



**Diseño y construcción de una plataforma móvil para implementación de  
algoritmos de localización empleados en la navegación**

Argoti Terán, Efraín Alexander

Departamento de Ciencias de la Energía y Mecánica

Carrera de Ingeniería en Mecatrónica

Trabajo de titulación, previo a la obtención del título de Ingeniero en Mecatrónica

Ing. Loza Matovelle, David César, MSc.







2 de agosto del 2021

# Curiginal

## Document Information

Analyzed document	Argoti Terán Efrain Alexander_ Tesis.pdf (D111350895)
Submitted	8/16/2021 5:41:00 PM
Submitted by	
Submitter email	dcloza@espe.edu.ec
Similarity	0%
Analysis address	dcloza.espe@analysis.arkund.com

## Sources included in the report

<b>SA</b>	<b>Moya_Carlos_Urkund.pdf</b> Document Moya_Carlos_Urkund.pdf (D29483592)	 2
<b>SA</b>	<b>Universidad de las Fuerzas Armadas ESPE / CHUQUITARCO-QUIMBITA, TESIS SLAM.docx</b> Document CHUQUITARCO-QUIMBITA, TESIS SLAM.docx (D52196501) Submitted by: djmendoza@espe.edu.ec Receiver: djmendoza.espe@analysis.arkund.com	 1
<b>W</b>	URL: <a href="https://docplayer.es/22305099-Departamento-de-ciencias-de-la-energia-y-mecanica-carrera-de-ingenieria-mecatronica.html">https://docplayer.es/22305099-Departamento-de-ciencias-de-la-energia-y-mecanica-carrera-de-ingenieria-mecatronica.html</a> Fetched: 11/18/2020 11:22:01 PM	 1
<b>SA</b>	<b>Universidad de las Fuerzas Armadas ESPE / PROYECTO DE TITULACION SIMBA.pdf</b> Document PROYECTO DE TITULACION SIMBA.pdf (D40111557) Submitted by: dcloza@espe.edu.ec Receiver: dcloza.espe@analysis.arkund.com	 1
<b>W</b>	URL: <a href="https://docplayer.es/58079756-La-version-digital-de-esta-tesis-esta-prottegida-por-la-ley-de-derechos-de-autor-del-ecuador.html">https://docplayer.es/58079756-La-version-digital-de-esta-tesis-esta-prottegida-por-la-ley-de-derechos-de-autor-del-ecuador.html</a> Fetched: 11/20/2019 9:24:36 PM	 1
<b>SA</b>	<b>Universidad de las Fuerzas Armadas ESPE / Tesis Alexis Carrera.pdf</b> Document Tesis Alexis Carrera.pdf (D38703363) Submitted by: wgaguilar@espe.edu.ec Receiver: wgaguilar.espe@analysis.arkund.com	 1



Ing. David César Loza Matovelle  
Director de la Carrera de Ingeniería Mecatrónica



**DEPARTAMENTO DE CIENCIAS DE LA ENERGÍA Y MECÁNICA**  
**CARRERA DE INGENIERÍA EN MECATRÓNICA**

**CERTIFICACIÓN**

Certifico que el trabajo de titulación, “**Diseño y construcción de una plataforma móvil para implementación de algoritmos de localización empleados en la navegación**” fue realizado por el señor **Argoti Terán, Efraín Alexander** el cual ha sido revisado y analizado en su totalidad por la herramienta de verificación de similitud de contenido; por lo tanto cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustente públicamente.

Sangolquí, 16 de agosto de 2021

Firma:

  
  
**Loza Matovelle, David César**  
C. C. 1708661549



**DEPARTAMENTO DE CIENCIAS DE LA ENERGÍA Y MECÁNICA**  
**CARRERA DE INGENIERÍA EN MECATRÓNICA**

**RESPONSABILIDAD DE AUTORÍA**

Yo, **Argoti Terán, Efraín Alexander**, con cédula de ciudadanía n°1723097117, declaro que el contenido, ideas y criterios del trabajo de titulación: **Diseño y construcción de una plataforma móvil para implementación de algoritmos de localización empleados en la navegación** es de mi autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos, y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

**Sangolquí, 16 de agosto de 2021**

Firma:

**Argoti Terán, Efraín Alexander**

C.C.:1723097117



**DEPARTAMENTO DE CIENCIAS DE LA ENERGÍA Y MECÁNICA**  
**CARRERA DE INGENIERÍA EN MECATRÓNICA**

**AUTORIZACIÓN DE PUBLICACIÓN**

Yo **Argoti Terán, Efraín Alexander**, con cédula de ciudadanía n°1723097117, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: **Diseño y construcción de una plataforma móvil para implementación de algoritmos de localización empleados en la navegación** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi responsabilidad.

**Sangolquí, 16 de agosto de 2021**

Firma:

**Argoti Terán, Efraín Alexander**

C.C.:1723097117

### **Dedicatoria**

El trabajo de titulación realizado lo dedico en memoria de mi padre Jaime Efraín Argoti (1946-2019), amigo, consejero y una extraordinario persona. Me diste uno de los mejores regalos al adoptarme como tu hijo, vi el orgullo en tus ojos el día que egrese de mi carrera, la cual logre estudiar por tu gran ayuda con el cambio de carrera. Siempre me mostrarte como un hombre debe comportarse, los modales primordiales que aprendí de ti nunca los olvidare, igual que tus regaños y la forma única que eras conmigo a pesar de no ser fruto de ti. Ansió lograr cuidar a mi familia como tú lo hiciste

Solo espero no haberte decepcionado por no cumplir el plazo que establecimos para titularme. Me has hecho mucha falta padre y siempre me harás falta en el resto de mi vida.

**Efraín Alexander Argoti Terán**

## **Agradecimiento**

En primero agradezco a mi padre Jaime Argoti y mis madres Jacqueline Argoti y Enma Terán, quienes demostraron una gran paciencia a cuidarme, criarme, educarme y apoyarme económicamente con la mayor parte de mis estudios superiores. A mi hermana Joselyn Jácome que tengo el placer de verle crecer y ver como cada día te esfuerzas para recuperarte, ver que esa alegría interna que posees no ha desaparecido, espero siempre ser tu apoyo.

Agradezco a mi tutor Ing. David Loza por guiarme y enseñarme en todo el proceso de titulación. De igual forma agradezco a todos los ingenieros de la Universidad que me aportaron conocimiento dentro y fuera de las horas de clases.

Agradezco a Alisson Quinapallo eres de las mejores personas que conozco me apoyaste en esos momentos tan difíciles cuidando de mi madre cuando no contaba con la ayuda de nadie.

Quiero agradecer a mi segunda familia, mis amigos, herman@s de distinta sangre, que tuve el placer de conocer durante mi trayecto universitario: Gabriel, Walter; Jonathan, Edwin, Kelly y Liseth, quienes compartieron conmigo varias experiencias tanto buenas como malas.

Y concluyendo mis agradecimientos solo quiero acotar, tal vez no tuve inicio de vida feliz, pero eso no me convierte quien soy, sino el resto de mi trayectoria, las lecciones de vida, las personas mencionadas con anterioridad que me guiaron, para saber quién soy, y soy quien yo decida ser, la mejor versión que puedo ser. Por ello finalizo agradeciendo a mis errores que me enseñaron como no debo hacer las cosas, como mejorar porque nadie nace sabiendo todo en la vida se aprende.

**Efraín Alexander Argoti Terán**

## Índice de contenido

Análisis Urkund .....	2
Certificado de director .....	3
Responsabilidad de autoría .....	4
Autorización de publicación .....	5
Dedicatoria .....	6
Agradecimiento .....	7
Índice de contenido .....	8
Índice de Tablas .....	12
Índice de Figuras .....	14
Resumen .....	18
Abstract .....	19
Capítulo I .....	20
Antecedentes .....	20
Antecedentes Internacionales .....	20
Antecedentes Locales .....	23
Planteamiento del problema .....	31
Justificación e importancia .....	31
Objetivos .....	32
Objetivo general .....	32
Objetivos específicos .....	32
Alcance .....	33



Construcción de la plataforma .....	33
Algoritmos para la localización .....	33
Implementación y evaluación .....	34
Estructura del Documento.....	34
Capítulo II Estado del arte .....	35
Robótica Móvil .....	35
Locomoción por ruedas .....	37
Cinemática de un robot móvil de tracción diferencial .....	40
Odometría .....	43
Localización .....	46
SLAM.....	47
Paradigmas del SLAM.....	50
Navegación.....	56
Planificación de trayectoria.....	58
Resumen del Capítulo .....	59
Capítulo III Metodología y Diseño.....	61
Definición de necesidades .....	61
Definición de las características técnicas.....	62
Definición de la funcionalidad del problema .....	63
Fuente de energía (Subsistema 1) .....	66
Sistema de Tracción (Subsistema 2) .....	69

	10
Soporte y Locomoción (Subsistema 3) .....	79
Controlador y procesador (Subsistema 4) .....	98
Algoritmos de Localización (Subsistema 5) .....	103
Técnicas de optimización grafica Graph-SLAM .....	120
Planificador de ruta (Subsistema 6).....	124
Especificaciones Finales del Diseño .....	126
Resumen .....	128
Capítulo IV Construcción y Pruebas .....	130
Simulación de los algoritmos para estimación de localización .....	130
Simulación de Filtro de Kalman Extendido (EKF) .....	131
Simulación Filtro de Partículas .....	132
Simulación Técnicas de optimización grafica Graph-SLAM.....	134
Construcción y funcionamiento de la plataforma móvil.....	135
Ensamblaje de la plataforma móvil .....	135
Software de la plataforma móvil .....	142
Interfaz Humano Maquina. ....	144
Pruebas y Resultados.....	147
Resumen .....	154
Capítulo V Conclusiones, Recomendaciones y Trabajos Futuros.....	156
Conclusiones y Recomendaciones .....	156
Trabajos Futuros.....	158

Bibliografía .....	160
Anexos .....	165

## Índice de Tablas

Tabla 1 Configuración de los sistemas de locomoción con ruedas.....	39
Tabla 2 Tablas de verdad -sentido de giro.....	45
Tabla 3 Necesidades de plataforma móvil.....	62
Tabla 4 Características técnicas de la plataforma móvil.....	63
Tabla 5 Subsistemas planteados de la plataforma móvil.....	65
Tabla 6 Análisis de los criterios para la fuente de Alimentación.....	66
Tabla 7 Ponderación con respecto al criterio Costo.....	67
Tabla 8 Matriz de Holmes de la fuente de alimentación.....	68
Tabla 9 Alimentación del Control.....	69
Tabla 10 Criterios para selección del tipo de Motor.....	70
Tabla 11 Matriz de Holmes para el tipo de Motor.....	72
Tabla 12 Criterio para selección del sensor.....	72
Tabla 13 Descripción matrices modelo dinámico.....	77
Tabla 14 Especificaciones del motor DC 5202 Series Yellow Jacket Planetary Gear Motor.....	79
Tabla 15 Criterios para selección del soporte de la plataforma móvil.....	80
Tabla 16 Matriz de Holmes para el tipo de forma de bases.....	81
Tabla 17 Criterios para selección de locomoción de la plataforma móvil.....	82
Tabla 18 Matriz de Holmes para la locomoción de la plataforma móvil.....	84
Tabla 19 Pesos de Elementos.....	86
Tabla 20 Propiedades del Material.....	87
Tabla 21 Criterios para la selección para el procesador.....	99
Tabla 22 Matriz de Holmes para la selección del procesador.....	99
Tabla 23 Criterios para la selección de tarjeta de control.....	100
Tabla 24 Matriz de Holmes para la selección de tarjeta de control.....	101

Tabla 25 Características Monster Moto Shield .....	101
Tabla 26 Especificaciones de la Tarjeta ESP 32 .....	102
Tabla 27 Especificaciones Intel NUC D54250WYKH.....	103
Tabla 28 Criterios para la selección de Algoritmo de Localización.....	103
Tabla 29 Matriz de Holmes para la selección de algoritmos para la localización .....	105
Tabla 30 Criterios para la selección de Planificador de ruta .....	124
Tabla 31 Matriz de Holmes para Planificador de ruta .....	125
Tabla 32 Módulos de la plataforma móvil .....	127
Tabla 33 Error de localización simulado utilizando EKF .....	132
Tabla 34 Error de localización simulado utilizando Filtro de Partículas .....	133
Tabla 35 Constantes PID motor derecho.....	143
Tabla 36 Constantes PID motor izquierdo. ....	144
Tabla 37 Resultados Odometría-Real.....	148
Tabla 38 Resultados Filtro de Kalman Extendido -Real.....	148
Tabla 39 Resultados Filtro de Partículas -Real.....	148
Tabla 40 Resultados Filtro de Kalman Extendido – Posición deseada.....	150
Tabla 41 Resultados Filtro de Partículas – Posición deseada. ....	150
Tabla 42 Resultados Técnica de Optimización Grafica Graph SLAM - Posición deseada. .....	151
Tabla 43 Resultados Filtro de Kalman Extendido – Trayectoria.....	153
Tabla 44 Resultados Filtro de Partículas – Trayectoria.....	153
Tabla 45 Resultados Técnica de Optimización Grafica Graph SLAM - Trayectoria.....	154

## Índice de Figuras

Figura 1 Método Gmapping realizado por RobChair 2.0.....	21
Figura 2 Esquema del algoritmo SLAM .....	22
Figura 3 Diagrama de flujo de vSLAM .....	23
Figura 4 Pruebas Pioneer P3-DX .....	25
Figura 5 Resultado del mapeo del entorno .....	26
Figura 6 Plataforma realizando SLAM .....	27
Figura 7 MRSLAM .....	28
Figura 8 Interfaz de recuperación de la ubicación .....	29
Figura 9 Trayectoria circular .....	30
Figura 10 Tipos de locomoción en un robot móvil.....	36
Figura 11 Opportunity de la NASA.....	37
Figura 12 Tipo de ruedas convencionales. ....	38
Figura 13 Cinemática -Tracción diferencial.....	40
Figura 14 Tricycle drive Kinematics .....	42
Figura 15 Robot con desplazamiento diferencial - Odometría .....	44
Figura 16 Medición de giro de la rueda.....	46
Figura 17 Crecimiento de la incertidumbre en un robot .....	47
Figura 18 Taxonomía SLAM.....	49
Figura 19 Ilustración de la construcción del gráfico .....	54
Figura 20 Esquema básico de la arquitectura de un robot móvil.....	56
Figura 21 Control de navegación básica.....	57
Figura 22 Planificación de trayectoria .....	59
Figura 23 Diagrama de Funcionalidad .....	64
Figura 24 Diagrama de subfunciones de la plataforma móvil.....	64
Figura 25 Diagrama de subsistemas de la plataforma móvil.....	65

Figura 26 Motor DC .....	71
Figura 27 Cinemática de una plataforma diferencial .....	73
Figura 28 Energía entregada por el sistema .....	75
Figura 29 Fuerzas del sistema .....	76
Figura 30 Formas de base de la plataforma móvil .....	81
Figura 31 Esquema de una plataforma diferencial con una rueda loca.....	83
Figura 32 Esquema de una Plataforma diferencial con dos ruedas locas .....	83
Figura 33 Esquema de ubicación de ruedas omnidireccionales .....	84
Figura 34 Forma de la base de la estructura .....	85
Figura 35 Análisis estático CAE -Esfuerzos Teoría de Von Mises .....	88
Figura 36 Deflexión de la base de la estructura.....	88
Figura 37 Forma geométrica de los acoples del motor .....	89
Figura 38 Análisis estático CAE -Esfuerzos Teoría de Von Mises en acople.....	90
Figura 39 Deflexión de los acoples.....	91
Figura 40 Forma geométrica final de la base de la plataforma móvil .....	92
Figura 41 Análisis estático CAE -Esfuerzos Teoría de Von Mises estructura.....	93
Figura 42 Deflexión de la estructura .....	93
Figura 43 Forma geométrica de la repisa .....	94
Figura 44 Análisis estático CAE -Esfuerzos Teoría de Von Mises Repisa .....	95
Figura 45 Deflexión de la repisa .....	95
Figura 46 Forma geométrica de la lámina para la cámara Kinect .....	96
Figura 47 Análisis estático CAE -Esfuerzos Teoría de Von Mises Soporte superior .....	97
Figura 48 Deflexión del soporte superior .....	97
Figura 49 Iteración entre Sensor y Mover.....	107
Figura 50 Curva Gaussiana.....	108
Figura 51 Corrección – Actualización de medición .....	109

Figura 52 Predicción – Actualización de movimiento .....	110
Figura 53 Multivariable Gaussiana .....	112
Figura 54 Partículas creadas randomicamente.....	117
Figura 55 Posicionamiento del robot en un entorno.....	121
Figura 56 Posición de dos nodos.....	121
Figura 57 Mediciones entre dos posiciones.....	122
Figura 58 Visualización de la anterior con la siguiente .....	122
Figura 59 Representación de los elementos en el algoritmo.....	123
Figura 60 Nodos ROS Filtro de Kalman Extendido - Simulado .....	131
Figura 61 Simulación Gazebo - Algoritmo EKF.....	131
Figura 62 Nodos ROS Filtro de Partículas - Simulado .....	132
Figura 63 Simulación mediante Gazebo con el algoritmo Filtro de Partículas.....	133
Figura 64 Nodos ROS Técnicas de optimización gráfica Graph-SLAM.....	134
Figura 65 Simulación Técnicas de optimización gráfica Graph-SLAM .....	135
Figura 66 Parámetros para realizar el corte laser .....	136
Figura 67 Estructura en acrílico de la plataforma móvil .....	137
Figura 68 Unión base- motores .....	137
Figura 69 Movimiento de la plataforma móvil.....	138
Figura 70 Placa PCB .....	139
Figura 71 Montaje de la placa PCB en la repisa .....	139
Figura 72 Acoples para la estructura de la plataforma móvil.....	140
Figura 73 Parte interna de la plataforma móvil .....	141
Figura 74 Ensamble final de la plataforma móvil .....	141
Figura 75 Nodos ROS – ESP32 .....	142
Figura 76 Ventana principal.....	144
Figura 77 Ventana Teleoperación.....	145



Figura 78 Ventana Planificación .....	146
Figura 79 Prueba en línea recta .....	147
Figura 80 Prueba - Cuadrado 1x1 .....	149
Figura 81 Resultado de la segunda prueba .....	151
Figura 82 Tercera prueba- Trayectoria de trabajo. ....	152
Figura 83 Modificación código planeación. ....	152

## Resumen

Los robots móviles poseen la capacidad de desplazarse dentro de un entorno por medio de sistemas de locomoción, con ayuda de sensores que estiman su localización o posición durante su navegación. En el presente trabajo se realiza el estudio y la implementación de algoritmos para estimar la localización de una plataforma móvil. Una forma común de estimar la localización en plataformas móviles es mediante la odometría, la cual presenta un aumento de incertidumbre a medida que avanza la trayectoria de la plataforma móvil. Para contrarrestar el error de localización se utilizan filtros probabilísticos los cuales son manejados en SLAM y en la navegación.

En SLAM encontramos que la mayoría de sus métodos tiene como base sus tres principales paradigmas, los cuales estudiaremos desde el punto de localización de modo que, tener los filtros probabilísticos (Filtro de Kalman Extendido y Filtro de partículas) y una técnica basada en optimización gráfica (Graph SLAM). El Filtro de Kalman Extendido (EKF) es usado en sistemas no lineales, mediante la aplicación de matrices jacobianas y teorema de Taylor, su fundamento matemático se basa en la curva gaussiana. El filtro de partículas es la creación de partículas por todo el entorno o área de trabajo predeterminado, mientras mayor sea el número de partículas creadas la exactitud de la estimación de localización mejorará; finalmente la técnica de optimización gráfica es una técnica off line la cual recopila la información entregada por la odometría cada cierta distancia y de dicha base de datos procede a evaluar los nodos hasta que convergen.

Para complementar el presente trabajo, se cuenta con un planificador de ruta A\*, el cual genera una secuencia de puntos cartesianos dentro de un entorno predeterminado, donde la plataforma móvil tiene la capacidad de llegar al punto deseado del mismo.

Palabras claves:

- **ESTIMACIÓN DE LOCALIZACIÓN**
- **FILTRO DE KALMAN EXTENDIDO.**
- **FILTRO DE PARTÍCULAS.**
- **TÉCNICA DE OPTIMIZACIÓN GRÁFICA.**
- **PLANIFICADOR DE RUTA.**

## Abstract

Mobile robots can move within an environment through locomotion systems, with the help of sensors that estimate their position or location while navigating. In the present work, the study and implementation of algorithms to estimate the location of a mobile platform are carried out. A common way to estimate location on mobile platforms is through odometry, which presents an increase in uncertainty as the trajectory of the mobile platform advances. To counteract the location error, probabilistic filters are used which are managed in SLAM and navigation.

In SLAM we find that most of its methods are based on its three main paradigms, which we will study from the point of location so that, having the probabilistic filters (Extended Kalman Filter and Particle Filter) and a technique based on graphic optimization (Graph SLAM). The Extended Kalman Filter (EKF) is used in non-linear systems, through the application of Taylor's theorem and Jacobian matrices, its mathematical foundation is based on the Gaussian curve. The particle filter is the creation of particles throughout the environment or predetermined work area, the greater the number of particles created, the accuracy of the location estimate will improve; finally, the graphical optimization technique is an off-line technique that collects the information provided by odometry every certain distance and proceeds from this database to evaluate the nodes until they converge.

To complement the present work, there is an A\* route planner, which generates a sequence of Cartesian points within a predetermined environment, where the mobile platform can reach the desired point of the same.

Keywords:

- **LOCATION ESTIMATION**
- **EXTENDED KALMAN FILTER.**
- **PARTICULATE FILTER.**
- **GRAPHIC OPTIMIZATION TECHNIQUE.**
- **ROUTE PLANNER.**

## Capítulo I

### INTRODUCCIÓN

El capítulo expone la necesidad del estudio de algoritmos para estimar la localización en la navegación, de plataformas móviles en interiores; con lo cual se cubre los paradigmas del SLAM. En el mismo se identificará la importancia y justificación para su desarrollo, investigando trabajos realizados al tema. A continuación, se plantea el problema, los objetivos, el alcance del proyecto enfocado al desarrollo de su tecnología y una breve descripción de su estructura.

#### **Antecedentes**

Se indaga los trabajos más representativos para la temática, donde se realiza un estudio de investigaciones tanto internacional como local. Donde se busca las distintas formas que han realizado los autores para conseguir la localización del robot, aplicando técnicas de SLAM.

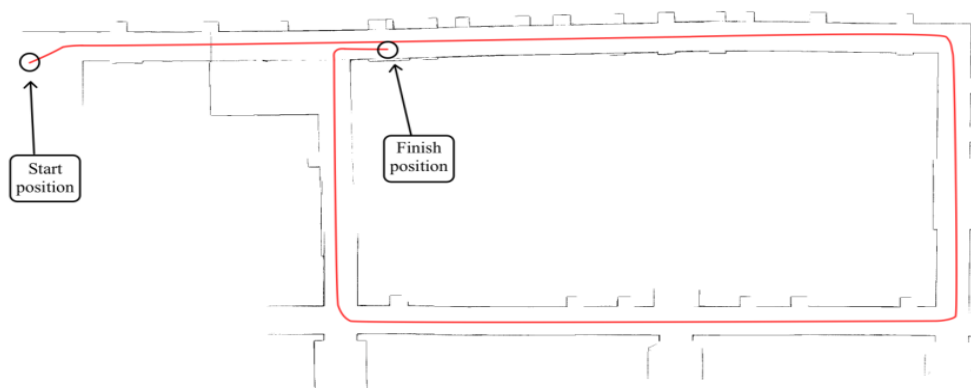
#### ***Antecedentes Internacionales***

Se busca en los trabajos internacionales el método de localización que utilizan en las técnicas de SLAM empleadas. De dicha manera, tener una idea más clara de la forma de localización que tienen los distintos robots.

En (da Silva, 2013) realiza un robot móvil denominado "RobChair 2.0", empleando una interfaz denominada cerebro-computadora (BCI). Mediante un módulo de procesamiento de datos recibe la información sensorial. En el proyecto para efectuar SLAM se utiliza un método gmapping y se procesa mediante nodos programados en ROS en lazo cerrado. SLAM es utilizado en 2D,

## Figura 1

*Método Gmapping realizado por RobChair 2.0*



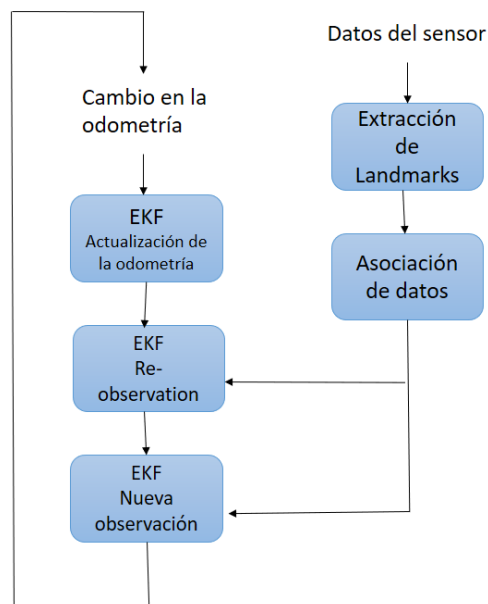
Nota: Resultado del mapeo y trayectoria del RobChair 2.0 (da Silva, 2013)

En la Figura 1 da conocer la trayectoria y el mapeo del RobChair 2.0 aplicando la metodología gmapping en un corredor de la universidad, se detecta el error de cierre de bucle por parte del SLAM aplicado.

En (Torres, 2016), propone un estudio de distintos métodos de SLAM con el propósito de llegar a una comparación entre estos, donde se utilizan tanto métodos de mapeo 2D como en 3D, usando el entorno para estimar la posición del robot. Donde al presentar poca fiabilidad la odometría, para la corrección se utiliza landmarks (puntos de referencia) para la aplicación de Filtro de Kalman Extendido (EKF) esto se encarga de actualizar la estimación de la posición en la que se encuentra el robot móvil.

## Figura 2

### Esquema del algoritmo SLAM



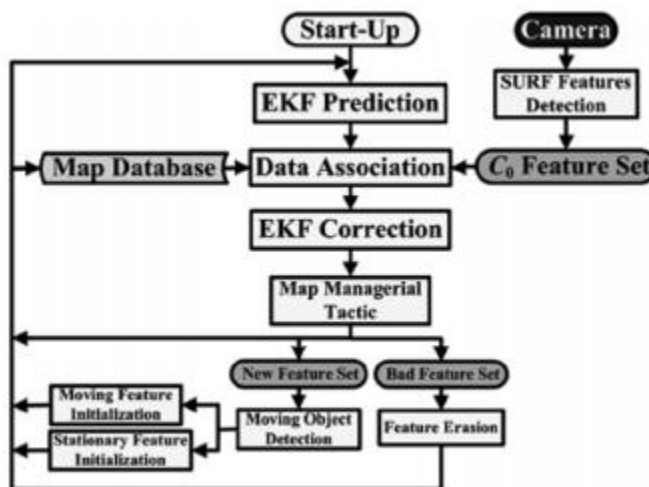
Nota: En la figura se muestra el esquema del algoritmo SLAM empleando EKF (Torres, 2016)

En la Figura 2 se observa un esquema del funcionamiento del algoritmo SLAM de manera general, al moverse el robot el valor de su odometría cambia y su actualización se realiza mediante EKF. (Torres, 2016)

(García, 2019) plantea el uso de vSLAM (Visual Simultaneous Location and Mapping) como herramienta del AR (Augmented Reality), donde se utiliza una cámara monocular móvil para obtener la aplicación en tiempo real. Maneja los datos de la trayectoria para estimar la posición de la cámara; así misma estima la posición relativa de la distancia o profundidad que se puede encontrar un objeto con relación a la posición de la cámara.

**Figura 3**

Diagrama de flujo de vSLAM



Nota: En la figura se muestra el esquema del flujo que sigue vSLAM empleando EKF. (García, 2019)

Observamos en la **Figura 3** como se emplea el EKF para saber la posición de un objeto detectado por una cámara, manipulando EKF como parte de vSLAM. (García, 2019) menciona la existencia métodos para solventar la parte del SLAM, entre los que se incluyen el filtro de partículas, Filtro de Kalman y Graph SLAM.

### **Antecedentes Locales**

El objetivo principal de indagar en los antecedentes locales es buscar la existencia de una investigación sobre la parte de localización en el SLAM. Enfocando la búsqueda en plataformas móviles o robots móviles con aplicaciones en SLAM.

En (Granda & Vásconez, 2012), se desarrolla un sistema de teleoperación para la realización de mapas de entornos mediante navegación difusa en una plataforma PIONER P3-DX. Con el cual se realiza un control de velocidad y programación solo para estimación del mapeo del entorno. La plataforma Pioneer P3-DX posee un software predeterminado MOBILE-EYES con lo que permite realizar su tele operación y un modelo

de su entorno, donde su trayectoria es teleoperada. La plataforma se localiza mediante el uso de la odometría generada por los encoders de los motores.

Presenta las siguientes ventajas y desventajas.

#### Ventajas

- Facilidad de adaptación a situaciones particulares con mínimas variaciones de los parámetros.
- Habilidad para combinar expresiones lingüísticas con datos numéricos.
- No es necesario conocer el modelo matemático del sistema.
- No requiere algoritmos sofisticados para su implementación.

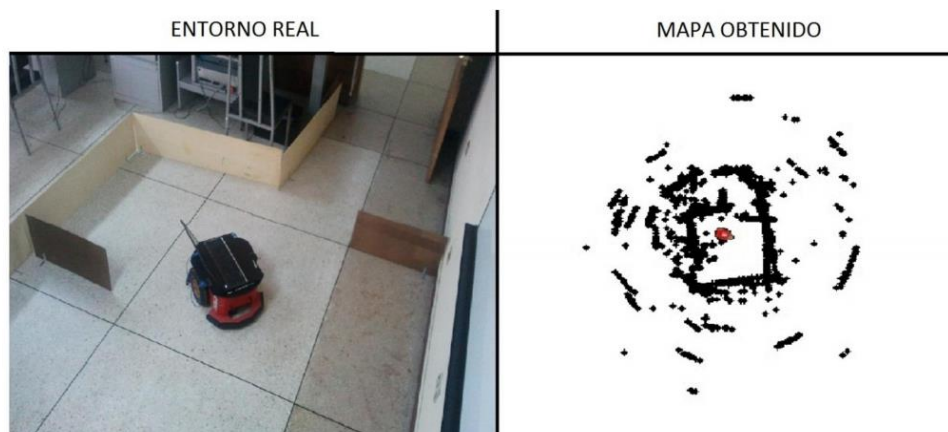
#### Desventajas

- Debido a que ARIA no permite el manejo del comando ARMUTEX con Python, se puede utilizar lenguaje C para el desarrollo de las aplicaciones.
- Para generar los mapas se recomienda usar OpenCv, ya que es una librería liviana y rápida, con esto se evita que las aplicaciones colapsen.
- Se debe verificar las direcciones IP tanto del servidor como del cliente, caso contrario no será posible realizar la comunicación Cliente-Servidor para ejecutar las aplicaciones.
- A su vez las aplicaciones se las realiza en Python 2.7 y debido a esto es preferible correrlas en esta versión ya que para versiones de Python 3.X la sintaxis de programación varía.



## Figura 4

### Pruebas Pioneer P3-DX



Nota: Resultado del Pioneer P3-DX sometido a pruebas. (Granda & Vásquez, 2012)

En la Figura 4 se muestra el mapeo realizado por la plataforma Pioneer P3-DX donde se denota el entorno obtenido de una habitación vacía de paredes de hormigón y madera.

En (Gallardo & Gaona, 2015) se elabora una plataforma robótica móvil “SIMBA” en el departamento de Energía y Mecánica de la universidad, consta de motores a pasos configurados para un desplazamiento diferencial, la cual puede ser teleoperada mediante un mando remoto, con forma de octágono.

Presenta las siguientes ventajas y desventajas.

#### Ventajas

- Controles independientes de cada motor
- Los motores no requieren de sensores para medir la velocidad de rotación
- Las baterías tienen capacidad de almacenamiento de energía

#### Desventajas

- Los motores deben tener las mismas características operativas para su desplazamiento diferencial.
- Los motores son de difícil adquisición

- Peso considerable de las baterías

## Figura 5

*Resultado del mapeo del entorno*



Nota: SIMBA resultados obtenidos. (Gallardo & Gaona, 2015)

En la Figura 5 se muestra el prototipo SIMBA versión 1 en su modo octogonal y el resultado del mapeo en 2D. Se realiza una plataforma robótica para exteriores en (Garzón & Obando, 2016), donde se utiliza el SLAM para realizar la navegación del robot enfocándose en la robótica móvil integrando un GPS. Dicha plataforma realizada en acero es teleoperada y monitoreada por una computadora, con un sensor se maneja un LIDAR, utilizado para la recepción de datos de localización.

Presenta las siguientes ventajas y desventajas.

### Ventajas

- GPS permite saber la ubicación en todo el planeta.
- La posición del GPS no se ve afectada por cambios de altura.
- El sensor LIDAR tiene respuestas inmediatas
- La cámara Web posee conexión Plug and Play

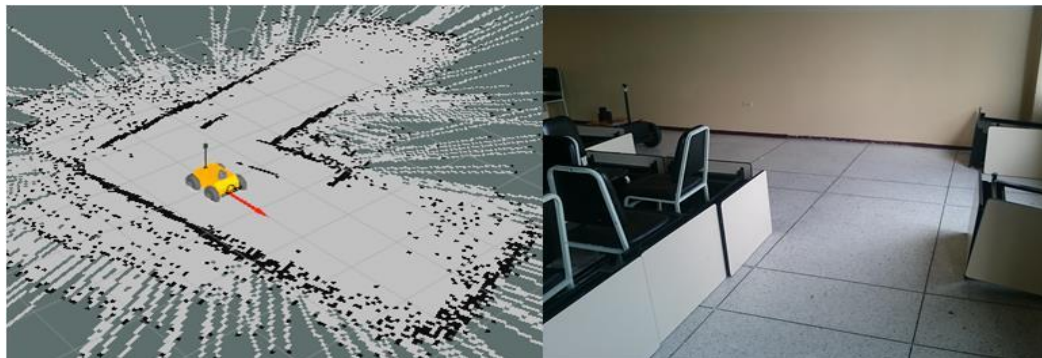
### Desventajas

- La precisión del módulo GPS depende directamente del costo.
- El GPS Trabaja únicamente a cielo abierto.
- El sensor LIDAR da mediciones inexactas con objetos transparentes

- Conexión de la cámara Web se realiza mediante un cable

## Figura 6

*Plataforma realizando SLAM*



Nota: Navegación de plataforma en un entorno cerrado. (Garzón & Obando, 2016)

En la Figura 6 se observa el resultado del mapeo a utilizar la función predefinida en SLAM como Hector, la cual es la que tiene mejor respuesta en mapeo 2D.

En (Moya, 2017) se realiza un trabajo de robot móvil autónomo donde se utiliza un sistema MRSLAM en los laboratorios de electrónica. Cuyo principal objetivo es realizar un robot para medir niveles de temperatura en un ambiente controlado y el desarrollar un algoritmo de control centralizado.

Las ventajas y desventajas que presenta la plataforma, se muestran a continuación:

### Ventajas

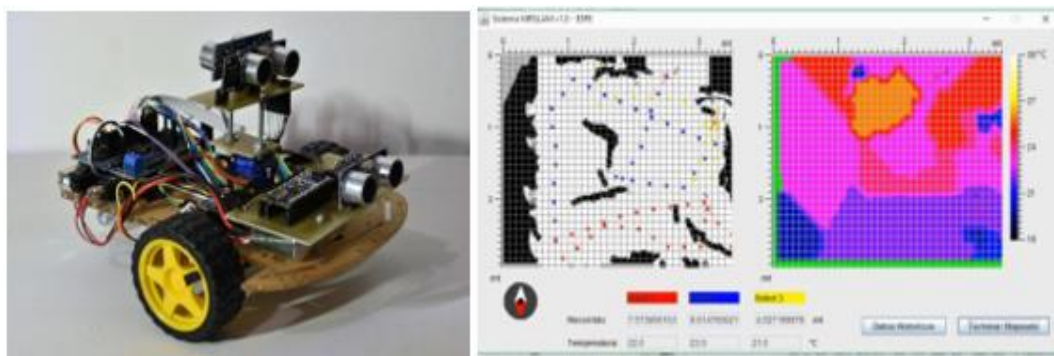
- Utiliza un driver L298N de bajo voltaje de saturación en los transistores de salida.
- Sensor ultrasónico debido a que no presenta algún tipo de iluminación para sensor correctamente.
- Se utilizan dos sistemas de control en cada plataforma, dedicados al desplazamiento angular y rectilíneo de los agentes robóticos.

