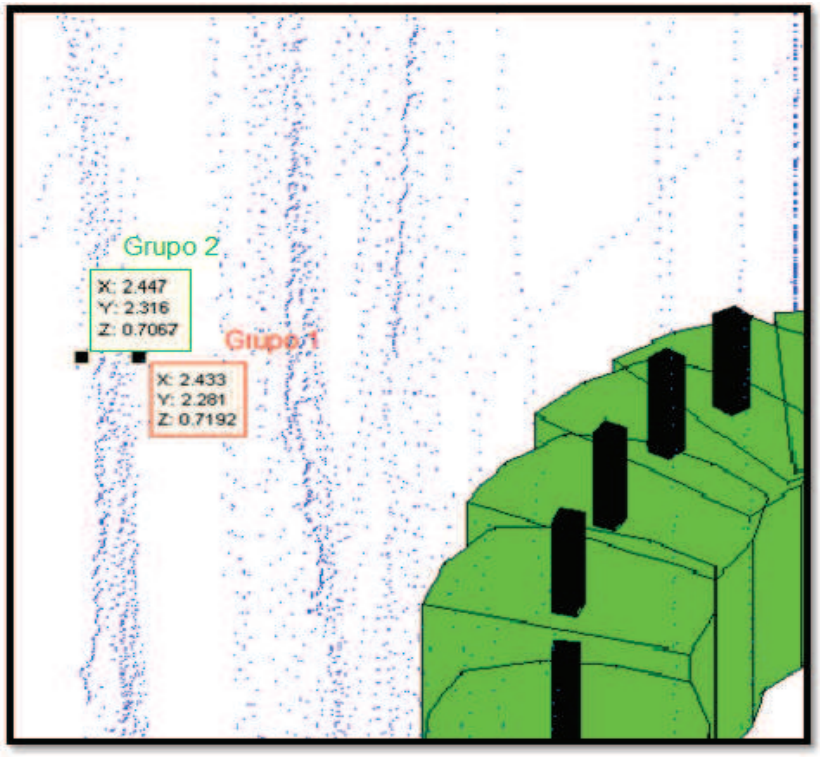


Fig. 4.10 Coordenadas del punto del primer y segundo grupo de datos del Mapa 1 correspondientes a la Comparación 3.

4.3.1.4 Comparación 4

Tabla 4.5 Cálculo de la diferencia entre 2 grupos de datos correspondientes a un mismo punto, techo del pasillo del pasillo del Mapa 1.

Coordenada	Grupo 1[m]	Grupo 2 [m]	Error[%]
X	2.84	2.846	0.21
Y	1.664	1.583	4.86
Z	3.011	2.981	0.99

Los puntos analizados en la Tabla 4.5 corresponden a las coordenadas del techo donde el error sobre el eje Z es de 0.99 %.

La Fig. 4.11 muestra las coordenadas del punto del primer y segundo grupo de datos del Mapa 1, correspondiente al techo del pasillo.

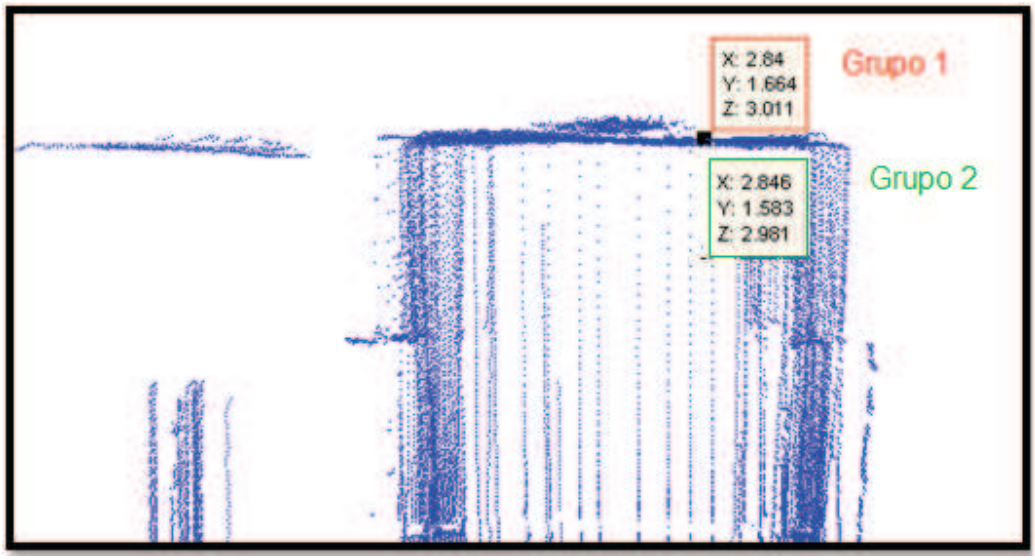


Fig. 4.11 Coordenadas de los puntos del primer y segundo grupo de datos del Mapa 1 correspondientes a la Comparación 4.

4.3.2 Mapa 2

La Fig. 4.7 muestra el Mapa 2 donde se puede apreciar en azul las paredes y el techo de un pasillo de la Facultad de Ingeniería Eléctrica y Electrónica, de color verde se observa la figura del Robotino® y su trayectoria. En la Fig. 4.13 se puede observar la imagen del pasillo mapeado.

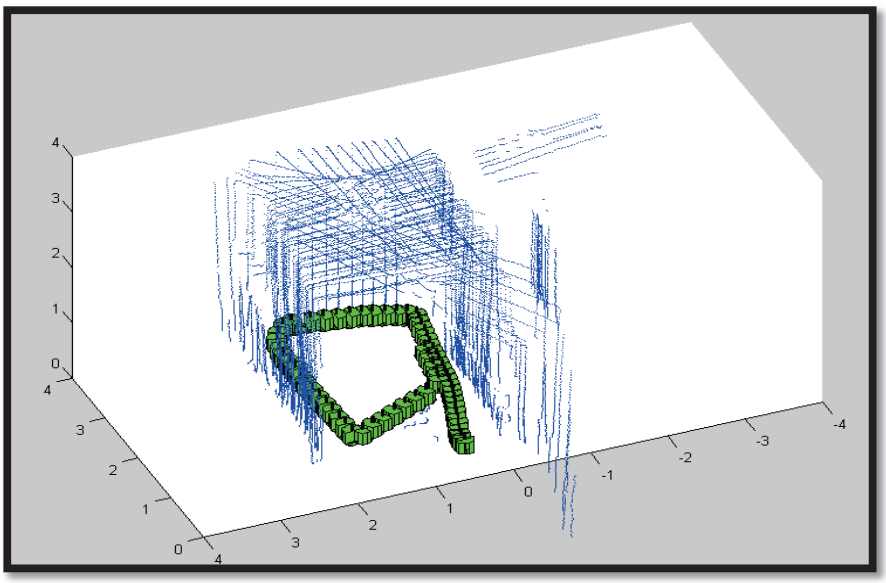


Fig. 4.12 Mapa 2.

Fig. 4.13 Pasillo de la Facultad de Ingeniería Eléctrica y Electrónica.

En las siguientes tablas se muestra la diferencia entre coordenadas de puntos pertenecientes al Mapa 2 realizado por el robot móvil al recorrer el mismo lugar por dos ocasiones.

4.3.2.1 Comparación 1

Tabla 4.6 Cálculo de la diferencia entre 2 grupos de datos correspondientes a un mismo punto, pared izquierda en la parte superior del pasillo del Mapa 2.

Coordenada	Grupo 1[m]	Grupo 2[m]	Error[%]
X	1.45	1.453	0.2
Y	-0.364	-0.2445	32.82
Z	2.951	3.014	2.13

El error entre las coordenadas sobre el eje Y de los puntos escogidos es de aproximadamente 33 %, mientras que para las coordenadas sobre el eje X es de 0.2 % y en el eje Z es de 2.13 %.

La Fig. 4.14 muestra las coordenadas de los puntos del primer y segundo grupo de datos de la pared izquierda superior del pasillo mencionado anteriormente.

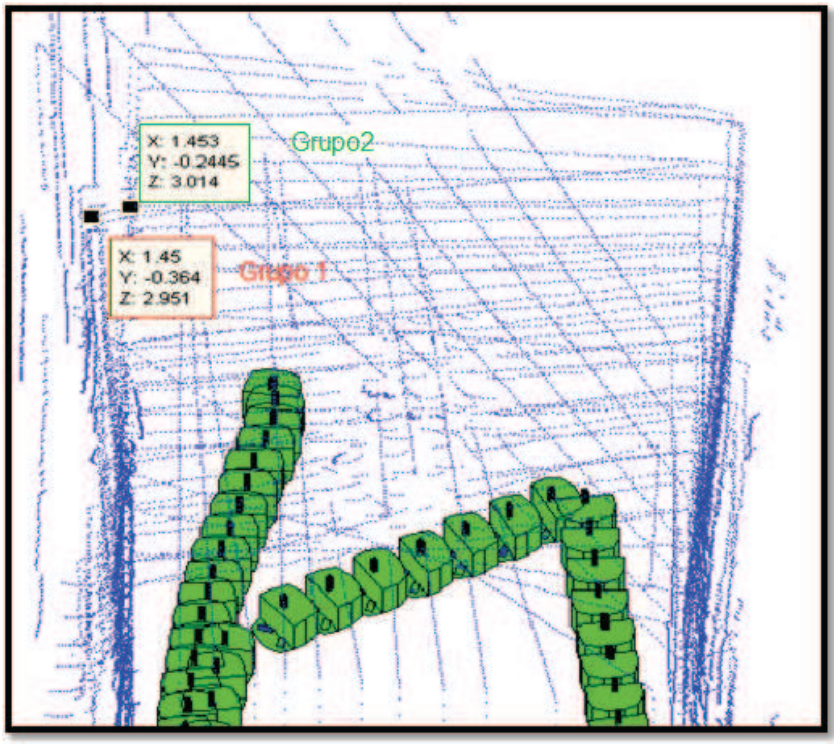


Fig. 4.14 Coordenadas de los puntos del primer y segundo grupo de datos del Mapa 2 correspondiente a la Comparación 1.

4.3.2.2 Comparación 2

Tabla 4.7 Cálculo de la diferencia entre 2 grupos de datos correspondientes a un mismo punto, base de la pared izquierda del pasillo del Mapa 2.

Coordenada	Grupo1[m]	Grupo2[m]	Error[%]
X	2.617	2.626	0.34
Y	-0.3675	-0.3119	15.12
Z	0.3773	0.3719	1.43

El error entre dos puntos escogidos correspondientes a la misma pared y a la misma altura en su coordenada sobre el eje Y es de aproximadamente 15.12 %, para la coordenada en el eje X es de 0.34 % y en el eje Z de 1.43 %..

La Fig. 4.15 muestra las coordenadas de los puntos del primer y segundo grupo de datos de la base de la pared izquierda del pasillo.

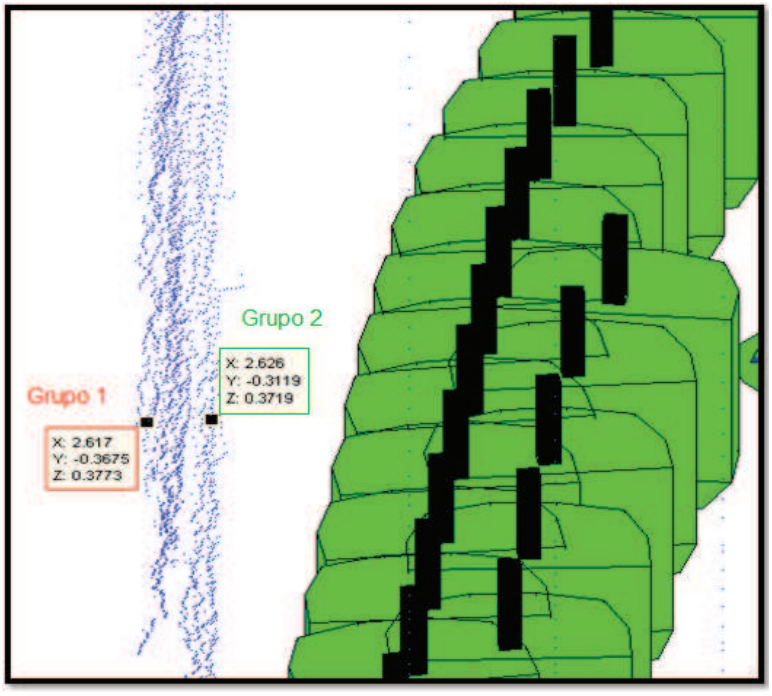


Fig. 4.15 Coordenadas del punto de los puntos del primer y segundo grupo de datos del Mapa 2 correspondiente a la Comparación 2.

4.3.2.3 Comparación 3

Tabla 4.8 Cálculo de la diferencia entre 2 grupos de datos correspondientes a un mismo punto, pared del fondo del pasillo del Mapa 2.

Coordenada	Grupo1[m]	Grupo2[m]	Error[%]
X	3.37	3.346	0.71
Y	-1.737	-1.865	7.36
Z	0.2427	0.2473	1.9

El error entre los puntos escogidos pertenecientes a la pared del fondo del pasillo es de 7.36 % en su coordenada sobre el eje Y, mientras que en el eje X es de 0.71%, y en el eje Z de 1.9 %.

