



**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**

**TEMA: “IMPLEMENTACIÓN DE REGLAS DE COMPORTAMIENTO
SOCIAL EN UNA PLATAFORMA ROBÓTICA DE TELEPRESENCIA, A
TRAVÉS DEL RECONOCIMIENTO DE GESTOS”**

AUTOR: ANDRANGO GUALLICHICO, FRANCISCO JAVIER

DIRECTOR: ING. MEJÍA CAMPOVERDE, PAÚL HERNÁN

SANGOLQUÍ

2020



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

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

CERTIFICACIÓN

Certifico que el trabajo de titulación, **“IMPLEMENTACIÓN DE REGLAS DE COMPORTAMIENTO SOCIAL EN UNA PLATAFORMA ROBÓTICA DE TELEPRESENCIA, A TRAVÉS DEL RECONOCIMIENTO DE GESTOS”** fue realizado por el señor **ANDRANGO GUALLICHICO FRANCISCO JAVIER** el mismo que ha sido revisado en su totalidad, analizado por la herramienta de verificación de similitud de contenido; por lo tanto cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustente públicamente.

Sangolquí, 14 de Enero de 2020

ING. PAÚL HERNÁN MEJÍA CAMPOVERDE
DIRECTOR DEL TRABAJO DE TITULACIÓN
C. C.: 1714090386



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

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

AUTORÍA DE RESPONSABILIDAD

Yo, **ANDRANGO GUALLICHICO FRANCISCO JAVIER**, con cédula de identidad N° 2300094824, declaro que el contenido, ideas y criterios del trabajo de titulación: **“IMPLEMENTACIÓN DE REGLAS DE COMPORTAMIENTO SOCIAL EN UNA PLATAFORMA ROBÓTICA DE TELEPRESENCIA, A TRAVÉS DEL RECONOCIMIENTO DE GESTOS”** es de mi autoría y responsabilidad, cumpliendo con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Consecuentemente el contenido de la investigación mencionada es veraz.

Sangolquí, 14 de Enero de 2020

Javier Andrango
.....
ANDRANGO GUALLICHICO
FRANCISCO JAVIER
C.C.: 2300094824



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

AUTORIZACIÓN

Yo, **ANDRANGO GUALLICHICO FRANCISCO JAVIER**, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: **“IMPLEMENTACIÓN DE REGLAS DE COMPORTAMIENTO SOCIAL EN UNA PLATAFORMA ROBÓTICA DE TELEPRESENCIA, A TRAVÉS DEL RECONOCIMIENTO DE GESTOS”** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi responsabilidad.

Sangolquí, 14 de enero de 2020

Javier Andrango
.....
ANDRANGO GUALLICHICO
FRANCISCO JAVIER
C.C.:2300094824

DEDICATORIA

A mis padres, Luis y María.

AGRADECIMIENTO

A mis padres y hermanos, de los cuales siempre recibí su apoyo.

A mis compañeros, que de una u otra forma, se involucraron en este proyecto.

A mi Director de Tesis, Ing. Paúl Mejía, que gracias a su conocimiento y consejos me orientó en este trabajo.

ÍNDICE DE CONTENIDOS

CONTENIDO

CERTIFICACIÓN.....	i
AUDITORIA DE RESPONSABILIDAD	ii
AUTORIZACIÓN	iii
DEDICATORIA.....	iv
AGRADECIMIENTO.....	v
ÍNDICE DE CONTENIDOS	vi
ÍNDICE DE TABLAS	xi
ÍNDICE DE FIGURAS	xiii
RESUMEN	xvi
ABSTRACT	xvii

CAPÍTULO I GENERALIDADES

1.1. Introducción.....	1
1.2. Planteamiento del problema	1
1.3. Justificación e importancia	5
1.4. Alcance del proyecto.....	6
1.5. Objetivos.....	7

1.5.1.Objetivo General.....	7
1.5.2.Objetivos Específicos	7
1.6.Estructura del documento.....	8