### Desventajas

- Se trata SLAM de forma elemental, para priorizar en primer lugar el diseño de la plataforma, el medio de comunicación y la interfaz de visualización
- Errores presentados por la construcción de la plataforma robótica.
- Solo se realiza mapeo en 2D.

**Figura 7**

*MRSLAM*



Nota: Plataforma robótica móvil y mapa generado con MRSLAM. (Moya, 2017)

Se puede verificar en la Figura 7 que la plataforma móvil carece de un estudio mecánico y es netamente concentrado en la parte electrónica y de control. Mediante 3 agentes robóticos es ejecutado el mapa generado con MRSLAM después de la calibración de dirección (Moya, 2017).

En (Carrera, 2018) se desarrolla el trabajo de un robot móvil en el caso de raptó, se requiere la utilización de la herramienta SLAM con recuperación de ubicación, para lo cual se parte de un mapa existente y con la información existente se localiza dentro del mapa. El sensor utilizado es un LIDAR, el cual es eficiente para realizar mapeo en 2D.

Exhibe las siguientes ventajas y desventajas.

#### Ventajas

- Uso de motores a pasos configurados en locomoción tipo Skid-Steer.
- Método de medición de entorno es Rplidar 360° y la codificación utilizada para SLAM es Héctor SLAM.

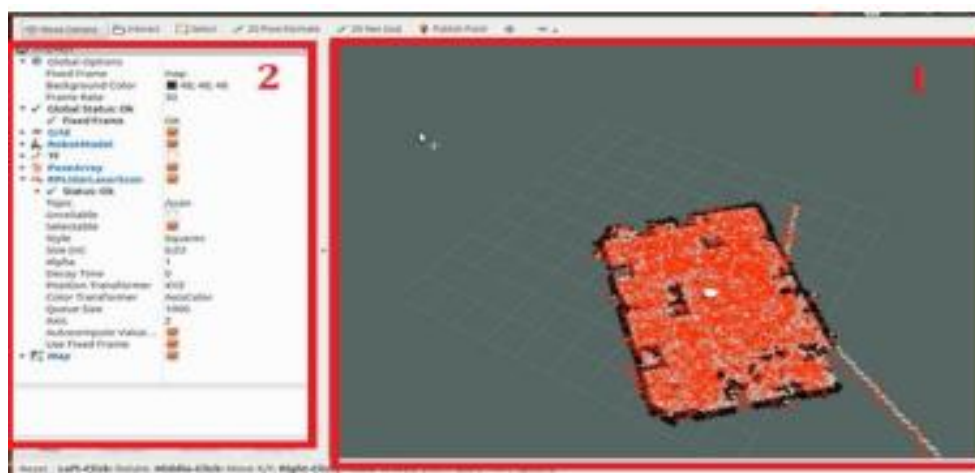
- Posee buena precisión, exactitud y robustez.

### Desventajas

- Solo se realiza mapeo en 2D.
- Necesita tener un mapa conocido para realizar la recuperación.
- Debe tener un controlador con altas prestaciones.

## Figura 8

### *Interfaz de recuperación de la ubicación*



Nota: Se denota los puntos rojos donde muestras las posiciones posibles del robot. (Carrera, 2018)

En la figura 8 se muestra la recuperación que realiza el robot móvil, mediante un sensor de laser LIDAR este realiza su ubicación, con un mapeo previamente realizado. Para encontrar la ubicación del robot utiliza el sistema AMCL.

(Quinaluisa & Toapanta, 2018) plantea un rediseño en la plataforma SIMBA en su forma y configuración de movimiento, aumentando una estación de carga, con la capacidad de realizar SLAM utilizando el método FastSlam 2.0; método incluido en el paquete Gmapping ROS.

Muestra las siguientes ventajas y desventajas.

### Ventajas

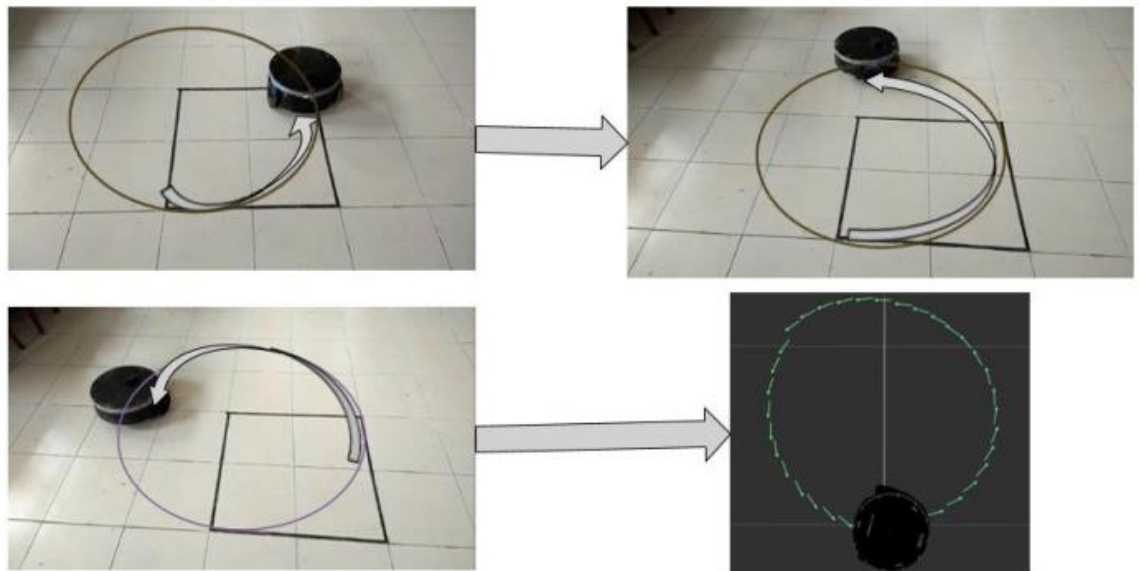
- Velocidad optima de mapeo a 0.17 m/s controlado por Joystick
- La localización es una fusión entre un sensor IMU y la odometría mediante un filtro de Kalman Extendido paquete encontrado en RobotLocalizationROS.
- Posee una estación de carga para el robot móvil.
- Mejora de mapas creados en ambientes reales con respecto a su antecesor.

### Desventajas

- Solo se realiza mapeo en 2D.
- Problemas en la navegación a presentar objetos pequeños en el piso.

**Figura 9**

*Trayectoria circular*



Nota: Se denota el seguimiento de la plataforma robótica móvil en una trayectoria circular.

(Quinaluisa & Toapanta, 2018)

Se observa en la Figura 9 como el rediseño de la plataforma SIMBA efectúa una trayectoria circular, (Quinaluisa & Toapanta, 2018).

## **Planteamiento del problema**

La localización se utiliza en el desarrollo de los temas expuestos en antecedentes, como una herramienta parte del SLAM más no como un tema de investigación. Debido a la poca profundidad que le dan a la localización se ha tornado la necesidad de la comprensión de los algoritmos inmersos en la temática.

En las investigaciones realizada dentro de la Universidad se denota una deficiencia en estudios de localización para plataformas móviles. Dichas investigaciones demuestran precedentes aplicativos con SLAM, en el cual la implementación de SLAM se denota como un instrumento para la obtención del mapa por donde transita la plataforma móvil y su respectiva localización, dentro de los paquetes de ROS.

Teniendo en cuenta que existe una falta de validación en los algoritmos de localización utilizados en SLAM (Mapeo y Localización Simultanea), aplicados en los diferentes ámbitos de la robótica en nivel local, nos da la pauta para realizar un estudio de los diferentes aspectos en los cuales son empleados los algoritmos como EKF, Filtro de partícula y técnicas de optimización gráfica, enfocados a lo localización de la plataforma móvil.

Donde se busca aportar conocimiento con respecto al entendimiento matemático y el manejo de los tres paradigmas de SLAM que son el Filtro de Kalman Extendido, Filtro de partículas y técnicas de optimización gráfica, aplicados en una plataforma móvil. Para ello se analizará los requerimientos tanto a nivel de hardware como software de las codificaciones que presenta el SLAM en la localización.

## **Justificación e importancia**

En vista del estudio realizado con anterioridad denotamos una deficiencia de investigación en la temática de localización, parte fundamental del SLAM; al indagar que las aplicaciones desarrolladas usan directamente paquetes predeterminados situados en

(ROS wiki, s.f.) o (GitHub, s.f.) para ejecutar SLAM. Siendo necesario su entendimiento para satisfacer la necesidad de un robot para ubicarse en un entorno real.

Con el fin del entendimiento de la resolución del problema de localización dentro de SLAM, se cubriera los tres paradigmas de SLAM (EKF, F.P. y técnicas de optimización gráfica) y se analizará su funcionamiento en un entorno predeterminado. Para lo cual serán puestos a prueba cada paradigma en una plataforma móvil diferencial para poder obtener una comparación.

El estudio ayudará a entender cuál es la mecánica detrás de los algoritmos utilizados para la realización de la localización de una plataforma móvil, dando como ventaja un entendimiento más profundo en el ámbito de robótica a nivel institucional con el fin de desarrollar robots con la capacidad de estimar su localización.

## **Objetivos**

### ***Objetivo general***

Diseñar y construir una plataforma móvil para la implementación de algoritmos de localización empleados en la navegación.

### ***Objetivos específicos***

- Desarrollar el modelo dinámico del sistema con el fin de obtener un correcto funcionamiento de la plataforma.
- Dimensionar y construir una plataforma diferencial la cual soportará diferentes elementos.
- Programar los algoritmos para estimar la localización de la plataforma móvil durante su desplazamiento.
- Comparar los resultados obtenidos durante la ejecución de cada algoritmo para estimar la localización de la plataforma móvil.



## **Alcance**

El proyecto tendrá un alcance en la comparación de los distintos algoritmos que embarcan los paradigmas del SLAM orientado en la parte de localización, de un robot móvil de avance tipo plataforma diferencial con odometría, que realizará la localización con la ayuda de sensores especializados en este campo. En el cual mediante los diversos algoritmos se obtendrá la mejor resolución en la estimación de la localización de la plataforma móvil.

## ***Construcción de la plataforma***

Los componentes necesarios para la construcción de la estructura de esta plataforma móvil, se basará en un tipo de plataforma diferencial para su desarrollo e integración con los otros subsistemas. Se solventará problemas de distribución de elementos mecánicos tanto para la movilidad como el soporte de la estructura. Tendrá locomoción que integrará dos ruedas motrices y una tercera rueda loca la cual facilitará el movimiento del robot. Para la emisión y recepción de instrucciones se utiliza una tarjeta de control con la capacidad necesaria para efectuar el trabajo.

## ***Algoritmos para la localización***

La utilización de los sensores de localización nos dará paso a tener una ubicación durante la movilización de la plataforma. La odometría es el dato de información a corregir con los algoritmos para la obtención de una localización estimada. La programación de los distintos algoritmos se realizar en un software Open Source, con esto se logrará tener una mejor aceptación para futuras prácticas.

Se plantea implementar los tres paradigmas del SLAM orientado a la localización:

- Filtro de Kalman Extendido
- Filtro de Partículas
- Técnicas de Optimización de Gráfica – Graph SLAM

### ***Implementación y evaluación***

La odometría se implementará para la obtención de la posición estimada de la plataforma móvil, corregida por un paradigma del SLAM. Se evaluará la capacidad de cada algoritmo de su paradigma correspondiente, ubicando a la plataforma móvil en un mismo entorno de trabajo y se comparará los resultados obtenidos.

### **Estructura del Documento**

El presente trabajo de titulación se desarrollará en cinco capítulos, indicando como se desarrolló el trabajo. El primer capítulo se enfoca a la búsqueda del problema, donde se inicia con una investigación en los trabajos realizados dentro de la Universidad y su desarrollo de la temática en otros países, en donde se describe las generalidades como la importancia del trabajo, los objetivos y los alcances que se desea conseguir.

El segundo capítulo se indaga los temas relacionados con la robótica móvil, plataforma diferencial, plataformas móviles y navegación de plataformas móviles enfocado a la estimación de la localización de las plataformas móviles.

El tercer capítulo se desarrolla, con una idea clara, como se implementará el trabajo de titulación, mediante normas para desarrollar sistemas mecatrónicos y el diseño concurrente. Obteniendo como resultado un modelo teórico tanto a nivel de hardware como software.

El cuarto capítulo empieza la implementación de la plataforma móvil donde se construirá dicha plataforma, su parte estructural, electrónica y control. Aquí también se cubre la programación de los algoritmos para la estimación de posición y su conexión con un procesador central. Concluida la implementación a nivel de hardware y software se procede a realizar las respectivas pruebas.

El quinto capítulo conclusión y trabajos futuros; finaliza el trabajo de titulación donde se detalla los logros y problemas obtenidos durante todo el trabajo.

## Capítulo II

### Estado del arte

El capítulo aborda la descripción y análisis de los conceptos básicos para la conformación de una plataforma móvil, para lo cual se estudia las definiciones sobre la robótica móvil. Dentro de la robótica móvil se realizará un enfoque a la odometría, localización, locomoción dando paso a los paradigmas que conforman SLAM vistos desde el punto de estimación de la localización y culminando con la navegación.

#### Robótica Móvil

En el transcurso de los años ochenta y noventa la robótica móvil ha asumido una importancia progresiva. La mayor diferencia de los robots móvil es su desplazamiento por agua, tierra inclusive aire a comparación con los robots manipuladores convencionales, teniendo su aspecto principal en relación con el desplazamiento autónomo o navegación. (Baturone, 2005).

Los robots móviles tienen la característica de desplazarse conforme a su programación por medio de patas, ruedas u orugas como se observa en la Figura 10. Recolectan información mediante sus sistemas de sensores que poseen y procesan según la necesidad, empleados en la industria para el transporte de mercadería, un mayor campo de utilización es para exploración espacial, investigación a lugares distantes a veces de difícil acceso como rescates de submarinos. (Tapia García & López Hernández, 2017).

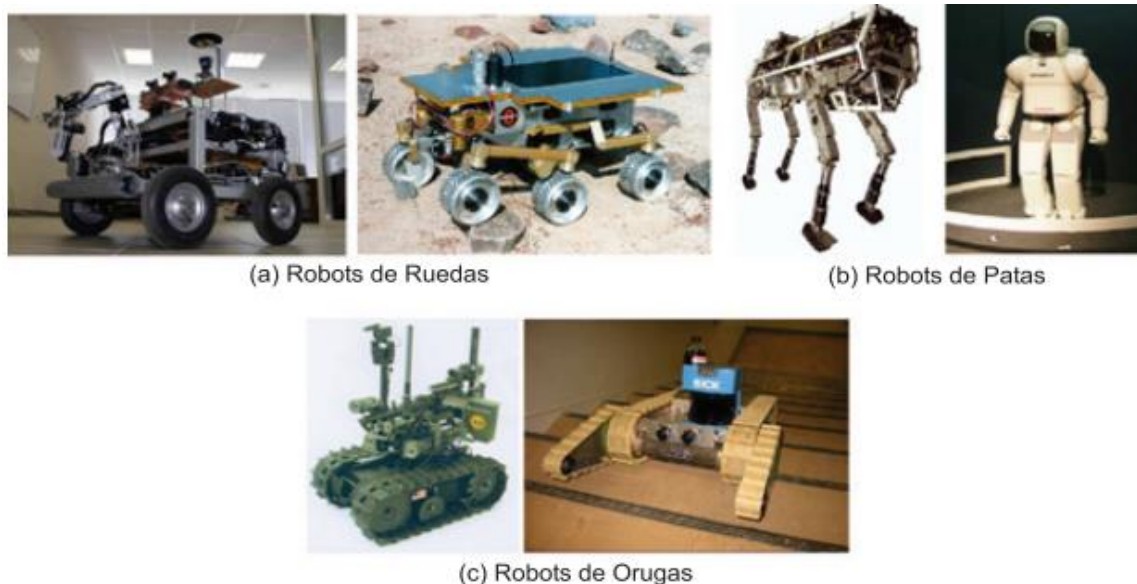
El mayor desarrollo presentado, es en los robots móviles con ruedas (RMR) a pesar del amplio estudio de la locomoción por orugas y patas. La ventaja de los RMR según (Barrientos Sotelo, García Sanchez, & Silva Ortigoza, 2007) es,

- Eficiencia energética en superficies lisas.
- No presenta desgaste en las superficies donde los robots se mueven.

- Menor complejidad en estructura, en relación a la locomoción por oruga y patas.

## Figura 10

*Tipos de locomoción en un robot móvil*



Nota: En la imagen se denota los tipos de locomoción que permite a los robots desplazarse.

(Barrientos Sotelo, García Sanchez, & Silva Ortigoza, 2007)

El sistema tracción y dirección de los robots móviles se encuentran distribuidas sobre el eje de las ruedas de acuerdo a la necesidad de maniobrabilidad, características del terreno y velocidad. La obtención del destino requerido depende de la precisión y rapidez del robot móvil, involucra un sistema de dirección para lograr una maniobrabilidad adecuada del robot y un confiable sistema de tracción. (Álvarez Gutiérrez & Jiménez López, 2019)

Las aplicaciones de los robots móviles cubren una gran área desde el servicio doméstico, industrial, agro-industrial hasta militar; los sectores mencionados son los que promueven su desarrollo (Una Introducción a los Robots Móviles, 2008) . Una de las aplicaciones más representativas es el robot autónomo Opportunity creado por la NASA

como se observa en la Figura 11, su principal ocupación es la navegación en un entorno desconocido careciendo de una precisa información de su ubicación. (Andrade & Llofriú)

### **Figura 11**

*Oportunity de la NASA*



Nota: Imagen del robot Oportunity de la NASA en un entorno marciano. (Andrade & Llofriú)

### ***Locomoción por ruedas***

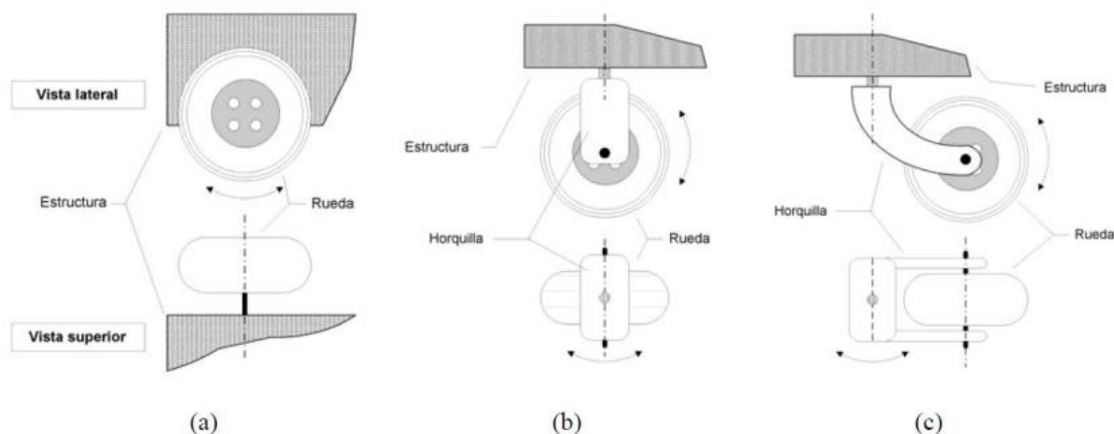
Una de las capacidades que poseen los seres vivos es la locomoción la cual consiente en trasladarse voluntariamente de un sitio a otro. La locomoción según (Trujillo, 2015) se divide en dos niveles superior e inferior.

- Nivel superior: Se relaciona con la voluntad, encargado de navegación y planificación de trayectorias
- Nivel inferior: Encargado del control y la coordinación (robots-actuadores) para lograr el desplazamiento.

(Trujillo, 2015) señala el contacto entre la rueda y el suelo es en un único punto del plano, distingue que existen tres tipos de ruedas convencionales.

## Figura 12

*Tipo de ruedas convencionales.*



Nota: (a) Rueda fija, (b) Rueda orientable centrada y (c) Rueda Loca. (Trujillo, 2015)

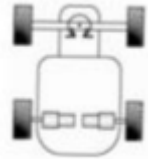

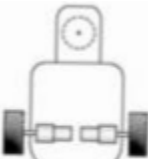

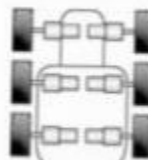



En la Figura 12 denota los tipos de rueda convencional, donde (Trujillo, 2015) declara:

- La rueda fija es asociada con el sistema de tracción del robot.
- La rueda orientable centrada cumple la función de dirección o tracción dirección.
- La rueda loca da estabilidad a la estructura mecánica del robot.

La locomoción por rueda presenta su principal ventaja en su mecánica fácil y simple con capacidad de cargar peso, así como su desventaja de un mal funcionamiento al encontrarse con objetos que supere el radio de la rueda. (Trujillo, 2015). Las distintas configuraciones de locomoción de las ruedas presentan diferentes características y propiedades (Gómez , 2020). En la Tabla 1 señala las distintas configuraciones que puede tener la locomoción por ruedas en un robot móvil.

**Tabla 1**

*Configuración de los sistemas de locomoción con ruedas.*

<b>SISTEMA</b>	<b>CARACTERSTICAS</b>	<b>ESQUEMA</b>	<b>EJEMPLO</b>
<b>Ackerman</b>	Basado en dos ruedas traseras motoras, mientras que las ruedas delanteras son del tipo direccionamiento, y se utilizan para seguir la trayectoria del robot.		 RB-CAR
<b>Tracción diferencial</b>	Tiene dos ruedas montadas, pero estas son propulsadas y controladas de manera independiente. El direccionamiento se da por a diferencia en la velocidad en las ruedas, adicional existen una o más ruedas para el soporte.		 RB-2 BASE
<b>Skid steer</b>	Este sistema dispone de varias ruedas en cada lado del vehículo y actúan de forma simultánea, su movimiento es el resultado de combinar las velocidades de las ruedas de la izquierda con las de la derecha.		 SUMMIT XL-HL
<b>Tracción omnidireccional</b>	Basado en la utilización de tres ruedas directrices y motrices. Posee tres grados de libertad, por lo que puede realizar cualquier movimiento, y posicionarse en cualquier posición u orientación.		 SMMIT XL- TEEL

Nota: La tabla es recuperada de una tabla considerada en (Gómez , 2020)

El estudio de cada configuración de los sistemas de locomoción por rueda inicia en la cinemática del robot, para el caso de estudio presente se enfoca en un robot de Tracción diferencial.

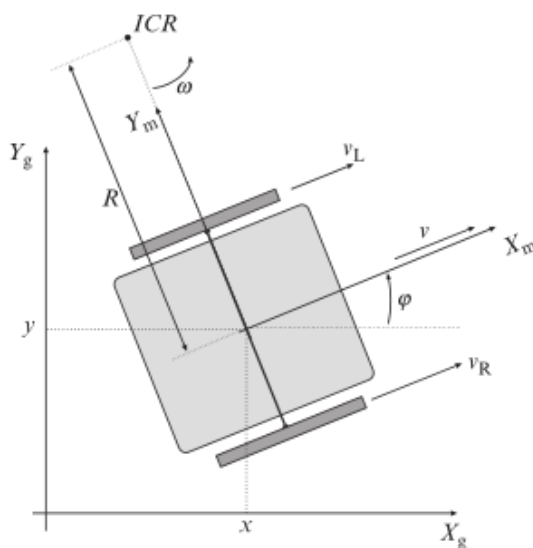
### ***Cinemática de un robot móvil de tracción diferencial***

Los robots móviles de tracción diferencial definen (KLANCAR, ZDEŠAR, BLAŽIC, & ŠKRJANC, 2016) como un mecanismo simple utilizado con gran frecuencia en práctica para robots pequeños, teniendo una o más ruedas giratorias para que el robot no se incline. Se tiene en cuenta que las ruedas principales deben estar en el mismo eje.

Con un concepto claro de que es la tracción diferencial se procede a realiza un análisis cinemático del mismo, donde (Siciliano & Khatib, 2016) señala a la cinemática como el movimiento de los cuerpos en un mecanismo robótico independiente de la fuerza y momentos de torsión que dan el movimiento. Se tiene en cuenta que la cinemática es la base fundamental del diseño, análisis, control y simulación de robots.

#### **Figura 13**

##### *Cinemática -Tracción diferencial*



Nota: Figura recuperada de (KLANCAR, ZDEŠAR, BLAŽIC, & ŠKRJANC, 2016)

En un robot de tracción diferencial la velocidad de cada rueda es controlada por un motor independiente. Como se observa en la Figura 13 las variables de control o entrada del robot (KLANCAR, ZDEŠAR, BLAŽIC, & ŠKRJANC, 2016) indica que son:

- $v_R(t)$ = velocidad de la rueda derecha



- $v_L(t)$  = velocidad de la rueda izquierda

Y sus otras variables que dependen dichas velocidades son:

- $\omega$  = velocidad angular
- ICR = centro instantáneo de rotación
- $L$  = distancia entre las ruedas
- $R$  = radio instantáneo de trayectoria
- $r$  = radio de la rueda

(KLANCAR, ZDEŠAR, BLAŽIC, & ŠKRJANC, 2016) toma en cuenta que, en cada instancia de tiempo, ambas ruedas tienen la misma velocidad angular  $\omega$  alrededor de ICR. Expresamos la velocidad angular  $\omega(t)$  con respecto al ICR(t) para cada rueda

$$\omega(t) = \frac{v_L(t)}{R(t) - \frac{L}{2}} \quad (1)$$

$$\omega(t) = \frac{v_R(t)}{R(t) + \frac{L}{2}} \quad (2)$$

Despejamos  $\omega(t)$  y  $R(t)$  de la ecuación (1) y (2) e igualamos:

Igualando  $R(t)$ :

$$\omega(t) = \frac{v_R(t) - v_L(t)}{L} \quad (3)$$

Igualando  $\omega(t)$

$$R(t) = \frac{L}{2} \times \frac{v_R(t) + v_L(t)}{v_R(t) - v_L(t)} \quad (4)$$

Calculando la velocidad tangencial es  $v = \omega(t) \times R(t)$ , tomando este concepto tenemos las velocidades tangenciales de cada rueda:

$$v_R(t) = \omega_R(t) \times r \quad (5)$$

$$v_L(t) = \omega_L(t) \times r \quad (6)$$

Expresamos la cinemática interna del robot (Coordenadas locales) en forma matricial se tiene:

$$\begin{bmatrix} \dot{x}_{local}(t) \\ \dot{y}_{local}(t) \\ \dot{\theta}(t) \end{bmatrix} = \begin{bmatrix} v_{X_{local}}(t) \\ v_{Y_{local}}(t) \\ \omega(t) \end{bmatrix} = \begin{bmatrix} \frac{r}{2} & \frac{r}{2} \\ 0 & 0 \\ -\frac{r}{L} & \frac{r}{L} \end{bmatrix} \begin{bmatrix} \omega_L(t) \\ \omega_R(t) \end{bmatrix} \quad (7)$$

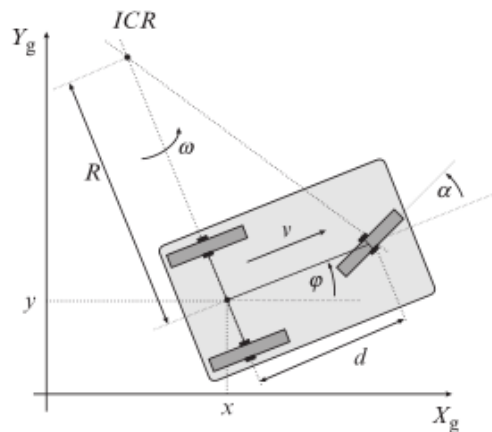
Debido a que el robot siempre se mueve en sentido  $X_{local}$  no encontramos velocidad en el eje  $Y_{local}$

La cinemática externa del robot es:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (8)$$

## Figura 14

### Tricycle drive Kinematics



Nota: en la figura se denota la cinemática del tricycle driver con respecto al ICR (KLANCAR, ZDEŠAR, BLAŽIC, & ŠKRJANC, 2016)

Una de las configuraciones más común según (KLANCAR, ZDEŠAR, BLAŽIC, & ŠKRJANC, 2016) es el tricycle drive debido a que su cinemática es como la de bicycle drive, ya que con sus tres ruedas hace que el robot sea estable en una dirección vertical como se observa en la Figura 14.

(KLANCAR, ZDEŠAR, BLAŽIC, & ŠKRJANC, 2016) declara las siguientes ecuaciones cinemáticas:

$$\begin{cases} \dot{x} = v_s(t) \times \cos(\alpha(t)) \times \cos(\varphi(t)) \\ \dot{y} = v_s(t) \times \cos(\alpha(t)) \times \sin(\varphi(t)) \\ \dot{\varphi} = \frac{v_s(t)}{d} \times \sin(\alpha(t)) \end{cases} \quad (9)$$

Donde:

$$v = v_s(t) \times \cos(\alpha(t))$$

$$\omega(t) = \frac{v_s(t)}{d} \times \sin(\alpha(t))$$

$v_s$  = la velocidad de la llanta del volante

### **Odometría**

La odometría se usa para estimar posición y no determinar la posición en los robots móviles, con respecto a su localización inicial argumenta (Ramírez, 2020). La odometría se implementa con codificadores (encoder) ópticos situados en las ruedas, dichos codificadores ópticos contabilizan el número de pulsos por giro de la rueda afirma (Musña Toapanta, Zapata Chancusig, Oñate Cadena, & Campusano Nieto, 2015).

Las ventajas que presenta la odometría son proveer una buena precisión, su simplicidad, bajo costo y admite altas tasas de muestreo según (Una Introducción a los Robots Móviles, 2008) y (Ramírez, 2020), así mismo, durante el tiempo de recorrido conduce a una acumulación de errores en la precisión, describe (Musña Toapanta, Zapata Chancusig, Oñate Cadena, & Campusano Nieto, 2015) que son:

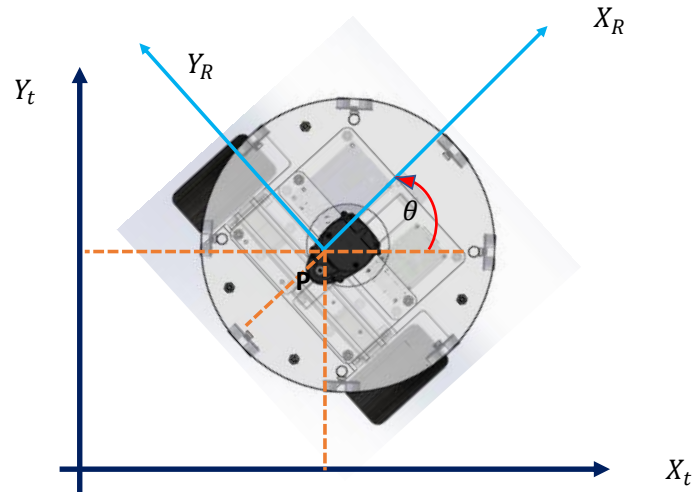
- Diámetros desiguales de las ruedas
- Mala alineación de las ruedas

- Resolución discreta de los codificadores

En (Carrera, 2018) indica que las ecuaciones presentadas en la Figura 15, consigue la ubicación del robot en un plano global.

### Figura 15

*Robot con desplazamiento diferencial - Odometría*



Nota: Imagen autor, información de (Ortiz , 2014), (Carrera, 2018)

En (Ortiz , 2014) y (Carrera, 2018) señalan las siguientes ecuaciones para la odometría:

$$\Delta S = \frac{\Delta s_r + \Delta s_l}{2} \quad (10)$$

$$\Delta S = \frac{\Delta s_r + \Delta s_l}{b} \quad (11)$$

$$\Delta x = \Delta s \cos(\theta + \Delta\theta) \quad (12)$$

$$\Delta y = \Delta s \sin(\theta + \Delta\theta) \quad (13)$$

Donde:

b=distancia entre ruedas

$$\begin{pmatrix} x(t+\Delta t) \\ y(x+\Delta t) \\ 0(t+\Delta t) \end{pmatrix} = f(x, y, 0, \Delta s_r, \Delta s_t) \begin{pmatrix} x(t) \\ y(t) \\ 0(t) \end{pmatrix} + \begin{pmatrix} \frac{\Delta s_r + \Delta s_l}{2} \cos(o(t) + \frac{\Delta s_r - \Delta s_l}{b}) \\ \frac{\Delta s_r + \Delta s_l}{2} \sin(o(t) + \frac{\Delta s_r - \Delta s_l}{b}) \\ \frac{\Delta s_r - \Delta s_l}{b} \end{pmatrix} \quad (14)$$

Con la integración de los codificadores las revoluciones de la rueda se pueden traducir a un desplazamiento lineal en relación al terreno (Ramírez, 2020). Uno de los codificadores utilizados para robótica son los encoders de cuadratura de tipo de encoder rotativo incremental, (Sásig) afirma, tiene la capacidad de indicar la posición, velocidad y dirección de movimiento del eje del motor.

Para la determinación del giro si es positivo o negativo, se hace referencia el desfase entre las señales AB, donde se asume la salida A como bit más significativo y como bit menos significativo la salida de B declara (Sásig), para observar la dirección de giro hacemos referencia a la Tabla 2, donde se compara el estado actual con el anterior.

**Tabla 2**

*Tablas de verdad -sentido de giro*

	<b>AB</b>		<b>ANTERIOR</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>
2	10	<b>ACTUAL</b>	<b>AB</b>	<b>00</b>	<b>01</b>	<b>10</b>	<b>11</b>
3	11	0	00	0	-1	+1	E
1	01	1	01	+1	0	E	-1
0	00	2	10	-1	E	0	+1
		3	11	E	+1	-1	0

Nota: Por motivo de trabajar con dos canales tenemos 4 estados. (Sásig)

(Sásig) considera para el cálculo de resolución de para una precisión cuádruple (R) utilizar la siguiente ecuación:

$$R = mH \times s \times r \quad (15)$$

Donde:

mH= Número de cuentas por revolución del eje del motor.

$s$ = Numero de estados generados.

$r$ = Relación de caja reductora

Se debe recalcar según (Ramírez, 2020) que a pesar de las limitaciones que posee la odometría es una parte esencial para el sistema de navegación y de un robot.

### **Localización**

En (Armigol Moreno, 1997) considera la estimación de la localización en los robots móviles como parte fundamental para la navegación, en cada instante de tiempo de manera de permitir conocer la posición y orientación del robot.

Una forma de determinar según (Borrero Cruz, 2016) la localización del robot móvil es mediante la velocidad angular o la medición de giro de las ruedas de tracción como se observa la **Figura 16**.

### **Figura 16**

*Medición de giro de la rueda*

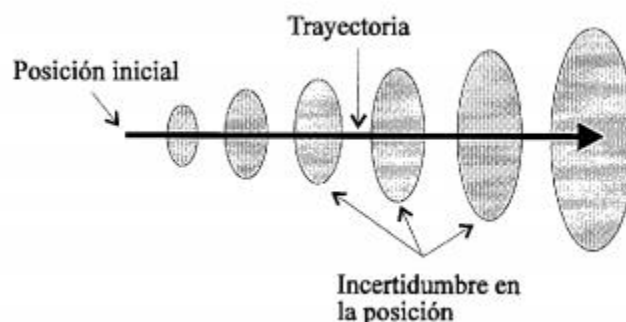
$$\Delta\alpha \Rightarrow \Delta s = R_w (\Delta\alpha) \rightarrow \begin{cases} \Delta s_r : \text{rueda dcha} \\ \Delta s_l : \text{rueda izda} \end{cases}$$

Nota: Recuperado de (Ortiz , 2014) (Borrero Cruz, 2016)

Un método incremental de localización tradicional, es el uso de la odometría teniendo como inconveniencia la acumulación de errores en toda su trayectoria como indica la **Figura 17**.

## Figura 17

*Crecimiento de la incertidumbre en un robot*



Nota: Crecimiento de la incertidumbre en la posición de un robot con la odometría. (Armigol Moreno, 1997)

Los métodos de localización (Silva, 2013) los clasifica en dos grupos principales: seguimiento de posición y localización global. Acotando que la localización global es más compleja.

El seguimiento de posición o localización local, (Morales & Sucar) manifiesta que la posición inicial del robot es conocida, estimando la localización con la odometría. (Silva, 2013) menciona la compensación de pequeños errores de odometría que ocurren durante la navegación del robot.

En la localización global, (Silva, 2013) considera una colocación del robot en un entorno desconociendo la posición inicial del robot y de este modo determinar su posición.