La Fig. 4.16 muestra las coordenadas de los puntos del primer y segundo grupo de datos de la pared del fondo del pasillo.

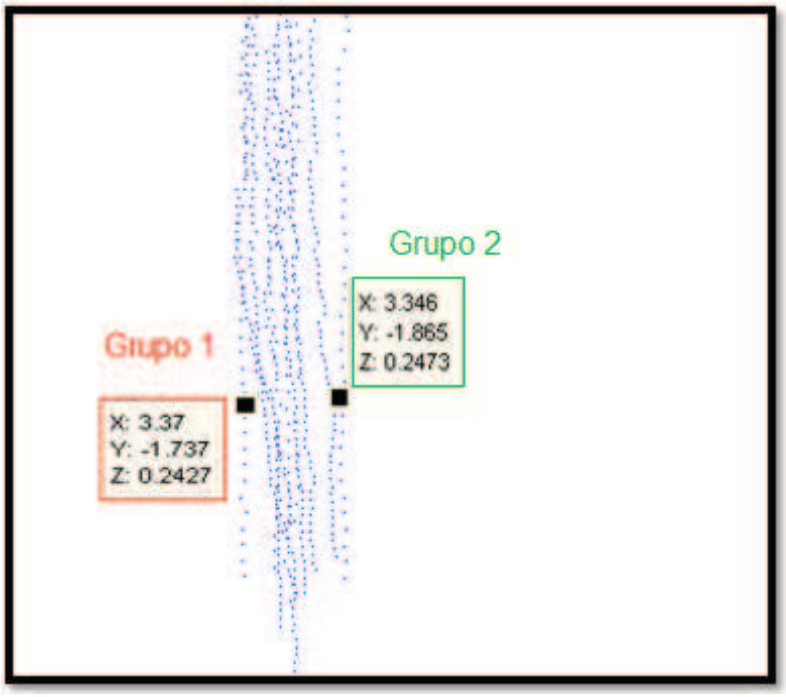


Fig. 4.16 Coordenadas de los puntos pertenecientes al primer y segundo grupo de datos del Mapa 2, correspondiente a la Comparación 3.

4.3.2.4 Comparación 4

Tabla 4.9 Cálculo de la diferencia entre 2 grupos de datos correspondientes a un mismo punto, techo del pasillo del Mapa 2.

Coordenada	Grupo 1[m]	Grupo 2[m]	Error[%]
X	1.483	1.465	1.21
Y	0.2869	0.2006	30.08
Z	2.992	3.017	0.83

El error entre los puntos escogidos correspondientes al techo es de 0.83 % en su coordenada sobre el eje Z, mientras que en el eje X es de 1.21 % y sobre el eje Y es de 30 %.

La Fig. 4.17 muestra las coordenadas de los puntos del primer y segundo grupo de datos del techo del pasillo ya mencionado.

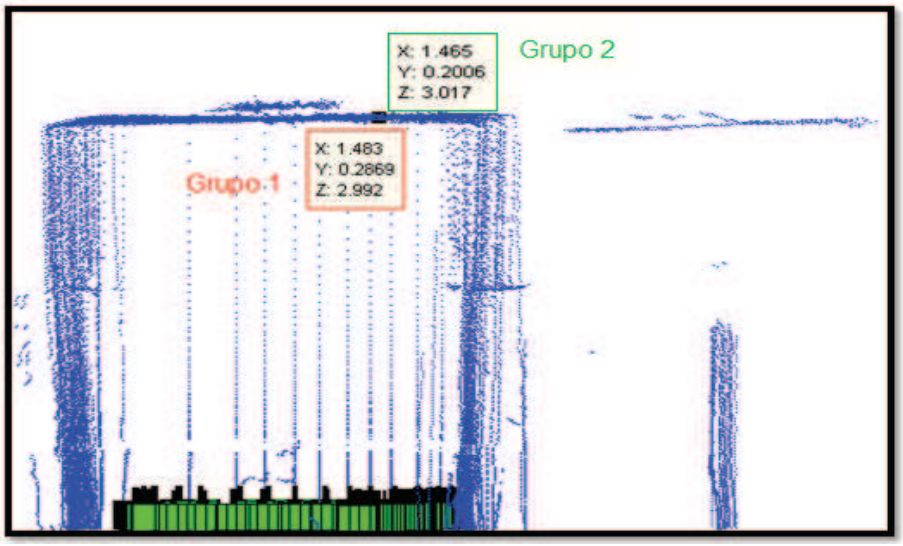


Fig. 4.17 Coordenadas de los puntos del primer y segundo grupo de datos del Mapa 2 correspondiente a la Comparación 4.

4.3.3 Detección de marcas (landmarks) en el entorno

La Fig. 4.18 muestra la Detección de marcas 1 donde se puede apreciar en verde la información de los puntos de las paredes detectadas, en color azul se observa la

recta que representa las paredes y en color negro se observa las esquinas detectadas. En la Fig. 4.13 se puede observar la imagen del pasillo mapeado.

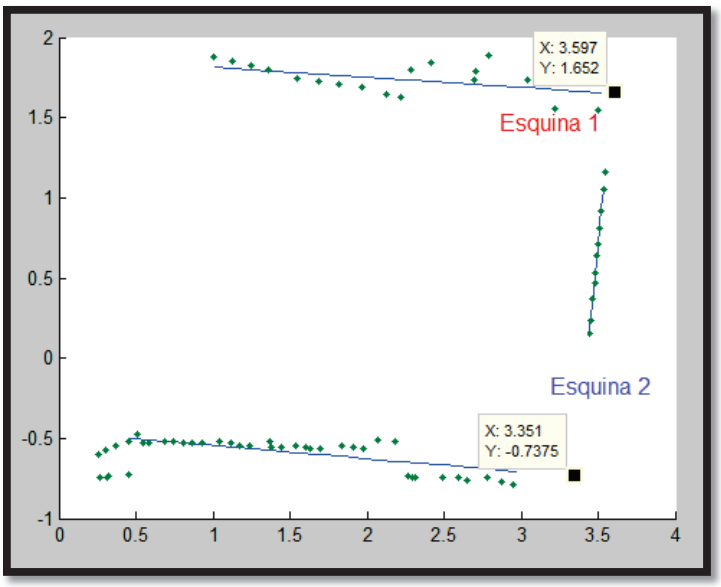


Fig. 4.18 Detección de marcas 1.

Tabla 4.10 Cálculo de error de la posición de la Esquina 1 detectada con respecto a la posición real.

Coordenada	Esquina 1 real [m]	Esquina 1 detectada [m]	Error [%]
X	3	3.597	20
Y	1.35	1.652	22.37

En la Tabla 4.10 se muestra los errores que se obtuvo en la detección de la Esquina 1, los mismos son muy aceptables debido a que se ha considerado que la ubicación de la esquina es la intersección de dos rectas que representan a las paredes, pero en la realidad la esquina está conformada por la intersección de dos paredes, pero en este pasillo existe una puerta, la cual provoca un desnivel en la adquisición de puntos como se muestra en la Fig. 4.18, lo que provoca que la recta calculada se desvíe y la esquina también lo haga.

Tabla 4.11 Cálculo de error de la posición de la Esquina 2 detectada con respecto a la posición real.

Coordenada	Esquina 2 real [m]	Esquina 2 detectada [m]	Error [%]
X	3	3.351	11.7
Y	-0.6	-0.7375	22.92

En la Tabla 4.11 se muestra el cálculo de error de la Esquina 2, los errores calculados para la posición de esta esquina están dentro de un rango aceptable, en conjunto con la Tabla 4.10 los errores que se manejan para la detección de ambas esquinas no sobrepasa el 22%.

La Fig. 4.19 muestra la Detección de marcas 2 donde se puede apreciar en verde la información de los puntos de las paredes detectadas, de color azul se observa la recta que representa las paredes y en color negro se observa las esquinas detectadas. En la Fig. 4.13 se puede observar la imagen del pasillo mapeado.

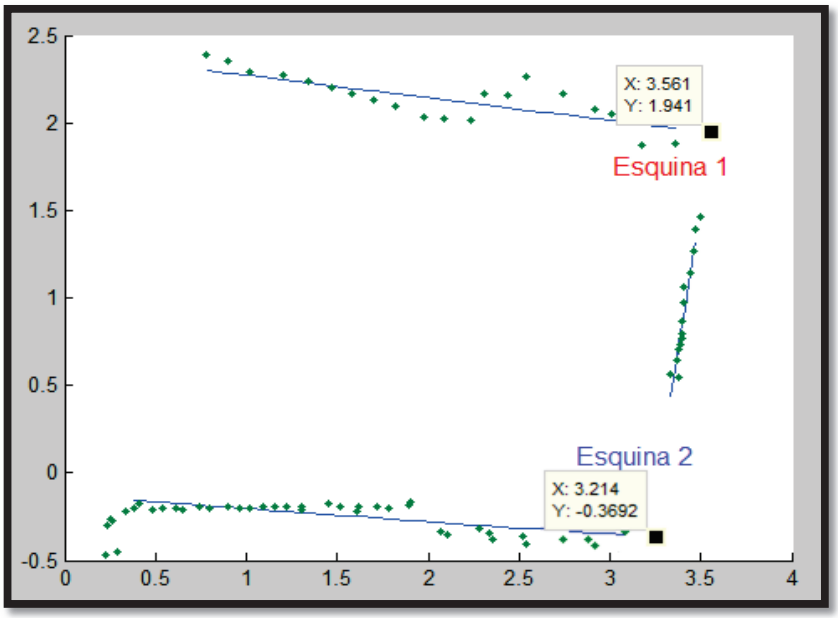


Fig. 4.19 Detección de marcas 2.

Tabla 4.12 Cálculo de error de la posición de la Esquina 1 detectada con respecto a la posición real.

Coordenada	Esquina 1 real [m]	Esquina 1 detectada [m]	Error [%]
X	3	3.651	21.7
Y	1.35	1.941	21.31

En la Tabla 4.12 se muestra el error de posición de la Esquina 1 en un nuevo mapeo, en relación con las Tablas 4.11 y Tabla 4.12 se maneja en el mismo porcentaje de error, con lo que se puede afirmar que las esquinas detectadas corresponden con la realidad.

Tabla 4.13 Cálculo de error de la posición de la Esquina 2 detectada con respecto a la posición real.

Coordenada	Esquina 2 real [m]	Esquina 2 detectada [m]	Error [%]
X	3	3.214	7.13
Y	-0.6	-0.3692	38.47

En la Tabla 4.13 se muestra el error de la Esquina 2 en un nuevo mapeo, se obtiene un error en la coordenada sobre el eje Y mayor a los mostrados en las Tablas 4.11, Tabla 4.12 y Tabla 4.13 debido a que el robot tiene una ligera desviación cuando avanza en una trayectoria recta, una de las posibles razones por las que sucede esta desviación es por el deterioro de las partes mecánicas del Robotino®.

4.4 ANÁLISIS DE RESULTADOS DE LOCALIZACION Y MAPEO SIMULTANEOS EN 3D (SLAM)

En los mapas realizados mediante el Robotino®, se implementó los algoritmos para localización mediante EKF con el fin de corregir los datos que provienen de la

odometría, para la localización de las características pertenecientes al entorno se utilizó el algoritmo que rota y traslada los puntos obtenidos por parte del sensor láser.

En la Fig. 4.20 se puede observar el mapeo de un pasillo de la Facultad de Ingeniería Eléctrica y Electrónica, el mismo que se puede observar en la Fig. 4.13. Se nota claramente la cartelera, luminaria y paredes del lugar.

En el mapa de la Fig. 4.20 se encuentran dos personas cerca de la cartelera, sus siluetas fueron capturadas por los barridos realizados mediante el sensor láser. El robot móvil realizó un movimiento en línea recta por lo que se puede apreciar barridos verticales y paralelos entre sí.

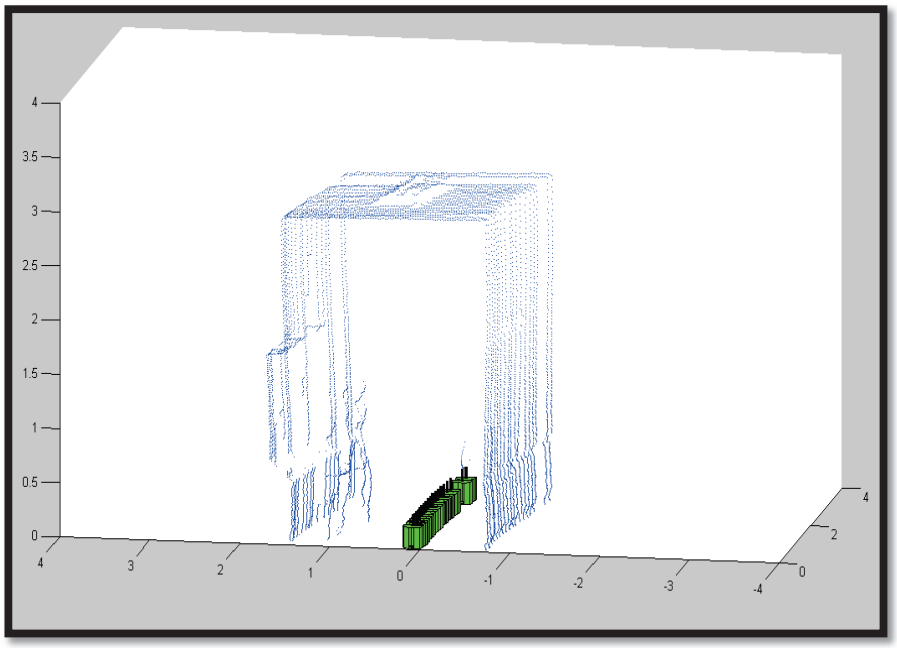


Fig. 4.20 Mapeo del entorno con movimiento en línea recta.