CAPÍTULO II ESTADO DEL ARTE

2.1. Introducción	9
2.2. Introducción al uso de visión Artificial en la robótica.....	9
2.3. Machine learning aplicada a la visión robótica.	10
2.3.1. Técnicas de Machine Learning aplicadas a la visión artificial	11
2.4. Reglas de comportamiento social aplicadas en la robótica	14
2.4.1. Gestos corporales.....	15
2.4.2. Reconocimiento de gestos mediante visión artificial.....	16
2.5. Uso del sensor Kinect para reconocimiento de gestos	17
2.5.1. Aspectos técnicos del sensor Kinect.....	17
2.5.2. OpenNI y NITE.....	18
2.5.3. Proceso de reconocimiento de gestos	19
2.6. Robots móviles	20
2.7. Navegación en robots móviles	21
2.7.1. Dispositivo de percepción móvil LIDAR	22
2.7.2. IMU (Inertial Measurement Unit).....	23

2.7.3. Localización	23
2.7.4. Mapeo	24
2.7.5. Planeación de trayectoria	26

CAPÍTULO III DISEÑO MECATRÓNICO

3.1. Introducción	27
3.2. Diseño de subsistemas	28
3.3. Subsistema Mecánico	28
3.3.1. Componentes del subsistema mecánico	29
3.3.2. Modelo matemático de un robot móvil de configuración diferencial.....	30
3.3.3. Selección y dimensionamiento de actuadores	33
3.3.4. Diseño CAD	39
3.3.4.1. Diseño de la base	39
3.3.4.2. Diseño del sistema de tracción diferencial.....	41
3.3.4.3. Distribución de elementos en la plataforma robótica	41
3.4. Subsistema Electrónico	43
3.4.1. Componentes del subsistema electrónico	44
3.4.2. Control de actuadores y sensores	45
3.4.3. Autonomía Energética	47
3.4.4. Diagrama de conexiones	50

3.5. Subsistema de Control.....	50
3.5.1. Componentes del subsistema de control	51
3.5.1.1. Reconocimiento de las juntas del cuerpo humano.....	52
3.5.1.2. Navegación de la plataforma robótica.....	53
3.5.1.3. Navegación Autónoma.....	54
3.5.1.4. Modelo virtual de la plataforma robótica	55
3.5.1.5. Desarrollo de la interfaz gráfica de interacción humano-máquina	56

CAPÍTULO IV PRUEBAS Y RESULTADOS

4.1. Gestos Corporales	58
4.1.1. Seguimiento de esqueleto	59
4.1.2. Base de datos de gestos realizados	60
4.1.3. Red Neuronal Pre alimentada.....	61
4.1.4. Predicción de Gestos.....	61
4.2. SLAM.....	63
4.2.1. Elaboración de mapas	63
4.3. Navegación Autónoma.....	65

CAPÍTULO V CONCLUSIONES, RECOMENDACIONES Y TRABAJOS FUTUROS

5.1. Introducción	69
-------------------------	----

	x
5.2. Conclusiones	69
5.3. Recomendaciones	71
5.4. Trabajos futuros.....	71
BIBLIOGRAFÍA.....	73
ANEXOS.....	78