(Carrera, 2018) menciona realizar correcciones probabilísticas como filtro de Kalman, Filtro de Kalman Extendido y Filtro de partículas, para proporcionar una estimación de posición más fiable ante los errores que puede tener la odometría.

## SLAM

(Andrade & Llofriu) refiere el principal problema en los robots móviles de los investigadores o desarrolladores, es la localización y elaboración de mapas de un entorno.

El estudio de SLAM ayuda a solventar el problema referido con anterioridad. En (Torres, 2016) indica una conferencia de IEEE en 1986 surge SLAM (Simultaneous Localization And Mapping) donde se trató sobre métodos probabilísticos en la robótica y la inteligencia artificial, en los años consecutivos se publicaron artículos para describir una relación entre puntos de referencias (landmarks) y la manipulación de la incertidumbre geométrica. Simultáneamente se dio los primeros avances en el campo de navegación utilizando algoritmos basados en Filtro de Kalman.

(Torres, 2016) también señala la presentación por primera vez de la estructura, acrónimo y el resultado convergente de SLAM, en 1995 en el International Symposium on Robotics Research.

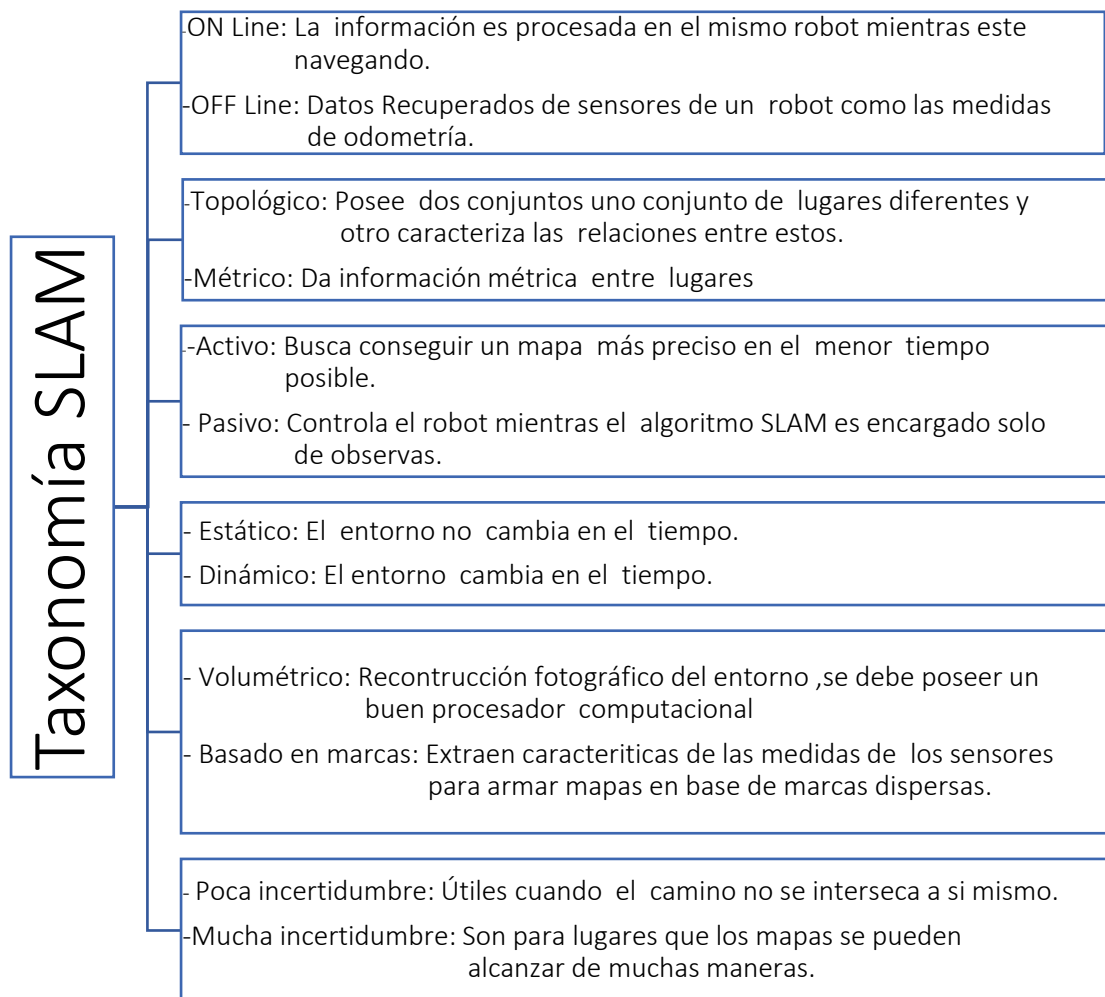
SLAM es una técnica la cual aborda el problema de la percepción de un robot móvil mientras navega por un entorno desconocido según (Siciliano & Khatib, 2016, págs. 1153-1176). Uno de los usos de SLAM es buscar un sentido preciso de la localización de un robot móvil

En señala las distinciones en varias dimensiones diferentes procedentes del SLAM, para el detalle de dichas distinciones se realizó un resumen observado en la **Figura 18**, las fuentes de la información son (Siciliano & Khatib, 2016, págs. 1153-1176) y (Andrade & Llofriu).



Figura 18

## Taxonomía SLAM



Nota: La figura es realizada por autor.

Lo más utilizado según (Andrade & Llofriu) actualmente es de modo OFF Line y se ha deteriorado la utilización del método topológico. Se debe tener en cuenta la capacidad de la computadora para proceder a seleccionar si será volumétrico, dinámico y con mucha o poca incertidumbre.

La mayor cantidad de métodos publicados para SLAM se derivan de los tres principales paradigmas deduce (Siciliano & Khatib, 2016, págs. 1153-1176).

### **Paradigmas del SLAM**

Estudiaremos los paradigmas de SLAM vistos desde su punto de estimación de localización. El primer paradigma de estudio es EKF SLAM, declara (Siciliano & Khatib, 2016, págs. 1153-1176) , a pesar de ser históricamente el primero en la robótica, se ha dejado de usar debido a sus propiedades computacionales limitantes y los problemas de realizar solo linealizaciones simples. (Silva, 2013) manifiesta, el algoritmo EKF SLAM es basado en filtro de Kalman Extendido

El segundo paradigma utiliza técnicas de filtrado estadístico no paramétrico según (Siciliano & Khatib, 2016, págs. 1153-1176) , conocida como filtros de partículas. Método más utilizado para SLAM en línea, el cual proporciona una perspectiva abordando el problema de asociación de datos en SLAM.

El tercer y último paradigma declarado por (Siciliano & Khatib, 2016, págs. 1153-1176) se basa en métodos dispersos de optimización no lineal al problema SLAM y resuelve el problema SLAM completo.

#### **Filtro Extendido de Kalman**

En (Jiménez Través, 2015) menciona sobre el filtro de Kalman debe mantener una solución de distribución normal en toda la trayectoria del proceso y tener la propiedad de linealidad, siendo una propiedad escasa en entornos reales.

(Andrade & Llofriu) señala, para la eliminación de restricción de transición lineal se realiza una extensión al filtro original obteniendo el Filtro de Kalman Extendido (EKF).

(Andrade & Llofriu) e (Jiménez Través, 2015) indican para EKF las siguientes ecuaciones para el modelo de sensado y odometría como funciones no lineales.

$$x_t = g(x_{t-1}, u_t) + r_t \quad (16)$$

$$z_t = h(x_t) + w_t \quad (17)$$

Debido que la ecuación 11 y 12 son no lineales es imposible dar de resultado una gaussiana, para su linealización se usa el teorema de Taylor indica (Jiménez Través, 2015).

$$g(x_{t-1}, u_t) = g(\mu_{t-1}, u_t) + G_t \cdot (x_{t-1} - \mu_{t-1}) \quad (18)$$

$$h(x_t) = h(\bar{\mu}_t) + H_t \cdot (x_t - \bar{\mu}_t) \quad (19)$$

Podemos expresar  $G_t$  y  $H_t$  mediante Jacobianos como se ve en la ecuación 15.

$$\begin{cases} G_t = \frac{\partial g(x_{t-1}, u_t)}{\partial x_{t-1}} \\ H_t = \frac{\partial h(x_t)}{\partial x_t} \end{cases} \quad (20)$$

A continuación, se presenta el algoritmo EKF, indica (Jiménez Través, 2015), se puede observar la etapa de predicción y actualización por la información dada de los sensores.

#### Algoritmo de Filtro de Kalman Extendido

- 1: Algorithm Extended\_Kalman filter  $\mu_{t-1}, \Sigma_{t-1}, u_t, z_t,$
- 2:  $\bar{\mu}_t = g(u_t, \mu_{t-1})$
- 3:  $\Sigma_t = G_t \Sigma_{t-1} G_t^T + R_t$
- 4:  $K_t = \bar{\Sigma}_t H_t^T (H_t \bar{\Sigma}_t H_t^T + Q_t)^{-1}$
- 5:  $\mu_t = \bar{\mu}_t + K_t (Z_t - h(\bar{\mu}_t))$
- 6:  $\Sigma_t = (I - K_t H_t) \bar{\Sigma}$
- 7: return  $\mu_t, \Sigma_t$

#### Filtro de Partículas

(Jiménez Través, 2015) menciona sobre los filtros no paramétricos donde se encuentran los filtros de partículas, los cuales dan solución con finitos números de

valores, podrá ser una perfecta solución si logran tener números infinitos, de manera de resolver problemas lineales y no lineales. Además, tolera sistemas multimodales.

En (Andrade & Llofriú) menciona, el filtro de partículas se acerca a una distribución de probabilidad del estado del sistema  $x_t$  manejando métodos de Montecarlo. Se emplea un conjunto de N muestras (samples) de partículas de la distribución a actualizar medida avance la odometría o sensado.

Según el número de partículas en el entorno se puede alcanzar una exactitud razonable tomando en cuenta a mayor número de partículas mayor carga computacional, declara (Jiménez Través, 2015).

Para realizar la actualización de distribución de probabilidad debe constar de dos pasos según (Andrade & Llofriú):

El paso de predicción modifica la función de la densidad, la cual refleja un movimiento ejecutado por el robot. Para modificar la posición de la partícula se utiliza la ecuación 16 donde  $f$  representa el modelo de movimiento a manejar indica (Andrade & Llofriú).

$$x_t = f(x_{t-1}, u_t) \quad (21)$$

El paso de la actualización ajusta la información de la densidad al último sensado obtenido asignando un peso a cada partícula, de acuerdo a la cercanía del sistema con la partícula como indica (Andrade & Llofriú) en la ecuación 17.

$$\omega_i \sim p(z_t | x_t) \quad (22)$$

Un consiguiente paso es realizar resampling lo cual es dar prioridad a las partículas de mayor peso propone (Jiménez Través, 2015). De esta manera obtenemos el algoritmo de filtro de partículas como se muestra:

1: Algorithm Particle Filter ( $X_{t-1}, \mu_t, Z_t$ ):

2:  $\bar{X}_t = X_t = \emptyset$

3: *for*  $m = 1$  to  $M$  *do*

4: *sample*  $X_t^{[m]} \sim p(X_t | u_t, X_{t-1}^{[m]})$

5:  $W_t^{[m]} = p(Z_t | x_t^{[m]})$

6:  $\bar{X}_t = \bar{X}_t + \langle X_t^{[m]}, W_t^{[m]} \rangle$

7: *end for*

8: *for*  $m = 1$  to  $M$  *do*

9: *draw*  $i$  with probability  $\propto W_t^{[i]}$

10: *add*  $X_t^{[i]}$  to  $X_t$

11: *end for*

12: *Return*  $X_t$

### **Técnicas de Optimización de Gráfica – Graph SLAM**

(Siciliano & Khatib, 2016) indica, Graph SLAM es una técnica offline con la capacidad de resolver el problema de FULL SLAM.

La resolución de Graph SLAM según (Andrade & Llofriu) se basa en marcas, dando la solución, modelando el problema de SLAM como un gráfico, donde:

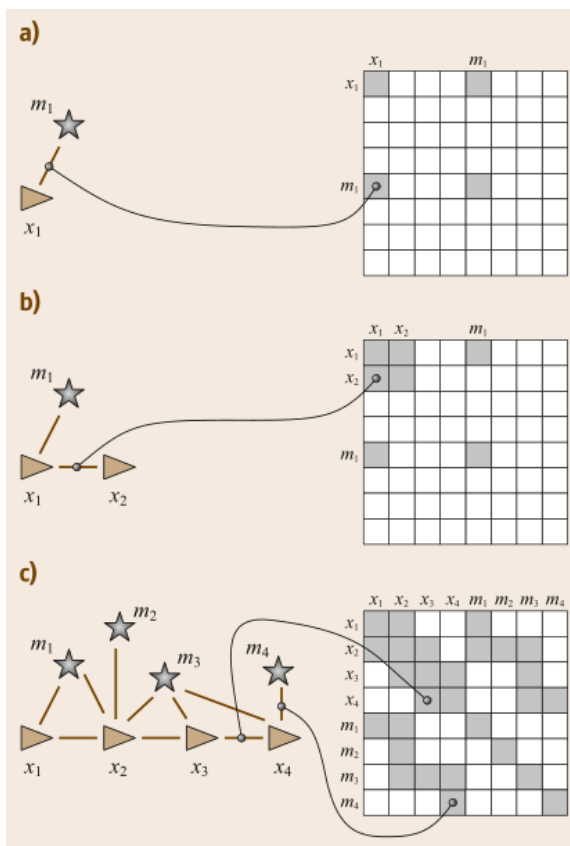
$x_i$ = posiciones del robot

$m_i$ = landmark – marcas – puntos de referencia

(Siciliano & Khatib, 2016) señala que los puntos de referencia y localización de robot se toman como nodos en un gráfico, los pares consecutivos son juntados al borde que representa la información transportada por la lectura de odometría como se observa en la Figura 19.

## Figura 19

Ilustración de la construcción del gráfico



Nota: a) Observación del primer landmark. b) movimiento del robot de  $x_1$  a  $x_2$ . c) varios pasos después (Siciliano & Khatib, 2016).

Se optimiza las posiciones sucesivas del robot (trayectoria) y la posición de los landmarks (mapa), indica (Andrade & Llofriu), se debe respetar lo mayor posible un conjunto de restricciones suaves. Las restricciones son distancias, deducidas de la observación de los datos de la odometría y landmarks.

En (Grisetti, Kümmerle, Stachniss, & Burgard, 2010) nos indica un algoritmo para Graph SLAM y se detalla a continuación.

**Require:**  $\tilde{x} = \tilde{x}_{1:T}$ : initial guess.  $C = \{(e_{ij}(\cdot), \Omega_{ij})\}$ : constrains

**Ensure:**  $x^*$ : new solution,  $H^*$  new information matrix//

find the maximum likelihood solution

while  $\rightarrow$  converged do

$b \leftarrow 0$   $H \leftarrow 0$

for all  $\langle e_{ij}, e_{ij} \rangle \in C$  do

//compute the Jacobian  $A_{ij}$ ,  $B_{ij}$ , of the error function

$A_{ij} \leftarrow \left( \frac{\partial e_{ij}(x)}{\partial x_i} \Big|_{x=x} \right)$   $B_{ij} \leftarrow \left( \frac{\partial e_{ij}(x)}{\partial x_i} \Big|_{x=x} \right)$

//

Compute the contribution of this constraint to the linear system

$H_{[ii]} += A_{ij}^T \Omega_{ij} A_{ij}$      $H_{[ii]} += A_{ij}^T \Omega_{ij} B_{ij}$

$H_{[ii]} += B_{ij}^T \Omega_{ij} A_{ij}$      $H_{[ii]} += B_{ij}^T \Omega_{ij} B_{ij}$

//Compute the coefficient vector

$b_{[i]} += A_{ij}^T \Omega_{ij} e_{ij}$      $b_{[j]} += B_{ij}^T \Omega_{ij} e_{ij}$

end for

$H_{[11]} += I$

// Solve the linear system using sparse Cholesky factorization

$\Delta x \leftarrow \text{solve}(H \Delta x = -b)$

//update the parameter

$\tilde{x} += \Delta x$

end while

$x^* \leftarrow \tilde{x}$

$H^* \leftarrow H$     //Release the first node

$H_{11}^* = I$

return  $\langle x^*, H^* \rangle$

## Navegación

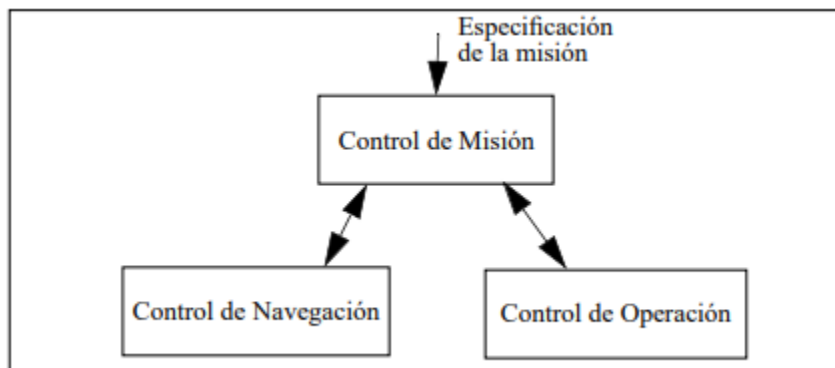
(Muñoz, 1999) describe la navegación como una metodología (o arte), la cual consiste en guiar a un robot a través de un ambiente con obstáculos, de tal forma, logre llegar seguro a su destino. Una de sus principales cualidades debe ser desenvolverse ante una situación inesperada o medios no estructurados. En (Una Introducción a los Robots Móviles, 2008) menciona el problema de la navegación se resume en las siguientes preguntas:

- ¿Dónde estoy?
- ¿A dónde voy?
- ¿Cómo debo llegar allí?

Para lo cuál en (Muñoz, 1999) declara que un robot móvil posee las características de efectuar desplazamientos en serie(navegación) y he interactuar con distintos elementos de su medio (operación). Como consecuencia el robot móvil posee una arquitectura coordinando los sistemas sensoriales, control de movimiento y operación, de manera correcta y eficaz.

### Figura 20

*Esquema básico de la arquitectura de un robot móvil.*



Nota: El esquema indica la interacción entre los módulos que componen la arquitectura. (Muñoz, 1999)



La Figura 20 indica la coordinación al controlar el desplazamiento (control de misión), con la interacción del control de elementos en el entorno (control de operación) considerando que el control de misión realiza un análisis del problema dando como resultado un plan de navegación y de operación (Muñoz, 1999).

(Preciado, 2018) e (Muñoz, 1999) indica que la navegación del robot móvil debe cumplir las siguientes tareas:

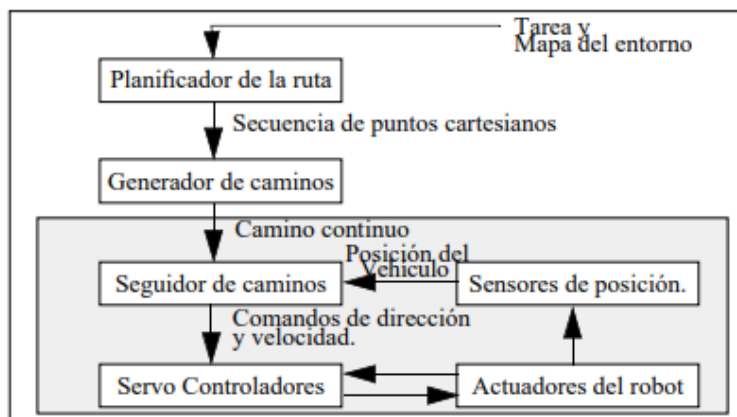
- Percepción del entorno a través de sensores.
- Planificación de trayectoria para alcanzar el destino deseado
- Guía del vehículo
- Seguimiento del camino

En la navegación deliberativa presenta métodos los cuales conforman un paradigma jerárquico dentro de la robótica.

Un requisito para la navegación deliberativa (Preciado, 2018) plantea, que es la presencia de un mapa y su resolución, así como la precisión de la localización o ubicación del robot en el mapa.

### Figura 21

*Control de navegación básica.*



Nota: En la figura se observa la estructura de navegación básica de un robot móvil. (Muñoz, 1999)

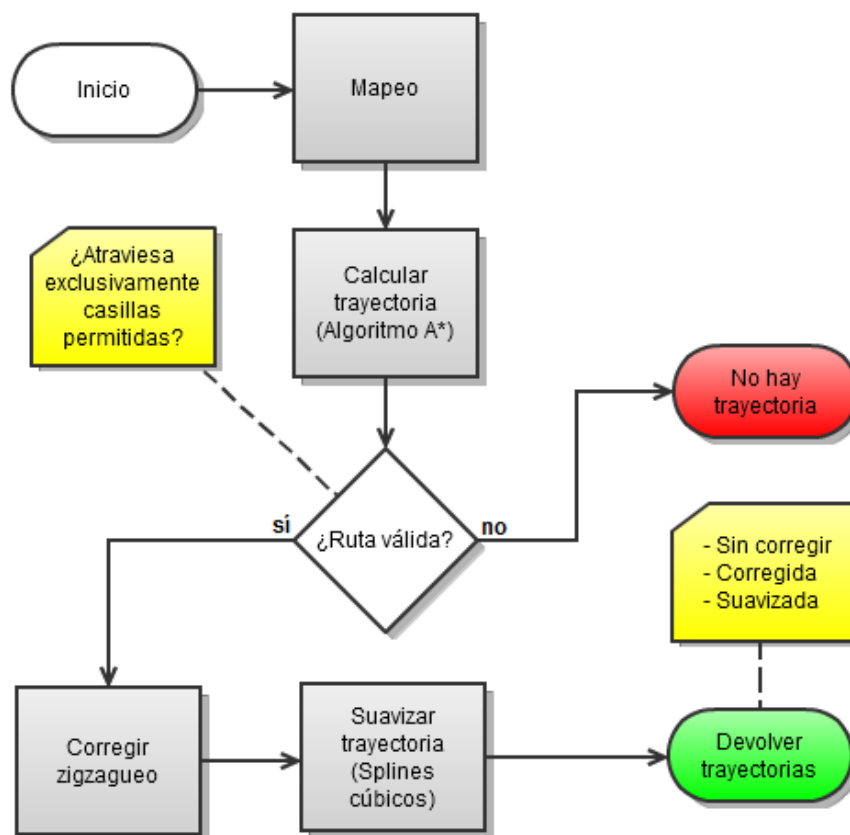
El esquema de la Figura 21 muestra la navegación partiendo de un entorno conocido y una tarea dada; debido a que la planificación de trayectoria toma estos datos para generar la ruta por donde se desea que el robot móvil se desplace, mediante la activación de los actuadores. La posición estimada del robot se da en conjunto con los sensores internos y las técnicas odométricas, que es el feedback del seguidor de caminos afirma (Muñoz, 1999).

### ***Planificación de trayectoria***

Planificar es decidir y prever las acciones las cuales nos permitan llegar desde un punto de partido a un objetivo deseable, donde (Fernández, Fernández, & Valmaseda, 2010) manifiesta que planificar no involucra realizar predicción hacia el futuro, sino una toma de decisiones pertinentes para llegar a ese futuro deseado.

Un objetivo de la planificación de trayectoria es la desvinculación de la cinemática y dinámica del robot para la generación de una ruta sin obstáculos señala (Fernández, Fernández, & Valmaseda, 2010) .

(Muñoz, 1999) señala un método de planificación clásica es utilizar algoritmos de búsqueda en grafos (tipo A\*), el cuál realiza una construcción de grafos en un espacio libre (información dada por el entorno) para consiguiente encontrar un camino óptimo a base de función de coste. En (Fernández, Fernández, & Valmaseda, 2010) señala un diagrama de flujo de cómo se realiza A\* contenido en el motor truevision3d, se observa en la Figura 22 .

**Figura 22***Planificación de trayectoria*

Nota: A\* en motor gráfico truevision3d. Recuperado de (Fernández, Fernández, & Valmaseda, 2010)

### Resumen del Capítulo

Los robots móviles poseen la capacidad de desplazarse en un entorno por medio de sistemas de locomoción con ayuda de sensores, los cuales les ayudan a estimar su posición durante su navegación. Una configuración de locomoción por ruedas bastante popular en robots pequeños es la tracción diferencial, donde gracias al estudio de su cinemática, observamos que cada rueda es controlada por un motor independiente. Los cálculos de la cinemática no tienen en cuenta ni la fuerza ni el torque del sistema.

Una forma de estimar la posición en robots móviles es mediante la odometría, la cual estima la localización del robot mediante sensores (encoder) encargados de traducir el giro o las revoluciones de la rueda a un desplazamiento lineal. Cabe recalcar que la odometría estima la posición y no la determina.

A medida como avanza la trayectoria del robot móvil, la odometría se encarga estima la localización local del mismo con un aumento de incertidumbre. Para contrarrestar el error de localización se utiliza filtros probabilísticos los cuales son manejados en SLAM y la navegación.

En SLAM encontramos que la mayoría de sus métodos tiene como base sus tres principales paradigmas los cuales estudiaremos desde el punto de localización de modo de tener los filtros probabilísticos (Filtro de Kalman Extendido y Filtro de partículas) y una técnica basada en optimización grafica (Graph SLAM).

El Filtro de Kalman Extendido (EKF) es un filtro de la extensión del Filtro de Kalman solventando el problema no lineal que un filtro de Kalman no puede resolver. En el mundo real es muy difícil tener sistemas lineales por dicho motivo se realiza esta extensión, resolviendo la no linealidad mediante teorema de Taylor y matrices jacobianas.

## Capítulo III

### Metodología y Diseño

En el capítulo, se presenta el desarrollo a nivel técnico de la plataforma móvil, basado en la norma (Künzel, 2005) la cual describe como desarrollar sistemas mecatrónicos, mediante sus tres fases implementa el diseño concurrente. El diseño a desarrollar inicia con la especificación del cliente o las necesidades que se requieren, para continuar con el modelado y simulado. El diseño de concepto y las especificaciones de funcionalidad se basa en el diseño y desarrollo de productos de (Ulrich & Eppinger, 2013); donde indica el modo de análisis de ventajas y desventajas mediante tablas, comparando los distintos conceptos para cada sistema que conforman la plataforma móvil, de esta manera cumplir con las especificaciones planteadas.

La selección de conceptos se tomará en cuenta una parte esencial, el modularidad de la parte física en la plataforma móvil, de este modo al someterse a cualquier cambio en su parte estructural permitirá la conservación de la parte de potencia y las tarjetas de control.

#### **Definición de necesidades**

La plataforma móvil se adapta a las necesidades requeridas, con el fin de efectuar los objetivos propuestos y requerimientos del usuario, a partir de estos se proyecta las especificaciones y características técnicas, con el fin de tener un diseño final y óptimo.

Entonces, con el objetivo del proporcionar la localización empleada para la navegación, se maneja algoritmos para obtener una mejor estimación de la localización de la plataforma móvil. Los algoritmos a utilizar según (Siciliano & Khatib, Spriger Handbook of Robotics, 2016) cubre los tres principales paradigmas de SLAM desde el punto de localización.

Se elaborará una plataforma móvil la cual satisfaga las necesidades más viables para el usuario. El análisis se otorga a las necesidades obtenidas con el fin de una mejora continua en el proyecto.

### **Tabla 3**

#### *Necesidades de plataforma móvil*

<b>Núm.</b>	<b>Necesidad</b>
1	Estructura Liviana
2	Implementar distintos componentes
3	Modelo Amigable
4	Transporte
5	Control
6	Ensamblaje sencillo
7	Capacidad de localización
8	Velocidad de desplazamiento uniforme en interiores

Nota: La tabla indica los requerimientos dados por el usuario (el cliente) para el desarrollo de la plataforma móvil.

La Tabla 3, muestra la recepción de la información dada por el usuario; el objetivo es obtener una plataforma móvil con la capacidad de realizar la localización de una plataforma móvil en la navegación. Se tiene en cuenta las necesidades solicitadas para la construcción de la misma, al igual que conseguir un desarrollo factible tanto de diseño como de mantenimiento.

#### **Definición de las características técnicas**

Al partir de las necesidades se precisa las características técnicas; como producto final siendo una plataforma móvil que llegue a satisfacer al usuario en los respectivos requerimientos. Las métricas se establecen prudentemente en el mejor de los casos para cada necesidad planteada.

**Tabla 4**

*Características técnicas de la plataforma móvil.*

<b>Núm. Métrica</b>	<b>Núm. de Necesidad</b>	<b>Métrica</b>	<b>Magnitud</b>	<b>Unidad</b>
1	4, 5	Diámetro de la rueda	5,2	Pulgadas
2	2, 6	Tiempo de ensamblaje	10	Min
3	2, 3, 6	Costo de Mantenimiento	370	US \$
4	1, 6, 5	Masa	10	Kg
5	3, 4	Diámetro del robot	360	Mm
6	3, 7	Altura	<200	Mm
7	4, 8	Velocidad de desplazamiento	0,19	m/s
8	4, 8	Tiempo de duración de batería	1	Hora
9	4, 8	Amperaje de batería	7	A
10	4, 8	Tensión de batería	12	V

Nota: Representa el cambio de especificación del usuario al entendimiento técnico.

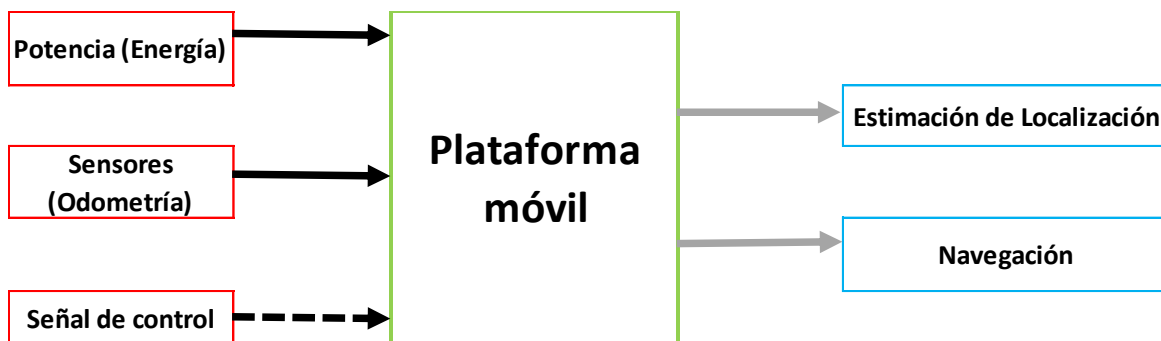
Denotando en la Tabla 4 las características técnicas, con los valores ideales para la plataforma móvil basadas en los requerimientos entregados o solicitados por el usuario para su previo desarrollo y funcionamiento.

#### **Definición de la funcionalidad del problema**

Para seguir con nuestro desarrollo es esencial entender la funcionalidad del problema (Ulrich & Eppinger, 2013), una de las formas más cotidianas para determinar la funcionalidad es descomponiendo el problema por partes para un mejor entendimiento. El primer paso a desarrollar es entendiendo al sistema como se muestra en la Figura 23, donde se trata el problema como una caja negra constando de las entradas y salidas que requiere la plataforma móvil para su correcto funcionamiento.

**Figura 23**

*Diagrama de Funcionalidad*

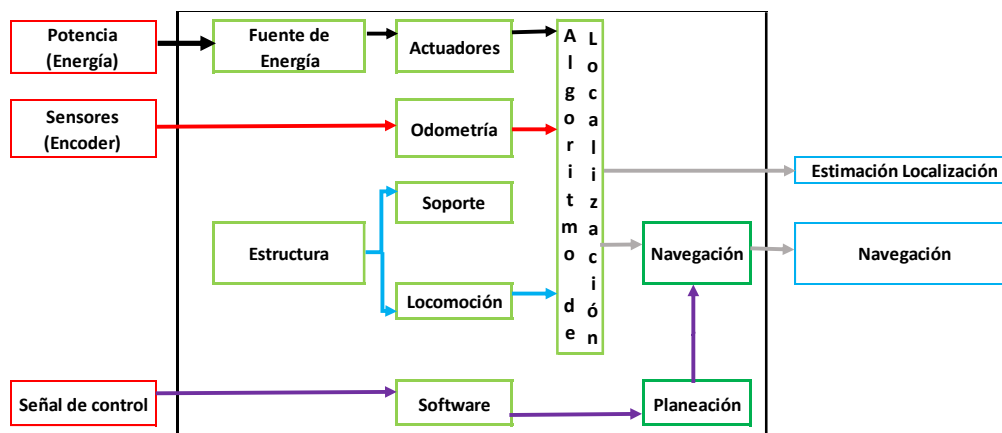


Nota: En la figura se muestra la descomposición del problema de la plataforma móvil. Realizada por autor.

Ahora entendido el ente de la caja negra, se continua con la división del problema general (plataforma móvil) en subfunciones. Las subfunciones se seguirán dividiendo hasta tener subfunciones más sencillas para trabajar con ellas como se muestra en la Figura 24, se detalla los flujos que se correlacionan entre sí para lograr la salida requerida, consiguiendo las operaciones necesarias. Consecuencias de las acciones se puede conocer los subsistemas requeridos.

**Figura 24**

*Diagrama de subfunciones de la plataforma móvil*



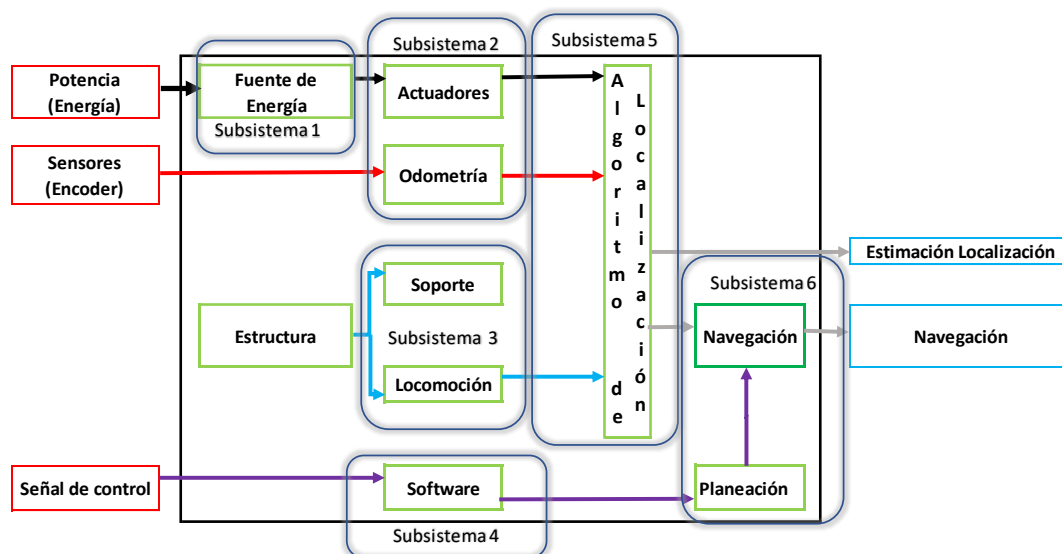
Nota: En la figura se denota la forma más sencilla de las subfunciones pertenecientes al problema. Realizada por autor.



En la Figura 25 se declaran los subsistemas con los que se trabajará en los siguientes pasos, los cuales se manejan para realizar la construcción de la plataforma móvil; estos subsistemas podrán dividirse en más subsistemas al ser necesario. Se desglosa en la Tabla 5. los subsistemas obtenidos por la división de las subfunciones, para la conformación de la plataforma móvil y obtención de las posibles soluciones a cada uno de estos subsistemas.

**Figura 25**

*Diagrama de subsistemas de la plataforma móvil*



Nota: En la figura se denota la forma más los subsistemas que solventan el problema de la plataforma móvil. Realizada por autor.

**Tabla 5**

*Subsistemas planteados de la plataforma móvil.*

Número	Subsistema	Función
1	Fuente de Energía	Fuente de Energía
2	Sistema de Tracción	Selección de actuadores, control y potencia
3	Soporte y locomoción	Soporte y desplazamiento
4	Controlador y procesador	Adquisición y procesamiento de datos
5	Localización	Algoritmos para la estimación localización
6	Planificador de ruta	Movilización

Nota: Desglose de los subsistemas obtenidos al realizar la división de la funcionalidad.

Plantea (Ulrich & Eppinger, 2013) para mayor facilidad de diseño se manejará en dos etapas. La primera etapa se establecerá especificaciones, la cual mediante una matriz de priorización se evaluará cada subsistema obtenido con alternativas de diseño y la segunda fase con ayuda de las alternativas ganadoras del paso anterior, obtendremos las especificaciones finales del diseño.

### ***Fuente de energía (Subsistema 1)***

La alimentación del sistema debe cumplir con la capacidad de abastecer a la plataforma móvil, la energía entregada debe satisfacer a las necesidades de los subsistemas de tracción y estructura. El subsistema fuente de energía conforma la parte esencial para el movimiento de la plataforma.