En la Fig. 4.21 Se puede apreciar el mapa realizado por parte del Robotino® cuando existe la presencia de una persona, la silueta de ésta se reconoce al igual que su sombra proyectada, un aspecto que distingue este mapa de los anteriores, es que los puntos obtenidos por parte del sensor láser están unidos a través de líneas, el mapa muestra mayor uniformidad pero la desventaja es que se pierde detalle de las características del entorno.

En la Fig. 4.22 se observa el mismo mapa desde un ángulo diferente. Gracias al diseño de la representación del robot móvil, es posible distinguir a qué lado se encuentra la persona, de igual forma saber cuál es el punto inicial y final en la trayectoria.

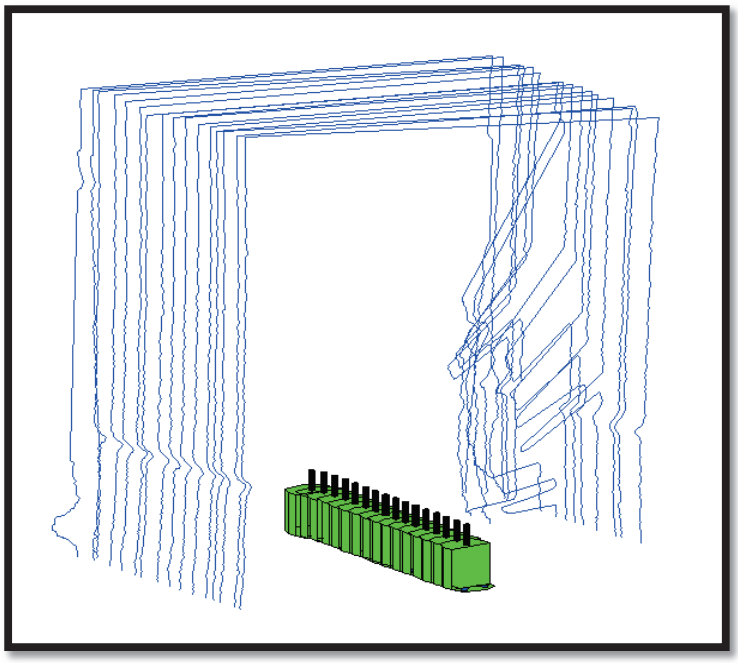


Fig. 4.21 Mapa del entorno realizado con líneas (vista frontal) con la presencia de una persona.

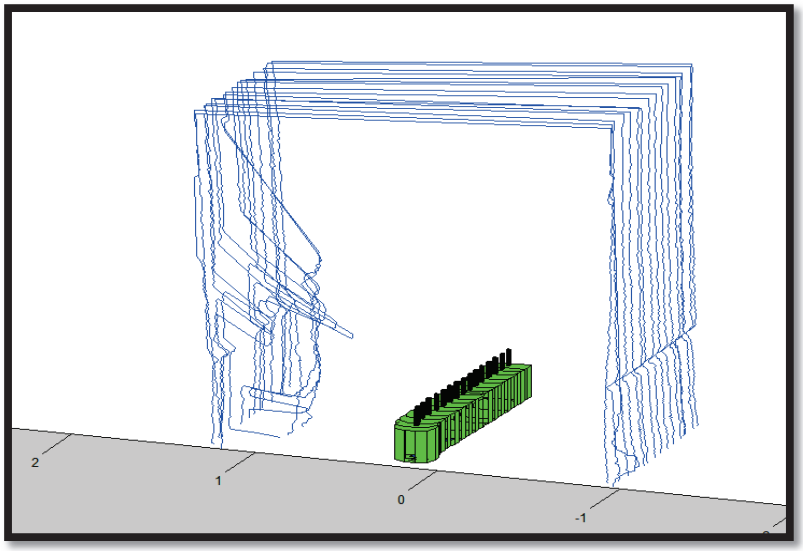


Fig. 4.22 Mapa del entorno realizado con líneas (vista posterior) con la presencia de una persona.

Las Fig. 4.23, Fig. 4.25 y Fig. 4.26 muestran mapas con mayor cantidad de detalles, en los mismos se puede observar las siguientes características:

- Luminaria de techo: se aprecia la forma rectangular de la luminaria en la Fig. 4.23 y Fig. 4.26, mientras que en la Fig. 4.24 se distingue que la luminaria está empotrada en el techo, por lo que tiene una diferente altura.
- Paredes y techo del pasillo: en las tres figuras se observa la forma del pasillo, por lo que se ubica fácilmente las paredes y el techo que lo conforman.
- Puerta y techo de una oficina adyacente: En la Fig. 4.26 se aprecia la puerta semi abierta y parte del techo del Laboratorio de Sistemas Digitales que se encuentra junto al pasillo.
- Persona: en las tres imágenes del mapa realizado se distingue la presencia de una persona en el pasillo, en las Fig. 4.23 y Fig. 4.24 se observa a la mencionada persona de perfil, mientras que en la Fig. 4.26 se la puede ver de espaldas con los brazos abiertos.

Las Fig. 4.25 y Fig. 4.27 muestran las imágenes reales del pasillo mapeado con las características descritas anteriormente.

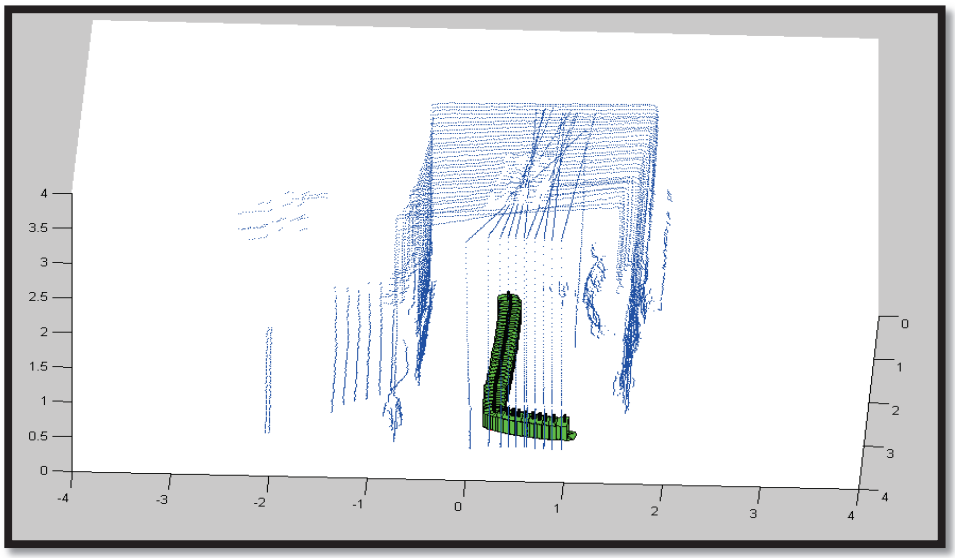


Fig. 4.23 Mapeo del entorno 1.

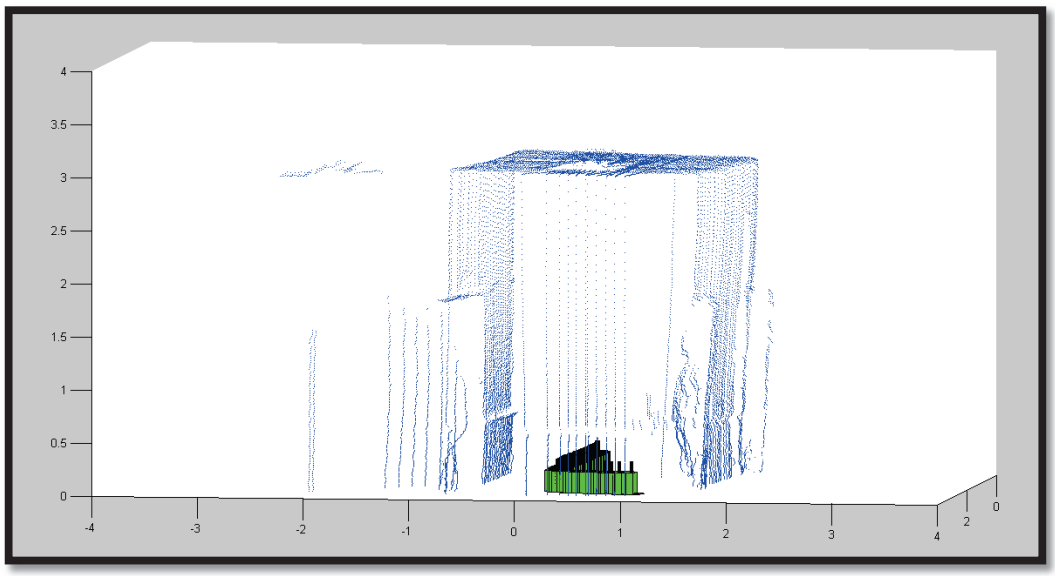


Fig. 4.24 Mapeo del entorno 2.

Fig. 4.25 Imagen del entorno con la presencia de una persona 1.

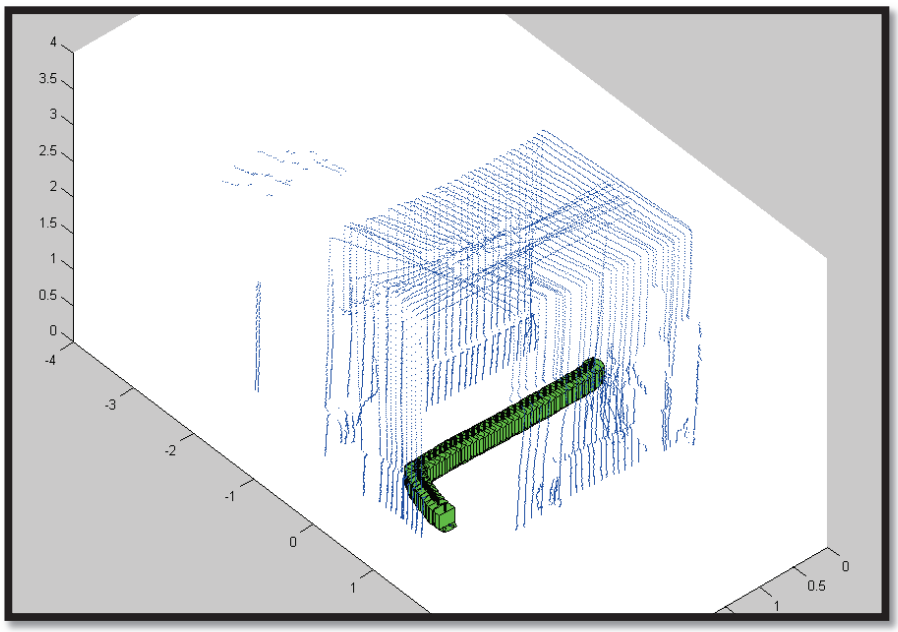


Fig. 4.26 Mapeo del entorno 3.

Fig. 4.27 Imagen del entorno con la presencia de una persona 2.

En la Fig. 4.29 se observa la imagen real en base al que se realizó el mapa de la Fig. 4.28.

Las características que se pueden distinguir corresponden a las de la luminaria colocada sobre el pasillo, las puertas cerradas de las oficinas adyacentes, una caja y a una persona que se encuentra levantando los brazos. La Fig. 4.28 presenta bastante detalle, está gran resolución es gracias al láser, el cual posee una precisión de ±1% de la medida.

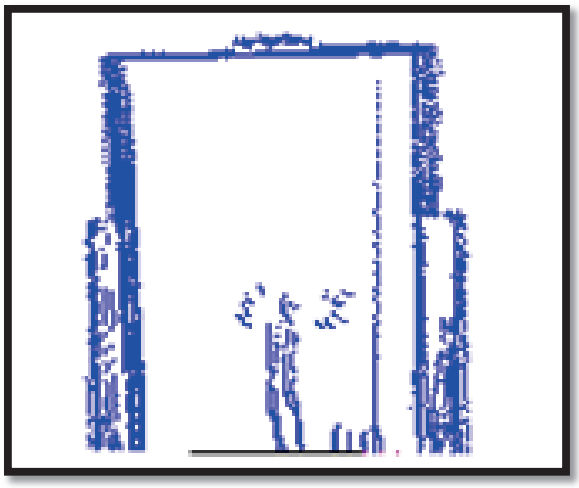


Fig. 4.28 Mapeo del entorno 4.