ÍNDICE DE TABLAS

Tabla 1. <i>Aplicaciones de Machine Learning y visión artificial en robots móviles.....</i>	11
Tabla 2. <i>Aplicaciones con técnicas de Machine Learning para visión artificial.....</i>	13
Tabla 3. <i>Especificación técnica sensor Kinect para Windows modelo 1517.....</i>	18
Tabla 4. <i>Identificación de requerimientos. Subsistema Mecánico.....</i>	28
Tabla 5. <i>Listado de métricas. Subsistema mecánico</i>	29
Tabla 6. <i>Cálculo de masa total de la plataforma robótica</i>	34
Tabla 7. <i>Consideraciones de diseño de la plataforma robótica.....</i>	35
Tabla 8. <i>Criterio de diseño para la selección de actuadores.....</i>	37
Tabla 9. <i>Conceptos para la selección de actuadores.....</i>	38
Tabla 10. <i>Matriz de decisión para la selección de actuadores.....</i>	38
Tabla 11. <i>Análisis estático. Base de la plataforma robótica</i>	40
Tabla 12. <i>Sistema de tracción diferencial</i>	41
Tabla 13. <i>Descripción de componentes plataforma robótica.....</i>	42
Tabla 14. <i>Identificación de requerimientos. Subsistema Electrónico</i>	43
Tabla 15. <i>Listado de métricas. Subsistema electrónico</i>	44
Tabla 16. <i>Criterio para la selección del controlador PC</i>	45
Tabla 17. <i>Conceptos para la selección del controlador PC.....</i>	45
Tabla 18. <i>Matriz de decisión para la selección del controlador PC.....</i>	46
Tabla 19. <i>Elementos del subsistema electrónico</i>	47
Tabla 20. <i>Consumo Energético de Potencia.....</i>	48
Tabla 21. <i>Consumo Energético de Control</i>	49
Tabla 22. <i>Elementos de consumo energético</i>	49

Tabla 23. <i>Identificación de requerimientos. Subsistema de Control</i>	51
Tabla 24. <i>Listado de métricas. Subsistema de Control</i>	51
Tabla 25. <i>Distancia de detección usuario-robot</i>	59
Tabla 26. <i>Base de datos de los gestos</i>	60
Tabla 27. <i>Matriz de confusión</i>	62
Tabla 28. <i>Desempeño de la Red Neuronal</i>	62
Tabla 29. <i>Configuración de parámetros para mapeo</i>	65
Tabla 30. <i>Pruebas de navegación</i>	67

ÍNDICE DE FIGURAS

Figura 1. Robot de asistencia para seguir personas.....	2
Figura 2. Seguimiento de persona objetivo.	3
Figura 3. Gestos ejecutados por el usuario	4
Figura 4. Robots móviles Universidad de las Fuerzas Armadas –ESPE	5
Figura 5. Robot social Pepper	10
Figura 6. Técnicas de Machine Learning	12
Figura 7. Comportamiento social de un robot de telepresencia.....	15
Figura 8. Gestos realizados con el cuerpo.....	16
Figura 9. Detección de caídas utilizando sensor Kinect	17
Figura 10. Articulaciones del esqueleto. NITE	19
Figura 11. Proceso de reconocimiento de gestos con Kinect	19
Figura 12. Clasificación de robots móviles con ruedas.....	21
Figura 13. RPLIDAR A1 esquema de trabajo	22
Figura 14. Descripción general de un sensor IMU	23
Figura 15. Esquema general de localización para robots móviles	24
Figura 16. Mapa de ocupación	25
Figura 17. Planeación de trayectorias	26
Figura 18. Norma VDI 2206 Modelo V para diseño de sistemas mecatrónicos.....	27
Figura 19. Proceso de desarrollo de productos	28
Figura 20. Componentes del subsistema mecánico	30
Figura 21. Sistema global de coordenadas de un robot móvil de configuración diferencial.	30