**Tabla 6**

*Análisis de los criterios para la fuente de Alimentación*

<b>Criterio</b>	<b>Costo</b>	<b>Peso</b>	<b>Amperaje</b>	<b>Suma +1</b>	<b>Ponderación</b>
Costo	-	5	5	11	0.56
Peso	0.2	-	0.2	1.4	0.07
Amperaje	0.2	5	-	7.2	0.37
<b>Total</b>				19.6	1

Nota: La tabla muestra una matriz formada por los criterios de peso, costo y amperaje. Se agrega la columna de "Suma +1" debido a que al compararse un principio consigo mismo se anula de manera que al sumar los valores en forma de fila se compensa.

En lo que respecta a la alimentación se busca una opción que permita el funcionamiento de manera óptima de la plataforma sin alterar su estructura o peso. Los criterios a considerar en el subsistema para la plataforma móvil se concentran en los siguientes factores:

- Peso
- Costo

- Amperaje

En la Tabla 6 se observa que el criterio con una mayor predominancia es el costo con un valor de 56%. A continuación, se selecciona las alternativas para el subsistema, como se necesita autonomía en la plataforma móvil se consideran tipos de baterías que nos brinde el voltaje DC requerido, para lo cual se presenta las siguientes soluciones:

- **Tipo Lipo:** Es una batería recargable conformada por celdas, cada una de ellas provee 3.7 V. La máxima corriente que puede producir es de 3 A. Teniendo una gran desventaja en costos.
- **Recargable Sellada:** Es usada para diferentes aplicaciones, con una capacidad de 12 V y 7 A con una durabilidad de hasta 1000 ciclos/descarga. Sin embargo, el peso es una desventaja.

Se realiza la ponderación de las alternativas planteadas para solventar los requerimientos del subsistema y se procede a evaluar cada criterio con las alternativas planteadas con anterioridad. En la Tabla 7 se muestra cómo se evalúa las alternativas para el primer criterio (costo) y así sucesivamente en las siguientes tablas se evaluará cada criterio para las alternativas.

**Tabla 7**

*Ponderación con respecto al criterio Costo*

<b>Costo</b>	<b>Sellada</b>	<b>Lipo</b>	<b>Suma +1</b>	<b>Ponderación</b>
Sellada	-	5	6	0.83
Lipo	0.2	-	1.2	0.17
<b>Total</b>			<b>7.2</b>	<b>1</b>

Nota: La matriz se enfoca en el criterio costo con lo que se busca minimizar el costo.

Se procede a la realiza la evolución para los criterios restantes como se lo hizo el del costo, para finalizar con la construcción de la matriz de Holmes.

**Tabla 8**

*Matriz de Holmes de la fuente de alimentación*

	<b>Costo</b>	<b>Peso</b>	<b>Amperaje</b>	<b>Suma</b>	<b>Prioridad</b>
Sellada	0.465	0.012	0.333	0.810	1
Lipo	0.095	0.058	0.037	0.19	2
			<b>Total</b>	1	

Nota: La matriz es formada por los criterios en las columnas mientras que las filas se encuentran las diferentes opciones de la fuente de alimentación.

En la Tabla 8 se denota la matriz de Holmes donde los valores se obtienen del producto de la ponderación de cada criterio con cada valor de la evaluación, entre las alternativas y el criterio. Como se observa en la tabla la batería sellada, se presenta como opción de prioridad a pesar de su gran desventaja en el peso.

Para los siguientes subsistemas se realizará la misma forma de selección de alternativas, basándonos en la matriz de priorización de criterios. Para lo cual, solo se mostrará los criterios tomados en cuenta para cada subsistema, las alternativas de selección y la matriz de selección final (Matriz de Holmes).

### **Consumo energético**

Se realiza una conexión en serie para el control y la potencia de la plataforma móvil, el procesador central (NUC) no se toma en cuenta en el consumo debido a su conexión directa con el tomacorriente.

Para saber cuánto amperaje necesita la batería de alimentación se procede a realizar una tabla con el amperaje necesario para la alimentación de todo el circuito.

Como se observa en la Tabla 9 el consumo de la parte de controla es menor a 1ª y con un voltaje de 5V por dicho motivo se realiza el regulador de voltaje (ver Anexo C), para poder suministrar con una batería de 12V lo suficiente para alimentar a los motores.

**Tabla 9***Alimentación del Control*

<b>Dispositivos</b>	<b>Voltaje(V)</b>	<b>Amperaje (mA)</b>
ESP 32	5	500
Monster Moto Shield	5	20
MPU 6050	5	3.8
Encoder Motor derecho	5	3
Encoder Motor Izquierdo	5	3
<b>Total:</b>	5	519.8

Nota: La matriz indica el consumo de Voltaje y Amperaje de la parte de control.

***Sistema de Tracción (Subsistema 2)***

El movimiento de la plataforma se basa en la conversión de energía eléctrica a energía rotacional, para realizar la traslación de la plataforma móvil. La base fundamental del subsistema es la selección de los actuadores rotacionales.

El sistema de tracción se realiza mediante los motores; los motores permiten realizar la traslación de la plataforma a diversos puntos del entorno en que navega. Para cumplir con los requerimientos detallados con anterioridad los motores generaran una velocidad optima y constante con un peso aproximado de 10 kg; por lo tanto, los motores deben tener un alto torque y bajas revoluciones.

Deben tener la factibilidad de ser controlados de una manera fácil y no poseer circuito de potencia y control complicados. Para la selección de este subsistema tendremos los siguientes criterios a valorar:

- Costo
- Facilidad de circuitos de potencia y control (Control).
- Disponibilidad en el mercado.
- Torque

**Tabla 10***Criterios para selección del tipo de Motor*

<b>Criterio</b>	<b>Costo</b>	<b>Control</b>	<b>Disponibilidad</b>	<b>Torque</b>	<b>Suma +1</b>	<b>Ponderación</b>
Costo	-	5	1	0.2	7.2	0.20
Control	0.2	-	5	10	16.2	0.44
Disponibilidad	1	0.2	-	0.2	2.4	0.06
Torque	5	0.1	5	-	11.1	0.3
<b>Total</b>					<b>36.9</b>	<b>1</b>

Nota: La tabla muestra una matriz formada por los criterios de costo, control, disponibilidad y torque. Dando la ponderación respectiva a cada criterio.

Se procederá a la evaluación de los criterios para determinar su ponderación correspondiente como se indica en la Tabla 10.

Como se planteó con anterioridad se presentará las alternativas a continuación y de ahí se procederá a la resolución dada por la matriz de Holmes.

- **Motor DC:** Los motores DC encontrados en el mercado poseen una caja reductora para minimizar el rpm y aumentar el torque entregado por el motor. En la **Figura 26** se observa un modelo de motor DC con eje céntrico con aplicación en la robótica.

## Figura 26

### Motor DC



Nota: 5202 Series Yellow Jacket Planetary Gear Motor (188:1 Ratio, 30 RPM, 3.3 - 5V Encoder).

(SERVOCITY, 2021)

- **Servomotores:** Se encuentran en el mercado con más frecuencia el tipo de rango de gira limitado, permitiendo una rotación de 180°. Las características esenciales que poseen estos motores según (González, 2016) es su facilidad de controlar su posición y velocidad de giro, debido a esta facilidad de control los motores vienen con tres cables.
- **Motor a pasos:** Según (Ingeniería Mecafenix, 2017) los pasos pueden variar de 1.8° a 90° todo depende de la secuencia indicada por el controlador. Los motores tienen la propiedad de quedar en una posición fija si una o más bobinas están energizadas.

Dadas las alternativas para el subsistema se debe proceder al análisis de cada una como se realizó en el subsistema anterior, como se especificó se presentará los resultados de prioridad de la matriz de Holmes para este subsistema.

**Tabla 11***Matriz de Holmes para el tipo de Motor*

	<b>Costo</b>	<b>Control</b>	<b>Disponibilidad</b>	<b>Torque</b>	<b>Suma</b>	<b>Prioridad</b>
Motor DC	0.12	0.264	0.008	0.135	0.527	1
Servomotor	0.067	0.145	0.008	0.027	0.247	2
A Pasos	0.014	0.031	0.042	0.135	0.222	3
<b>Total</b>					<b>1</b>	

Nota: Los valores se obtienen del producto de la ponderación con cada uno de los valores en cada tabla.

Como se observa en la Matriz de Holmes de la Tabla 11 se presenta como elección prioritaria al motor DC, con predominancia de los criterios de control y torque.

Así mismo encontrar sensores o dispositivos, los cuales permitan tomar la odometría, de esta manera la plataforma móvil sensa el desplazamiento; con esta medición y los subsistemas 4 y 5 la plataforma móvil cumplirá sus objetivos. Analizamos la obtención de la odometría utilizando como sensor encoder.

**Tabla 12***Criterio para selección del sensor*

<b>Criterio</b>	<b>Costo</b>	<b>Precisión</b>	<b>Transmisión</b>	<b>Control</b>	<b>Suma +1</b>	<b>Ponderación</b>
Costo	-	1	10	5	17	0,39
Precisión	1	-	10	0,1	12,1	0,28
Transmisión	0,1	0,1	-	1	2,2	0,05
Control	0,2	10	1	-	12,2	0,28
<b>Total</b>					<b>43,5</b>	<b>1</b>

Nota: Debido a que ambas alternativas presentan valores altos se analiza la presente una menor inversión



Para la selección de este subsistema tendremos los siguientes criterios a valorar:

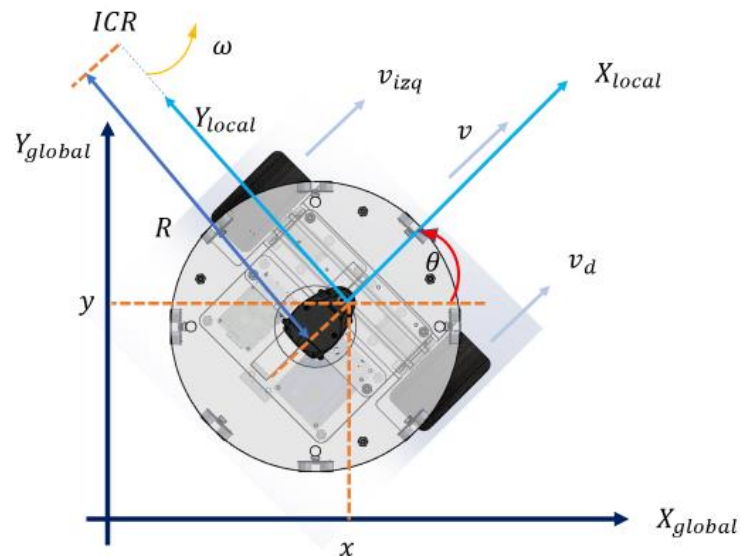
- Costo
- Precisión
- Capacidad de transmisión de datos. (Transmisión)
- Capacidad de adaptarse a los controladores comerciales. (Control)

La ponderación para cada criterio establecido se muestra en la Tabla 12 , donde el criterio más importante es el costo con una ponderación de 39,1%. Algunos motores vienen incluidos con enconder tipo Magnético (Hall Effect). Con lo cual se procedería a la obtención de motores con enconder incluido para ahorrar costos debido a su prioridad como criterio y un control más sencillo debido a su integración.

### Modelamiento cinemático

**Figura 27**

*Cinemática de una plataforma diferencial*



Nota: En la figura se observa los parámetros del robot y la ubicación de las coordenadas locales y globales. Figura realizada por autor.

El modelamiento cinemático describe la relación entre el comportamiento del estado de espacio y la entrada del sistema. El estudio de los modelos cinemáticos es suficiente para diseñar las estrategias de locomoción y describe las velocidades que presenta el sistema mediante un conjunto de ecuaciones de primer orden. (KLANCAR, ZDEŠAR, BLAŽIC, & ŠKRJANC, 2016)

En la Figura 27 se denota que ambas ruedas tienen la misma velocidad angular  $\omega$  alrededor de ICR (centro instantáneo de rotación)

Expresamos la cinemática interna del robot (Coordenadas locales) en forma matricial

$$\begin{bmatrix} \dot{x}_{local} \\ \dot{y}_{local} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v_{x_{local}} \\ v_{y_{local}} \\ \omega \end{bmatrix} = \begin{bmatrix} \frac{r}{2} & \frac{r}{2} \\ 0 & 0 \\ -\frac{r}{L} & \frac{r}{L} \end{bmatrix} \begin{bmatrix} \omega_{izq} \\ \omega_d \end{bmatrix} \quad (23)$$

Debido a que el robot siempre se mueve en sentido  $X_{local}$  no encontramos velocidad en el eje  $Y_{local}$

La cinemática externa del robot la expresamos con las coordenadas globales como se muestra en la Figura 27.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (24)$$

### Modelamiento dinámico

El modelo dinámico describe un movimiento al someter al sistema a una fuerza externa. El modelamiento dinámico incluye las fuerzas energías, masa del sistema, la inercia y los parámetros de velocidad. El modelo es representado por ecuaciones diferenciales de segundo orden.

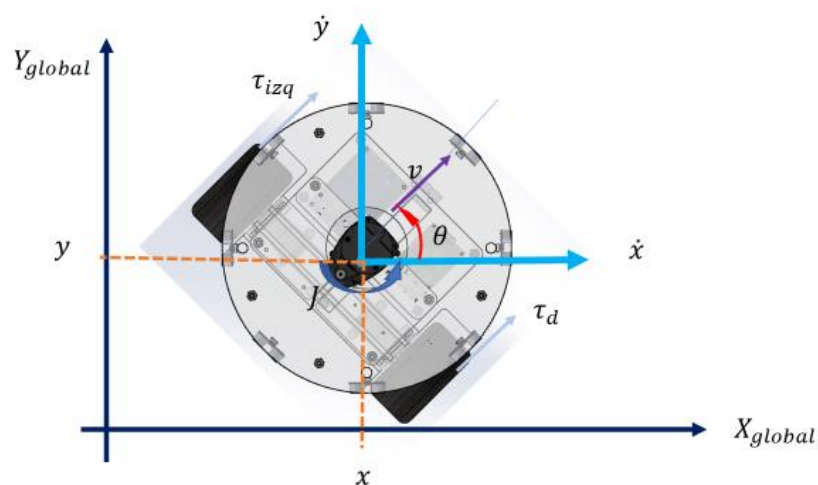
Se desarrolla el modelamiento dinámico mediante la formulación del Lagrange utilizada especialmente para la descripción de sistemas mecánicos.

Debido a que en el plano la energía potencial es constante entonces  $U=0$ , debido a que la masa de la rueda es insignificante con respecto a la masa de la estructura será despreciada por ende la inercia polar ( $J$ ) responde a la estructura igual la masa  $m$ .

Analizamos las energías presentadas en la Figura 28 con respecto al eje de coordenadas global.

### Figura 28

*Energía entregada por el sistema*



Nota: En la figura se muestra las energías dadas por el sistema. Figura realizada por autor.

$$L = \frac{1}{2} m \dot{x}^2 + \frac{1}{2} m \dot{y}^2 + \frac{1}{2} J \dot{\theta}^2 \quad (25)$$

Donde:

$T$ =Suma de la energía cinética del sistema

$U$ = Suma de la energía potencial del sistema

$m$ =Masa del sistema

$J$ =Inercia del sistema

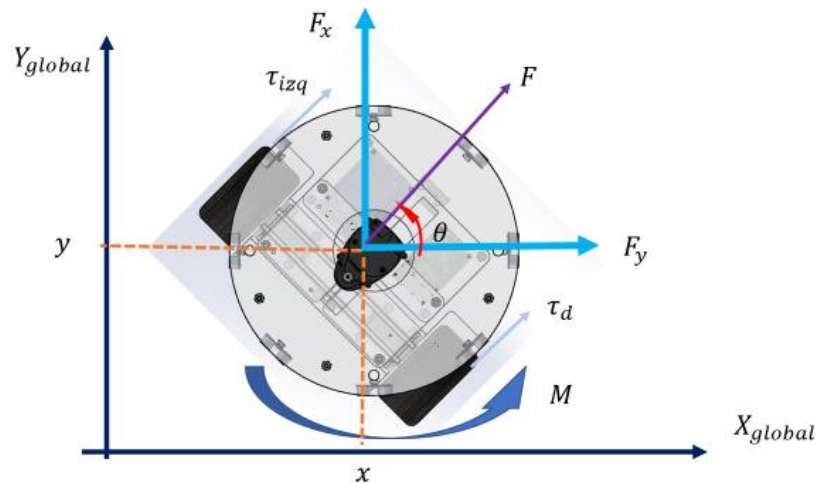
$\dot{x}$ =Velocidad lineal en el eje x global

$\dot{y}$ =Velocidad lineal en el eje y global

$\dot{\theta}$ =Velocidad angular del sistema

### Figura 29

*Fuerzas del sistema*



Nota: La figura muestra las fuerzas que intervienen en el robot. Figura realizada por autor.

$$f_k = F_x + F_y + M \quad (26)$$

Donde:

$f_k$ = Sumatoria de las fuerzas que interviene en el sistema

$F$ =Fuerza del robot

$M$ =Momento del robot

En la Figura 29 se observa las fuerzas que interactúan con el sistema, donde la fuerza está dada por los torques que entrega cada motor y el momento está dado por el torque que presenta el cuerpo, entonces tenemos la siguiente relación

$$F_x = \frac{1}{r} (\tau_d + \tau_{izq}) \cos(\theta) \quad (27)$$

$$F_y = \frac{1}{r}(\tau_d + \tau_{izq}) \sin(\theta) \quad (28)$$

$$M = \frac{L}{2r}(\tau_d - \tau_{izq}) \quad (29)$$

$L$ = Distancia entre las ruedas

$\tau_d$ = Torque motor derecho

$r$ = Radio de la rueda

$\tau_{izq}$ = Torque motor izquierdo

**Tabla 13**

*Descripción matrices modelo dinámico*

<b>Matriz</b>	<b>Descripción</b>
$q$	Vector de coordenadas generales (nx1)
$M(q)$	Matriz positiva define las masas e inercias(nxn)
$V(q, \dot{q})$	Vector de Coriolis y fuerza centrífuga (nx1)
$F(\dot{q})$	Vector de fricción y fuerzas dumping (nx1)
$G(q)$	Vector de fuerzas y toques dado por la gravedad (nx1)
$\tau$	Vector de perturbaciones desconocidas que no se incluyen en el modelo dinámico(nx1)
$E(q)$	Matriz de transformación del espacio del actuador al espacio de coordenadas generalizado(nxr)
$u$	Vector de entradas (rx1)
$A^T(q)$	Matriz del coeficiente de las restricciones cinemáticas(mxn)
$\lambda$	Vector de las restricciones de fuerzas (Langrange Multiple) (mx1)

Nota: En la tabla se detalla el significado de las matrices en el modelo dinámico (KLANCAR,

ZDEŠAR, BLAŽIC, & ŠKRJANC, 2016)

Se desprecia el damping y la fricción dadas por la rotación de las ruedas ( $P=0$ ), y se procede analizar la ecuación general de LaGrange para cada estado. Obteniendo las ecuaciones del modelo dinámico son:

$$m \ddot{x} - \frac{1}{r}(\tau_d + \tau_{izq}) \cos(\theta) - \lambda_1 \sin(\theta) = 0 \quad (30)$$

$$m \ddot{y} - \frac{1}{r}(\tau_d + \tau_{izq}) \sin(\theta) + \lambda_1 \cos(\theta) = 0 \quad (31)$$

$$J \ddot{\theta} - \frac{L}{2r}(\tau_d + \tau_{izq}) = 0 \quad (32)$$

Modelo dinámico de forma matricial es:

$$M(q)\ddot{q} + V(q, \dot{q}) + F(\dot{q}) + G(q) + \tau = E(q)u - A^T(q)\lambda$$

Donde las matrices se describen en la Tabla 13:

### Dinámica inversa del modelo del robot

El modelo más común que intervine la cinemática y la dinámica del robot está determinado por las siguientes matrices:

$$\tilde{M} = S^T M S \quad (33)$$

$$\tilde{M} = \begin{bmatrix} m & 0 \\ 0 & J \end{bmatrix} \quad (34)$$

$$\tilde{V} = S^T M \dot{S} v + S^T (V + F * G) \quad (35)$$

$$\tilde{E} = \frac{1}{r} \begin{bmatrix} 1 & 1 \\ L & -L \\ 2 & -2 \end{bmatrix} \quad (36)$$

El sistema puede ser descrito en los espacios de estados desde  $\dot{x} = f(x) + g(x)u$ , donde el vector de estado es  $x = [q^T, v^T]^T$

$$\dot{x} = \begin{bmatrix} S v \\ -\tilde{M}^{-1} \tilde{V} \end{bmatrix} + \begin{bmatrix} 0_{n \times r} \\ \tilde{M}^{-1} \tilde{E} \end{bmatrix} u \quad (37)$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{v} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} v \cos(\theta) \\ v \sin(\theta) \\ \omega \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{\tau_d + \tau_{izq}}{mr} \\ \frac{L(\tau_d - \tau_{izq})}{2Jr} \end{bmatrix} \quad (38)$$

Despejando los torques tenemos las ecuaciones

$$\tau_d = \frac{\dot{v} mr}{2} + \frac{\dot{\omega} Jr}{L} \quad (39)$$

$$\tau_{izq} = \frac{\dot{v} mr}{2} - \frac{\dot{\omega} Jr}{L} \quad (40)$$

El torque a soportar de la plataforma será tal analizado en el modelo dinámico, encontrado el torque se procede a buscar un motor DC que cumplan con el toque máximo deseado de 1.32 Nm. Las características del motor que se utilizara se presentan en la Tabla 14 y se muestra su representación gráfica en la Figura 26. Una gran ventaja de obtención de estos motores es el encoder que viene incorporado.

**Tabla 14**

*Especificaciones del motor DC 5202 Series Yellow Jacket Planetary Gear Motor*

<b>Especificación</b>	<b>Magnitud</b>	
Voltaje:	12	VDC
Velocidad sin carga @ 12VDC:	30	Rpm
Corriente sin carga@ 12VDC:	0.25	A
Torque de bloqueo @ 12VDC:	20	± kg.cm
Corriente de bloqueo @ 12VDC:	9.2	A
Eje:	6	mm
Encoder: ciclos por revolución (eje del motor):	7	(Rises of Ch A)
Encoder: Eventos contables por revolución (eje del motor):	28	(Rises & Fall)
Tipo de Encoder- sensor magnético (Hall Effect)	Relativo, cuadratura	

Nota: En la tabla se detalla las especificaciones más relevantes de funcionamiento del motor y el encoder entregadas por (SERVOCITY, 2021)

### **Soporte y Locomoción (Subsistema 3)**

La parte soporte del subsistema se enfoca en la forma más adecuada para la distribución de los distintos componentes y materiales de los cuales se van a conformar

la plataforma. La plataforma debe garantizar una distribución de elementos confortable para el usuario el cual puede tener facilidad de manipulación de la misma, para ello los criterios a estudiar en el son:

- costo
- tamaño
- ergonomía

En la Tabla 15 se realizará la evaluación de los criterios establecidos, con esto podremos ver la ponderación de cada criterio y cuál de los criterios predomina sobre los demás.

**Tabla 15**

*Criterios para selección del soporte de la plataforma móvil*

<b>Criterio</b>	<b>Costo</b>	<b>Tamaño</b>	<b>Ergonomía</b>	<b>Suma +1</b>	<b>Ponderación</b>
Costo	-	10	5	16	0,79
Tamaño	0,1	-	1	2,1	0,10
Ergonomía	0,2	1	-	2,2	0,11
<b>Total</b>				<b>20,3</b>	<b>1</b>

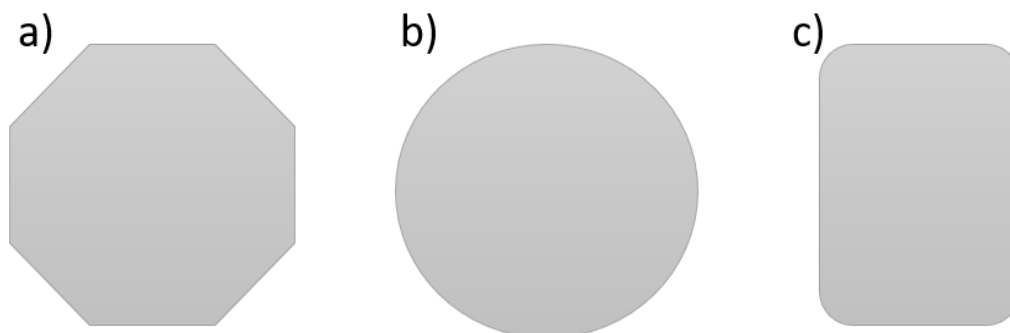
Nota: La tabla evalúa la importancia de los criterios con respecto a cada uno de ellos.

Para determinar las alternativas de la forma de la plataforma móvil, nos basamos en los antecedentes indagados y notamos en la Figura 30 que las formas de bases más común para las plataformas móviles son:



### Figura 30

*Formas de base de la plataforma móvil*



Nota: a) Opción 1: Base octogonal b) Opción 2: Base circular c) Opción3: Base rectangular.

Realizada por autor.

Una vez obtenidas las opciones para solventar el subsistema se hace su respectiva evaluación para cada criterio, como en el subsistema uno. A continuación, presentamos la resolución dada por la matriz de Holmes.

En la Tabla 16 se observa que la mejor base de forma se utiliza después de ser evaluado por los criterios es la forma de base circular.

**Tabla 16**

*Matriz de Holmes para el tipo de forma de bases*

	<b>Costo</b>	<b>Tamaño</b>	<b>Ergonomía</b>	<b>Suma</b>	<b>Prioridad</b>
Octogonal	0,07	0,01	0,01	0,09	3
Circular	0,36	0,07	0,07	0,51	1
Rectangular	0,36	0,01	0,03	0,40	2
			<b>Total</b>	<b>1</b>	

Nota: La matriz indica la prioridad de las opciones después de ser evaluada por los criterios correspondientes.

Una subdivisión del subsistema es la locomoción de la plataforma móvil; La configuración de la locomoción corresponde a una subdivisión del subsistema donde se analiza el tipo de locomoción para la plataforma se tiene los siguientes criterios:

- Costo
- Movilidad
- Estabilidad
- Peso
- Mínima fricción. (Fricción)

**Tabla 17**

*Criterios para selección de locomoción de la plataforma móvil*

<b>Criterio</b>	<b>Costo</b>	<b>Movilidad</b>	<b>Estabilidad</b>	<b>Peso</b>	<b>Fricción</b>	<b>Suma +1</b>	<b>Ponderación</b>
Costo	-	1	1	5	1	9	0,17
Movilidad	1	-	5	5	10	22	0,42
Estabilidad	1	0,2	-	5	5	12,2	0,23
Peso	0,2	0,2	0,2	-	5	6,6	0,13
Fricción	1	0,1	0,2	0,2	-	2,5	0,05
<b>Total</b>						52,3	1

Nota: La tabla evalúa la importancia de los criterios con respecto a cada uno de ellos.

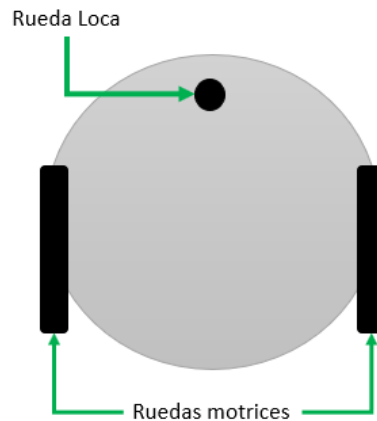
En la Tabla 17 indica la valoración de los criterios para saber qué criterio tiene una mayor ponderación.

Se debe tener en consideración para tener estabilidad en el sistema se debe tener mínimo 3 apoyos y debido a que se manejara la plataforma móvil como una plataforma diferencial debe tener dos motores por estos motivos se consideran las siguientes opciones para esta subdivisión.

- **Plataforma diferencial con una rueda loca:** La configuración más utilizada para plataformas móviles, el cual el robot utiliza dos ruedas motrices como se observa en la Figura 31 con una rueda loca para mantener la estabilidad.

**Figura 31**

*Esquema de una plataforma diferencial con una rueda loca*

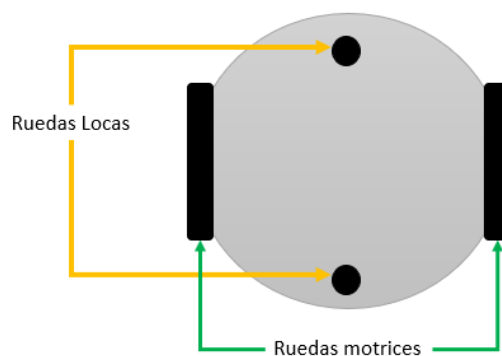


Nota: Figura realizada por autor.

- **Plataforma diferencial con dos ruedas locas:** Una de las formas más convencionales debido a su distribución como se muestra en la Figura 32, donde posee dos ruedas motrices y dos ruedas locas.

**Figura 32**

*Esquema de una Plataforma diferencial con dos ruedas locas*

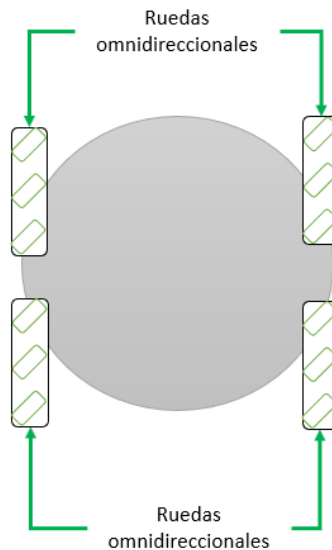


Nota: Figura realizada por autor.

- **Plataforma con cuatro ruedas omnidireccionales:** Sus cuatro ruedas son matrices y al ser omnidireccional la plataforma se traslada a cualquier dirección de espacio, su distribución se observa en la Figura 33

**Figura 33**

*Esquema de ubicación de ruedas omnidireccionales*



Determinado las opciones para analizar, las evaluamos como en el primer subsistema. La Tabla 18 tal presenta la resolución de prioridad desarrollada en la matriz de Holmes.

**Tabla 18**

*Matriz de Holmes para la locomoción de la plataforma móvil*

	Costo	Movilidad	Estabilidad	Peso	Fricción	Suma	Prioridad
P.D. con una rueda	0,117	0,191	0,042	0,075	0,022	0,45	1
P.D. con dos ruedas	0,045	0,191	0,134	0,042	0,022	0,43	2
Omnidireccionales	0,010	0,038	0,057	0,009	0,004	0,12	3
						<b>Total</b>	<b>1</b>

Nota: La tabla presenta la resolución de prioridad de las opciones.

El desarrollo del diseño inicia con el análisis de la forma de la base seleccionada con anterioridad, dentro de la forma se toma en cuenta la posición de las ruedas motrices y la rueda loca, donde se analizará las partes de la estructura mediante una herramienta CAE.

### **Diseño de la base de la estructura**

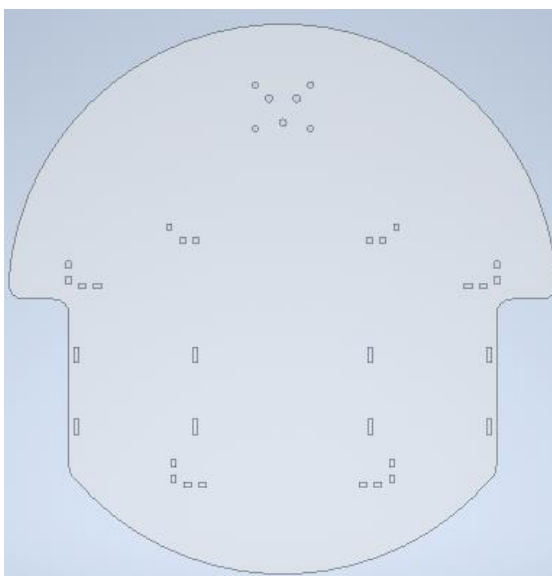
El diseño de la base de la plataforma iniciara con las consideraciones de unión de las diversas partes de la plataforma móvil para seguir con el análisis estático.

#### **Geometría de la Base**

Al realizar la geometría de la base de la plataforma móvil consideramos la distribución de los distintos elementos que son parte de la plataforma como consta en la Figura 34, donde denota la distribución de la rueda para el movimiento o tracción a una distancia desde el centro de cada una a 357mm y con respecto a la rueda loca tomando como referencia el centro de la distancia entre ruedas, se ubica a de 162mm.

### **Figura 34**

*Forma de la base de la estructura*



Nota: En la figura denota la forma de la base de la plataforma móvil, con agujeros para la adaptación de los distintos componentes. Figura realizada por autor.

### Análisis de resistencia de la base

La base de la estructura de la plataforma móvil debe soportar la carga del peso de los distintos componentes detallados en la Tabla 19. Al tener una gran incertidumbre en el diseño se recomienda en multiplicar por un factor de seguridad de 1.5 el peso resultante de la plataforma.

**Tabla 19**

*Pesos de Elementos*

	Masa-Unitario (g)	Cantidad	Masa Total (g)	Peso Total (N)
Batería	2405	1	2405	23,59
Ruedas	255	2	510	5,00
NUC	690	1	690	6,77
Motores	495	2	990	9,71
Cuerpo	1120	1	1120	10,99
Tarjetas Control	40	1	40	0,39
Electrónica	250	1	250	2,45
		<b>Total</b>	6005	58,91
		<b>F.S.</b>		1,50
		<b>Total</b>		88,36

Nota: Detalla los pesos de los diferentes elementos que conforman la plataforma móvil

**Tabla 20***Propiedades del Material*

<b>Característica</b>	<b>Valor</b>
Material	Acrílico
Espesor	3 mm
Resistencia a la tracción (Sut)	72 MPa
Limite Elástico (Sy)	45 MPa

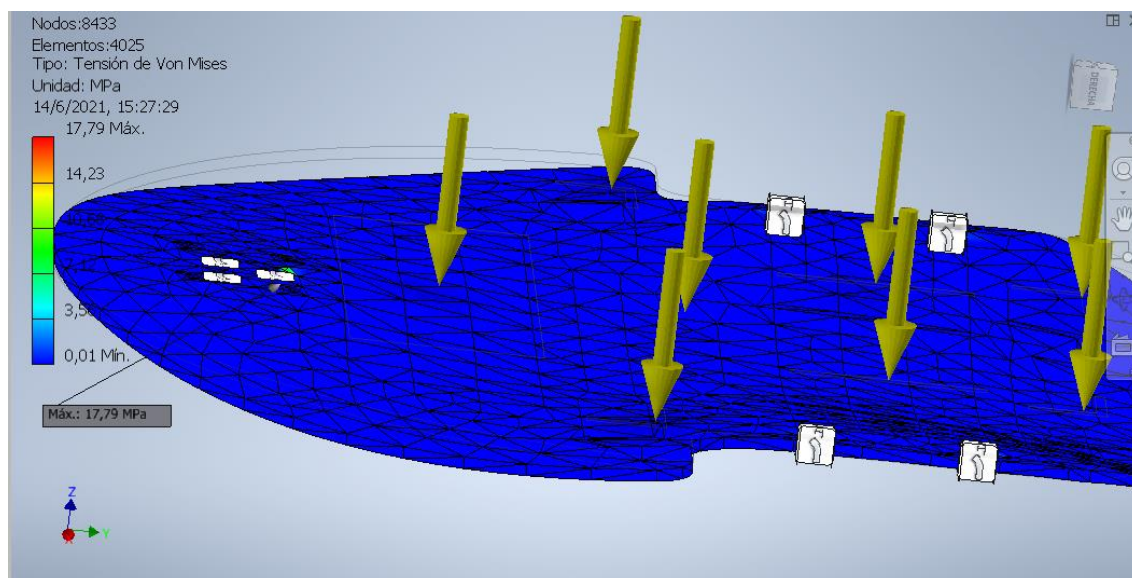
Nota: En la tabla se detalla las propiedades mecánicas, el tipo de material y el espesor a utilizar.

Para realizar el análisis de la resistencia de la base se utiliza programas CAE, asumiendo para todos los análisis un caso extremo donde se posiciona la NUC en la parte superior de la plataforma móvil. Se aplica cargas a cada punto de la base con una distribución aproximada. En la Tabla 20 se indica las propiedades del material acrílico.

Analizada la estructura de la base, se procede al análisis de resultados. En la Figura 35 se muestra los esfuerzos principales según la Teoría de esfuerzos de distorsión (Von Mises) en la base, tiene un valor máximo de 17.79 MPa, que surge de los concentradores de esfuerzos que presenta la estructura de la base. En la Figura 36 se muestra una deflexión de la base de 2.191 mm, lo cual consideramos que es una deflexión alta para la plataforma móvil y puede provocar ruptura.