Fig. 4.29 Imagen del entorno con la presencia de una persona y una caja.

En la Fig.4.30 se muestra el mapa realizado en un entorno de 200 m^2, se puede apreciar que el mapa tiene una leve curvatura, a pesar de que todos los sensores indican que el robot sigue una trayectoria recta, como se mencionó anteriormente

una posible razón es el deterioro de las partes mecánicas del robot como son: llantas, encoder y motores. En la Fig. 4.31 se presenta el pasillo de 200 m^2 mapeado.

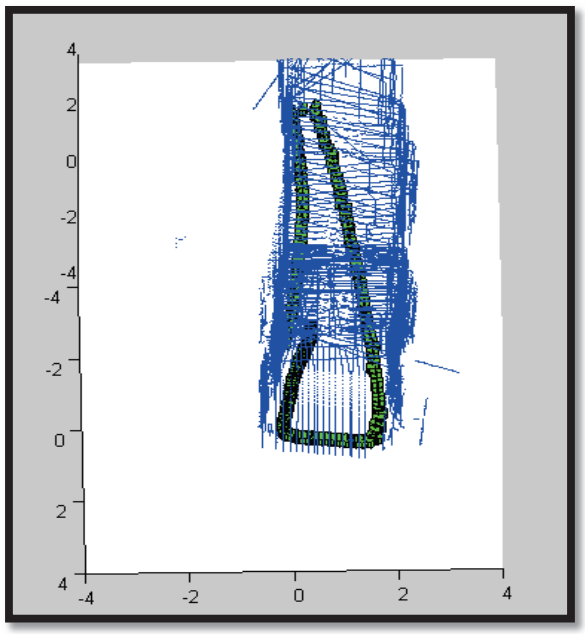


Fig. 4.30 Mapa del entorno en un alcance de 200m^2.

Fig. 4.31 Pasillo de la Facultad de Ingeniería Eléctrica y Electrónica

En la Fig. 4.32 se presenta el mapa realizado por el robot en otro pasillo con diferente arquitectura respecto al pasillo mapeado anteriormente, se puede observar claramente que el mapa tiene muchos más detalles como son las gradas y las

paredes que las rodean. Debido a que se realiza un barrido vertical se tiene la parte frontal de las gradas más no la parte superior. En la Fig. 4.33 se tiene la imagen real del pasillo mapeado.

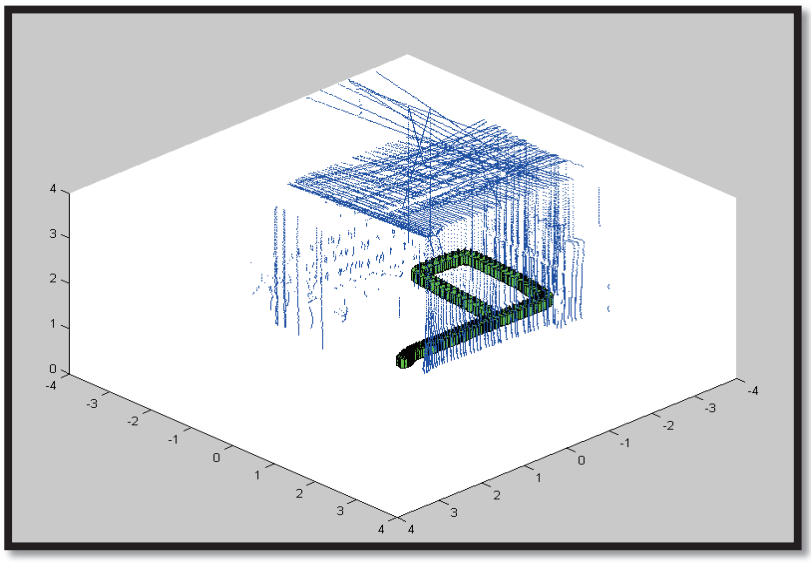


Fig. 4.32 Mapeo del entorno 5.

Fig. 4.33 Imagen del ingreso al pasillo de la FIEE.

4.5 ANÁLISIS DE RESULTADOS DE DETECCIÓN DE MARCAS (LANDMARKS) EN EL ENTORNO

En esta sección se describe los resultados de las pruebas de la detección de marcas del entorno, se escogió detectar paredes y esquinas para dar un realce y complemento al SLAM 3D.

En la Fig. 4.34 se muestra la detección de las paredes del pasillo en color azul y en color rojo se observa las esquinas detectadas, se puede afirmar que el algoritmo de control para detección funciona eficientemente, ya que no se observan esquinas falsas ni tampoco existen paredes en donde no corresponde. La Fig. 4.34 corresponde claramente a la Fig. 4.13 que es la imagen del pasillo mapeado.

La información que entrega el láser es fundamental, ya que el algoritmo se basa en comparación de puntos, además se debe tomar en cuenta que tanto el mapa y la detección es referenciada al eje absoluto que es el punto de partida del robot, con esta aclaración se justifica por que las paredes se encuentran levemente desplazadas.

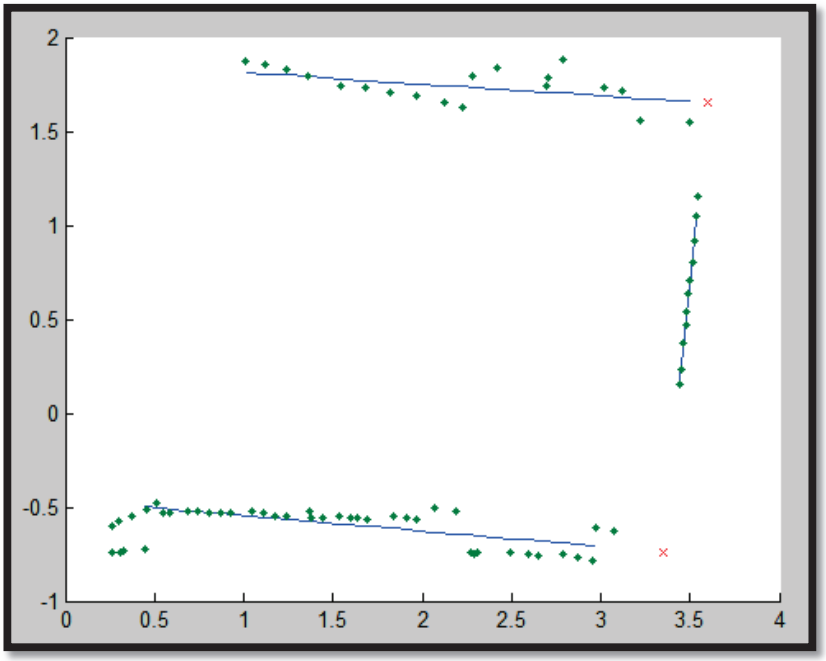


Fig. 4.34 Paredes y esquinas del pasillo

En la Fig.4.35 se observa el pasillo de la Fig. 4.13 con un detalle adicional, éste es que la puerta del Laboratorio de Sistemas Digitales se encuentra abierta, anteriormente se habló sobre los errores que tiene la posición de las esquinas, se mencionó que el error se daba porque existía un desnivel por la puerta cerrada, lo que producía que la nube de puntos se baje un poco y la recta que representa la pared se desvié levemente y la esquina también lo haga, pero en esta imagen se logra apreciar como el algoritmo responde a cabalidad, ya que si tiene la información apropiada éste se ajusta más a la realidad.

Se puede observar que se detectó tres esquinas, la esquina 1 y 2 corresponden a las esquinas del pasillo, en cambio la esquina 3 es un falso positivo, ya que debido a las restricciones del algoritmo se da esta esquina, no se puede eliminarla porque si se observa la distancia del espacio que corresponde a la puerta es similar al espacio que hay entre la pared 1 y la pared 2, y si se restringe más la compuerta del algoritmo no se detectaría la esquina 2.

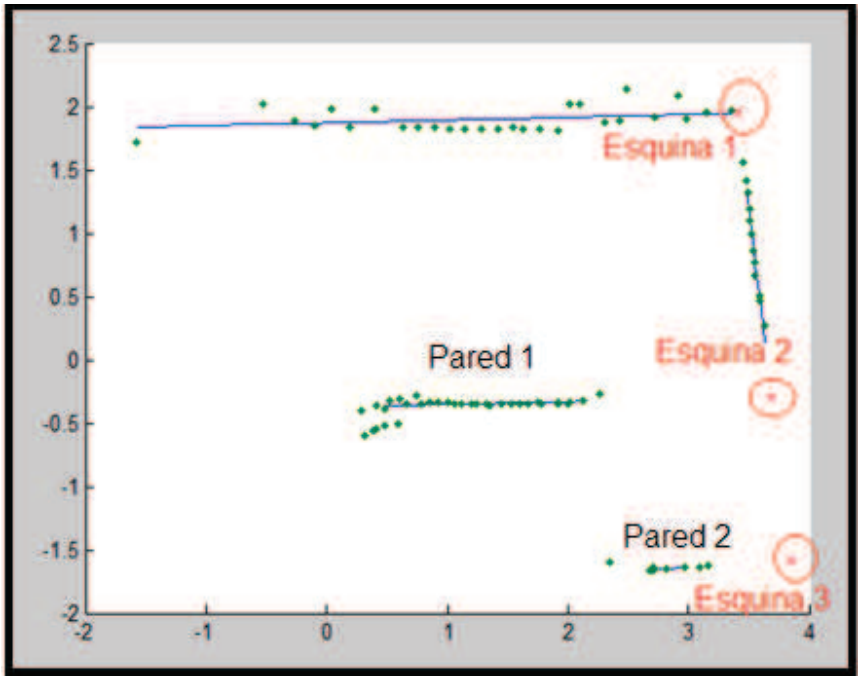


Fig. 4.35 Paredes y esquinas del pasillo con una puerta abierta

4.6 RELACIÓN CON TRABAJOS ANTERIORES

En el presente Proyecto de Titulación se utilizó las herramientas computacionales y accesorios necesarios para conseguir un mapeo con el menor error posible, donde se pueda apreciar la mayor cantidad de características del entorno y que sirva como base para trabajos posteriores, en los cuales se pueda usar una plataforma robótica móvil con sensores de mayor precisión.

Las limitaciones que se tiene al usar los sensores actuales se deben al rango de cobertura, específicamente el sensor NS2, ya que éste no permite que el robot móvil se desplace con total libertad, ya que debe estar dentro del alcance del emisor.

El punto más importante es la localización, razón por la cual se implementó un algoritmo que permita administrar y corregir los datos proporcionados por los sensores. En términos generales, los resultados son bastante aceptables, pero se podrían mejorar utilizando un sensor que posea un menor error y sobretodo que éste no incremente en el tiempo.

La ventaja que posee el SLAM 3D realizado en este Proyecto de Titulación sobre el SLAM 2D radica principalmente en el tipo de detalles que éste puede llegar a adquirir, ya que por ejemplo en un SLAM 2D se puede distinguir características tales como puertas, paredes y esquinas, pero no se puede conocer la altura del entorno, lo que es una limitante cuando se explora un entorno desconocido.

En el SLAM 3D además se puede reconocer características del entorno tales como ventanas, puertas, paredes, luminarias o cualquier otro objeto que se encuentra por sobre el plano que se desplaza el robot, pero por sobre todo, en el SLAM 3D se puede distinguir personas que son la característica más importante que debería ser reconocida en el caso de que un mapeo se realice con el fin de rescate, con éste se podría saber si una persona está presente, en qué posición se encuentra e incluso con la comparación de varios mapas del mismo sitio se podría determinar si dicha persona se encuentra en movimiento, esta tarea la podría realizar una persona entrenada en el reconocimiento de formas características.

Se realizó la detección de marcas (landmarks) en el entorno estructurado, estas marcas consisten en paredes, esquinas y techo. En trabajos anteriores también se realizó esta detección de marcas en base a algoritmos de reconocimiento de imagen, mientras que en este trabajo se realizó mediante un algoritmo de comparación entre puntos adquiridas por el sensor láser.

Finalmente, de todo lo expuesto anteriormente, se puede afirmar que sí se obtuvo excelentes resultados realizando SLAM 3D con un sensor láser, la gran cantidad de mapas con una resolución considerable y errores de posición que no sobrepasan los 30 cm corroboran la factibilidad de desarrollar SLAM 3D con sensor láser.

4.6.1 Trabajos futuros

Los trabajos futuros de SLAM 3D podrán enfocarse en utilizar varios sensores de adquisición de características al mismo tiempo, tales como son las cámaras de video junto con los sensores láser, ya que juntos pueden complementarse entre sí. Los sensores láser entregarían las características de paredes, techo, puertas mientras que la cámara se enfocaría en los objetos o personas. La ventaja de los sensores láser es que éstos podrían funcionar sin la presencia de luz, en cuanto a las cámaras, ofrecerían un mayor detalle como es el color de los objetos y una mayor resolución.

Con una plataforma apropiada se podría ingresar a lugares donde la movilidad sea restringida, para esto se debería tomar en cuenta el nuevo modelo a ser implementado en el EKF. La mejor opción de plataforma sería un quadrotor, ya que éste posee mayor capacidad de desplazamiento.