**Figura 35**

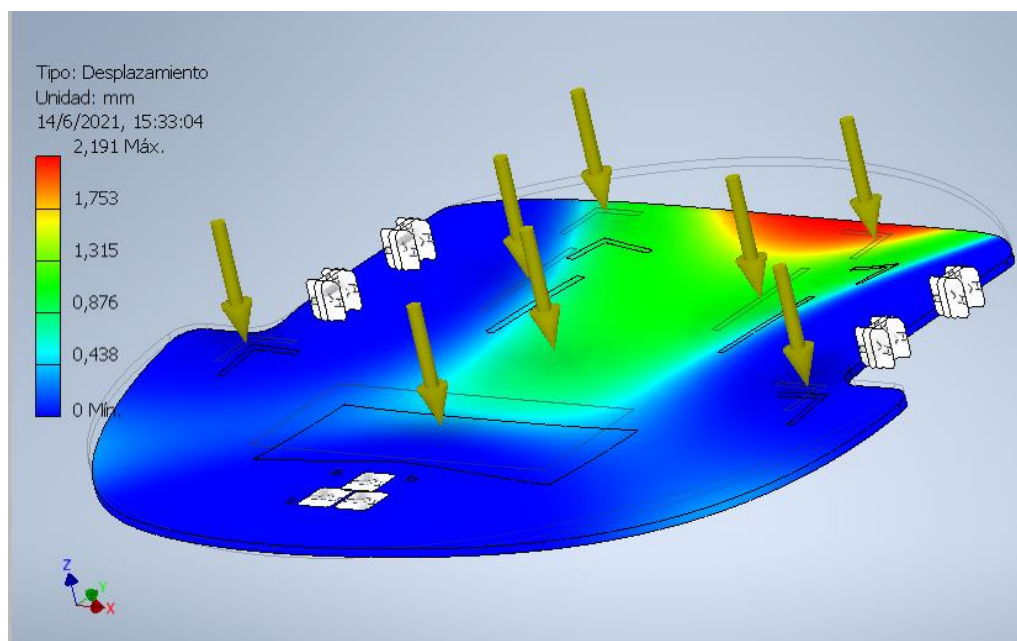
*Análisis estático CAE -Esfuerzos Teoría de Von Mises*



Nota: Figura realizada por autor.

**Figura 36**

*Deflexión de la base de la estructura*



Nota: Figura realizada por autor



### Diseño acople motor- estructura

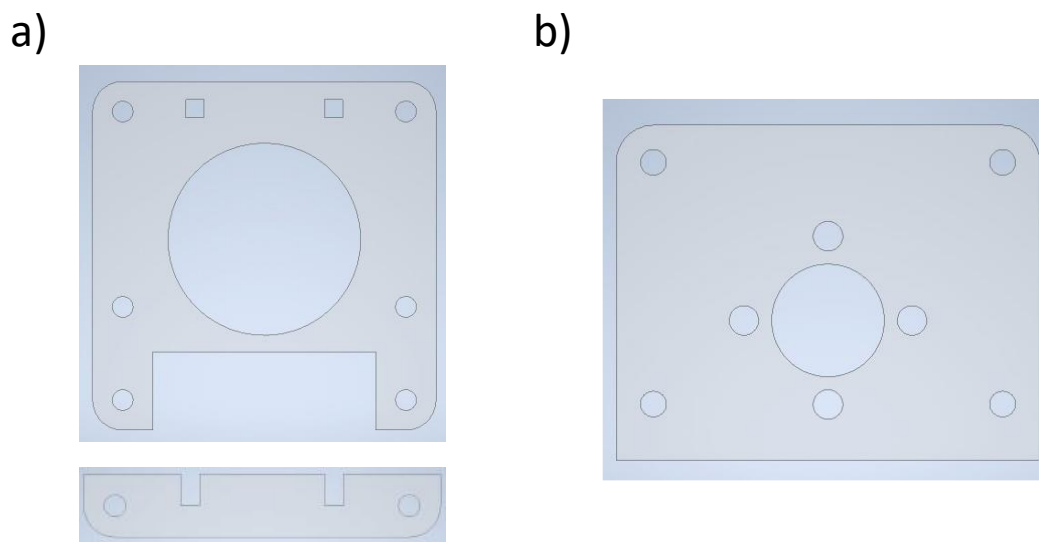
Analizaremos los concentradores de esfuerzo que pueden existir en esta área de contacto debido, a las fuerzas generadas pueden existir fallas como ruptura.

### Geometría del acople

Al realizar la geometría del acople consideramos la fijación del motor a la estructura de la base, garantizando el contacto con la base. Para ello se propone la siguiente geometría indicada en la Figura 37, la cual consta de dos partes la unión con la plataforma base y la unión con el motor. Se denota en la figura los agujeros principales para la adaptación del motor y los agujero para la unión entre estos con tornillos M3

**Figura 37**

*Forma geométrica de los acoples del motor*



Nota: a) acoples para unión con la base b) acople para unión con el motor Figura realizada por autor.

### Análisis de resistencia del acople

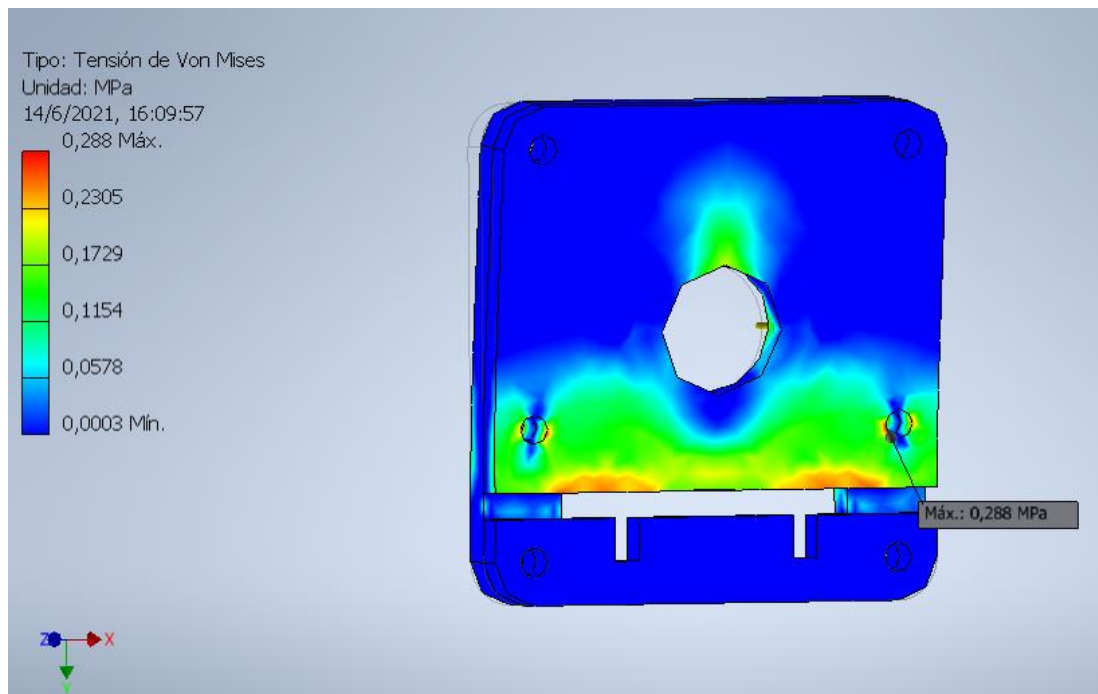
Para realizar el análisis de la resistencia del acople se utiliza programas CAE, se aplica una carga igual a al que genera el peso de las llantas de 2.5 N por cada llanta y un

peso de motor de 4.86 N. En la Tabla 20 se indica las propiedades del material acrílico que vamos a usar para su elaboración.

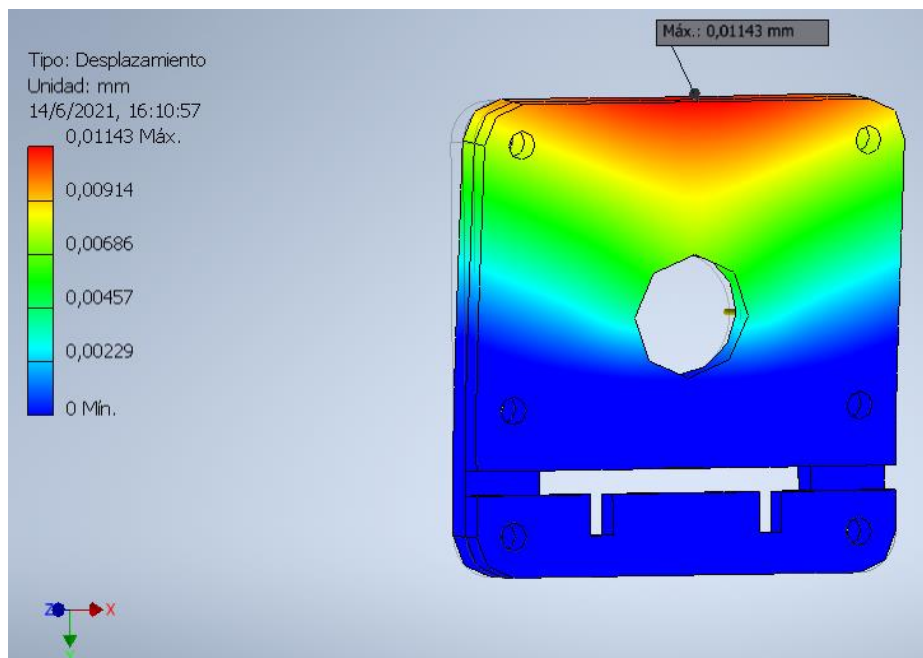
Finalizado el análisis del acople se procede al análisis de resultados. En la Figura 38 se muestra el valor máximo de 0.288 MPa de esfuerzo, obtenido por el estudio de la teoría de Von Mises en los acoples, estos esfuerzos surgen de los concentradores de esfuerzos que presenta en los acoples. En la Figura 39 se muestra la deflexión máxima de 0.01143 mm en los acoples.

### Figura 38

*Análisis estático CAE -Esfuerzos Teoría de Von Mises en acople*



Nota: Figura realizada por autor.

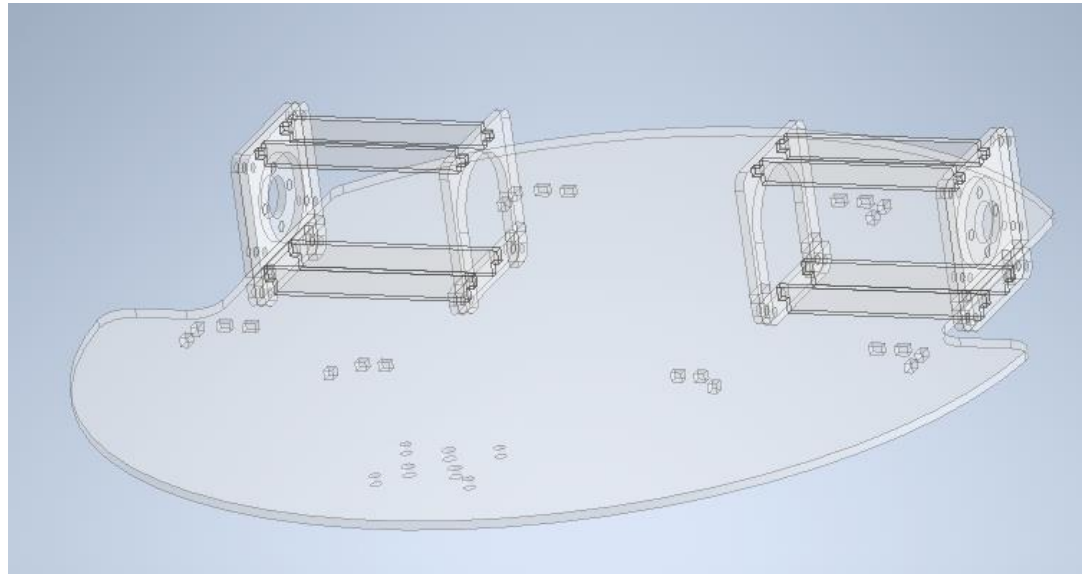
**Figura 39***Deflexión de los acoples*

Nota: Figura realizada por autor.

Para evitar la deflexión que podría ocasionar ruptura, por el peso de los distintos componentes, se propone una estructura base como se indica en la Figura 40, donde se puede apreciar que se centra la inercia de los acoples para evitar la deflexión y evitar algún esfuerzo que perjudique a la plataforma móvil.

**Figura 40**

*Forma geométrica final de la base de la plataforma móvil*

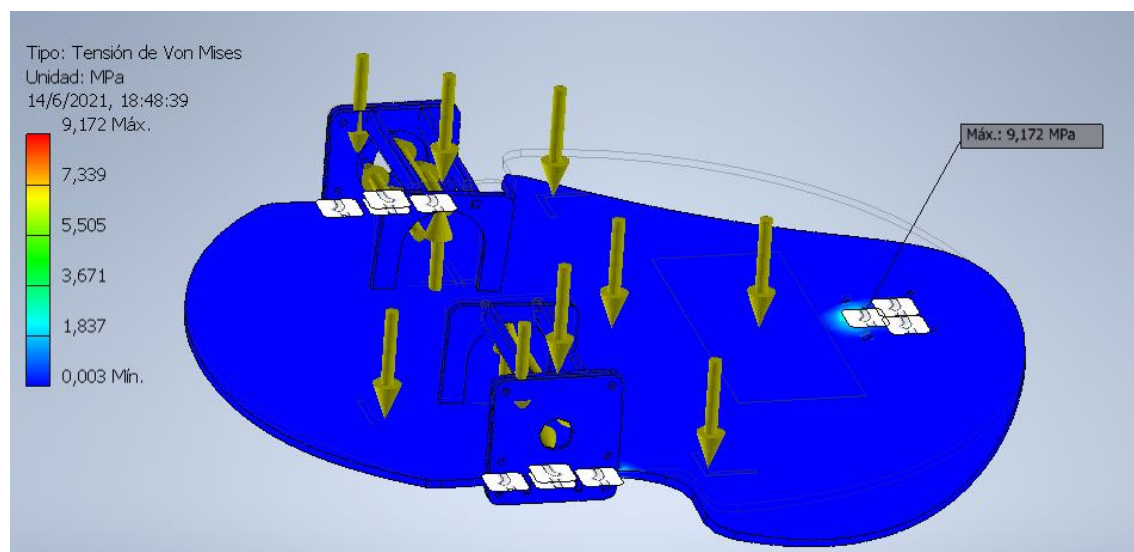


Nota: En la figura se denota el resultado final de la base de la plataforma donde consta el acople para los motores con la estructura y poniendo soportes para evitar deflexiones. Figura realizada por autor.

Con el aumento de estos acoples no solo baja el esfuerzo como se nota en la Figura 41, mostrando un valor máximo de 9.172MPa de esfuerzo, obtenido por el estudio de la teoría de Von Mises. En la Figura 42 se muestra la deflexión máxima a soportar de 0.7851 mm, la cual consideramos que es una deflexión admisible para el tipo de material.

**Figura 41**

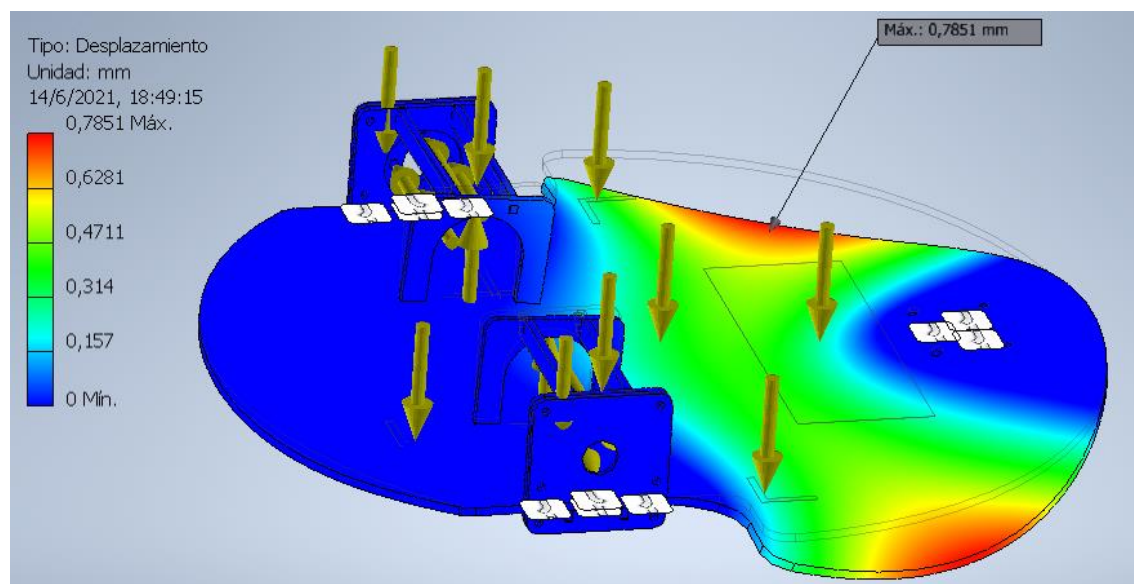
*Análisis estático CAE -Esfuerzos Teoría de Von Mises estructura*



Nota: Figura realizada por autor.

**Figura 42**

*Deflexión de la estructura*



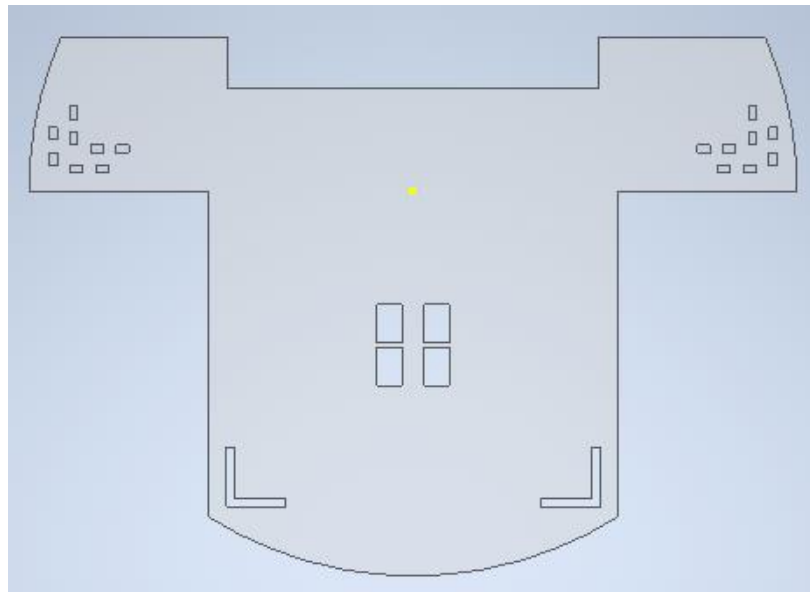
Nota: Figura realizada por autor.

### Geometría de la Repisa

La creación de las repisas es para ordenar los distintos elementos de control y potencia, para tener una mejor distribución de peso y una mejor ergonomía. Las repisas se muestran en la Figura 43.

#### Figura 43

*Forma geométrica de la repisa*



Nota: Figura realizada por autor.

#### Análisis de resistencia de la lámina

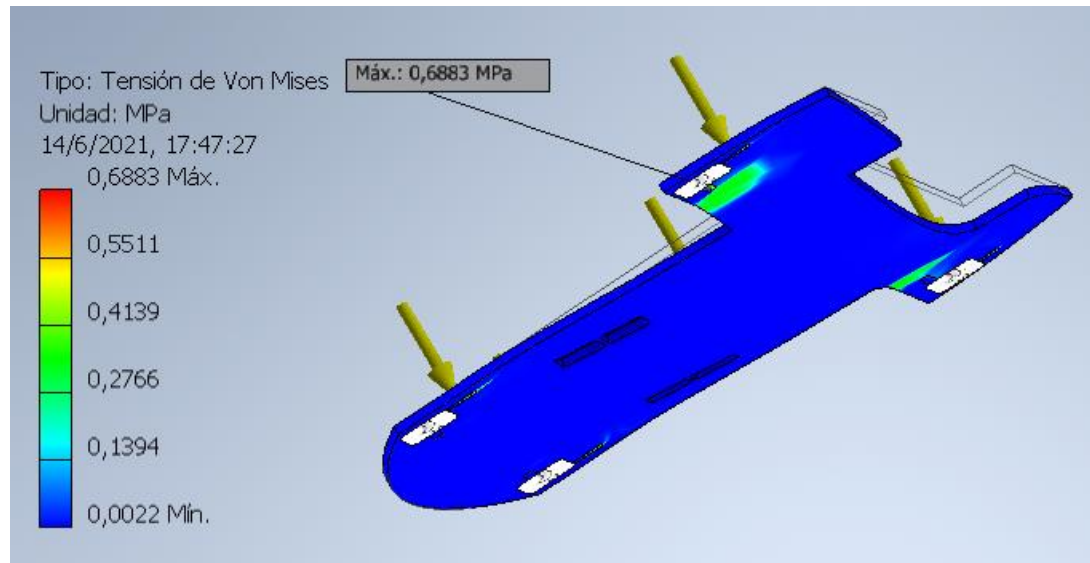
Se toma en cuenta el peso de las tarjetas de control 0.39N, procesador por el caso máximo 6.77N y los cables de 2.45N.

El análisis a realizar será estático el cual se desarrolla en programas CAE en la que se asigna carga estática a cada punto, en la Tabla 20 se encuentra las propiedades del material para el análisis CAE del material acrílico. Concluido el análisis estático se procede al estudio de los resultados, donde en Figura 44 se denota que la repisa soporta un esfuerzo máximo de 0.6883MPa. La repisa presenta una deflexión máxima de

0.161mm como muestra en la Figura 45, el valor de la deflexión máxima se presenta al colocar la NUC en la superior de la plataforma

**Figura 44**

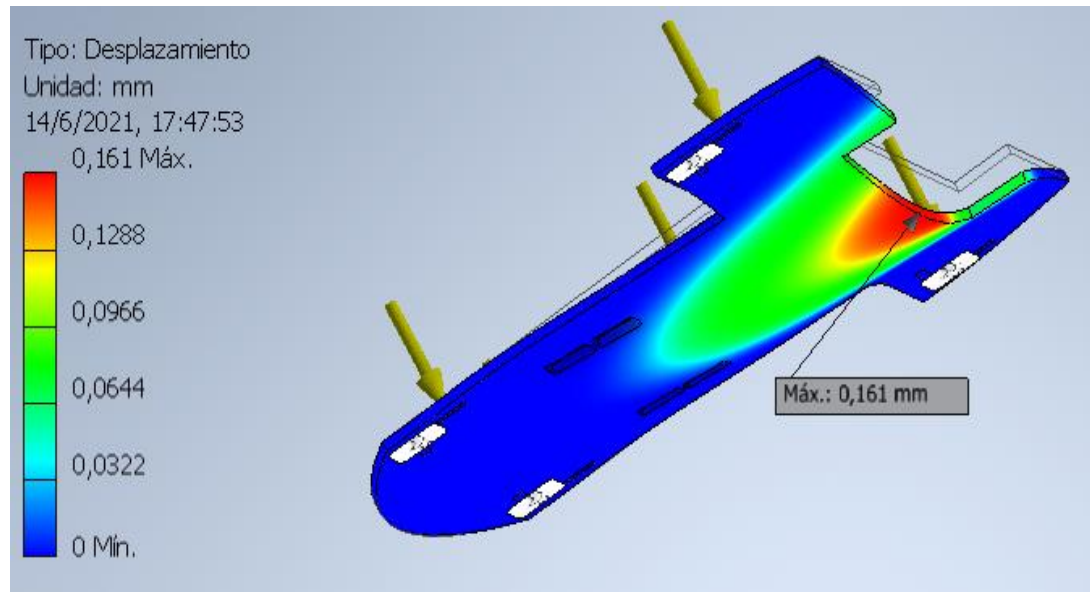
*Análisis estático CAE -Esfuerzos Teoría de Von Mises Repisa*



Nota: Figura realizada por autor.

**Figura 45**

*Deflexión de la repisa*



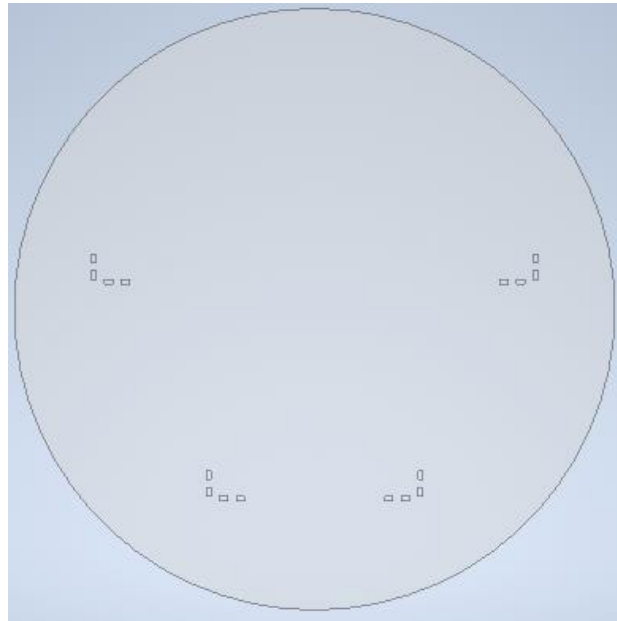
Nota: Figura realizada por autor.

### Geometría de la lámina

La lámina de acrílico tiene una forma circular como la primera prioridad dada en el subsistema de soporte y agujero para acoplar, a la plataforma móvil como se muestra en la Figura 46 la forma de la lámina con la que se trabajara.

#### Figura 46

*Forma geométrica de la lámina para la cámara Kinect*



Nota: Figura realizada por autor.

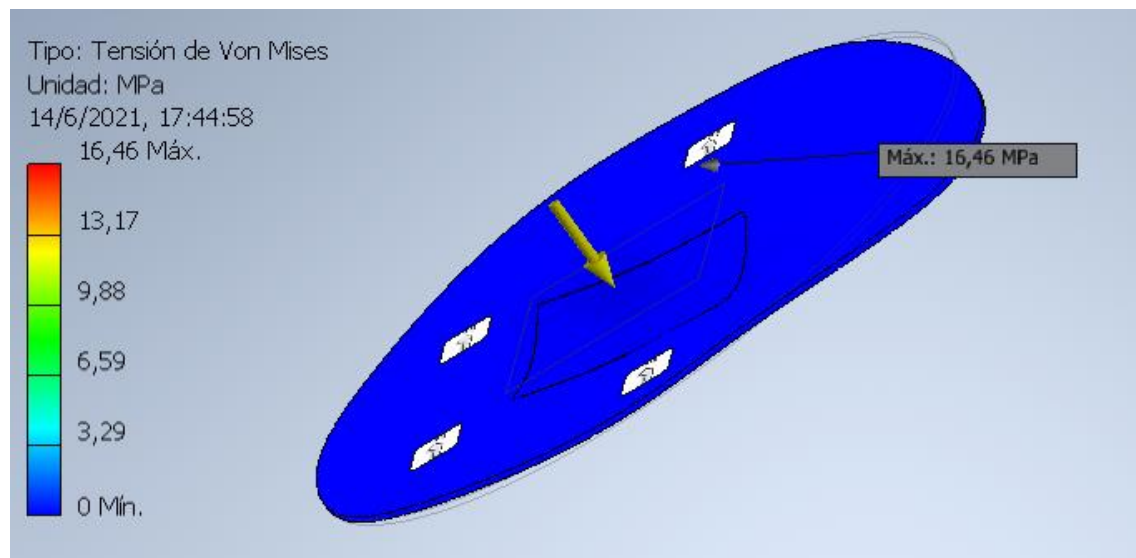
### Análisis de resistencia de la lámina

El análisis solo se realiza como caso extremo, donde la plataforma móvil debe soportar la NUC. El análisis estático se realiza en programas CAE, teniendo en cuenta las propiedades del material especificadas en la Tabla 20. Luego de ejecutar el análisis CAE procedemos con el análisis de los resultados. En la Figura 47 se denota un esfuerzo máximo de 16.46 MPa, obtenido este valor al realizar el análisis mediante la teoría de Von Mises. La deflexión máxima encontrada es de 0.3132 mm denotada en la Figura 48, donde la mayor deflexión se encuentra en el posicionamiento de la NUC.



**Figura 47**

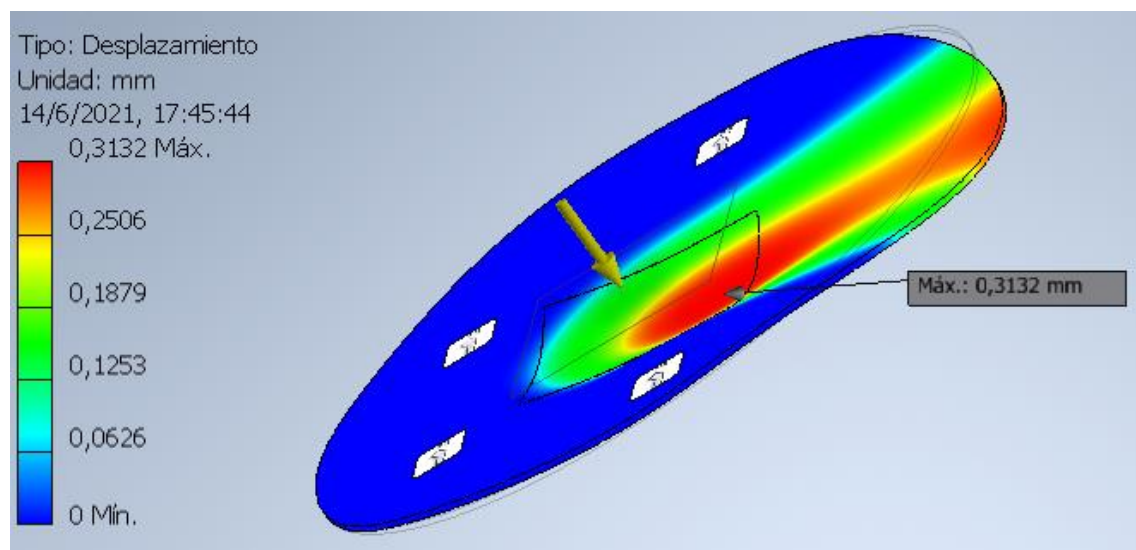
*Análisis estático CAE -Esfuerzos Teoría de Von Mises Soporte superior*



Nota: Figura realizada por autor.

**Figura 48**

*Deflexión del soporte superior*



Nota: Figura realizada por autor.

### **Controlador y procesador (Subsistema 4)**

El controlador a utilizar debe tener la capacidad de adquirir los datos de la parte física, enviar la odometría y recibir la señal de desplazamiento. Linux es un desarrollador de sistemas operativos libres, el cual ofrece una gran variedad de programas para poder utilizarlos sin necesidad de licencia, para la aplicación a generar se utilizará Ubuntu 18.04, la cual en la actualidad presenta pocas fallas de compatibilidad con los distintos controladores.

Ubuntu es uno los sistemas operativos más utilizados en el campo de la robótica debido a la compatibilidad que tiene con ROS, las distintas librerías desarrolladas para la utilización de ROS nos dan la factibilidad de uso del sistema operativo. ROS ofrece varias opciones para su instalación, para nuestro desarrollo decidimos utilizar ROS Kinect por recomendación (ROBOTIS e-Manual, 2020). Para la conexión entra el software y hardware se necesita una tarjeta de control para su comunicación la cual se evaluará bajo los siguientes criterios:

- Costo
- Velocidad de Procesamiento. (Velocidad)
- Facilidad de uso. (Facilidad)

En la Tabla 21 se presenta la ponderación de los 3 criterios indicados.

Las opciones que se presenta para el controlador son tarjetas de adquisición y procesamiento de datos, entra las cuales se plantean los tipos de tarjetas o procesadores que tengan la posibilidad de conectarse con ROS.

- **Intel NUC D54250WYKH:** Procesador Intel Core i5
- **Jetson Nano:** Procesador de 4 núcleos ARM A57 de 1.43 GHz.
- **Raspberry Pi 3 B+:** Procesador ARM v8 de 64 bits con una memoria RAM de 1 GB.

Una vez obtenido las opciones se realizará la evaluación para cada criterio como se presenta en el primer subsistema. En la Tabla 22 se encuentra la matriz de Holmes del subsistema controlador, en el cual nos indica la prioridad entre los procesadores señalados.

**Tabla 21**

*Criterios para la selección para el procesador*

Criterio	Costo	Velocidad	Facilidad	Suma +1	Ponderación
Costo	-	5	0.2	6.2	0.32
Velocidad	5	-	1	7	0.36
Facilidad	5	1	-	6.2	0.32
<b>Total</b>				19.4	1

Nota: Las opciones de los criterios llevan a definir la velocidad con que los procesadores pueden realizar procesos junto con la programación de las mismas.

**Tabla 22**

*Matriz de Holmes para la selección del procesador*

	Costo	Velocidad	Facilidad	Suma	Prioridad
Raspberry Pi 3 B+	0.179	0.014	0.058	0.25	3
Intel NUC	0.022	0.205	0.182	0.41	1
Jetson Nano	0.118	0.140	0.08	0.34	2
<b>Total</b>				1	

Nota: Producto entre los valores obtenidos de cada criterio comprados entre si con los valores individuales de cada opción en cada criterio.

Al mismo tiempo se analiza el procesador debe tener la capacidad de comunicación de la parte física con los programas desarrollado en la parte de software, para ello se considera las tarjetas de control que tengan la capacidad de usar un software

libre y se puedan sincronizar con ROS, para su manipulación y se seleccionarán según el criterio de:

- Costo.
- Fácil uso. (Programación)
- Disponibilidad en el mercado.

En la Tabla 23 se muestra la ponderación de cada criterio.

**Tabla 23**

*Criterios para la selección de tarjeta de control*

<b>Criterio</b>	<b>Costo</b>	<b>Programación</b>	<b>Disponibilidad</b>	<b>Suma +1</b>	<b>Ponderación</b>
Costo	-	5	10	16	0,78
Programación	0,2	-	1	2.2	0,11
Disponibilidad	0,1	1	-	2.2	0,11
<b>Total</b>				20,4	1

Nota: Las opciones de los criterios llevan a definir la facilidad de uso de la tarjeta de control.

Se utiliza una tarjeta de control que tenga software libre y de fácil acceso por ello se utiliza Arduino para adquirir los datos de los sensores y controlar el PWM con el driver de los motores, el driver se selecciona con el amperaje que consume cada uno de los motores a utilizar. Entre las opciones del Arduino a utilizar tenemos:

- **Arduino Mega:** Procesador de AVR ATMEGA2560.
- **ESP 32:** Posee un módulo Wifi incorporado procesador Tensilica Xtensa LX6 160 MHz.

Planteado las alternativas para la tarjeta de controlar, donde, procedemos a evaluarlas con cada criterio como se lo realizo en el primer subsistema. La Tabla 24 nos indica la tarjeta de control con mayor prioridad.

**Tabla 24**

*Matriz de Holmes para la selección de tarjeta de control*

	<b>Costo</b>	<b>Programación</b>	<b>Disponibilidad</b>	<b>Suma</b>	<b>Prioridad</b>
ESP 32	0,39	0,091	0,055	0,536	1
Arduino Mega	0,39	0,019	0,055	0,464	2
<b>Total</b>				<b>1</b>	

Para el control de la velocidad de los motores seleccionaremos el controlador (driver) que soporte la corriente necesaria para los dos motores, buscando compatibilidad con las tarjetas de control Arduino o ESP 32. Teniendo en cuenta estos parámetros se selecciona el driver Monster Moto Shield el cual tiene 2 puentes-H VNH2SP30, el módulo presente es un puente dual permitiendo el manejo de motores DC soportando una corriente constante de 30 A; como se observa en la Tabla 25 donde se presenta su especificación técnica del módulo.

**Tabla 25**

*Características Monster Moto Shield*

<b>Características</b>	<b>Magnitud</b>
Controlador	VNH2SP30
Canales	2
Tensión máxima	16 V
Resistencia	19 mOhm
Corriente recomendada sin disipador:	14 a
Corriente máxima	30 a
Frecuencia PWM máxima	20 kHz

Nota: La tabla presenta un resumen de las características técnicas del driver (Mercado Libre Ecuador, 2021)

Cada motor presenta un requerimiento de corriente de bloqueo de 9.2 A, como se ve en la Tabla 14, lo cual no tendrá inconveniente con el driver seleccionado. El módulo puede manipular hasta 2 motores.

Continuando con el módulo de movimiento de la plataforma se inicia con el estudio de envío y recepción de datos de la plataforma móvil se utilizará una tarjeta de control de fácil acceso comercial. La placa de control a utilizar es ESP 32, en la Tabla 26 se indica las especificaciones técnicas de la tarjeta de control.

### Tabla 26

#### *Especificaciones de la Tarjeta ESP 32*

<b>Procesador Tensilica Xtensa</b>	32 bits LX6 hasta 240MHz
Wifi	802.11b /g/n/e/l) (802.11 @2.4 Ghz hasta 150 Mbit/s
Bluetooth	v4.2 BR/EDR y Bluetooth low energy (BLE)
Rom	448 KiB
SRAM	520 KiB
RTC slow SRAM	8 KiB
RTC fast SRAM	8 KiB
eFuse	1 Kbit
Flash embebido	2MiB (ESP32-D2WD chip); 4MiB (ESP32 - PICO - D4 SIP module)
Periféricos compatibles	ADC, DAC, I2C, UART, INTERFAZ CAN 2.0, SPI, I2S, RMII y PWM
Seguridad tipo	IEEE 802.11, WFA, WPA /WPA2 y WAPI
Encriptacion	Memoria flash
Criptografía	Soportada por acelerador de hardware AES, SHA-2, RSA, ECC, RNG
Voltaje de trabajo	3.3 VDC
Energía y datos	Vía conector Micro USB 5VDC

Nota: La tabla detalla las características de la ESP 32. (Sigma Electronic, 2021)

Al tener que procesar gran cantidad de datos por el filtro de partículas, debemos tener un procesado con la capacidad de soportar las N partículas creadas y dar la solución a un tiempo admisible se utiliza el procesador Intel NUC D54250WYKH el cuál presenta las características indicadas en la Tabla 27.

**Tabla 27***Especificaciones Intel NUC D54250WYKH*

<b>Intel NUC D54250WYKH</b>	
Procesador	Core™ i5-4250U Processor (3M Cache, up to 2,60 GHz)
RAM	4 GB
Tarjeta gráfica	Intel HD Graphics 5000
Alimentación	19 V, 65 W

Nota: La tabla detalla las características de la Arduino Mega 2560. (Arduino Mega 2560

Características, Especificaciones, 2018).

### **Algoritmos de Localización (Subsistema 5)**

El subsistema 5 se enfoca en los algoritmos para encontrar la localización estimada, con el progreso de los subsistemas 2,3 y 4. El subsistema embarca el desarrollo de los tres principales paradigmas del SLAM estudiando su parte de estimación de localización, los cuales dan la posición estimada de la plataforma móvil por ende solventa el objetivo.

**Tabla 28***Criterios para la selección de Algoritmo de Localización*

<b>Criterio</b>	<b>Costo</b>	<b>Implementación</b>	<b>Entendimiento</b>	<b>Suma +1</b>	<b>Ponderación</b>
Costo	-	0,2	0,2	1,4	0,09
Implementación	5	-	1	7	0,45
Entendimiento	5	1	-	7	0,45
<b>Total</b>				<b>15,4</b>	<b>1</b>

El subsistema es la integración entre los datos que envían los sensores y la evaluación con el posicionamiento de la plataforma móvil, los algoritmos se desarrollaron en Python, por este motivo se utiliza los criterios analizar son:

- Coste computacional. (Costo)
- Facilidad de Implementación. (Implementación)
- Facilidad de entendimiento matemático. (Entendimiento)

En la Tabla 28 se muestra la ponderación de cada criterio.

Un robot en movimiento tiene la habilidad de estimar su localización dentro de un entorno, este es el principal propósito del trabajo, el entendimiento de los algoritmos de localización para lo cual se enumera dichos algoritmos, su desarrollo se realizará posteriormente.

- **Filtro de Kalman Extendido (EKF):** El filtro es derivado del filtro de Kalman, el algoritmo es desarrollado para sistemas no lineales.
- **Filtro de partículas (PF)** Este filtro crea N partículas randomicas y realiza una valoración de que tan cerca esta de la ubicación del robot.
- **Técnicas de Optimización de Gráfica Graph -SLAM:** Técnica usada para la implementación de mapas a partir de sus puntos de localización tomados cada cierto rango.

Planteado las alternativas de los algoritmos para estimar la localización procedemos a evaluarlas con cada criterio como se lo realizo en el primer subsistema. La matriz de Holmes nos indica la priorización de los algoritmos de localización basándonos en la ponderación dada por los criterios.

La Tabla 29 nos indica que la priorización en algoritmos es el filtro de partículas, el cual es la base del algoritmo desarrollado para encontrar la ubicación llamando AMCL (Algoritmo de localización de Monte Carlo).



**Tabla 29**

*Matriz de Holmes para la selección de algoritmos para la localización*

	Costo	Implementación	Entendimiento	Suma	Prioridad
EKF	0,06	0,15	0,08	0,30	2
FP	0,02	0,27	0,26	0,55	1
Graph-SLAM	0,01	0,03	0,11	0,15	3
<b>Total</b>				<b>1</b>	

Para el entendimiento de los algoritmos para la estimación de la localización se explicará la matemática que existe atrás de cada algoritmo.

### **Teorema de Bayes**

El teorema de Bayes desarrollado a continuación, aclara el problema de corrección, el cual se refiere a la actualización de medida corrigiendo el error presentado por la precisión.

$$p(x_i|z) = \frac{p(z|x_i) \times p(x_i)}{p(z)} \quad (41)$$

Donde:

$p(x_i)$ = probabilidad a priori

$p(z|x_i)$ = probabilidad de medida

$p(z)$ = normalización

$p(x_i|z)$ =probabilidad a posteriori

$x_i$ =una celda del grid;  $z$ = medidas

El producto solo de la probabilidad de medida con la probabilidad a priori obtenemos una distribución posteriori no normalizada:

$$\bar{p}(x_i|z) = p(z|x_i) \times p(x_i) \quad (42)$$

Para la normalización se realiza la suma de todos los índices  $i$ , pero del producto de la ecuación anterior:

$$\alpha = \sum_i \bar{p}(x_i|z) \quad (43)$$

De esta manera se obtiene el teorema de Bayes:

$$p(x_i|z) = \frac{p(z|x_i) \times p(x_i)}{\alpha} \quad (44)$$

### **Teorema de Probabilidad total**

El teorema presente representa a la convolución dado un movimiento de nuestra plataforma móvil

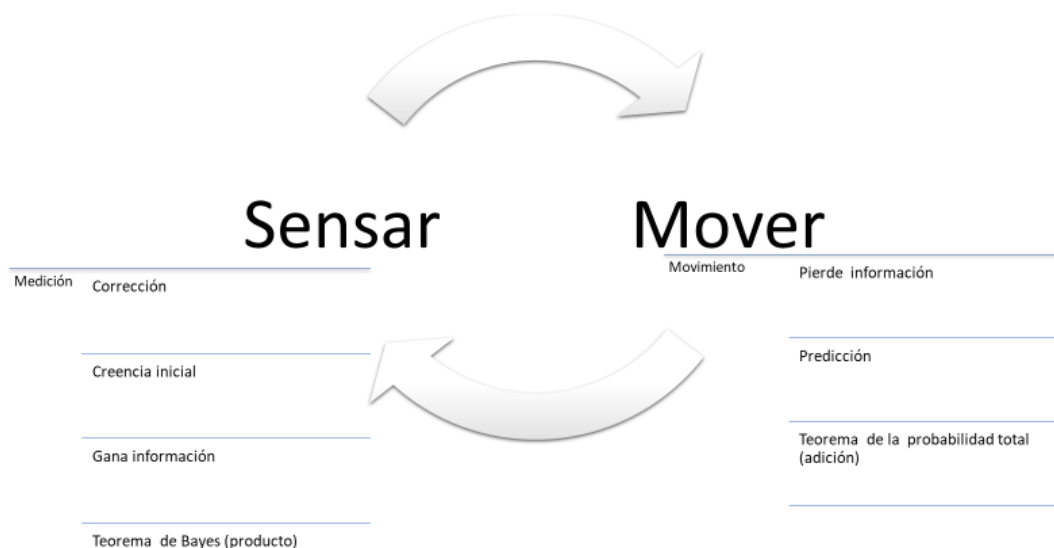
$$p(x_i^t) = \sum_j p(x_j^{t-1}) \times p(x_i|x_j) \quad (45)$$

Generaliza en los libros se encuentra de la siguiente manera:

$$p(A) = \sum_B p(A|B) \times p(B) \quad (46)$$

### **Estimación de la Localización**

La plataforma móvil al moverse genera incertidumbre, la cual con el paso del tiempo se hace inexacta la localización, a pesar del sensado obtenido por el robot. La localización es el resultado de la iteración de sensar y mover, el cual se detalla en la Figura 49.

**Figura 49***Iteración entre Sensar y Mover*

Nota: En la figura se detalla cómo funciona la iteración entre sensar y mover, y las cualidades que presenta cada uno. Figura realizada por autor.

**Filtro de Kalman**

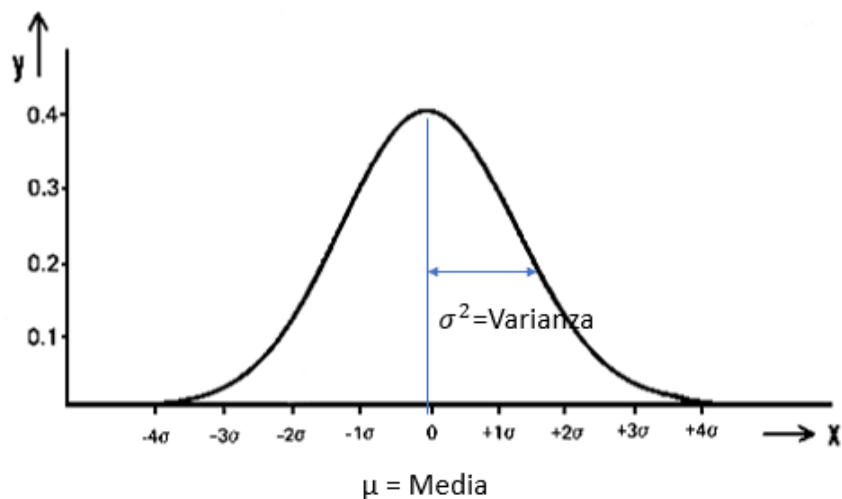
La base del filtro de Kalman es la probabilidad Gaussiana, el filtro de Kalman es usado en estado continuos Uni-modal, lo cual significa que puede tener solo un impulso en la curva Gaussiana

**Matemática del Filtro de Kalman**

El Filtro de Kalman se inicia con el entendimiento de la probabilidad Gaussiana, para proceder con el filtro de Kalman.

**Probabilidad Gaussiana**

En la Figura 50 describe la distribución gaussiana, representando la media y la covarianza de la cura, hay que tomar en cuenta que el área bajo la curva es igual a 1.

**Figura 50***Curva Gaussiana*

Nota: Figura realizada por autor.

La curva gaussiana es la representación gráfica de la ecuación de la campana de Gauss la cuál es:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \times e^{-\frac{1(x-\mu)^2}{2\sigma^2}} \quad (47)$$

Donde:

$\mu = \text{Media}$

$\sigma^2 = \text{Varianza}$

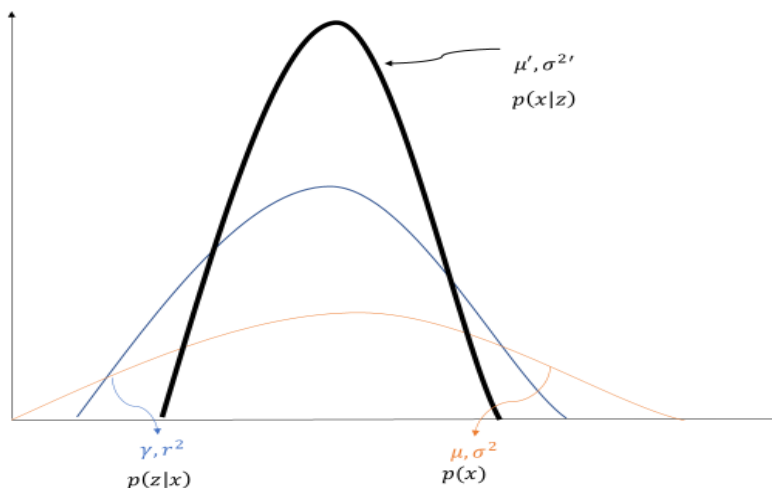
$\frac{1}{\sqrt{2\pi\sigma^2}}$  Variable para normalizar

### **Corrección – Actualización de medición**

La obtención de medidas genera una gran varianza y distinta media, por ello con la actualización de medida disminuimos la varianza como se ve en la Figura 51, obteniendo una varianza más pequeña, una media entre las dos medidas y así una medida más confiable

**Figura 51**

Corrección – Actualización de medición



Nota: En la figura muestra el Teorema de Bayes de dos medidas tomadas por el sensor y mostradas como campanas de Gauss. Figura realizada por autor.

Datos de las curvas:

$\mu, \sigma^2$  = media y covarianza de la curva una

$\gamma, r^2$  = media y covarianza de la curva dos

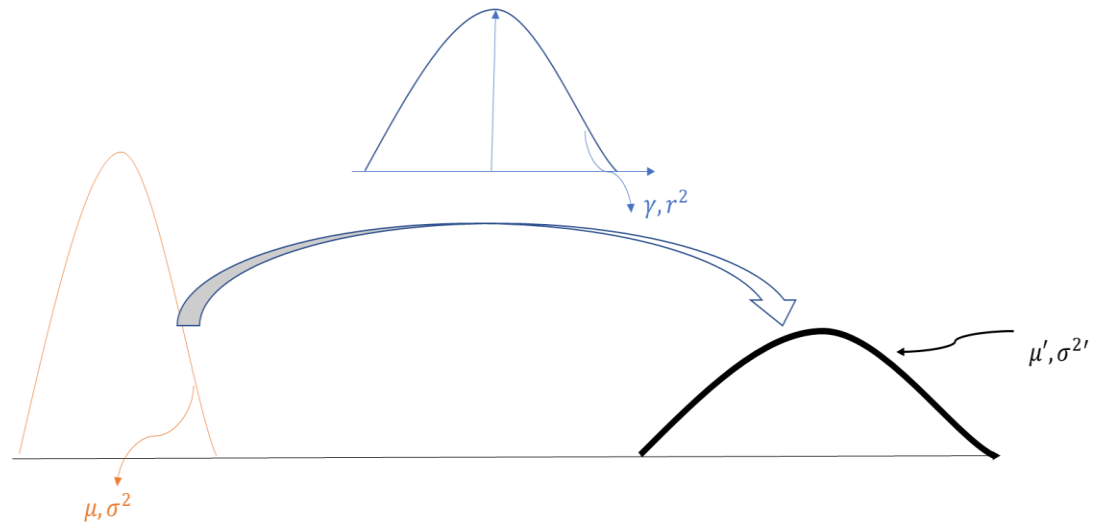
Para la obtención de la actualización de medida se aplica las siguientes fórmulas para la obtención de la nueva media y varianza

$$\mu' = \frac{r^2\mu + \sigma^2\gamma}{r^2 + \sigma^2} \quad (48)$$

$$\sigma^{2'} = \frac{1}{\frac{1}{r^2} + \frac{1}{\sigma^2}} \quad (49)$$

### Predicción- Actualización de movimiento

Al moverse el robot aumenta la incertidumbre como se muestra en la Figura 52, hay un aumento en la varianza obtenida después del movimiento:

**Figura 52***Predicción – Actualización de movimiento*

Nota: En la figura muestra el aumento de media y varianza de la campana de Gauss luego de realizar un movimiento. Figura realizada por autor.

Donde se tiene los siguientes datos:

$\mu, \sigma^2$  = media y varianza antes del movimiento

$\gamma, r^2$  = media y varianza dada del movimiento

$\mu'$  = media resultante de la actualización del movimiento

$\sigma'^2$  = Varianza resultante de la actualización del movimiento

Para encontrar la nueva media y varianza se realiza la suma respectiva:

$$\mu' = \mu + \gamma \quad (50)$$

$$\sigma'^2 = \sigma^2 + r^2 \quad (51)$$

### Secuencia lógica Filtro de Kalman Unidimensional

1. Posición inicial del robot ( $x$ )
2. Medición inicial dada por el sensor ( $Z$ ), precisión del sensor ( $\sigma^2$ ) y varianza de movimiento ( $\sigma^2+1$ )
3. Corrección de medida  $\mu', \sigma^{2'}$

$$\mu' = \frac{r^2\mu + \sigma^2\gamma}{r^2 + \sigma^2} \quad \sigma^{2'} = \frac{1}{\frac{1}{r^2} + \frac{1}{\sigma^2}}$$

$$m[i] = (\mu', \sigma^{2'})$$

4. Predicción de movimiento  $\mu', \sigma^{2'}$

$$\mu' = \mu + \gamma \quad \sigma^{2'} = \sigma^2 + r^2$$

$$m[i+1] = (\mu', \sigma^{2'})$$

5. Volver a paso 3 hasta N veces sea necesario

### **Filtro de Kalman Multivariable**

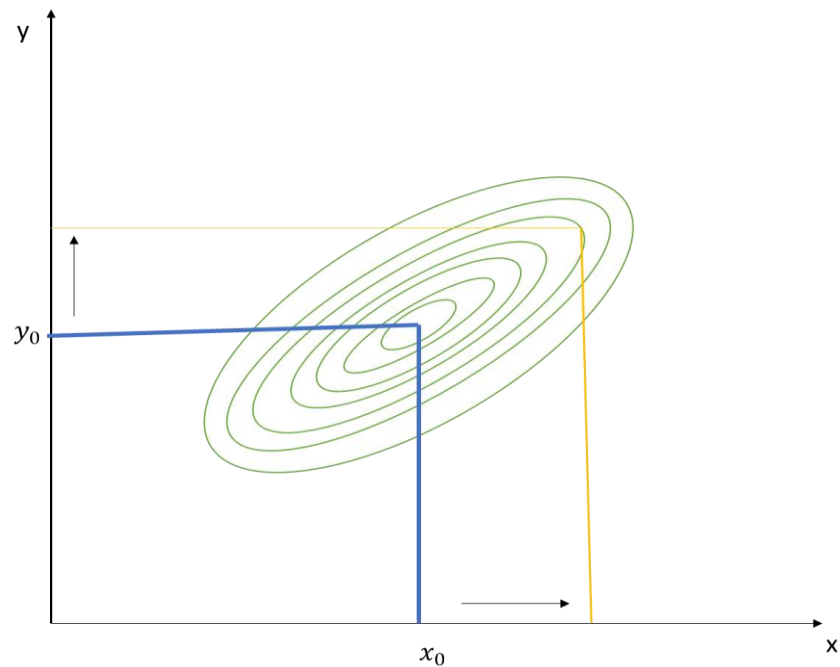
La Gaussiana multivariable tiene un alto grado de dificultad ya que se trabaja en N dimensiones representando su media:

$$\mu = \begin{pmatrix} \mu_0 \\ \mu_1 \\ \vdots \\ \mu_D \end{pmatrix} \quad (52)$$

La covarianza está compuesta por D filas D columnas, donde D es el número de dimensióne.

La curva gaussiana multivariable se representa en la Figura 53, es este caso es representando dos dimensiones. La curva gaussiana multivariable busca la correlación, donde se refiere a la probabilidad de que un valor en x muestre un valor en y. La fórmula se define mediante la siguiente ecuación:

$$f = (2\pi)^{-D/2} |\Sigma|^{-1/2} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)} \quad (53)$$

**Figura 53***Multivariable Gaussiana*

Nota: En la figura muestra la correlación entre el valor de  $x$  y el valor de  $y$ . Figura realizada por autor.

### **Diseño de filtro de Kalman**

Las variables de filtro de Kalman a menudo se nombran estados porque reflejan estados para su construcción se requiere:

- Estimación inicial (Localización)
- Incertidumbre (Caracterizado por la covarianza)
- Matriz de transición de estados (Construida desde las variables de estados)
- Función de Medida

### **Matemática**

Para obtener la nueva medida se arma las funciones de estado:



$$x' = x + \dot{x} \quad (54)$$

$$\dot{x}' = \dot{x} \quad (55)$$

$$\begin{pmatrix} x' \\ \dot{x}' \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ \dot{x} \end{pmatrix} \quad (56)$$

$$z = (1 \quad 0) \begin{pmatrix} x \\ \dot{x} \end{pmatrix} \quad (57)$$

Donde nuestras matrices de transición (F) y de medida (H) son:

$$F = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \quad (58)$$

$$H = (1 \quad 0) \quad (59)$$

Procedemos a la construcción de las matrices de Kalman:

Predicción - Actualización del movimiento

Se integra la matriz de movimiento u

$$x = F \times x + u \quad (60)$$

$$P = F \times P \times F^T \quad (61)$$

Donde:

$x$ = Estimación de posición

$F$ =Matriz de transición de estado

$u$ = Vector de movimiento

Corrección - Actualización de medida

Para realizar la corrección de medida para el diseño del filtro se procede con los siguientes pasos:

1. Mido el error del sistema

$$y = z - H \times x \quad (62)$$

## 2. Mapeo el error

$$S = H \times P \times H^T + R \quad (63)$$

## 3. Consigo la ganancia de Kalman

$$k = P \times H^T \times S^{-1} \quad (64)$$

## 4. Actualización de Matrices

$$x' = x + (k \times y) \quad (65)$$

$$P' = (I - k \times H)P \quad (66)$$

Donde:

$z$ = Medida del sensor

$H$ = Función de medida

$R$ = Ruido de la medición

$P$ = Covarianza de incertidumbre

$y$ = Error

$I$ =Matriz identidad

$x'$ =Nueva predicción

$P'$ = Actualización de la covarianza de incertidumbre

### Secuencia Lógica Filtro de Kalman

1. Posición inicial del robot ( $x_0, y_0$ )
2. Declaro la matriz de covarianza inicial alta debido a la incertidumbre
3. Predicción de la matriz de estimación (localización y velocidad) y de covarianza

$$x = F \times x + u$$

$$P = F \times P \times F^T$$

4. Medición inicial dada por el sensor (z)
5. Corrección de posición

$$y = z - H \times x$$

$$S = H \times P \times H^T + R$$

$$k = P \times H^T \times S^{-1}$$

$$x' = x + (k \times y)$$

$$P' = (I - k \times H)P$$

Volver a paso 3 hasta N veces de medidas tomadas-

### Filtro de Kalman Extendido

Los robots hacen frente a problemas más realistas donde el modelado de proceso es no lineal. Debido al modelado las ecuaciones cinemáticas son no lineales y por ende se modifica las matrices de medida y de transición obteniendo las matrices jacobianas para cada una, y se mantiene la secuencia de las matrices de Kalman.

$$f(u, x) \approx f(u, \mu) + \frac{\delta f(u, \mu)}{\delta x} (x - u) \quad (67)$$

$$h(x) \approx h(\mu) + \frac{\delta h(\mu)}{\delta x} (x - u) \quad (68)$$

Donde las matrices jacobianas vienen expresadas por

$$J_F = \frac{\delta g(\mu)}{\delta x} \quad (69)$$

$$f(x) = \begin{pmatrix} f_1(x) \\ \vdots \\ f_m(x) \end{pmatrix} \quad (70)$$

$$J_F(x) = \begin{pmatrix} \frac{\delta f_1}{\delta x_1} & \dots & \frac{\delta f_1}{\delta x_n} \\ \vdots & \dots & \vdots \\ \frac{\delta f_m}{\delta x_1} & \dots & \frac{\delta f_m}{\delta x_n} \end{pmatrix} \quad (71)$$

$$J_H = \frac{\delta h(u, \mu)}{\delta x} \quad (72)$$

Donde:

$J_H$  = Jacobiano de la función de Medida

$J_F$  = Jacobiano de la Matriz de transición de estado

### Secuencia Lógica Filtro de Kalman Extendido

1. Posición inicial del robot  $(x_0, y_0)$
2. Declaro la matriz de covarianza inicial alta debido a la incertidumbre
3. Predicción de la matriz de estimación (localización y velocidad) y de covarianza

$$x = f(u, x)$$

$$P = J_F \times P \times J_F^T$$

4. Medición inicial dada por el sensor (Z)
5. Corrección de posición
6. Volver a paso 3 hasta N veces de medidas tomadas

$$y = z - h(x)$$

$$S = J_H \times P \times J_H^T + R$$

$$k = P \times J_H^T \times S^{-1}$$

$$x' = x + (k \times y)$$

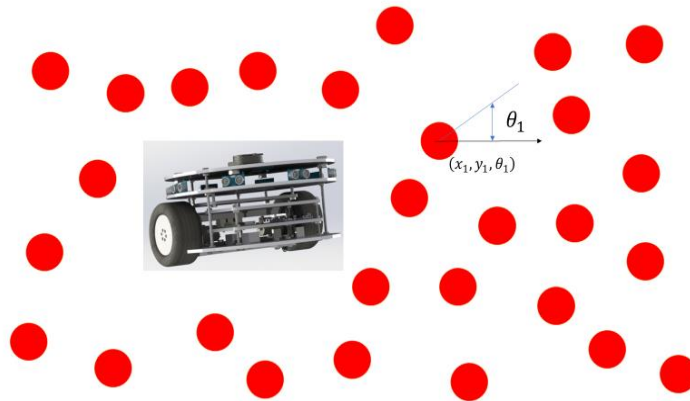
$$P' = (I - k \times J_H)P$$

### **Filtro de Partículas**

El filtro de partículas es uno de las bases de algoritmos más utilizados por su flexibilidad, su aplicación unimodal. El cuál consiste en la creación de N partículas randomicas, cada partícula tiene su propia orientación desde el eje x, con un valor de coordenada x y y como indica la Figura 54. El algoritmo modela el ruido como un Gaussiana con media cero.

## Figura 54

*Partículas creadas randomicamente*



Nota: Figura realizada por autor.

### Matemáticas

A continuación, se presenta la matemática a implementar en el filtro de partículas:

Creación de las partículas aleatoriamente

$$p(x) = (x, y, \text{orientación}) \quad (73)$$

Donde:

$p(x)$ = Partícula creada

$x$ = valor de coordenada x

$y$  = valor de coordenada y

*orientación*= ángulo de la partícula, medido desde el eje x positivo

Para facilitar el filtro de partículas se realiza un desajuste de la medición real y la medición predicha mediante la importancia, donde cada partícula tendrá un peso llamado importancia donde se considera dar mayor importancia a las partículas más cercanas. Para considerar la importancia de las partículas creadas nos ayudaremos con los landmarks y las formularemos bajo la probabilidad gaussiana.

$$\mu = \sqrt{(m_x - \text{Landmark}_x)^2 + (m_y - \text{Landmark}_y)^2} \quad (74)$$

Donde:

$\mu$ = distancia al landmark

$m_x$ =medida entregada en el eje x

$\text{Landmark}_x$ = valor de landmark del eje x

$m_y$ =medida entregada en el eje y

$\text{Landmark}_y$ = valor de landmark del eje y

La función Gaussian o ecuación de la campana de Gauss nos ayudara a encontrar el peso de cada partícula:

$$f = \frac{1}{\sqrt{2\pi\sigma^2}} \times e^{-\frac{1(\mu-m)^2}{2\sigma^2}} \quad (75)$$

Se desprecia la variable de normalización para le evaluación de cada partícula para la obtención de su peso:

$$p = e^{-\frac{1(\mu-m)^2}{2\sigma^2}} \quad (76)$$

Donde:

$\mu$ = distancia al landmark

$x$ = medida del robot

$\sigma^2$ = ruido de la medida

El peso de la partícula se obtiene del valor de la campana de Gauss sin normalizar de cada partícula, de esta manera realizamos el desajuste y vemos la importancia de peso. Para normalizar los pesos se suman todos los pesos y se divide cada uno por la suma total.

$$W = \sum_i \omega_i \quad (77)$$

$$\alpha_i = \frac{\omega_i}{W} \quad (78)$$

Consiguiente se realiza el remuestreo, donde las partículas con mayor peso son las que más cercanas se encuentran a la ubicación de la plataforma, las partículas con mayor peso se dibujan hasta obtener las N partículas.

<i>Particula</i>	<i>Peso</i>	<i>Normalizacion de peso</i>
$(x_1, y_1, \theta_1)$	$\omega_1$	$\alpha_1 = \frac{\omega_1}{W}$
$\vdots$	$\vdots$	$\vdots$
$(x_N, y_N, \theta_N)$	$\omega_N$	$\alpha_N = \frac{\omega_N}{W}$
$(x, y, \text{orientación})$	$W = \sum_i \omega_i$	$\sum_i \alpha_i = 1$

Lo realizado con el filtro de partículas con anterioridad conforma la relación con los teoremas de Bayes y de teorema de probabilidad total como se indica:

Teorema de Bayes:

$$p(x|z) = \frac{\overbrace{p(z|x)}^{\text{Peso de importancia}} \times \overbrace{p(x)}^{\text{Conjunto de partículas}}}{\underbrace{p(z)}_{\text{Remuestreo}}}$$

Partículas Correctas

Teorema de Probabilidad total:

$$p(x') = \underbrace{\sum}_{\text{Nuevo conjunto de Partículas}} \overbrace{p(x'|x)}^{\text{Ruido del sensor}} \times \underbrace{p(x)}_{\text{Conjunto de partículas}}$$

Muestreo de la suma tomando una particula

### Secuencia Lógica de Filtro de Partículas

1. Medición inicial dada por el sensor (Z), Numero de partículas a crear
2. Peso de cada partícula:
3. Remuestreo:
  1. Escojo una partícula de forma randomica
  2. Comparo con un máximo valor de peso
  3. Y comienzo a crear una nueva matriz de partículas con el mismo número de partículas
4. Vuelvo al paso 2 hasta que las partículas tengan valores semejantes

Para verificar la probabilidad si alguna partícula no es incluida en el remuestreo se utiliza la siguiente formula:

$$p = \frac{\alpha_i}{\sum_i \alpha_i}^N$$

Donde:

$p$ = Probabilidad de que la partícula no sea incluida

$\sum_i \alpha_i$ =Suma de los pesos de todas las partículas menos de la que se va analizar

$N$ = Número de las partículas

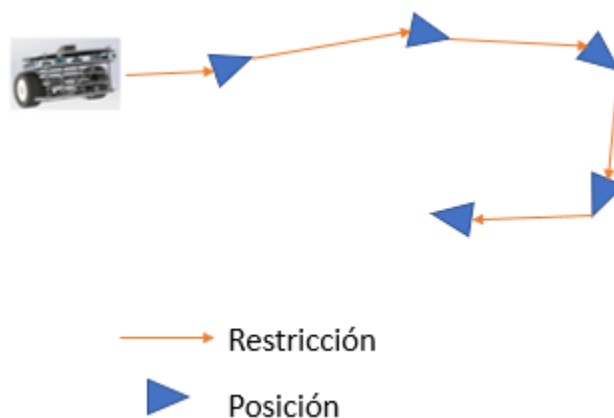
### ***Técnicas de optimización grafica Graph-SLAM***

Es un método OFF- Line debido a que recopila los datos de los sensores, donde guarda los datos de su entorno tomado cada cierto tiempo o cada cierta distancia, como se ve en la Figura 55, donde se representa la marca de posición o nodo y las restricciones



### Figura 55

*Posicionamiento del robot en un entorno*

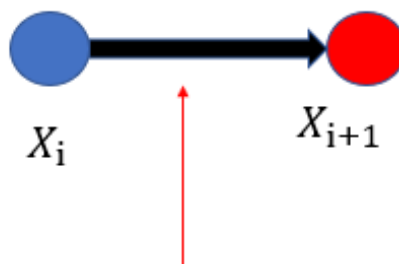


Los nodos son la representación del estado, con lo que calculamos la predicción. Con la observación del nodo anterior se relacionan estos, buscando una configuración de los nodos de manera que los valores reales y las predicciones sean lo más similar posible.

La posición consiste de  $n$  nodos  $X = X_{1:n}$  y cada  $X_i$  es una transformación 2D o 3D como muestra Figura 56.

### Figura 56

*Posición de dos nodos*

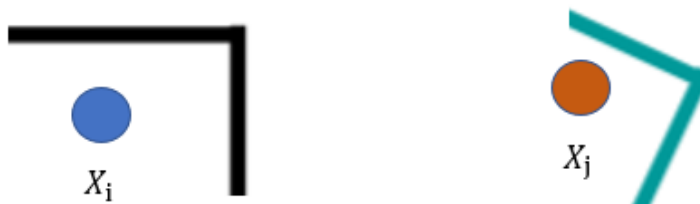


El borde representa la medida de odometría

El algoritmo inicia con la construcción de la medida virtual de las posiciones como se ve en la Figura 57:

**Figura 57**

*Mediciones entre dos posiciones*



Nota: Medición de la posición inicial y la siguiente representadas por el círculo de color azul y color naranja respectivamente.

Subsecuentemente se establece como la posición anterior determina a la Figura 58:

**Figura 58**

*Visualización de la anterior con la siguiente*



Nota: Representación como la posición actual observa a la siguiente posición.

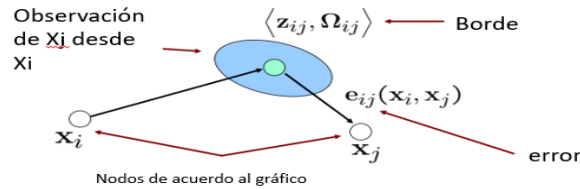
Las posiciones se expresan mediante matrices de transformación ( $X_i$  y  $X_j$ ), dado que  $X_i^{-1}$  es la transformación inversa de  $X_i$  es posible expresar la transformación relativa mediante  $X_i^{-1}X_j$  además se agrega la matriz de observación  $Z_{ij}$  que se determina por las mediciones de las posiciones  $i$  y  $j$ , es decir  $t_j - t_i$ . (Grisetti, Kümmerle, Stachniss, & Burgard, 2010)

Con lo anterior establecido el error en función de las matrices se expresa de la siguiente manera  $e_{ij} = Z_{ij}^{-1}(X_i^{-1}X_j)$  de este proceso se obtiene la siguiente matriz:

$$e_{ij} = \begin{pmatrix} R_{ij}^T \{ R_i^T (t_j - t_i) - t_{ij} \} \\ (\theta_j - \theta_i) - \theta_{ij} \end{pmatrix}$$

**Figura 59**

*Representación de los elementos en el algoritmo*



Nota: El círculo verde representa la observación combinada de i y j con lo que podemos establecer el error.

Dado que el objetivo de la técnica por optimización por gráficos es reducir estos errores entre el modelo de movimiento y el modelo de observación mediante el uso de errores de cuadrados ponderados y un método de mínimos cuadrados (algoritmo de Gauss-Newton) con una estructura de gráficos dispersa. La función es la suma de los errores cuadrados ponderados  $e_{ij}(x_i, x_j)$  a lo largo de los datos observados:

$$F = e_{ij}^T(x, t) \Omega e_{ij}(x, t) \quad (79)$$

La función de error se somete a linealización, se determina las constantes  $b$  y  $H$  mediante Gauss-Newton y el jacobiano. (Grisetti, Kümmerle, Stachniss, & Burgard, 2010)

$$J_{ij} = \frac{\partial e_{ij}(x)}{\partial x} \quad (80)$$

Al establecer la linealidad se puede representar la solución para obtener la posición de la siguiente manera:

$$e_{ij}(x + \Delta x) = e_{ij}(x) + J_{ij} \Delta x \quad (81)$$

$$= (e_{ij} + J_{ij} \Delta x)^T \Omega_{ij} (e_{ij} + J_{ij} \Delta x) \quad (82)$$

$$= e_{ij}^T \Omega_{ij} e_{ij} + 2e_{ij}^T \Omega_{ij} J_{ij} \Delta x + \Delta x^T J_{ij}^T \Omega_{ij} J_{ij} \Delta x \quad (83)$$

$$= c_{ij} + 2b\Delta x + \Delta x^T H \Delta x \quad (84)$$

$$H\Delta x = -b \quad (85)$$

### **Planificador de ruta (Subsistema 6)**

La plataforma debe tener la capacidad de navegar en un entorno conocido aplicando algún planificador de ruta.

El subsistema se encarga de la navegación la cual tiene como factores para su realización la planificación y el mapa por donde le robot va a circular. Con esto en cuenta se toma los siguientes criterios:

- Coste computacional. (Costo)
- Facilidad de implementación. (Facilidad)
- Confiabilidad

En la Tabla 30 evaluaremos los criterios presentados con anterioridad, para tener una mejor percepción del algoritmo de planificación que se podrá aplicar a nuestra plataforma móvil. De esta manera aseguramos que el algoritmo de planeación de ruta nos ayudará a tener un mejor resultado con los algoritmos que vamos a trabajar en la plataforma móvil.