Adicionalmente, en base a resultados obtenidos en este Proyecto de Titulación, se sugiere como trabajo adicional diseñar una planeación de trayectorias a partir de la información entregada por el mapa en un entorno no estructurado.

CAPÍTULO 5
CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

- La plataforma móvil Robotino® de la compañía Festo, permite ser programada por una variedad de software como son: Java, MATLAB, LabView, RobotinoView, ya que posee un sistema operativo desarrollado en Linux. Esta flexibilidad hace posible que se haya desarrollado este Proyecto de Titulación en MATLAB manejando archivos del tipo script, permitiendo hacer uso del hardware que posee el Robotino®, ya que cuenta con librerías dedicadas a cada uno de sus periféricos.
- El Robotino® es un robot móvil del tipo omnidireccional. Se utiliza el modelo matemático en base a su construcción física y mediante este modelo se pudo calcular las posiciones relativas del Robotino®, acumulando estos datos para posteriormente tratarlos con el Filtro Extendido Kalman (EKF).
- El láser Hokuyo URG-04LX-UG01. obtiene los datos del entorno estructurado. La confiabilidad de los datos entregados por el láser respecto al entorno tienen un margen de error de milímetros, lo que corresponde a las especificaciones dadas por el fabricante (±1% de la medida), siendo el error aportado por éste, mínimo en comparación con los errores obtenidos por los otros sensores involucrados en el mapa final.
- Los datos obtenidos a través de las pruebas realizadas al sistema de localización NorthStar2 (NS2) arrojan resultados nada confiables, debido a que no existe ninguna coherencia entre la posición real del robot en los ejes X y Y, lo que no ocurre con la orientación que para todas las alturas de techo presenta resultados similares a la orientación real. Se utilizó esté ángulo de orientación para el desarrollo de los algoritmos.
- Para el desarrollo del Proyecto de Titulación se unió la característica más confiable del NS2 (ángulo de orientación) y la odometría intrínseca del Robotino®. Con esto se logró obtener errores en los mapas en un rango entre

20 y 30 cm cuando se compara dos mapas del mismo entorno adquiridos consecutivamente. Este error es la resultante de varios factores como por ejemplo error de la medida del láser, error del ángulo del NS2, error de la odometría, la no uniformidad del suelo, desgaste de las ruedas y la posición física del láser sobre el Robotino® el cual no se encuentra exactamente a cero grados.

- La razón por la cual la odometría acarrea un error considerable es debido a que su encoder incremental tiene una resolución por vuelta de 512, mientras que en otras plataformas móviles investigadas tienen una resolución por vuelta de 55184.
- El uso del Filtro Extendido de Kalman (EKF) permitió realizar una corrección de los datos obtenidos de posición por parte de odometría y NS2. Las constantes que se usa en el EKF se obtuvieron mediante prueba y error, ya que usualmente se lo realiza mediante *tuning*. En cuanto a este Proyecto de Titulación se obtuvo resultados muy aceptables, ya que los valores corregidos por el EKF están entre los datos de odometría y el modelo en el caso de la posición (Ver Fig. 4.3 y Fig. 4.4), en el caso de orientación entre el ángulo de NS2 y el modelo (Ver Fig.4.5).
- Haciendo referencia a la teoría de SLAM, se debe obtener un mapa y una localización con una minimización de errores, para ello se usó los sensores propioceptivos y exteroceptivos en conjunto con el EKF se obtuvo un mapa del tipo métrico absoluto y estático.
- La detección de marcas, específicamente paredes, se realizó mediante un algoritmo de comparación de puntos entregados por el láser, ésta comparación se obtuvo mediante el criterio de compuertas dinámicas, esto permitió obtener las ecuaciones que representan al lugar donde se encuentran las paredes del pasillo mapeado. En base a las ecuaciones obtenidas, se realizó un algoritmo para detección de esquinas.
- De todo el Proyecto de Titulación se puede afirmar que es fue factible realizar SLAM 3D con sensor láser (Hokuyo). De todas las pruebas realizadas se

obtuvo errores entre mapas aceptables en un alcance reducido (Capítulo 4) y las características del entorno estructurado son bastante apreciables, con lo que se cumple a cabalidad la teoría de SLAM; con lo que corresponde a la detección de marcas se obtuvo excelentes resultados.

5.2 RECOMENDACIONES

- Verificar que la versión del sistema operativo instalada en la CFcard soporte el accesorio que se desee usar en el Robotino®, caso contrario se deberá realizar una actualización de la tarjeta.
- Se debe tomar muy cuenta la posición física en la que está colocado el trípode emisor del NS2, ya que éste determina los ejes de referencia del ángulo de orientación. No se debe someter a una luz intensa directa sobre el lente del emisor y receptor del NS2, tampoco se debe mirar directamente al emisor, ya que puede ocasionar daños a la salud visual.
- No conectar directamente el láser al puerto USB del Robotino®, porque éste demanda una gran cantidad de corriente y el puerto no es capaz de abastecerla.
- No sacar la CFcard cuando el Robotino® se encuentre en funcionamiento o en proceso de encendido o apagado, ya que éste puede ocasionar daños al equipo.

REFERENCIAS BIBLIOGRÁFICAS

[1] **BURGOS** F, **CERÓN** M, “Diseño y programación de algoritmos de control en robots móviles. Estudio y aplicación a ROBOTINO – FESTO”, Universidad Austral de Chile, 2010

[2] **FESTO,** Página web oficial empresa FESTO S.A, 2012.
www.festo.com.

[3] **FESTO,** 544305 ES/FR, Ralph-Christoph Weber, Festo Didactic GmbH & Co. KG, 73770 Denkendorf, Germany, 2007.

[4] **LEMA** O, “Actualización del sistema operativo, Manual de operación y guía de prácticas para el sistema didáctico robótico móvil del laboratorio de mecatrónica del FIMCP”, Escuela Politécnica del Litoral, Ecuador, 2013.

[5] **FESTO,** ©Didactic GmbH & Co. KG, 73770 Denkendorf, Germany, April 2010
http://doc.openrobotino.org/download/RobotinoView/RobotinoView2_ES.pdf

[6] **AUAT** F,” Localización y Reconstrucción Simultánea de Entornos por un Robot Móvil basada en la Navegación Orientada a las Zonas de Máxima Incertidumbre”, UNIVERSIDAD NACIONAL DE SAN JUAN, San Juan, 2008

[7] **RODRÍGUEZ DEL RÍO** R, “Gráficas con MatLab”, Departamento de Matemática Aplicada, Universidad Complutense de Madrid.

[8] **ANÓNIMO**, “Introducción a los gráficos de 3 Dimensiones”. Departamento de Matemática Aplicada, Universidad Politécnica de Madrid.
http://www2.caminos.upm.es/departamentos/matematicas/Fdistancia/PIE/matlab/temasmatlab/TEMA%206.pdf

[9] **HOKUYO**, Scanning Laser Range Finder URG-04LX-UG01 (Simple-URG) Specifications

[10] **ANÓNIMO,** “Sensores de Posición”**.**
http://www.info-ab.uclm.es/labelec/Solar/Componentes/SPOSICION.htm

[11] **BAEDE** T.A, “Motion control of an omnidirectional mobile robot”, National University of Singapore Faculty of Engineering Department of Mechanical Engineering Control and Mechatronics Group, Eindhoven, September 18th, 2006

[12] **FESTO,** ©NorthstarTM Sensor 2 Manual, Página web oficial empresa FESTO S.A, 2012.

[13] **FESTO,** ©Didactic GmbH & Co. KG, 73770 Denkendorf, 2009
www.festo-didactic.com

[14] **MUÑOZ** P,”Aplicación del Filtro de Kalman al seguimiento de objetos en Secuencias De Imágenes”, Ingeniería técnica en informática de sistemas, Universidad del Rey Juan Carlos, 2002-2003.

[15] **CAMARENA** J. A., “El filtro de Kalman”,University of Michigan, FIE.

[16] Universidad Carlos III de Madrid, Doctorado en Tecnologías de las Comunicaciones - Procesado Digital de Señales en Comunicaciones (Curso 2003/04)

[17] **ORTIZ** G, “Introducción al filtro de Kalman”.

[18] **ARCOS,** J, “Sistemas de navegación y modelado del entorno para un robot móvil”, Universidad Politécnica de Madrid, Noviembre 2009.

[19] **ELECTROMATE,** “Encoder MR, type L, 256-1024CPT, Channels, with Line Driver”.
http://www.electromate.com/ftp/public/Maxon%20Product%20Catalogs/Maxon%20Catalog%202009-2010/265.pdf

[20] **DURRANT-WHYTE** H, **BAILEY** T, “Simultaneous Localization and Mapping: Part 1”, IEEE Robotics & Automatization Magazine.

[21] **GONZÁLEZ** R, “Simutaneous Localization and Mapping”, Universidad de Almeria, España, Noviembre 2007

[22] **RICCILLO** M, “Robots que juegan al fútbol”, Universidad autónoma de México, http://www.comoves.unam.mx/numeros/articulo/163/robots-que-juegan-al-futbol

[23] **ANOMIMO**, “Métodos Estadísticos en Ciencias de la Vida”, Argentina.

http://www.bioingenieria.edu.ar/academica/catedras/metestad/Cadenas%20de%20Markov-1.pdf

[24] **TEREJANU** G, “Extended Kalman Filter Tutorial”, Department of Computer Science and Engineering, University at Buffalo, Buffalo, NY 14260.

[25] **BRENNEKE** C**, WULF** O**, WAGNER** B, “Using 3D Laser Range Data for SLAM in Outdoor Environments”, Institute for Systems Engineering, University of Hannover, Germany.

ANEXOS

ANEXO A
MANUAL DE USUARIO

MANUAL DE USUARIO

Este manual busca ser una guía para una correcta utilización del Robotino® con el fin de realizar un SLAM en 3D, para lo cual se debe tener el hardware y software apropiado y correctamente instalado.

Instalación de Hardware

El láser debe ser colocado sobre el puente del Robotino® de modo que el barrido se realice de forma vertical, para esto se utiliza los accesorios incluidos con el láser, Fig. A.1, la disposición del láser se puede observar en la Fig. A.2.

La conexión del láser debe hacerse mediante un cable de mini USB a USB hacia el hub que es alimentado por las baterías del Robotino®. No se debe conectar directamente al láser hacia la entrada USB del Robotino® ya que la corriente demandada por éste es mayor a la que la entrada USB puede proveer.

Se debe tener en cuenta si el led indicador del sensor láser se encuentra titilando, ya que es señal de que no se reconoce el láser por parte del Robotino®, por lo que se debe desconectar y conectar de nuevo hasta que la luz permanezca encendida.

El sensor NS2 receptor es colocado en la parte de atrás del puente del Robotino® como se puede apreciar en la Fig. A.2, el cable utilizado es de mini USB a USB y es conectado directamente a la entrada USB del Robotino®.

El sensor NS2 emisor debe estar a una distancia apropiada del receptor NS2.

Adicionalmente se debe colocar una tarjeta CF card de 1 o 4 GB con la actualización 2.4 del sistema operativo.

Para la comunicación con el Robotino® se requiere un computador que pueda conectarse a redes inalámbricas.

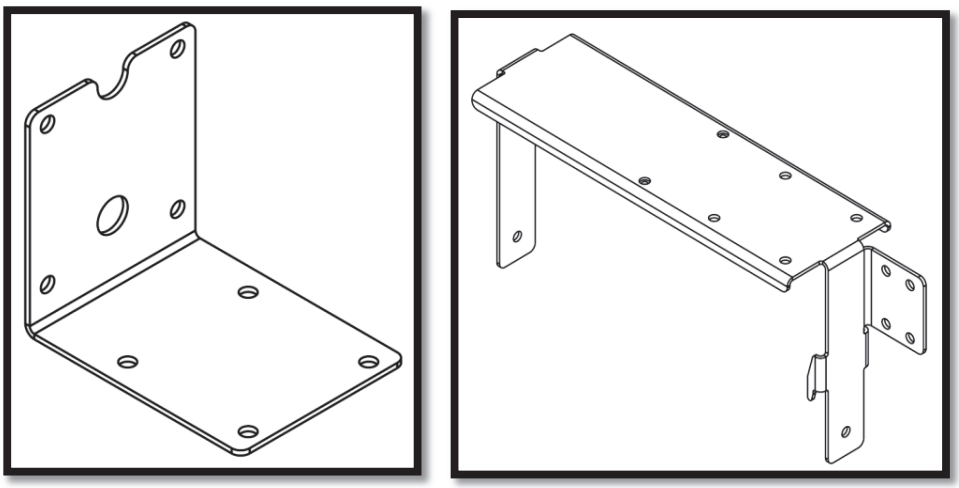


Fig. A.1 Accesorios usados para colocar el sensor láser sobre el Robotino®

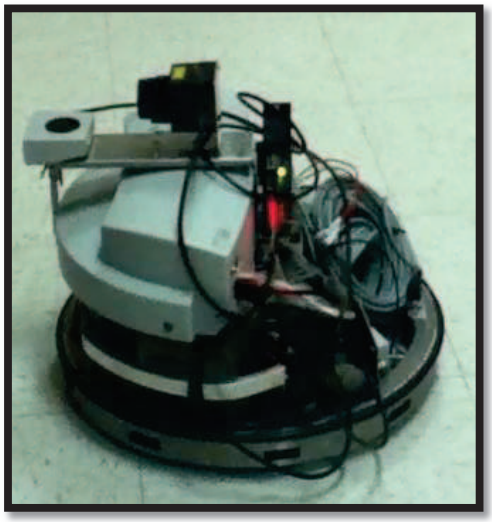


Fig. A.2 Disposición de sensor láser y NS2 sobre Robotino®

Instalación del software.

Al ingresar a la página de Robotino wiki:

http://wiki.openrobotino.org/index.php?title=Downloads (Ver Fig. A.3.)

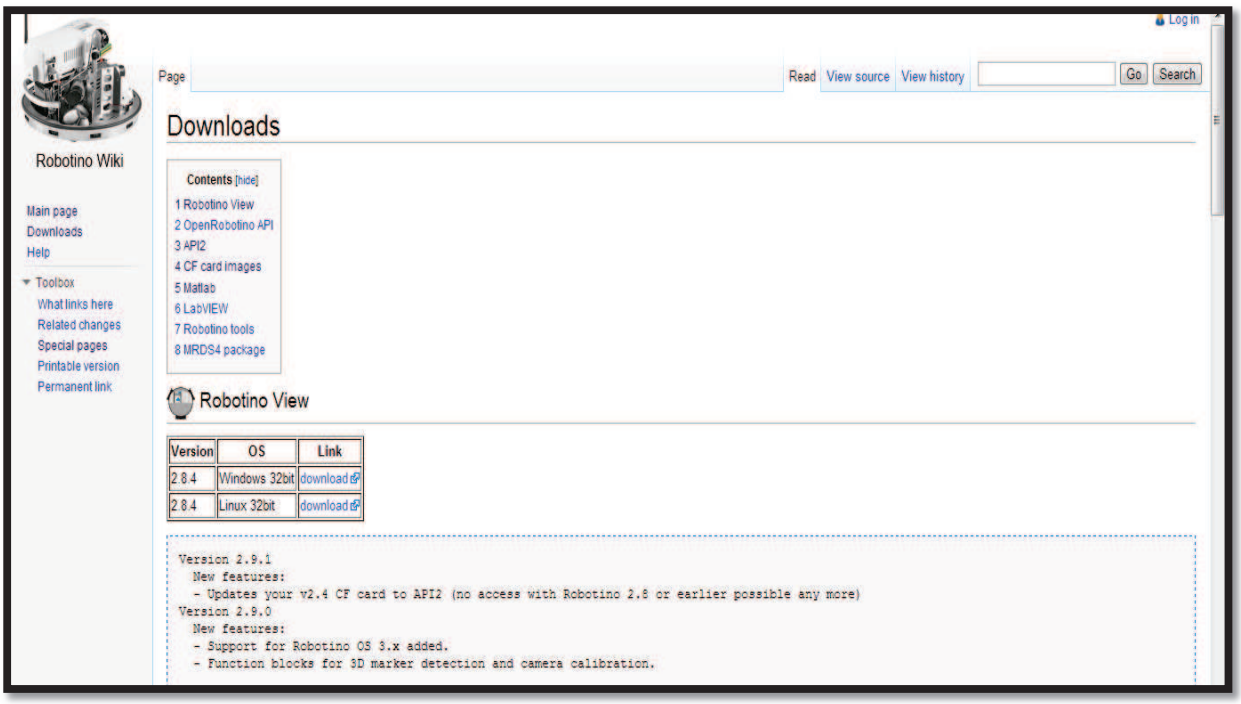


Fig. A.3 Captura de pantalla de página de descargas de software para Robotino®.

Se puede observar los diferentes links de descarga, en este caso se debe seleccionar el relacionado con MATLAB (Ver Fig. A.4), debido a que este programa será utilizado para la programación del Robotino®.

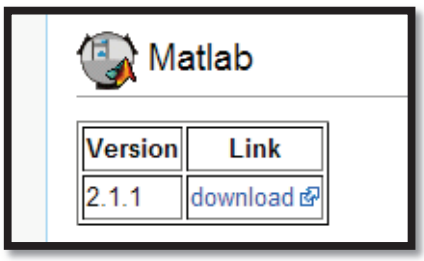


Fig. A.4 Link de descarga de componentes para manejo con MATLAB

Se debe descargar también el API2 (Ver Fig. A.5).

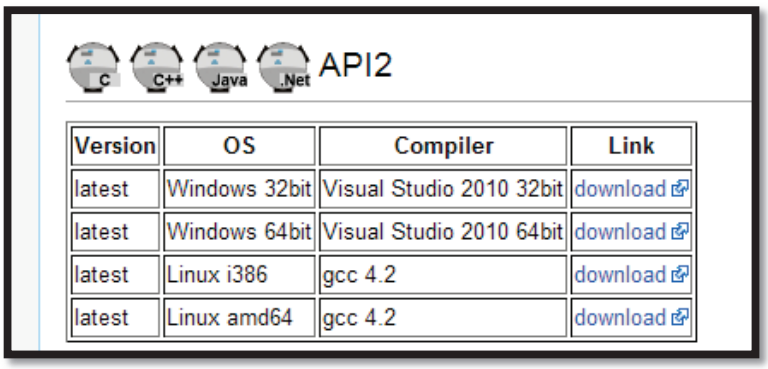


Fig. A.5 Links de descarga de API2

Adicionalmente se debe instalar en el computador el programa MATLAB, así como el compilador Visual Studio, en este caso se tiene las versiones 11b y 10 respectivamente.

Una vez descargado los toolboxes y blocksets de MatLab se procede a su instalación como también el API2.

El siguiente paso consiste en configurar como compilador predeterminado al Visual Studio 2010 para esto se debe:

Sobre el Command Window de MATLAB escribir el comando:

```
>> mex –setup
```

Se selecciona la opción [y] para observar los compiladores existentes en el equipo, (Ver Fig. A.6).

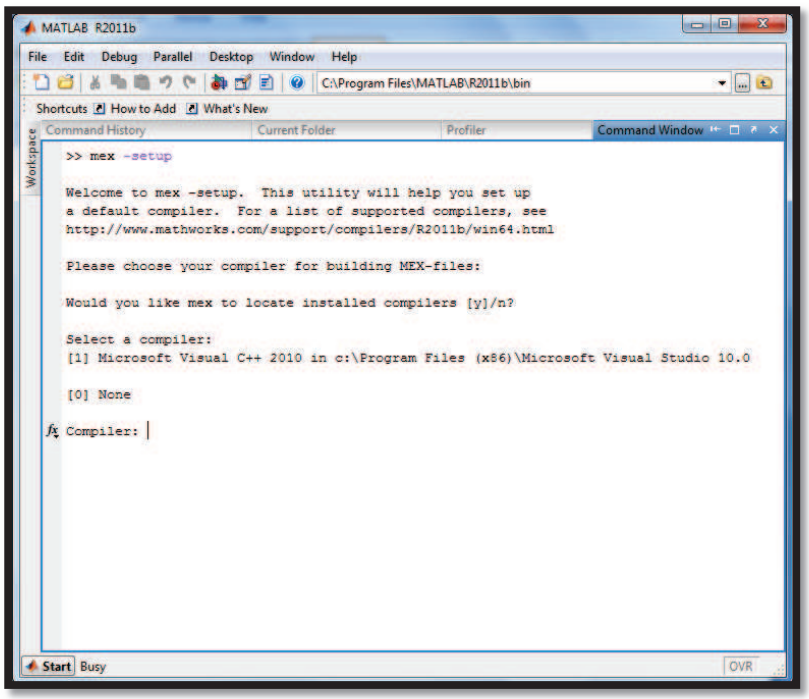


Fig. A.6 Compiladores existentes en el computador.

Se selecciona el compilador deseado escribiendo el número correspondiente y luego aparece la pantalla mostrada en la Fig. A.7.

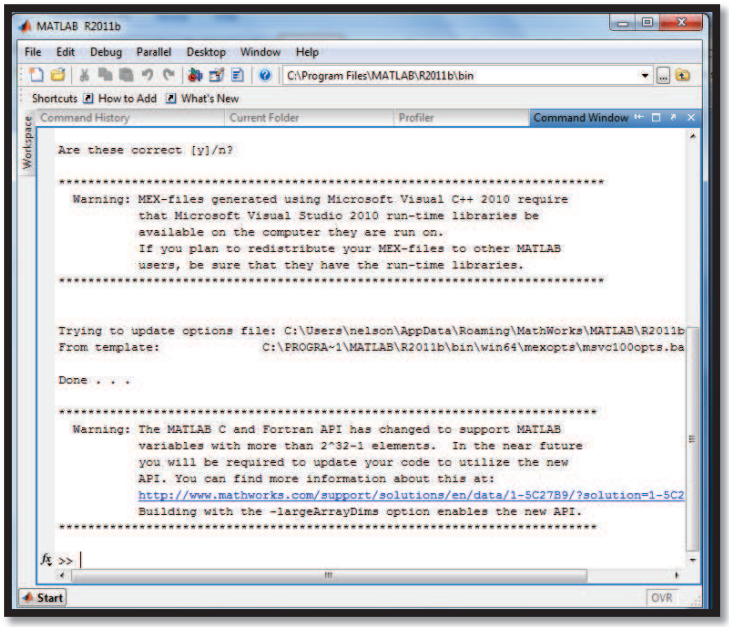


Fig. A.7 Mensaje cuando se elige el compilador deseado.

Después de terminados estos pasos se abre MATLAB y se selecciona como “espacio de trabajo” la carpeta de instalación de los controladores ‘ROBOTINOMATLAB_DIR’. Además se ejecutan los comandos:

```
>>addpath(strcat( getenv('ROBOTINOMATLAB_DIR'), '/blockset' ) );

>>addpath(strcat( getenv('ROBOTINOMATLAB_DIR'), '/toolbox' ) );

>>startup;
```

Para agregar los toolboxes y blocksets que permitan usar al Robotino® por medio de programas elaborados en MATLAB.

Actualización a sistema operativo 2.4 api2

Los Robotino® que se poseen en el laboratorio de control tenían el sistema operativo V2.0 en las compact flash, con el mismo no se podía manejar ciertos accesorios que se necesitan para el desarrollo de este trabajo, como son los sensores láser y el NorthStar, para estose debió actualizar las CF cards.

Requisitos para la actualización:

Se debe tener un programador de CF card r/w, una computadora instalada Windows Xp SP3 y descargar de la página oficial de Festo el Robotino®_CF_imagenv24, el cual es el archivo donde se encuentra el sistema operativo, en este caso se usa el sistema operativo V2.4, es recomendable tener las tarjetas de 1gb o 4gb como máximo.

Pasos a seguir:

- Extraer el archivo imagen usando WinZip o 7-zip (VerFig. A.8).

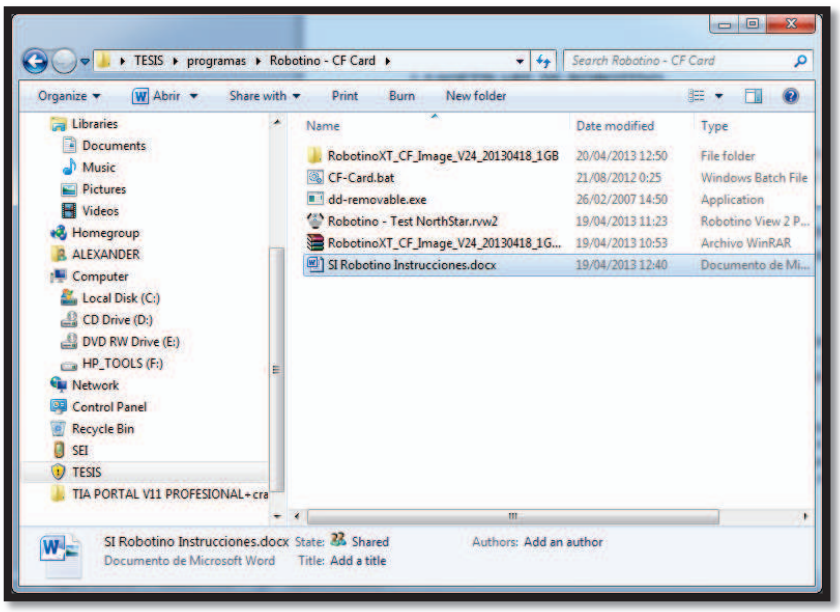


Fig. A.8 Extracción del archivo imagen.

- Correr el archivo CF-Card.bat que se encuentra en el directorio

Ya terminados todos los procesos del programa la actualización fue hecha y se puede contar con las diferentes mejoras que son mencionadas en la misma página donde se realiza la descarga.

Encendido

Para encender al Robotino® se debe mantener presionado el botón marcha/paro del teclado de membrana. El Robotino® se tardará unos minutos en inicializarse.

Dentro del menú se debe escoger la IP que llevará el Robotino®, es recomendable asignarle una IP estática, la misma que debe coincidir con la dirección escrita en el programa de MATLAB.