**Tabla 30**

*Criterios para la selección de Planificador de ruta*

<b>Criterio</b>	<b>Costo</b>	<b>Facilidad</b>	<b>Confiabilidad</b>	<b>Suma +1</b>	<b>Ponderación</b>
Costo	-	1	5	7	0,57
Facilidad	1	-	1	3	0,25
Confiabilidad	0,2	1	.	2,2	0,18
<b>Total</b>				12,2	1

La navegación a realizar es dependiente de un grid o mapa y la planificación de trayectoria que tenemos, las opciones de planificación a evaluar son:

- **A\***: Es el algoritmo con mayor factibilidad de entendimiento donde a partir de una coordenada final e inicial nuestro robot planificara la trayectoria evitando los obstáculos previstos en el grid (Udacity, 2021).
- **Planificación Dinámica**: Ofrece una planificación para cual punto de partida que sea posible que el robot inicie su trayectoria y asigna un movimiento a cada cuadrícula del grid (Udacity, 2021).

Una vez declarada las alternativas u opciones procedemos a evaluarlas para cada criterio del subsistema, como se realiza en el primer subsistema. A continuación, en la Tabla 31 se presenta la matriz de Holmes para ver el algoritmo de planificación con mayor prioridad

**Tabla 31**

*Matriz de Holmes para Planificador de ruta*

	<b>Costo</b>	<b>Facilidad</b>	<b>Confiabilidad</b>	<b>Suma</b>	<b>Prioridad</b>
A*	0,478	0,123	0,090	0,69	1
Dinámica	0,096	0,123	0,090	0,31	2
			<b>Total</b>	1	

Como resultado de prioridad se obtiene A\*, el cual es un planificador de ruta clásico basado en grafos y de los más usados.

La planeación trata de buscar una trayectoria donde podamos buscar una solución para partir de un punto A y llegar a un punto B. El algoritmo a implementar para la planeación de trayectoria será A-star el cual hace que el robot trabaje lo necesario para llegar a la ubicación deseada, para el cumplimiento de la planificación A-star utiliza funciones Heuristicas la cual busca subestimar o igualar a la distancia real del punto de

llegada, se plantea para el funcionamiento de la plataforma el algoritmo de (Udacity, 2021). A continuación, se presenta el pseudocódigo de A\*.

### Pseudocódigo de A\*

```

Creo punto de partida
Creo punto de llegada
Inicializamos arreglo OPEN
Inicializamos arreglo CLOSED
Agregamos punto de partida al arreglo OPEN
Mientras el arreglo no este vacío
{
    Obtenemos el punto n del arreglo OPEB con la más baja f(n)
    Añadimos n al arreglo CLOSED
    Si n es el mismo que el punto de llegada hemos encontrado la solución;
    retornamos solución (n)
    Generamos cada punto sucesor n' de n
    Para cada punto sucesor n' de n
    {
        Establecemos el
        Establecemos h(n') como la distancia heurística estimada al punto de
        llegada
        Establecemos g(n') como g(n) más el costo para llegar a n' desde n
        Establecemos f(n') en g(n') más h(n')
        Si n' está en el arreglo OPEN y la existente es bueno o mejor
descartamos      n' y continuamos
        Si n' está en el arreglo CLOSED y la existente es bueno o mejor,
descartamos n' y continuamos
        Eliminamos las apariciones de n' de OPEN y CLOSED
        Agregamos n' al arreglo OPEN
    }
}
Retorno error (Si llegamos a este punto hemos buscado todos los puntos accesibles y
aún no hemos encontrado la solución, por lo tanto, no existe)

```

### **Especificaciones Finales del Diseño**

Una vez analizado las alternativas para cada subsistema escogemos las que tienen mayor prioridad para solventar el funcionamiento de la plataforma móvil como se indica en la Tabla 32 y con esto obtenemos las especificaciones finales para proceder con el diseño de cada módulo. Para ciertos casos como el subsistema de medición del entorno no solo se tomará la alternativa con mayor prioridad, también las demás alternativas, debido a que la unión de estos dispositivos garantiza el funcionamiento de la plataforma.

**Tabla 32***Módulos de la plataforma móvil*

<b>Módulo</b>	<b>Subsistema</b>	<b>Alternativa de solución</b>
<b>Estructura de la plataforma móvil</b>	Soporte y locomoción	El sistema usara un soporte de forma circular con una locomoción de dos ruedas motrices y una rueda loca.
	Fuente de Energía	La alimentación viene de una batería sellada con un voltaje de 12 V con un amperaje por hora de 7 Ah.
<b>Movilidad de la plataforma móvil</b>	Sistema de tracción	Los motores a utilizar serán motores DC con caja de reducción los cuales ayudan a tener un mejor torque y enconder magnéticos incluidos.
	Controlador	El controlador a utilizar es un ESP 32 con un procesador Intel NUC, la cual nos brinda la factibilidad de ser programada en ROS para el procesamiento de los códigos.

<b>Funcionamiento de la plataforma móvil</b>	Localización	Se utilizará los tres algoritmos que planteamos en alternativas para encontrar la localización estimada
	Planificación de ruta	Se planificará la ruta por donde ira la plataforma usando A*.

Nota: En la tabla separamos los subsistemas en módulos para realizar su estudio separado posteriormente.

En el módulo estructura, se desarrollará el análisis de la base de la plataforma, en la cual se integran los elementos principales de la tracción, como son las ruedas. Teniendo para la aplicación dos ruedas motrices y una rueda loca.

El módulo de movilidad se analiza el movimiento de tracción para esto iniciaremos con el modelado matemático de la plataforma diferencial en su estudio cinemático y dinámico.

El módulo de funcionamiento de la plataforma móvil nos enfocamos en analizar los algoritmos para la estimación de la localización del robot y la planeación de ruta que tendrá el robot para realizar su navegación Planificación de ruta A\*

### **Resumen**

El proceso de metodología y diseño iniciamos tomando en cuenta las especificaciones dadas por el clientes , tomando estas especificaciones las pasamos a las especificaciones técnicas, para la cual nos basamos en el diseño de producto de (Ulrich & Eppinger, 2013) donde a partir de ver el problema como un diagrama de proceso , donde consideramos las entradas que tendrá nuestro problema y las salidas del mismo como si fuera una caja negra el problema, para luego realizar subfunciones del problema e irlas dividiendo a las subfunciones hasta tener claro que se realizara.



Una vez tenidas las subfunciones procedemos agrupar las subdivisiones que se relacionan, de este modo damos origen nuestros subsistemas del problema. Ahora se procede a establecerá especificaciones que se necesita para cada subsistema, de este modo tomamos los criterios que sean más acordes al subsistema y los evaluamos con las alternativas u opciones que se plantean para cada subsistema y con la ayuda de la matriz de Holmes obtenemos la prioridad de elección para solventar cada subsistema, estas prioridades conformaran las especificaciones finales del diseño.

El diseño inicia con el análisis de la estructura que tendrá la plataforma móvil, en el cual estudiamos la deflexión y los esfuerzos que tendrá el material acrílico con un análisis estático. Continuando con el análisis cinemático y dinámico de la plataforma , para realizar su modelamiento dinámico inverso para ver el torque requerido por la plataforma ayudando a la selección de motor DC. Teniendo en cuenta la potencia que requiere los motores para su funcionamiento , iniciamos con el análisis de los sensores y tarjetas de control para el funcionamiento adecuado de la plataforma , de este modo tener una fuentes de energía para la alimentación de la plataforma móvil. Finalizando con el estudio de los distintos algoritmos de localización y planificación de ruta por A\*.

## Capítulo IV

### Construcción y Pruebas

El capítulo detalla la implementación de los de los resultados obtenidos en el Capítulo 3 para la construcción de la plataforma móvil, donde se probarán los algoritmos para la estimación de la localización. Se establece como prioridad la comprobación de los algoritmos, para ello se inicia con una plataforma móvil simulada. Continuando con la construcción de la plataforma móvil.

Construida la plataforma se ejecuta una serie de pruebas para evaluar la estimación de la localización de cada algoritmo.

#### **Simulación de los algoritmos para estimación de localización**

Para probar los algoritmos de localización se utiliza Gazebo un entorno de simulación 3D, Gazebo permite observar el comportamiento de la plataforma móvil en un mundo virtual. Por otra parte, gazebo no presenta error en los sensores simulados, por dicho motivo se inserta ruido aleatorio de  $\pm 10$  mm a las medidas entregadas por la odometría, para conseguir una simulación medidas más reales.

La funcionalidad de la simulación se detalla a continuación: La planificación envía la ruta creada por A\* desde un grid predeterminado; la planificación alimenta al movimiento de la plataforma móvil, donde, mediante el algoritmo movimiento se publica la velocidad con la que avanzara la simulación de la plataforma móvil.

Otro requerimiento para el funcionamiento del movimiento es la estimación de posición entregado por el algoritmo EKF o filtros de partículas, en el caso de la técnica por optimización gráfica se recolecta la información de la odometría con ruido y se realiza la estimación de posición con dichos datos.

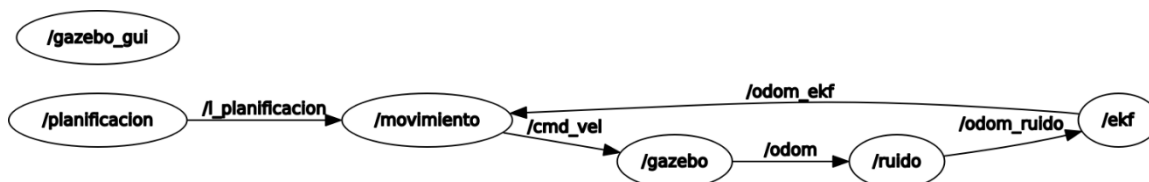
Los filtros EKF y de partículas se alimentan con la odometría con ruido; para la obtención de la odometría con ruido se desarrolla un algoritmo encargado de infiltrar ruido a las mediciones odométricas de la plataforma móvil.

## Simulación de Filtro de Kalman Extendido (EKF)

La covarianza del filtro de Kalman Extendido disminuye con cada movimiento, por su retroalimentación. La alimentación de los filtros se da por la odometría entregada por la plataforma móvil como indica en la Figura 60, donde, los nodos indican cómo funciona la simulación de la plataforma móvil.

**Figura 60**

*Nodos ROS Filtro de Kalman Extendido - Simulado*

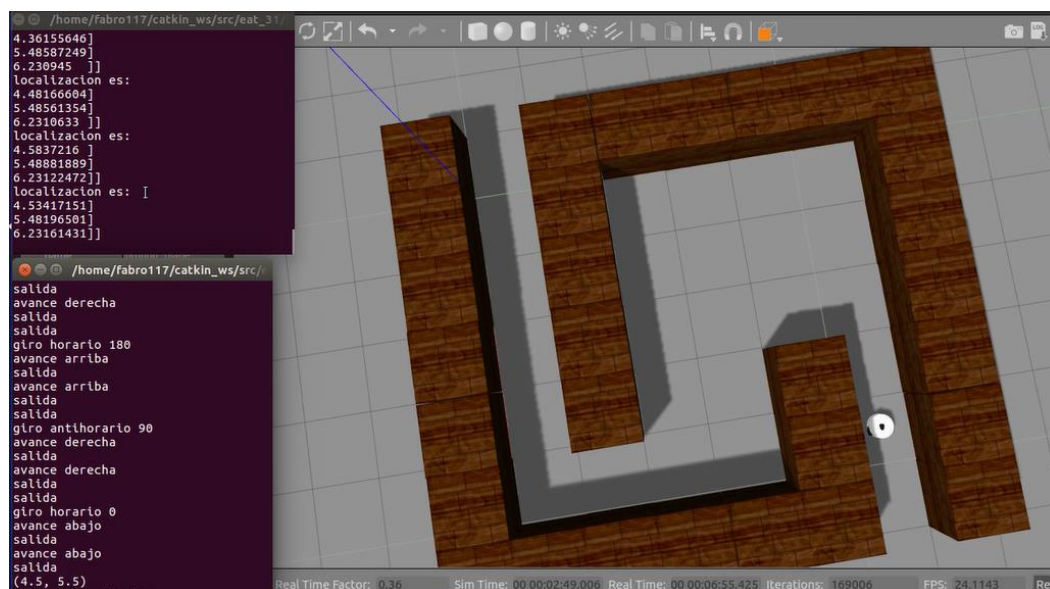


Nota: La figura muestra los nodos de simulación en Gazebo para el filtro EKF.

Con esta secuencia de algoritmos conectados mediante ROS da el funcionamiento simulado por Gazebo como se observa por la Figura 61.

**Figura 61**

*Simulación Gazebo - Algoritmo EKF*



Nota: La figura se observa a lado izquierdo las coordenadas resultantes del EKF y en el lado derecho la simulación en su mundo virtual.

Los resultados obtenidos simulados se indican en la Tabla 33, en la cual presenta el error localización estimada y la localización deseada del robot.

**Tabla 33**

*Error de localización simulado utilizando EKF*

Posición	Simulado (m)	Deseado (m)	Error
X	4,53	4,50	0,7%
Y	5,48	5,50	0,4%
orientación	6,23	6,28	0,8%

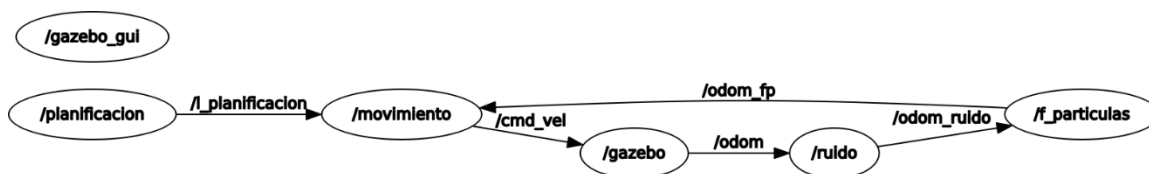
Nota: La tabla muestra el error relativo de la localización deseada vs la localización estimada por el filtro de Kalman Extendido

### **Simulación Filtro de Partículas**

La localización estimada del filtro de partículas depende del número de partículas creadas de forma random por todo el grid, donde cada partícula es evaluada y la partícula más cercana a la plataforma móvil, es la que se toma en cuenta como la estimación de la localización de la plataforma móvil. En la Figura 62 se detalla como los nodos se entrelazan para la simulación de la plataforma

**Figura 62**

*Nodos ROS Filtro de Partículas - Simulado*

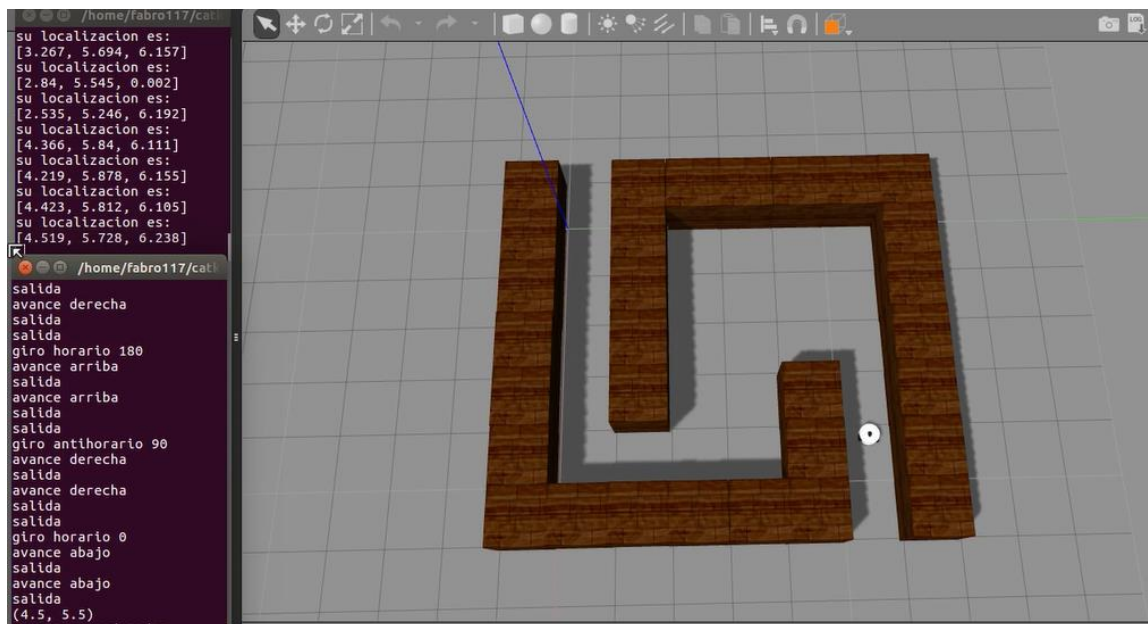


Nota: La figura muestra los nodos de simulación en Gazebo para el filtro de partículas.

La simulación de la plataforma móvil, se ve en la Figura 63 donde se obtiene la localización estimada de la plataforma diferencial implementando el filtro de partículas.

Figura 63

Simulación mediante Gazebo con el algoritmo Filtro de Partículas



Nota: La figura se observa a lado izquierdo las coordenadas resultantes de la aplicación del algoritmo del filtro de partículas y en el lado derecho la simulación en su mundo virtual.

Obtenido la localización estimada simulada en la Tabla 34 se compara la posición resultante del algoritmo de localización con la medida requerida o punto de llegada.

Tabla 34

Error de localización simulado utilizando Filtro de Partículas

Posición	Simulado (m)	Deseado (m)	Error
x	4,52	4,50	0,4%
y	5,73	5,50	4,2%
orientación	6,24	6,28	0,7%

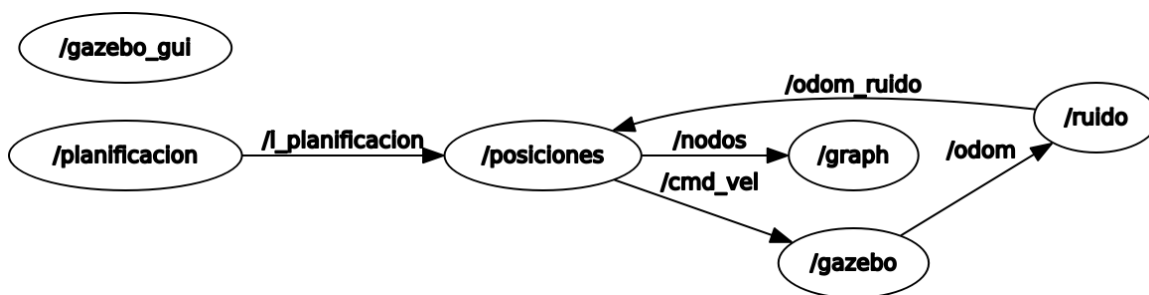
Nota: La tabla muestra el error relativo de la localización del punto de llegada vs la localización estimada por el filtro de partículas.

### **Simulación Técnicas de optimización gráfica Graph-SLAM**

La técnica de optimización gráfica Graph-SLAM, es una técnica donde se publica las poses del robot tomados por la odometría cada cierto trayecto. En la taxonomía del SLAM vista en el Capítulo 2, muestra que es una técnica OFF Line, debido a que recupera los datos de la odometría de forma métrica, con poca incertidumbre. En la Figura 64, se muestra los nodos de cómo realiza el movimiento la plataforma móvil, retroalimentada con la odometría con ruido, guardando los datos de la odometría con ruido para luego utilizar la técnica de optimización.

**Figura 64**

*Nodos ROS Técnicas de optimización gráfica Graph-SLAM*



Nota: La figura muestra los nodos de simulación en gazebo para la técnica de optimización gráfica Graph-SLAM

Recuperamos los datos de la odometría y lo analizamos en el código para optimización gráfica Graph-SLAM, dando como resultado la Figura 65 donde indica las convergencias de las posiciones obtenidas.

## Figura 65

### Simulación Técnicas de optimización gráfica Graph-SLAM

```

fabro117@fabro117:~$ roslaunch eat_31 prueba graph.launch
... logging to /home/fabro117/.ros/log/84956d7a-b0c0-11ea-a3a4-645a04a722d9/roslaunch-fabro117-10119.log
Checking log directory for disk usage. This may take awhile.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://192.168.0.9:34175/

*SUMMARY
*****
PARAMETERS
* /rostdistro: kinetic
* /rosverison: 1.12.14

NODES
/
graph (eat_31/graph_ros.py)

ROS_MASTER_URI=http://192.168.0.9:11311

process[graph-1]: started with pid [10138]
posicion estimada
[[1.575, 0.593, 0.0], [2.595, 0.554, 0.0], [4.529, 0.538, 0.0], [4.487, 1.565, 1.61], [2.542, 1.366, 1.61], [2.599, -0.6360000000000006, 3.1],
[2.4130000000000003, -2.6939999999999999, 1.551], [2.575, -1.577, -0.02]]
[[1.575, 0.593, 0.0], [2.595, 0.554, 0.0], [4.529, 0.538, 0.0], [4.487, 1.565, 1.61], [2.542, 1.366, 1.61], [2.599, -0.6360000000000006, 3.1],
[2.4130000000000003, -2.6939999999999999, 1.551], [2.575, -1.577, -0.02]]
posicion estimada
[[1.575, 0.593, 0.0], [2.595, 0.554, 0.0], [4.529, 0.538, 0.0], [4.487, 1.565, 1.61], [4.7620000000000005, -0.371, 1.61], [6.76, -0.2350000000000054, 3.1], [7.032, 1.8130000000000006, 1.551], [5.918, 1.9969999999999999, -0.02]]
[[1.575, 0.593, 0.0], [2.595, 0.554, 0.0], [4.529, 0.538, 0.0], [4.487, 1.565, 1.61], [4.7620000000000005, -0.371, 1.61], [6.76, -0.2350000000000054, 3.1], [7.032, 1.8130000000000006, 1.551], [5.918, 1.9969999999999999, -0.02]]
[graph-1] process has finished cleanly

```

Nota: La figura muestra como convergen las posiciones mediante la técnica de optimización gráfica Graph-SLAM

## Construcción y funcionamiento de la plataforma móvil

Se procede con la implementación de los resultados obtenidos en el Capítulo 3, iniciando con la construcción de la parte física o hardware de la plataforma móvil.

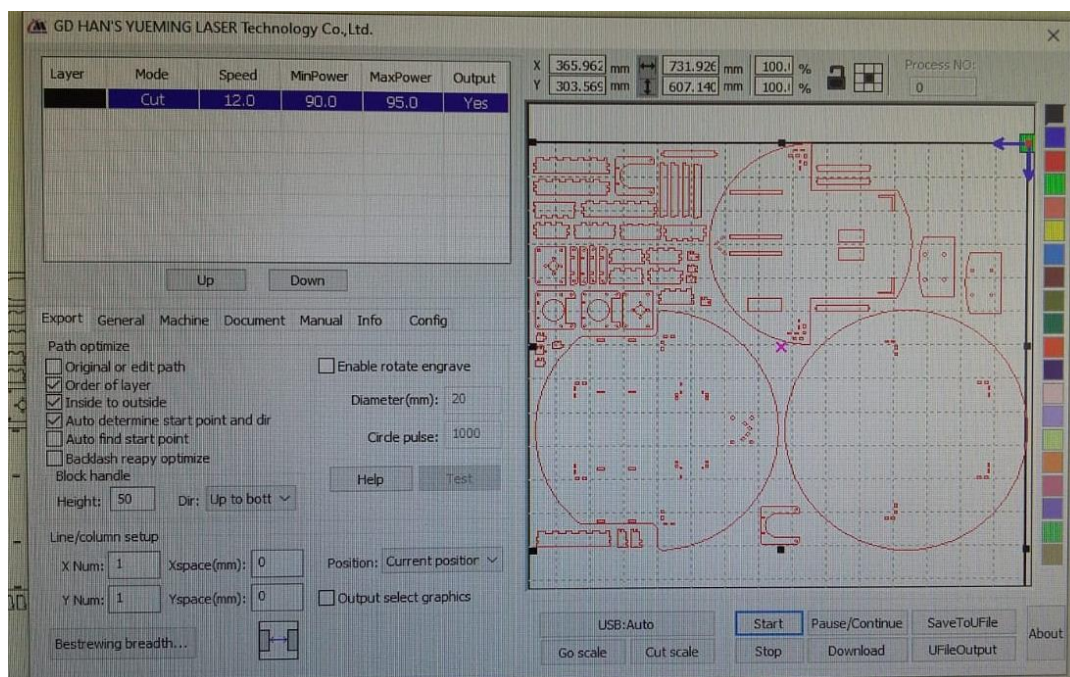
### ***Ensamblaje de la plataforma móvil***

La estructura o parte física de la plataforma móvil se la realiza de forma modular, debido a la facilidad de montaje y desmontaje de la estructura. La formación de la estructura empieza con cortar las diferentes piezas en acrílico de 3 mm, en una cortadora laser.

En la Figura 66 se observa los parámetros seteados para realizar el corte. Las distintas piezas a cortar se obtienen de la extracción de los distintos de los sólidos, loca cuales conforman la parte estructural de la plataforma móvil.

**Figura 66**

*Parámetros para realizar el corte laser*



Nota: La figura muestra el programa donde se introduce los parámetros de corte y las piezas a cortar.

Una vez realizado el corte laser se procede a ensamblar la parte estructural de la plataforma móvil iniciando con la base de la plataforma, los motores y los neumáticos, en la Figura 67 se observa las piezas de la estructura cortada.

Los motores se acoplarán a la plataforma móvil con 4 pernos Ansis Metric M4, recomendando por el fabricante, se muestra en la Figura 68; para la unión entre los neumáticos y los motores, se realiza con adaptadores para eje de salida de motores tipo D de 6 mm. En la Figura 69 a) vemos el neumático y el adaptador.

La rueda loca se coloca una rueda castor Pololu de 1 pulgada mostrado en la Figura 69 b). En la figura tal se observa como la rueda loca ayuda a equilibrar la plataforma móvil dando un punto de apoyo y nivelando su base como se observa en la Figura 69 c).



**Figura 67**

*Estructura en acrílico de la plataforma móvil*



Nota: La figura indica las piezas cortadas. Las cuales conformaran la estructura de la plataforma móvil

**Figura 68**

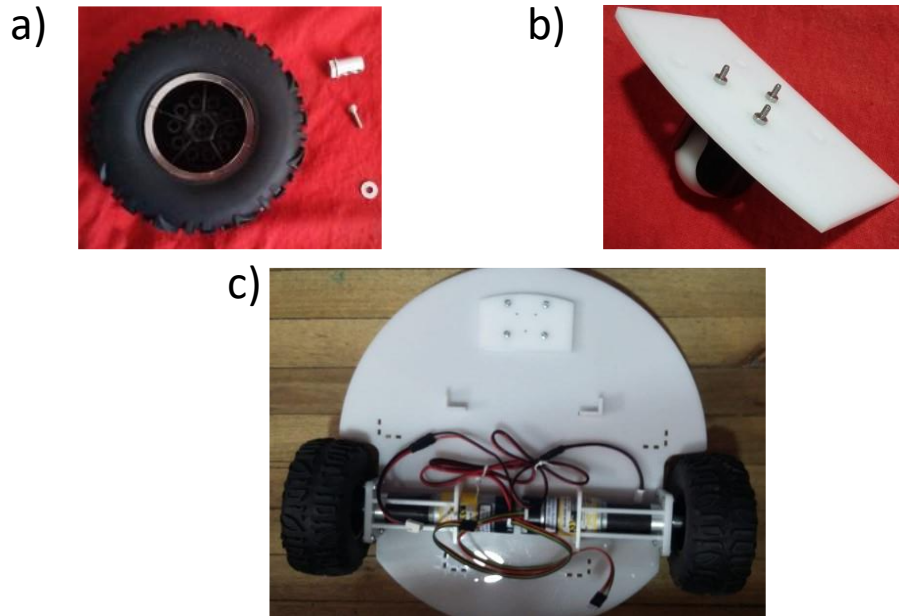
*Unión base- motores*



Nota: La figura indica como se une los motores con la base de la plataforma móvil.

## Figura 69

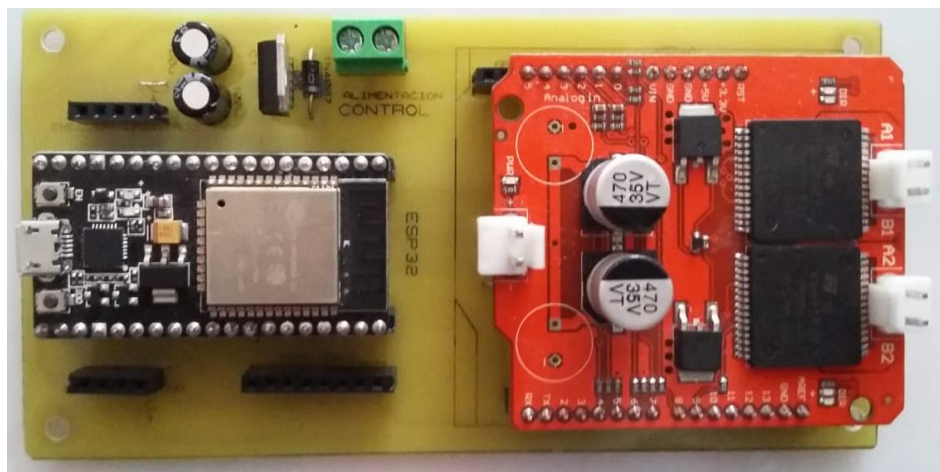
*Movimiento de la plataforma móvil*



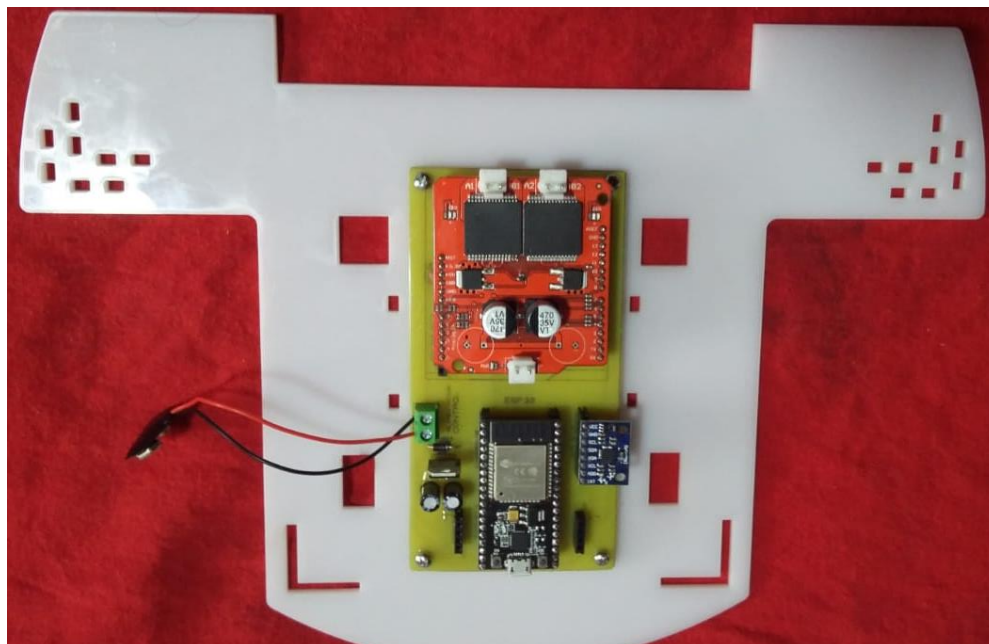
Nota: a) Neumático de la plataforma móvil con los adaptadores para unir al motor, b) Rueda loca pololu de 1", junto con el acople para unir a la base de la plataforma móvil, c) Unión de los distintos componentes con la base de la plataforma móvil.

Continuamos con el montaje de la repisa donde se pondrá la tarjeta PCB, en la cual se conectará los sensores como los encoders y la IMU, y se posiciona el driver Monster Moto Shield VNH2SP30. Todos los componentes mencionados se conectan a la placa de control ESP32, la cual envía la información (a través de su módulo Wifi integrado), al procesador (NUC). En la Figura 70 se indica la conexión de la placa PCB la cual evita problemas de cableado.

Además, para fijar la placa a la repisa de la plataforma diferencial se usa pernos Ansis Metric M3, como se muestra en la Figura 71.

**Figura 70***Placa PCB*

Nota: La figura indica la placa PCB de la plataforma móvil y la posición de los distintos componentes electrónicos

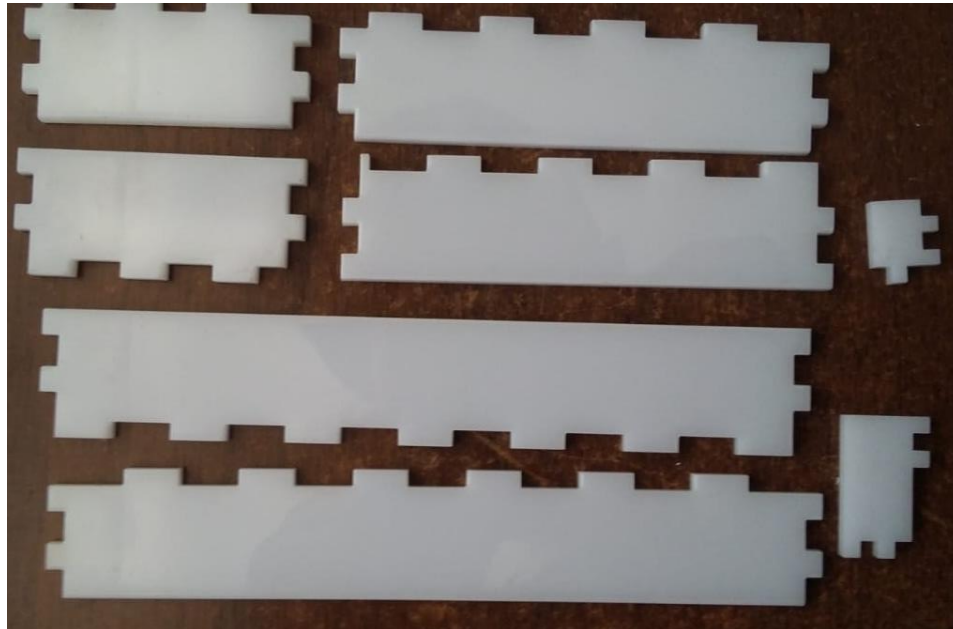
**Figura 71***Montaje de la placa PCB en la repisa*

Nota: La figura indica la placa PCB de la plataforma móvil y la posición de los distintos componentes electrónicos

Con respecto a la unión de la repisa y la parte superior de la plataforma se hace el uso de acoples cortados de igual forma en acrílico de 3mm, se pueden observar a continuación en la Figura 72. Así mismo dichos acoples ayudan a la fijación y posicionamiento de la alimentación (batería) dentro de la plataforma móvil.

### **Figura 72**

*Acoples para la estructura de la plataforma móvil*

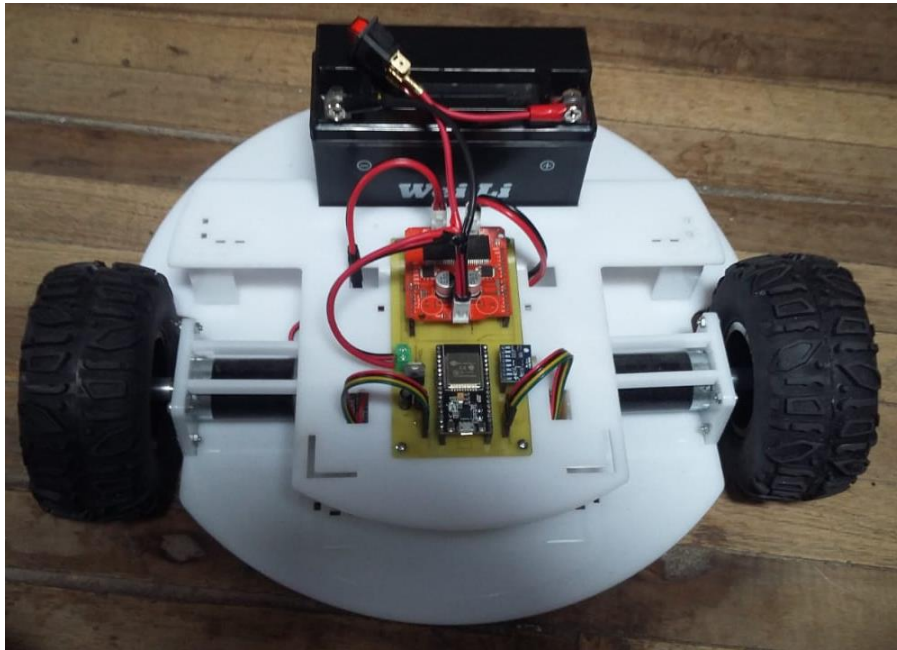


Nota: La figura se observa los acoples para la repisa parte superior de la plataforma y de la batería.