Apagado

Para el apagado del Robotino®, es recomendable navegar en el menú y elegir la opción apagar.

Ejecución del programa

Una vez que se tiene el hardware y software instalado, se puede ejecutar el programa, para esto se debe tener en cuenta que el programa principal como las funciones necesarias se encuentre en la misma carpeta.

Se procede a conectar al computador con el Robotino® vía wireless, debido a que el Robotino® genera su propia red se debe seleccionar la misma cuyo nombre es: RobotinoAPX.1.

Cuando ya se ha realizado la conexión se verifica que la dirección IP coincida con la indicada en el display del Robotino®, se procede a ejecutar el programa presionando el ícono de la Fig. A.9.

Fig. A.9 Ícono para ejecución de programa.

Aparecerá un mensaje en la pantalla en el que se debe elegir la opción add to path que se muestra en la Fig. A.10.

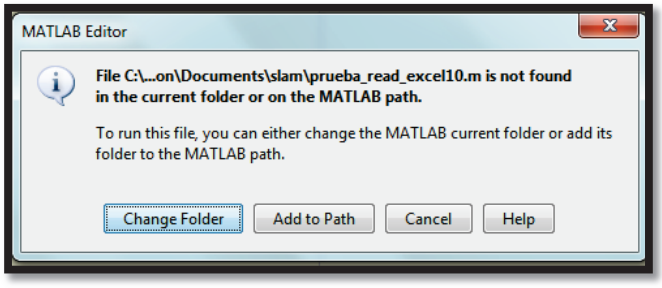


Fig. A.10 Mensaje mostrado al ejecutar el programa.

Al escoger la opción add to path se direcciona a MATLAB a la carpeta donde se encuentra el archivo .m que contiene el código del programa.

ANEXO B

TABLAS DE ERRORES

NS2

TABLAS DE ERRORES NS2

Tabla B.1 Datos de posición y orientación de NS2 y datos reales para una altura de techo de 1700 mm.

x real [mm]	y real [mm]	theta real [deg]	x ns2 [mm]	y ns2 [mm]	theta ns2 [deg]	error x [%]	error y [%]	error theta [%]	relación x real/ns2	relación y real/ns2
300	300	0	136	275	4,3	54,7	8,3	0,00	2,21	1,09
300	600	0	127	449	4	57,7	25,2	0,00	2,36	1,34
300	900	90	104	653	96,42	65,3	27,4	-7,13	2,88	1,38
600	300	90	341	159	90,98	43,2	47,0	-1,09	1,76	1,89
600	600	90	269	389	90,03	55,2	35,2	-0,03	2,23	1,54
600	900	90	417	508	81,47	30,5	43,6	9,48	1,44	1,77
900	300	-90	553	105	-93,91	38,6	65,0	-4,34	1,63	2,86
900	600	-90	570	226	-101,5	36,7	62,3	-12,78	1,58	2,65
900	900	90	627	225	70,89	30,3	75,0	21,23	1,44	4,00
1200	300	90	772	-41	79,04	35,7	113,7	12,18	1,55	-7,32
1200	600	90	799	28	76,52	33,4	95,3	14,98	1,50	21,43
1200	900	90	819	26	69	31,8	97,1	23,33	1,47	34,62
1500	300	90	906	-7	80,55	39,6	102,3	10,50	1,66	-42,86
1500	600	90	945	-27	78,05	37,0	104,5	13,28	1,59	-22,22
1500	900	90	949	28	71,8	36,7	96,9	20,22	1,58	32,14
1800	300	90	912	-64	81,1	49,3	121,3	9,89	1,97	-4,69
1800	600	90	915	36	79,6	49,2	94,0	11,56	1,97	16,67
1800	900	90	934	-17	72	48,1	101,9	20,00	1,93	-52,94

Tabla B.2 Datos de posición, orientación de Ns2 y posición, orientación reales para una altura de techo de 1800 mm.

x real [mm]	y real [mm]	theta real [deg]	x ns2 [mm]	y ns2 [mm]	theta ns2 [deg]	error x [%]	error y [%]	error theta [%]	relación x real/ns2	relación y real/ns2
300	300	0	148	256	3,5	50,7	14,7	0,00	2,03	1,17
300	600	0	123	430	1,7	59,0	28,3	0,00	2,44	1,40
300	900	90	90	687	98,61	70,0	23,7	-9,57	3,33	1,31
600	300	90	346	206	97	42,3	31,3	-7,78	1,73	1,46
600	600	90	246	399	93,22	59,0	33,5	-3,58	2,44	1,50
600	900	90	450	489	78,6	25,0	45,7	12,67	1,33	1,84
900	300	-90	530	101	-94,8	41,1	66,3	-5,30	1,70	2,97
900	600	-90	547	157	-106	39,2	73,8	-18,18	1,65	3,82
900	900	90	591	159	68,1	34,3	82,3	24,33	1,52	5,66
1200	300	90	745	-19	80	37,9	106,3	11,11	1,61	-15,79
1200	600	90	765	45	76,58	36,3	92,5	14,91	1,57	13,33
1200	900	90	781	44	71,04	34,9	95,1	21,07	1,54	20,45
1500	300	90	855	24	82,98	43,0	92,0	7,80	1,75	12,50
1500	600	90	877	65	80,3	41,5	89,2	10,78	1,71	9,23
1500	900	90	890	42	70,4	40,7	95,3	21,78	1,69	21,43
1800	300	90	929	-36	83,11	48,4	112,0	7,66	1,94	-8,33
1800	600	90	932	-29	75,37	48,2	104,8	16,26	1,93	-20,69
1800	900	90	927	-2	72,13	48,5	100,2	19,86	1,94	-450,00

Tabla B.3 Datos de posición, orientación de Ns2 y posición, orientación reales para una altura de techo de 2000 mm.

x real [mm]	y real [mm]	theta real [deg]	x ns2 [mm]	y ns2 [mm]	theta ns2 [deg]	error x [%]	error y [%]	error theta [%]	relación x real/ns2	relación y real/ns2
300	300	0	115	225	4,80	61,67	25,0	0,00	2,61	1,33
300	600	0	130	424	2,30	56,67	29,3	0,00	2,31	1,42
300	900	90	97	659	98,50	67,67	26,8	-9,44	3,09	1,37
600	300	90	352	173	92,70	41,33	42,3	-3,00	1,70	1,73
600	600	90	267	365	90,41	55,50	39,2	-0,46	2,25	1,64
600	900	90	451	510	79,33	24,83	43,3	11,86	1,33	1,76
900	300	-90	471	100	-92,52	47,67	66,7	-2,80	1,91	3,00
900	600	-90	484	171	-102,92	46,22	71,5	-14,36	1,86	3,51
900	900	90	515	201	73,53	42,78	77,7	18,30	1,75	4,48
1200	300	90	675	-65	77,41	43,75	121,7	13,99	1,78	-4,62
1200	600	90	684	15	75,13	43,00	97,5	16,52	1,75	40,00
1200	900	90	686	85	74,06	42,83	90,6	17,71	1,75	10,59
1500	300	90	778	-15	81,00	48,13	105,0	10,00	1,93	-20,00
1500	600	90	792	67	80,50	47,20	88,8	10,56	1,89	8,96
1500	900	90	772	-22	68,00	48,53	102,4	24,44	1,94	-40,91
1800	300	90	891	7	86,93	50,50	97,7	3,41	2,02	42,86
1800	600	90	919	102	84,00	48,94	83,0	6,67	1,96	5,88
1800	900	90	930	15	74,35	48,33	98,3	17,39	1,94	60,00

Tabla B.4 Datos de posición, orientación de Ns2 y posición, orientación reales para una altura de techo de 2200 mm.

x real [mm]	y real [mm]	theta real [deg]	x ns2 [mm]	y ns2 [mm]	theta ns2 [deg]	error x [%]	error y [%]	error theta [%]	relación x real/ns2	relación y real/ns2
300	300	0	97	210	9,9	67,67	30,00	0,00	3,09	1,43
300	600	0	139	414	3,03	53,67	31,00	0,00	2,16	1,45
300	900	90	92	663	98,4	69,33	26,33	-9,33	3,26	1,36
600	300	90	350	189	94	41,67	37,00	-4,44	1,71	1,59
600	600	90	250	385	92,2	58,33	35,83	-2,40	2,40	1,56
600	900	90	431	502	79,3	28,17	44,22	11,89	1,39	1,79
900	300	-90	425	80	-94,4	52,78	73,33	-4,89	2,12	3,75
900	600	-90	434	119	-107	51,78	80,17	-19,00	2,07	5,04
900	900	90	473	129	68,1	47,44	85,67	24,33	1,90	6,98
1200	300	90	604	-29	81,4	49,67	109,67	9,59	1,99	-10,34
1200	600	90	622	123	74,1	48,17	79,50	17,67	1,93	4,88
1200	900	90	625	18	69,3	47,92	98,00	23,06	1,92	50,00
1500	300	90	708	36	84,3	52,80	88,00	6,33	2,12	8,33
1500	600	90	722	45	79,9	51,87	92,50	11,19	2,08	13,33
1500	900	90	781	42	73,3	47,93	95,33	18,53	1,92	21,43
1800	300	90	927	-23	83,8	48,50	107,67	6,89	1,94	-13,04
1800	600	90	930	-14	76	48,33	102,33	15,56	1,94	-42,86
1800	900	90	931	-26	71,4	48,28	102,89	20,62	1,93	-34,62

Tabla B.5 Datos de posición, orientación de NS2 y posición, orientación reales para una altura de techo de 2400 mm.

x real [mm]	y real [mm]	theta real [deg]	x ns2 [mm]	y ns2 [mm]	theta ns2 [deg]	error x [%]	error y [%]	error theta [%]	relación xreal/ns2	relación y real/ns2
300	300	0	94	188	9,6	68,67	37,33	0,00	3,19	1,60
300	600	0	126	425	2,5	58,00	29,17	0,00	2,38	1,41
300	900	90	54	655	102	82,00	27,22	-13,82	5,56	1,37
600	300	90	350	230	97,2	41,67	23,33	-7,99	1,71	1,30
600	600	90	253	371	90,3	57,83	38,17	-0,33	2,37	1,62
600	900	90	437	495	78,8	27,17	45,00	12,49	1,37	1,82
900	300	-90	393	76	-94,7	56,33	74,67	-5,19	2,29	3,95
900	600	-90	406	125	-105	54,89	79,17	-17,00	2,22	4,80
900	900	90	440	141	70,8	51,11	84,33	21,33	2,05	6,38
1200	300	90	560	-20	81,8	53,33	106,67	9,07	2,14	-15,00
1200	600	90	566	-7	72,5	52,83	101,17	19,48	2,12	-85,71
1200	900	90	579	-25	66	51,75	102,78	26,71	2,07	-36,00
1500	300	90	626	23	84,3	58,27	92,33	6,38	2,40	13,04
1500	600	90	661	-9	75,2	55,93	101,50	16,42	2,27	-66,67
1500	900	90	771	31	71,5	48,60	96,56	20,56	1,95	29,03
1800	300	90	909	-5	85,1	49,50	101,67	5,44	1,98	-60,00
1800	600	90	929	-6	77	48,39	101,00	14,44	1,94	-100,00
1800	900	90	936	-25	71	48,00	102,78	21,11	1,92	-36,00

Tabla B.6 Datos de posición, orientación de NS2 y posición, orientación reales para una altura de techo de 2600 mm.

x real [mm]	y real [mm]	theta real [deg]	x ns2 [mm]	y ns2 [mm]	theta ns2 [deg]	error x [%]	error y [%]	error theta [%]	relación x real/ns2	relación yreal/ns2
300	300	0	82	169	7,8	72,67	43,67	0,00	3,66	1,78
300	600	0	126	417	2,9	58,00	30,50	0,00	2,38	1,44
300	900	90	66	679	99,85	78,00	24,56	-10,94	4,55	1,33
600	300	90	353	190	94,34	41,17	36,67	-4,82	1,70	1,58
600	600	90	265	409	91,05	55,83	31,83	-1,17	2,26	1,47
600	900	90	419	503	78,7	30,17	44,11	12,56	1,43	1,79
900	300	-90	364	60	-96	59,56	80,00	-6,68	2,47	5,00
900	600	-90	377	138	-102	58,11	77,00	-13,69	2,39	4,35
900	900	90	420	125	69,64	53,33	86,11	22,62	2,14	7,20
1200	300	90	526	-15	80,78	56,17	105,00	10,24	2,28	-20,00
1200	600	90	536	-15	72,06	55,33	102,50	19,93	2,24	-40,00
1200	900	90	534	-10	68,94	55,50	101,11	23,40	2,25	-90,00
1500	300	90	606	14	81	59,60	95,33	10,00	2,48	21,43
1500	600	90	608	17	77,3	59,47	97,17	14,11	2,47	35,29
1500	900	90	785	5	70	47,67	99,44	22,22	1,91	180,00
1800	300	90	933	-9	85,22	48,17	103,00	5,31	1,93	-33,33
1800	600	90	933	-76	73,42	48,17	112,67	18,42	1,93	-7,89
1800	900	90	922	0	72,01	48,78	100,00	19,99	1,95	#¡DIV/0!