Con los acoples presentados con anteriormente, se garantiza un fácil ensamblaje y desmontaje de la plataforma móvil, obteniendo una plataforma modular. Para finalizar la parte de ensamblaje, observamos como se ve la plataforma móvil con el montaje de todos sus componentes excepto la parte superior de la plataforma en la Figura 73 y como se ve la plataforma móvil al final de su ensamblaje en la Figura 74.

**Figura 73**

*Parte interna de la plataforma móvil*



Nota: La figura se aprecia la ubicación de todos sus componentes a nivel hardware

**Figura 74**

*Ensamble final de la plataforma móvil*



Nota: Producto final del trabajo a nivel hardware

### **Software de la plataforma móvil**

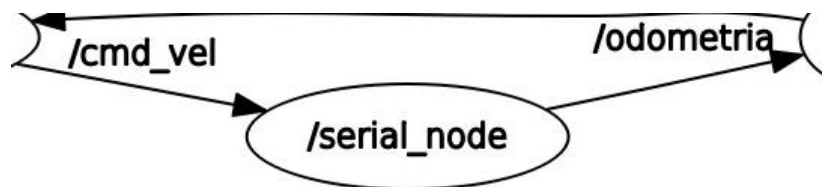
La programación de los algoritmos para la estimación de la localización es realizada en el lenguaje de programación Python, cada código de cada algoritmo para estimar la localización es detallado en el Anexo A.

De igual forma en el Anexo A podemos encontrar el código firmware utilizado en la ESP32; dicho componente es programado en Visual Studio Code, para ello se descarga su extensión PlatformIO, la cual ofrece una mayor facilidad de programar tarjetas de control como la ESP32.

En el firmware del ESP32 se suscribe al topico `/cmd_vel` mediante ROS, los datos ha receptor son la velocidad lineal y angular, y mediante la cinemática descrita en el Capítulo 2 y 3 se obtiene la velocidad de cada motor. El firmware se encarga de realiza de publicar en `/odometria`, datos obtenidos desde los encoders de cuadratura vistos en el Capítulo 2, donde se realiza la lectura de los flancos en subida, con las fórmulas detalladas para odometría en el Capítulo 2 convertimos la lectura de los flancos en distancia.

### **Figura 75**

*Nodos ROS – ESP32*



Nota: Denota los canales de comunicación entre el ESP32 y el procesador.

En el Anexo A se verifica como partiendo desde el dato `cmd_vel` como se observa en la Figura 75 ,es el setpoint del lazo de control. En el cual se controla la velocidad de la plataforma.

### Control de la plataforma móvil

Para la obtención de las funciones de transferencia de cada motor lo consideramos como una caja negra, tomando como entrada el Voltaje suministra el driver y su salida es la velocidad de cada motor. El software Matlab sirve como gran ayuda para la identificación de la planta con el método “arx”; se recalca, el tiempo de muestreo para la toma de datos es de 1 segundo.

En el motor derecho tiene una estimación de modelo de 93.94%, la función de transferencia correspondiente indica la siguiente ecuación:

$$Gz1 = \frac{0.0032 z^{-1}}{1 - 0.6744 z^{-1} + 0.352 z^{-2} - 0.0591 z^{-3}} \quad (86)$$

En el motor izquierdo tiene una estimación de modelo de 90.79%, la función de transferencia correspondiente indica la siguiente ecuación:

$$Gz1 = \frac{0.0007 z^{-1}}{1 - 0.3728 z^{-1} - 0.7774 z^{-2} + 0.301 z^{-3}} \quad (87)$$

Se tiene en cuenta que la toma de los datos se realiza con la plataforma móvil sin contacto sobre el suelo. Se realiza un control tipo PID con un código pre realizado en Matlab para la obtención de las constantes de control las cuales se detallan en la Tabla 35 y Tabla 36:

**Tabla 35**

*Constantes PID motor derecho.*

<b>Coefficientes</b>	<b>PID</b>
Ganancia proporcional (Kp)	1.18
Tiempo integral (ki)	0.2215
Tiempo derivativo (kd)	0.1

Nota: La tabla indica los parámetros del PID para el motor Derecho

**Tabla 36**

*Constantes PID motor izquierdo.*

<b>Coeficientes</b>	<b>PID</b>
Ganancia proporcional (Kp)	1.19
Tiempo integral (ki)	0.2635
Tiempo derivativo (kd)	0.1

Nota: La tabla indica los parámetros del PID para el motor izquierdo.

### ***Interfaz Humano Maquina.***

El interfaz humano maquina (HMI) facilita la interacción con la plataforma móvil, para acceder a la HMI en el terminal de Ubuntu tecleamos `rosrun hmi_tesis main.py`, antes de ello se verifica que se haya iniciado ROS y encendido la plataforma y su conexión con el procesador. Presentando en primera instancia la ventana de inicio o principal indicada en la Figura 76.

**Figura 76**

*Ventana principal*



Nota: Como se observa en la ventana principal podemos comunicarnos con la plataforma mediante teleoperación o planificación.



Al seleccionar el botón teleoperación se ingresa a la ventana observada en la Figura 77, donde, en su lado izquierdo se escoge la técnica de estimación de localización a realizar para poder teleoperar. Como se indica en el apartado de simulación, la técnica de optimización grafica funcionara luego de recolectar datos de la odometría.

**Figura 77**

*Ventana Teleoperación*



Nota: En la parte media inferior se observará posición estimada y la velocidad.

En el lado derecho de la ventana teleoperación podemos setear su velocidad dentro de 20 rangos, su función se describe a continuación:

- Si la velocidad es lineal se presiona arriba o abajo la velocidad seteada se dividirá por 100 y se obtendrá la velocidad lineal
- Si la velocidad es angular se elige derecha o izquierda, la plataforma girará sobre su eje con una velocidad angular de la velocidad seteada dividida entre 10

Al presionar el botón de STOP se detiene el programa de estimación de posición y se gráfica las posiciones donde estuvo la plataforma móvil.

La otra opción entregada por la ventana principal es la ventana de planificación, observada en la Figura 78. En el lado izquierdo de la ventana se tiene las mismas opciones vistas en la ventana de teleoperación y en el lado derecho 3 botones:

- Planificar: el cual activa el planificador de ruta predeterminado
- Movimiento: inicia el movimiento de la plataforma, el botón movimiento se activa solo si se pulso antes el botón planificar y se desactiva con el botón stop
- Stop: una vez realizada la trayectoria grafica de las estimaciones de posiciones de la plataforma móvil.

**Figura 78**

*Ventana Planificación*



Nota: El punto de llegada del robot es predeterminada en el código del planificador de ruta A\*

## Pruebas y Resultados

Para la realización las distintas pruebas en la plataforma móvil, el nodo que genera el ruido del sensor en la parte de simulación desaparece, debido que en el mundo físico los sensores entregan los datos medidos con ruido.

Se inicia la primera prueba, donde la plataforma móvil avanzara una trayectoria en línea recta, como muestra la Figura 79, de 1.5 m y repitiendo la prueba 5 veces. Cabe recalcar la conexión de envío y recepción de datos es de 1 segundo, debido a la velocidad del internet.

### Figura 79

*Prueba en línea recta*



Nota: La plataforma móvil deberá llegar o pasar los 1.5 m y detenerse de inmediato.

En la primera prueba tomamos los datos dado por la odometría de la plataforma móvil, la medición en el mundo real se toma como referencia la línea azul y es medido con una regla métrica. Los resultados obtenidos en la prueba para cada estimación de posición

**Tabla 37***Resultados Odometría-Real.*

N°	X_Odom (m)	Y_Odom (m)	X_Real (m)	Y_Real (m)	Error_X	Error_Y
1	1,653	0,021	1,68	0,02	1,6%	4,8%
2	1,614	0,042	1,64	0,04	1,6%	16,7%
3	1,534	0,012	1,545	0,015	0,7%	25,0%
4	1,562	0,023	1,58	0,02	1,2%	13,0%
5	1,604	0,017	1,62	0,015	1,0%	11,8%

Nota: La tabla indica la posición final de la plataforma en cada prueba.

**Tabla 38***Resultados Filtro de Kalman Extendido -Real.*

N°	X_EKF (m)	Y_EKF(m)	X_Real (m)	Y_Real (m)	Error_X	Error_Y
1	1,69	0,025	1,68	0,02	0,6%	20,0%
2	1,68	0,047	1,66	0,04	1,2%	41,2%
3	1,55	0,013	1,545	0,015	0,3%	15,4%
4	1,59	0,021	1,58	0,02	0,6%	4,8%
5	1,64	0,016	1,62	0,015	1,2%	6,3%

Nota: La tabla indica posición estimada usando EKF en cada prueba.

**Tabla 39***Resultados Filtro de Partículas -Real.*

N°	X_FP (m)	Y_FP(m)	X_Real (m)	Y_Real (m)	Error_X	Error_Y
1	1,746	0,076	1,68	0,02	3,8%	73,7%
2	1,832	0,043	1,64	0,04	10,5%	23,1%
3	1,552	0,23	1,545	0,015	0,5%	93,5%
4	1,574	0,042	1,58	0,02	0,4%	52,4%
5	1,632	0,107	1,62	0,015	0,7%	86,0%

Nota: La tabla indica posición estimada usando filtro de partícula en cada prueba.

La primera prueba no se realiza para las técnicas de optimización gráfica. Debido para el funcionamiento de la técnica se necesita la medición entre dos posiciones y en la prueba uno manejamos la posición de llegada.

El mayor error en X de 10.5% y Y en 93.5% da el filtro de partículas, debido a su limitante de creación de partículas y el tamaño del mundo, donde al crear N partículas

evalúa cual de todas ellas se ubica más cerca de la posición estimada de la plataforma móvil.

La segunda prueba a realiza es el seguimiento de una trayectoria cuadrada de 1X1 m, vista en la Figura 80 , donde se realiza una prueba por cada algoritmo de estimación de posición. Con el objetivo de concluir el cuadrado; la toma de datos para graficar será cada metro.

### Figura 80

*Prueba - Cuadrado 1x1*



Nota: La plataforma móvil debe acabar el cuadrado.

Para realizar la segunda prueba se toma en cuenta las siguientes consideraciones:

- Los filtros EKF y de partículas recibirán el nodo /odometría y cada filtro correspondiente enviará la estimación de posición al programa en el cual la plataforma se moverá en forma cuadrada
- Para obtener un resultado deseable en la Técnica de Optimización Gráfica – Graph SLAM, se envía cada nodo a converger, en otras palabras,

enviamos dos puntos para obtener la convergencia del segundo punto; con este segundo punto que convergió se envía el tercer punto y así sucesivamente.

Al querer realizar el cuadrado con la plataforma móvil, se debe tomar en cuenta el tiempo de transmisión de datos, y en el caso del filtro de partículas se le realizó un cambio en el mundo o la zona de trabajo, reduciendo a un tamaño de 1.5X1.5.

**Tabla 40**

*Resultados Filtro de Kalman Extendido – Posición deseada.*

<b>Puntos:</b>	<b>P1</b>	<b>P2</b>	<b>P3</b>	<b>P4</b>
X (m)	1,065	1,021	-0,033	-0,2
Y (m)	-0,033	1,093	0,92	-0,091
Angulo (rad)	-0,022	1,6	3,18	-1,64
X_deseado (m)	1	1	0	0
Y_deseado (m)	0	1	1	0
Angulo_desea (rad)	0	1,57	3,14	-1,57
Error_X	6,1%	2,1%	3,3%	20,0%
Error_Y	3,3%	8,5%	8,7%	9,1%
Error_Angulo	2,2%	1,9%	1,3%	4,3%

**Tabla 41**

*Resultados Filtro de Partículas – Posición deseada.*

<b>Puntos:</b>	<b>P1</b>	<b>P2</b>	<b>P3</b>	<b>P4</b>
X (m)	1,074	1,107	0,012	-0,05
Y (m)	-0,015	1,06	1,01	-0,071
Angulo (rad)	-0,304	2,05	3,2	-1,93
X_deseado (m)	1	1	0	0
Y_deseado (m)	0	1	1	0
Angulo_desea (rad)	0	1,57	3,14	-1,57
Error_X	6,9%	9,7%	1,2%	5,0%
Error_Y	1,5%	5,7%	1,0%	7,1%
Error_Angulo	30,4%	23,4%	1,9%	18,7%

**Tabla 42**

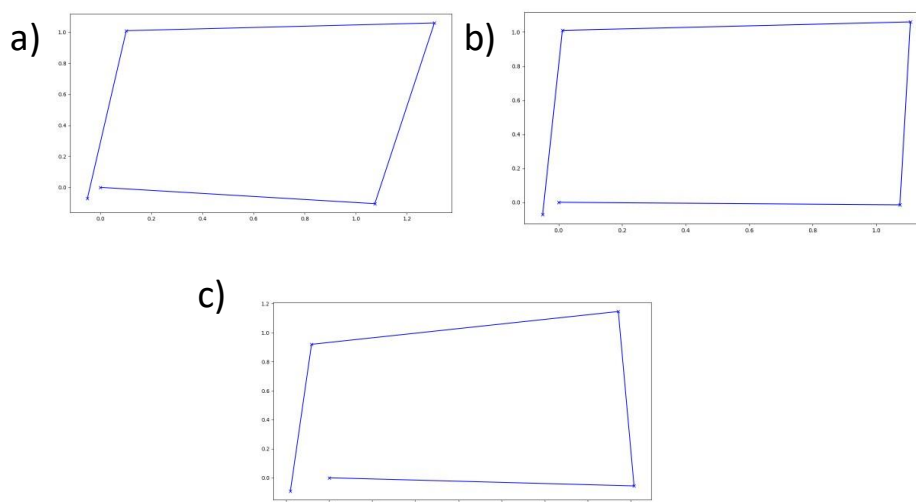
*Resultados Técnica de Optimización Grafica Graph SLAM - Posición deseada.*

<b>Puntos:</b>	<b>P1</b>	<b>P2</b>	<b>P3</b>	<b>P4</b>
X (m)	1,415	1,324	-0,082	-0,18
Y (m)	-0,056	1,146	0,904	-0,098
Angulo (rad)	-0,091	1,85	3,32	-1,517
X_deseado (m)	1	1	0	0
Y_deseado (m)	0	1	1	0
Angulo_desea (rad)	0	1,57	3,14	-1,57
Error_X	29,3%	24,5%	8,2%	18,0%
Error_Y	5,6%	12,7%	10,6%	9,8%
Error_Angulo	9,1%	15,1%	5,4%	3,5%

Se observa en las tablas un error mayor en las Técnicas de Optimización Gráfica con 24.5% de error en x y y con 12.6% una reducción de errores en el filtro de partículas comparado con la prueba anterior donde su realizo con un área de trabajo mayor. En la Figura 81, nos muestra como la plataforma siguió la trayectoria cuadrada.

**Figura 81**

*Resultado de la segunda prueba*



Nota: a) Prueba con Filtro de Kalman Extendido b) Prueba con filtro de partícula c) Prueba con Técnica de optimización grafica.

La tercera prueba que deberá realizar la plataforma móvil es llegar al punto deseado o lo más próximo posible, para ello se emplea un planificador de trayectoria A\*, y comprobamos el cumplimiento de la trayectoria como en el caso de la simulación. En la Figura 82, indica la trayectoria que debe cumplir la plataforma móvil.

### Figura 82

*Tercera prueba- Trayectoria de trabajo.*

a)



b)



Nota: a) Punto de partida de la plataforma. b) Prueba de llegada de la plataforma móvil.

El área de trabajo donde se prueba el planificador de trayectoria es muy reducida de aproximadamente 3X3.8m, por dicho motivo a diferencia de la simulación la plataforma móvil recorrerá menos trayectoria y se deberá modificar el grid y la función Heurística, como se observa en la Figura 83, se para que la plataforma móvil llegue al punto deseado.

### Figura 83

*Modificación código planeación.*

```
grid = [[0, 1, 1, 0],
        [0, 1, 0, 0],
        [0, 0, 0, 0]]
```

```
heuristic = [[5, 4, 3, 2],
             [4, 3, 2, 1],
             [3, 2, 1, 0]]
```

Nota: La imagen denota la parte del código a modificar



La tercera prueba se toma los resultados de la plataforma móvil al llegar al punto deseado de cada algoritmo para estimar su posición. Se tiene en cuenta que mediante código se realiza que el punto de llegada sea en  $X=0.7$  y  $Y=3.4$ , esto debido al tamaño del área de trabajo al tener.

**Tabla 43**

*Resultados Filtro de Kalman Extendido – Trayectoria*

	<b>X</b>	<b>Y</b>	<b>Angulo</b>
Pf_deseado	0,7	3,4	3,14
Pf_P1	0,654	3,56	3,101
Pf_P2	0,69	3,48	3,19
Pf_P3	0,673	3,6	3,21
Error_P1	7,0%	4,5%	1,3%
Error_P2	1,4%	2,3%	1,6%
Error_P3	4,0%	5,6%	2,2%

**Tabla 44**

*Resultados Filtro de Partículas – Trayectoria*

	<b>X</b>	<b>Y</b>	<b>Angulo</b>
Pf_deseado	0,7	3,4	3,14
Pf_P1	0,723	3,96	3,239
Pf_P2	0,685	3,232	3,08
Pf_P3	0,645	3,173	3,56
Error_P1	3,2%	14,1%	3,1%
Error_P2	2,2%	5,2%	1,9%
Error_P3	8,5%	7,2%	11,8%

**Tabla 45**

*Resultados Técnica de Optimización Gráfica Graph SLAM - Trayectoria.*

	<b>X</b>	<b>Y</b>	<b>Angulo</b>
Pf_deseado	0,7	3,4	3,14
Pf_P1	0,751	3,525	3,22
Pf_P2	0,891	3,025	3,19
Pf_P3	0,609	3,35	3,34
Error_P1	6,8%	3,5%	2,5%
Error_P2	21,4%	12,4%	1,6%
Error_P3	14,9%	1,5%	6,0%

El código movimiento lee la ruta planificada, y genera la secuencia de movimientos que debe realizar la plataforma para alcanzar su objetivo. La técnica de optimización gráfica aún conserva el mayor error entre los algoritmos para estimar la localización conservando el ángulo de la odometría y evaluando solo los puntos (x, y).

### **Resumen**

La construcción de la plataforma móvil es la implementación de los resultados finales de los subsistemas 1, 2, 3 y 4 detallados en el Capítulo 3. La estructura de la plataforma móvil es realizada en acrílico de 3 mm, para realizar los cortes del acrílico se usa la cortadora laser del Laboratorio de Mecatrónica y Sistemas Dinámicos; las medidas de cada pieza que forma parte de la estructura son obtenida de los archivos dibujados en un sistema CAD 3D.

En la implementación de la parte electrónica del sistema se realiza una placa PCB, donde se conectan el driver, el encoder de cada motor, la MPU 6050 y la tarjeta de control ESP 32. La placa PCB facilita la comunicación de los distintos componentes sin la necesidad de cables, los cuales pueden estar dañados y dar un problema al momento de implementar la parte de software de la plataforma móvil.

Para la unión entre la estructura la PCB y la fuente de alimentación de la plataforma móvil se utiliza acoples de acrílico y tornillos Ansis Metric M3. Dando como resultado la construcción del hardware de la plataforma móvil.

Al momento de iniciar la implementación de los subsistemas 5 y 6 se facilitó por parte del tutor una NUC, la cual sirve como procesador central de la plataforma móvil. En la NUC se desarrolla los códigos de los algoritmos para la estimación de la localización, el planificador de ruta A\* y el firmware de la ESP32.

En el procesador central se ejecuta ROS y realiza la conexión mediante Wifi entra la ESP32 y los distintos códigos desarrollados en Python para realizar las pruebas de los algoritmos de estimación de localización.

Una vez concluida la construcción del hardware y la programación del software de la plataforma móvil se inicia con las pruebas.

La primera prueba en realiza es un avance en línea recta donde vamos a probar la estimación de la posición dada por la plataforma en el mundo real con el dato de odometría y los filtros.

La segunda prueba es realizar un cuadrado de 1X1 m, donde se analiza como ayuda cada algoritmo para estimar la posición para ello el filtro se subscribe a la odometría dada por la plataforma y da como resultado la estimación de la localización. Con la estimación de la localización se procede a completar el cuadrado

La última prueba se realiza junto al planificador de trayectoria, donde el planificador indica el punto de llegada dentro del área de trabajo. Cada algoritmo para estimar la localización da como resultado la posición estimada en la que se encuentra la plataforma móvil en el área de trabajo.

## Capítulo V

### Conclusiones, Recomendaciones y Trabajos Futuros

#### Conclusiones y Recomendaciones

Para la elaboración de la plataforma se tomó en cuenta sus componentes principales, en este caso la batería de la cual se alimenta, razón por la cual la altura tomada desde el suelo hasta su parte superior es de 184mm, con los cuales se facilita el circuito de control y a su vez la potencia. En la obtención del diámetro de 360mm de la plataforma móvil se da por la longitud de los motores, para tener un posicionamiento lo más recto posible entre los dos motores. Con el modelamiento de la dinámica inversa de la plataforma se obtuvo que los motores necesitan un torque máximo de 1.32 Nm.

El material considerado para la elaboración de la estructura fue acrílico de 3mm, el mismo que tiene una resistencia moderada, dando la capacidad de desplazamiento de manera adecuada sin incrementar en gran parte el peso del resto de componentes, los análisis CAE nos demuestra que la deflexión máxima a soportar de 0.7851 mm, la cual consideramos que es una deflexión admisible para el tipo de material. La deflexión obtenida es realizada en un caso extremo, donde la plataforma soportara la NUC.

El filtro extendido de Kalman tiene su fundamento en la curva gaussiana, por lo cual se ha realizado una adaptación de los algoritmos de ubicación principalmente, ya que este modelo es utilizado para poder transformar un sistema no lineal en un sistema lineal para alcanzar una estimación más real.

La precisión de un filtro de partículas, dependerá del número de las partículas generadas, para la realización de las pruebas se creó 2500 partículas, por ello al tener mayor cantidad de partículas se obtendrá un menor error ya que estas partículas estarán distribuidas por todo mapa o grid y la cantidad que pueda generar ayuda a mejorar la exactitud.

La técnica de optimización gráfica graph SLAM es un método OFF line y estático, depende de la recolección de datos obtenidos por la odometría y son independiente de los puntos de referencia, obteniendo como resultado puntos de referencias (x, y). Los datos recolectados se realizan cada cierta distancia predeterminada en nuestro caso de 1 metro, donde crea nodos con el menor error.

En las pruebas se avance en línea recta el filtro de Kalman Extendido mostro un mejor resultado con un error de en  $X= 0.6\%$  y  $Y=4.8\%$  con respecto a las medidas tomadas en el mundo físico, mejorando las tomas de la odometría que presentaron errores de  $X = 1\%$  y  $Y=4.8\%$ , el filtro de partículas obtuvo el mayor error en esta prueba con  $X=3.8\%$  y  $Y=23.1\%$ ; debido a que su entorno, mundo o área de trabajo era de  $5X6$  m. No se realizó prueba con el algoritmo de técnicas de optimización grafica debido a que se necesita dos puntos y solo trabajamos con el punto de llegada en línea recta.

En la segunda prueba efectuada donde la odometría de cada filtro señalaba el movimiento de la plataforma móvil, se rebajó el área de trabajo del filtro de partículas a  $1.5 X 1.5$  m, dando un menor error con  $X=1.2\%$  y  $Y=1.0\%$ . El filtro de Kalman Extendido tuvo también un error aceptable de  $X=3.3\%$  y  $Y=3.3\%$ ; en tanto luego de recopilar datos la técnica de optimización gráfica presento el mayor error con  $X= 8.2\%$  y  $Y$  con  $5.6\%$ .

La última prueba realiza la planificación de ruta, donde la planeación de trayectoria A\* dice la ruta por donde ir la plataforma y el punto a donde llegar. Se aumento el área de trabajo para el filtro de partículas a  $3 X 4$  m, debido que aún se trabaja con las 2500 partículas el error incremento dando un  $X=2.2\%$  y  $Y= 5.2\%$ , en cambio el filtro de Kalman Extendido mantiene su error con un  $X=1.4\%$  y  $Y=2.3\%$ . En tanto la técnica de optimización grafica presento errores de  $X=6.8\%$  y  $Y=1.5\%$ .

Al tener una longitud de mensaje más grande que el buffer dado por la librería de ros\_lib, da un error de conexión, provocando desconexión entre la ESP32 y la NUC, se

solventa el problema realizamos el siguiente cambio en el archivo `node_handel.h` ubicado en: [Bibliotecas Arduino] ---> `ros_lib` ---> `ros` ---> `node_handle.h`

Si existe fallo en la conexión Wifi o la conexión con el procesador NUC, la plataforma móvil pierde el control en un lapso de 2 a 5 segundos pudiendo ocasionar algún accidente. Por dicho motivo se recomienda en plataformas móviles internas tener bajas velocidades; un ejemplo son las aspiradoras robots, según (Vorwerk International para España, 2018), el modelo Kobold Robot Aspirador VR100 posee una velocidad media de 0.20 m/s, la misma velocidad máxima que puede alcanzar la plataforma móvil

En los códigos de movimiento dados en el Anexo A tanto, se publica la velocidad angular de la plataforma en `avance.linear` y en vez de `avance.angular.z`, el motivo del cambio de canal es debido a un fallo de envío de datos en `avance.angular.z`. Se recuerda que ROS nos facilita con canales de comunicación de diferentes tipos, facilitando realizar dicho cambio siempre y cuando se tenga claro que se desee realizar con el dato.

### **Trabajos Futuros.**

Si existe la necesidad de tener una plataforma móvil para exteriores se deberá aumentar un modelo de amortiguamiento para la locomoción de la plataforma y utilizar un GPS para tener una localización global implementando los mismos algoritmos de estimación de localización.

La plataforma móvil no realiza mapeo ni una percepción del entorno solo es guiada mediante sensores de odometría, para tener percepción del entorno y poder mapear con la plataforma móvil se necesita sensores como Kinect o un Lidar, obteniendo como resultado un SLAM completo no solo desde el punto de la localización.

La trayectoria de planificación A\* es muy rígida, se puede suavizar utilizando algoritmos como Smoth, el cual da una ruta de llegada al punto deseado más suave.

Con los algoritmos de estimación y localización se puede realizar MultiSLAM, según (Velásquez Hernández, 2017) es interacción que se produce entre 2 o más robots

con la finalidad de obtener un mapa global de un entorno explorado, por medio de las reconstrucciones parciales que reportan los mismos.

De igual forma otra aplicación resultante de la estimación de localización en mapas semántico debido, a que el robot permite manejar la incertidumbre para poder interactuar con los objetos que perciben, según (Ruiz Sarmiento, 2016), los mapas semánticos ayudan al robot comprender su entorno de trabajo e interactuar con los objetos que forman parte de dichos mapas (representaciones del mundo).

## Bibliografía

- Álvarez Gutiérrez, E. L., & Jiménez López, F. R. (26 de Junio de 2019). *Scielo*. Obtenido de GENERACIÓN DE MAPA GLOBAL 2D Y SLAM USANDO LIDAR Y UNA ESTÉREO CÁMARA PARA EL SEGUIMIENTO DE MOVIMIENTO DE UN ROBOT MÓVIL:  
[http://www.scielo.org.co/scielo.php?script=sci\\_arttext&pid=S1692-17982019000200144&lang=es](http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S1692-17982019000200144&lang=es)
- Andrade, F., & Llofriu, M. (s.f.). SLAM Estado del Arte. Montevideo, Uruguay.
- Arduino Mega 2560 Características, Especificaciones*. (23 de Agosto de 2018).  
 Obtenido de Proyecto Arduino: <https://proyectoarduino.com/arduino-mega-2560/>
- Armigol Moreno, J. M. (1997). LOCALIZACIÓN GEOMÉTRICA DE ROBOTS MÓVILES AUTÓNOMOS. Leganés.
- Barrientos Sotelo, V. R., García Sanchez, J. R., & Silva Ortigoza, R. (2007). Robots Móviles: Evolución y Estado del Arte. *polibits*, 12-17.
- Baturone, A. O. (2005). *Robótica: Manipuladores y Robots Móviles*. Barcelona: marcombo ediciones técnicas.
- Borrero Cruz, E. (2016). MAPEO Y LOCALIZACIÓN BASADA EN ODOMETRÍA USANDO IMÁGENES COMO LAND MARKS. Bogota.
- Carrera, A. (2018). Diseño y construcción de un robot móvil para SLAM con recuperación de ubicación en el caso de Rapto.
- da Silva, D. (2013). RobChair 2.0 : Simultaneous Localization and Mapping and Hardware/Software Frameworks.
- Fernández, M. Á., Fernández, D., & Valmaseda, C. (2010). *PLANIFICACION DE TRAYECTORIAS PARA UN ROBOT MOVIL*. Obtenido de <https://eprints.ucm.es/id/eprint/11301/1/MemoriaProyectoSSII.pdf>



- Forcada, R. (2013). Implementación de algoritmo SLAM Basado en sensor Láser Hokuyo 04LX-UG01. Valencia.
- Gallardo, J., & Gaona, A. D. (2015). *Diseño y construcción de una plataforma robótica móvil para interiores capaz de realizar SLAM*. Sangolquí: Universidad de las Fuerzas Armadas- ESPE.
- García, A. M. (2019). Realidad Aumentada mediante SLAM en tiempo real con cámara monocular. Sevilla.
- Garzón, P., & Obando, C. (2016). *DESARROLLO DE UN SISTEMA DE UBICACIÓN EN LA PLATAFORMA ROBÓTICA PARA EXTERIORES TELEOPERADA DEL LABORATORIO DE MANUFACTURA*. Sangolquí.
- GitHub*. (s.f.). Obtenido de <https://github.com/>
- Gómez , D. A. (4 de Septiembre de 2020). Diseño e implementación de una celda colaborativa robotizada mediante robots móviles y humanoides para clasificación de objetos.
- González, A. G. (02 de 12 de 2016). *Panama Hitek*. Obtenido de <http://panamahitek.com/que-es-y-como-funciona-un-servomotor/>
- Granda, D., & Vásconez, D. (2012). *MAPAS DE ENTORNOS MEDIANTE NAVEGACIÓN DIFUSA Y SISTEMA DE TELEOPERACIÓN DE UNA PLATAFORMA PIONEER P3-DX*. Sangolquí.
- Grisetti , G., Kümmerle, R., Stachniss, C., & Burgard, W. (Diciembre de 2010). A tutorial on graph-based SLAM. *IEEE Intelligent Transportation Systems Magazine*.
- Grisetti, G., Kümmerle, R., Stachniss, C., & Burgard, W. (2010). A Tutorial Graph-Based SLAM. *IEEE*.
- Ingeniería Mecafenix*. (20 de Abril de 2017). Obtenido de <https://www.ingmecafenix.com/electricidad-industrial/motor-paso-a-paso/>

- Jiménez Través, M. (2015). Algoritmos de localización de robots móviles empleando filtros estadísticos. Sevilla.
- KLANCAR, G., ZDEŠAR, A., BLAŽIC, S., & ŠKRJANC, I. (2016). *Wheeled Mobile Robotics*. Slovenia: Butterworth-Heinemann.
- Künzel, G. (14 de 9 de 2005). *THE MECHATRONICS DESIGN PROCESS*. Mercado Libre Ecuador. (2021). Obtenido de [https://articulo.mercadolibre.com.ec/MEC-431362118-mgsystem-monster-moto-shield-puente-h-30a-vnh2sp30-arduino-\\_JM#redirectedFromParent](https://articulo.mercadolibre.com.ec/MEC-431362118-mgsystem-monster-moto-shield-puente-h-30a-vnh2sp30-arduino-_JM#redirectedFromParent)
- Morales, E., & Sucar, L. E. (s.f.). Introducción a la Robótica Móvil. *Inteligencia Artificial*.
- Moya, C. (2017). *SISTEMA ROBÓTICO MÓVIL AUTÓNOMO COOPERATIVO PARA MAPEO 2D DE OBSTÁCULOS Y NIVELES DE TEMPERATURA EN UN AMBIENTE CONTROLADO*. Sangolquí.
- Muñoz, V. F. (3 de Octubre de 1999). *Página personal de Victo Fdo. Muñoz Martínez*. Obtenido de Ingeniería de Sistemas y Automatica: <http://webpersonal.uma.es/~VFMM/PDF/cap2.pdf>
- Musña Toapanta, J., Zapata Chancusig, B., Oñate Cadena, L., & Campusano Nieto, G. (2015). Diseño y construcción de un robot móvil que permita la obtención de una nube de puntos del escaneo de habitaciones utilizando láser y webcams. *INGENIUS Revista de Ciencia y Tecnología*, 9.
- Ortiz , A. (2014). Navegación para robots móviles. Palma de Mallorca.
- Preciado, J. D. (2018). *PLANIFICACIÓN DE TRAYECTORIAS DE ROBOTS MÓVILES DE DIFERENTES ARQUITECTURAS EN ENTORNOS DINÁMICOS*. Obtenido de <https://repository.unimilitar.edu.co/bitstream/handle/10654/18006/PreciadoAguilarJuanDavid2018.pdf?sequence=2&isAllowed=y>

- Quinaluisa, E., & Toapanta, J. (2018). *IMPLEMENTACIÓN DE UN SISTEMA DE NAVEGACIÓN AUTÓNOMO BASADO EN SLAM Y NAVEGACIÓN REACTIVA*. Sangolquí.
- Ramírez, K. (2020). *Ph.D. Kryscia Ramírez*. Obtenido de Odometría:  
<http://www.kramirez.net/Robotica/Material/Presentaciones/Odometria.pdf>
- ROBOTIS e-Manual*. (2020). Obtenido de  
<https://emanual.robotis.com/docs/en/platform/turtlebot3/overview/>
- ROS wiki*. (s.f.). Obtenido de <http://wiki.ros.org/ROS/Tutorials>
- Ruiz Sarmiento, J. R. (2016). Probabilistic techniques in semantic mapping for mobile robotics. UMA Editorial.
- Sásig, E. (s.f.). Robótica con Arduino. Conectados online.
- SERVOCITY*. (2021). Obtenido de <https://www.servocity.com/5202-series-yellow-jacket-planetary-gear-motor-188-1-ratio-30-rpm-3-3-5v-encoder/>
- Siciliano, B., & Khatib, O. (2016). Springer Handbook of Robotics. En B. Siciliano, & O. Khatib, *Spirnger Handbook of Robotics* (págs. 11-36). Stanford: Springer.
- Siciliano, B., & Khatib, O. (2016). Springer Handbook of Robotics. En B. Siciliano, & O. Khatib, *Spirnger Handbook of Robotics* (págs. 1153-1176). Stanford: Springer.
- Sigma Electronic*. (2021). Obtenido de <https://www.sigmaelectronica.net/producto/esp-32/>
- Silva, D. d. (Septiembre de 2013). RobChair 2.0. Coimbra, Portugal.
- Tapia García, M. R., & López Hernández, J. M. (2017). ROBÓTICA MÓVIL. *JOVENES EN LA CIENCIA , REVISTA DE DIVULGACIÓN CIENTIFICA*, 2526-2530.
- Torres, P. (2016). Análisis de algoritmos para localización y mapeado simultáneo de objetos.
- Trujillo, V. (Septiembre de 2015). Entender los diferentes tipos de locomoción de los robots móviles (Locomoción con Ruedas). Zumpango de Ocampo.

Udacity. (2021). Obtenido de Artificial Intelligence for Robotics - Udacity:

<https://classroom.udacity.com/courses/cs373/lessons/48646841/concepts/486468390923>

Ulrich, K. T., & Eppinger, S. D. (2013). *Diseño y desarrollo de productos*. México: McGraw Hill.

Una Introducción a los Robots Móviles. (2008). IL Bambino.

Velásquez Hernández, C. A. (06 de Diciembre de 2017). *Desarrollo de algoritmo de mezclado de mapas por ocupación de celdas aplicado a la navegación y exploración colaborativa de entornos internos desconocidos*. Obtenido de <https://repositorio.unal.edu.co/handle/unal/62177?fbclid=IwAR3v4vRNCPyzjSnxqslPBixltvAVLG7p5HclqDgliy25jmQuvNiHvfm774>

Vorwerk International para España. (14 de Mayo de 2018). Obtenido de

<https://support.vorwerk.com/hc/es/articles/360000896605--Con-qu%C3%A9-velocidad-se-desplaza-el-robot-aspirador->

**Anexos**