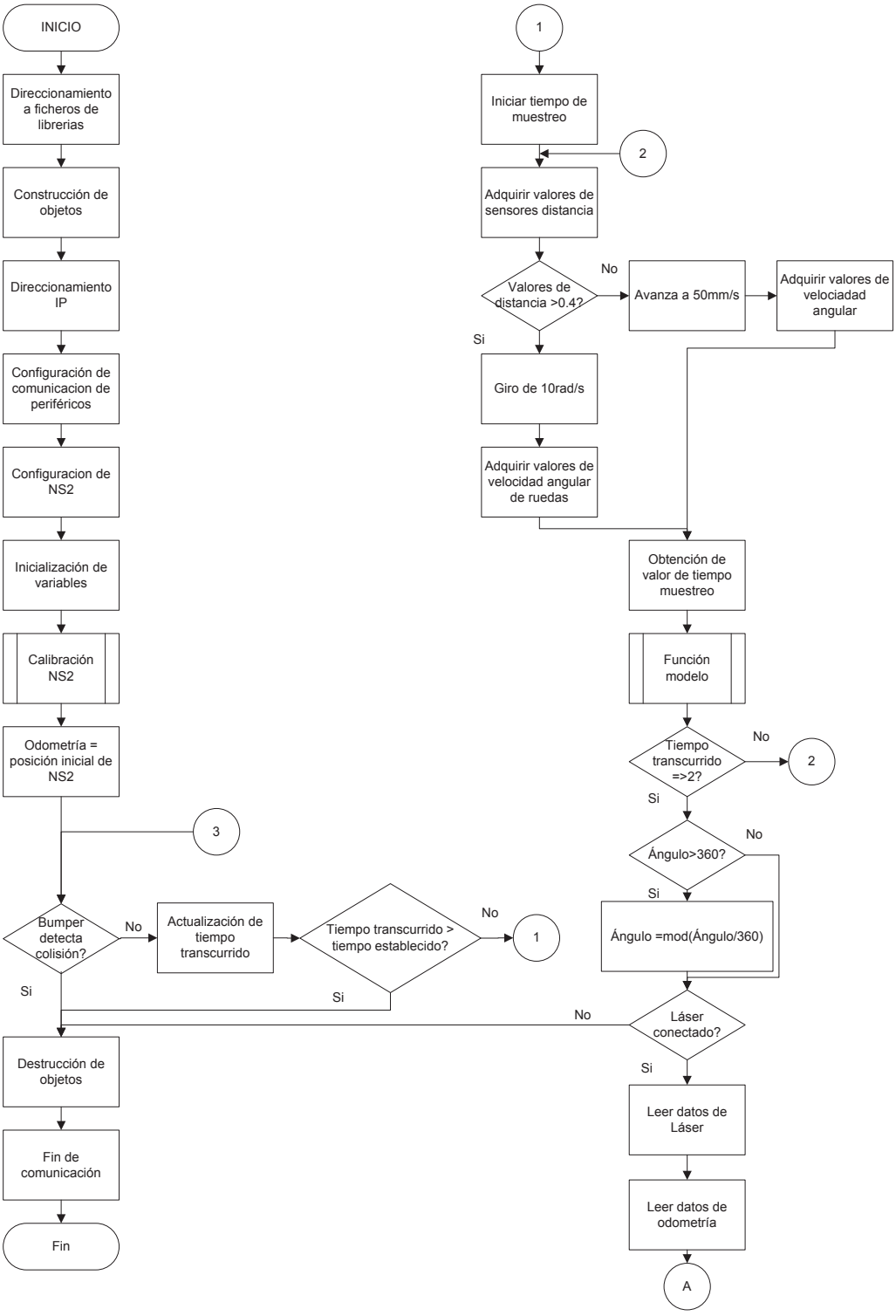
Tabla B.7 Datos de posición, orientación de NS2 y posición, orientación reales para una altura de techo de 2800 mm.

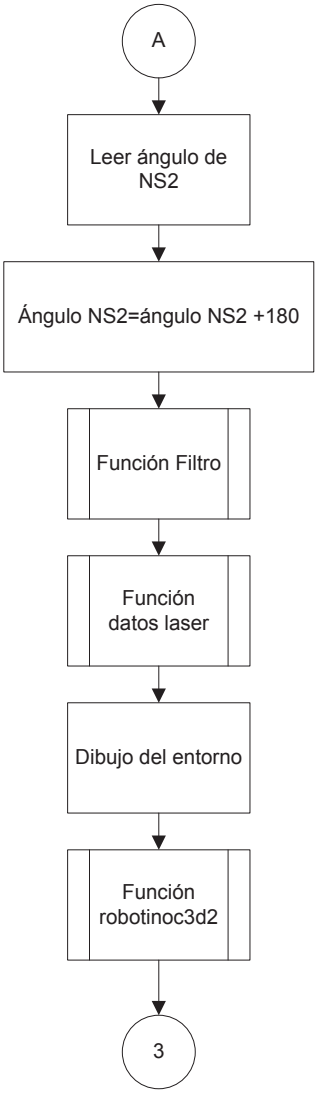
x real [mm]	y real [mm]	theta real [deg]	x ns2 [mm]	y ns2 [mm]	theta ns2 [deg]	error x [%]	error y [%]	error theta [%]	relación x real/ns2	relación y real/ns2
300	300	0	83	159	8,3	72,33	47,00	0,00	3,61	1,89
300	600	0	119	417	8	60,33	30,50	0,00	2,52	1,44
300	900	90	66	685	100,5	78,00	23,89	-11,67	4,55	1,31
600	300	90	349	195	94,59	41,83	35,00	-5,10	1,72	1,54
600	600	90	272	378	90,23	54,67	37,00	-0,26	2,21	1,59
600	900	90	433	501	78,97	27,83	44,33	12,26	1,39	1,80
900	300	-90	355	60	-94,1	60,56	80,00	-4,56	2,54	5,00
900	600	-90	347	115	-104	61,44	80,83	-15,41	2,59	5,22
900	900	90	375	129	71,53	58,33	85,67	20,52	2,40	6,98
1200	300	90	480	-43	77,48	60,00	114,33	13,91	2,50	-6,98
1200	600	90	485	7	74,93	59,58	98,83	16,74	2,47	85,71
1200	900	90	499	12	67,3	58,42	98,67	25,22	2,40	75,00
1500	300	90	563	12	83,52	62,47	96,00	7,20	2,66	25,00
1500	600	90	570	26	78,88	62,00	95,67	12,36	2,63	23,08
1500	900	90	784	44	72	47,73	95,11	20,00	1,91	20,45
1800	300	90	918	-29	85	49,00	109,67	5,56	1,96	-10,34
1800	600	90	930	61	80	48,33	89,83	11,11	1,94	9,84
1800	900	90	906	38	75	49,67	95,78	16,67	1,99	23,68

ANEXO C
DIAGRAMAS DE FLUJO

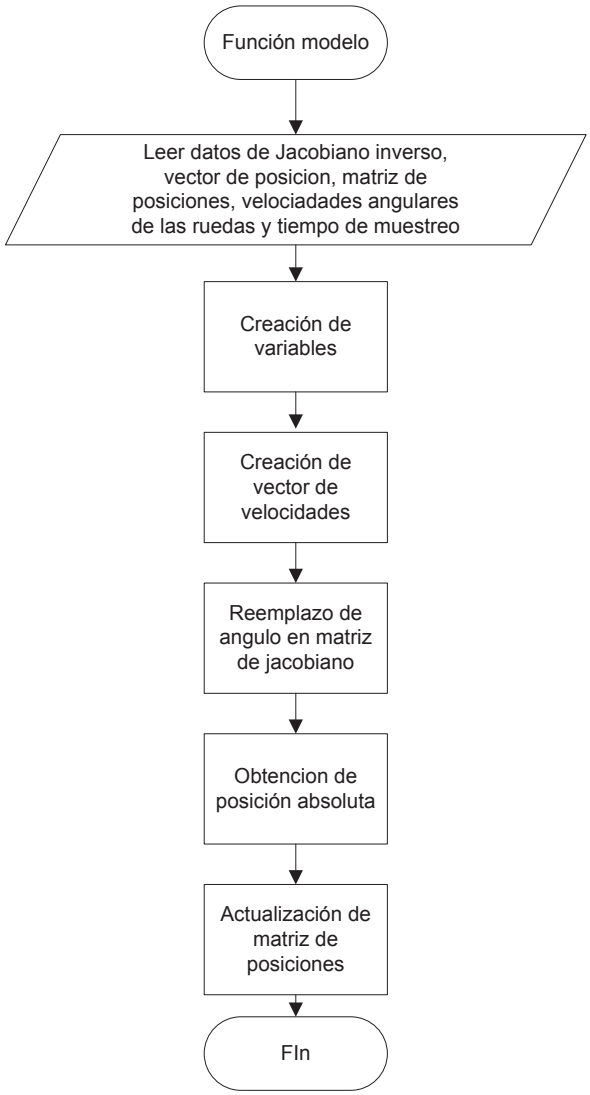
DIAGRAMAS DE FLUJO

PROGRAMA PRINCIPAL

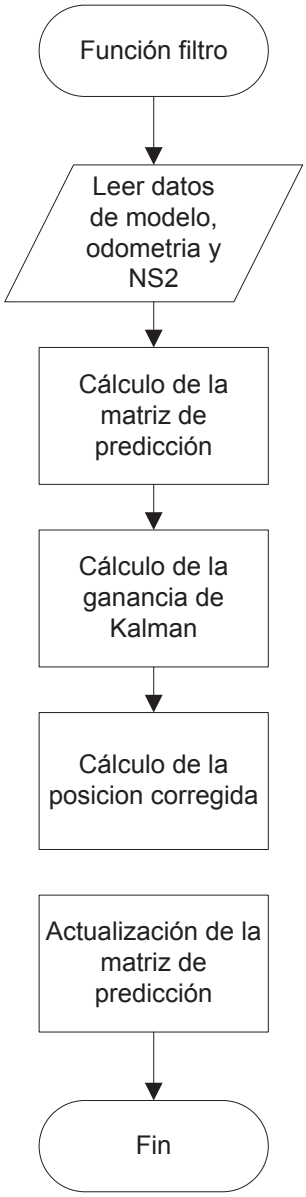



A
Leer ángulo de NS2
Ángulo NS2=ángulo NS2 +180
Función Filtro
Función datos laser
Dibujo del entorno
Función robotinoc3d2
3

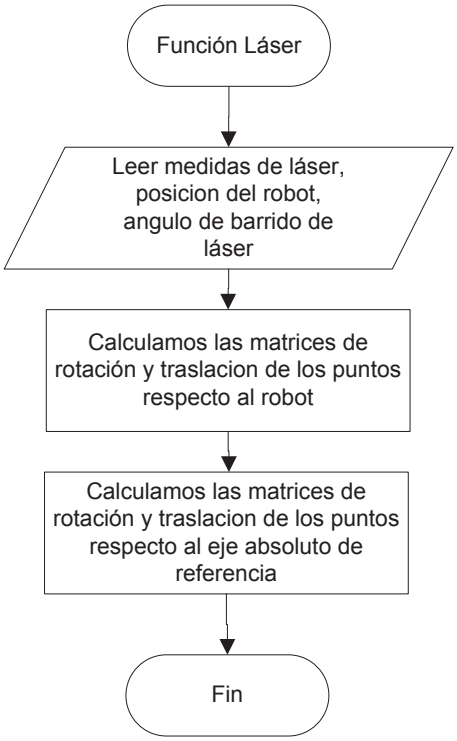
FUNCIÓN MODELO



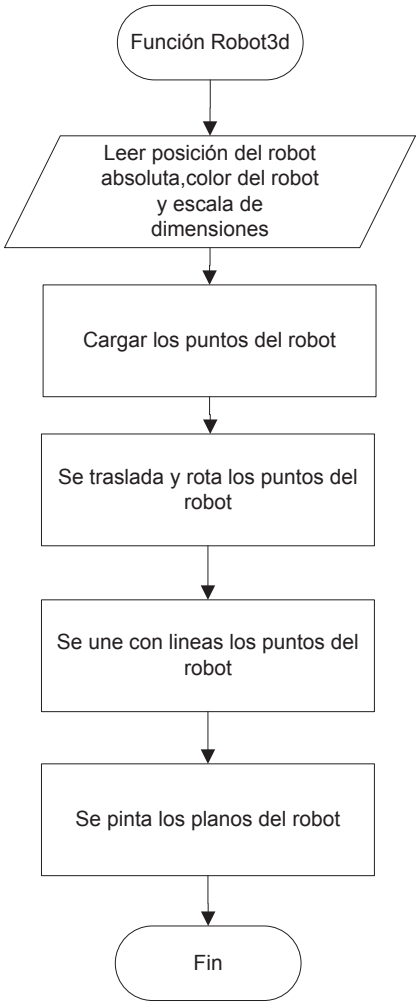
FUNCIÓN FILTRO



FUNCIÓN LÁSER



FUNCIÓN ROBOT3D



FUNCIÓN MARCAS