Figura 22. Diagrama detalle sistema de coordenadas robot diferencial.	31
Figura 23. Rotación del robot alrededor de I_{cc} en un tiempo δt	32
Figura 24. Diagrama de cuerpo libre de una rueda en plano inclinado.	35
Figura 25. Cálculo esfuerzos Von Mises	40
Figura 26. Cálculo de desplazamientos.	40
Figura 27. Acople de sistema de tracción diferencial.	41
Figura 28. Distribución de componentes plataforma robótica	42
Figura 29. Modelo final del prototipo robótico	43
Figura 30. Componentes del subsistema electrónico	44
Figura 31. Diagrama de bloques subsistema electrónico.	50
Figura 32. Componentes del subsistema de control.	52
Figura 33. Pose Psi para detección de juntas del cuerpo humano	52
Figura 34. Nodos del paquete differential_drive ROS.	54
Figura 35. Descripción del paquete ROS move_base	54
Figura 36. Modelo virtual de la plataforma robótica	55
Figura 37. Modelo virtual del robot.	56
Figura 38. Interfaz gráfica de interacción humano-máquina	57
Figura 39. Gestos realizados con el cuerpo.	58
Figura 40. Determinación del límite para detección de esqueleto	59
Figura 41. Base de datos de los gestos realizados	60
Figura 42. Total de aciertos y fallas en las observaciones	61
Figura 43. Laboratorio de Mecatrónica y Sistemas Dinámicos	63
Figura 44. Proceso de mapeo con Gmapping	64

Figura 45. Configuración parámetros Gmapping ROS	64
Figura 46. Circuito de calle cerrado, como medio de pruebas.....	65
Figura 47. Estimación de pose inicial	66
Figura 48. Estimación de trayectoria	66
Figura 49. Propuesta: Plataforma de telepresencia robusta	72
Figura 50. Propuesta: Sistema para seguimiento de Personas.....	72

RESUMEN

Uno de los principales objetivos de los robots de servicio pretende resolver el problema de la interpretación del comportamiento humano. De esta forma, el reconocimiento de gestos se vuelve esencial para la interacción humano-máquina. Siguiendo este principio en este trabajo se presenta una técnica que permite almacenar y reconocer gestos, utilizando un Kinect y bajo el supuesto de que pueda ser utilizado por usuarios desconocidos.

Inicialmente el sistema de reconocimiento de gestos está pensado para ser utilizado en robots de servicio, por lo cual, se desarrolló un prototipo de robot móvil, de bajo costo y con ciertas características de un robot de telepresencia médica, las cuales se detallan en la patente desarrollada por (Pinter, y otros, 2015).

El sistema robótico completo presenta una solución al problema de reconocimiento de gestos realizados por un usuario, la representación virtual del esqueleto de la persona registrada por el Kinect, el diseño de un modelo digital del robot y la navegación teleoperada y autónoma del robot en un ambiente controlado.

PALABRAS CLAVES:

- **RECONOCIMIENTO DE GESTOS.**
- **KINECT.**
- **ROBOT MÓVIL.**

ABSTRACT

One of the main targets of the robots service aims to solve the problem of the interpretation of human behavior. In this way, gesture recognition becomes essential for human machine interaction. Following this principle in this work is presented a technique that allows to store and recognize gestures, using a Kinect and under the assumption that it can be used by unknown users.

Initially the gesture recognition system is intended to be used in service robots, so, a mobile robot prototype was developed, low cost and with certain characteristics of a medical telepresence robot, which are detailed in the patent developed by (Pinter, y otros, 2015).

The complete robotic system presents a solution to the problem of gestures recognition made by a user, the virtual representation of the skeleton of the person registered by the Kinect, the design of a digital model of the robot and the tele operated and autonomous navigation of the robot in a checked environment.

KEYWORDS:

- **GESTURE RECOGNITION.**
- **KINECT.**
- **MOBILE ROBOT.**

CAPÍTULO I

GENERALIDADES

1.1. Introducción

El capítulo 1, plantea el problema a resolver, su importancia, alcance y objetivos; además, con el fin de delimitar el estudio, se hace una revisión de trabajos previos relacionados con el tema, para así desarrollar una base clara del proyecto. Al final del mismo, se hace una breve introducción a los capítulos presentes en el trabajo, describiendo brevemente la estructura del documento.

1.2. Planteamiento del problema

Para (Días Alabart, 2018), la robótica social fundamenta sus estudios en la interacción hombre-máquina, siguiendo comportamientos sociales establecidos, convirtiéndose en un campo de investigación emergente gracias al desarrollo de nuevas tecnologías tanto de hardware como de software, lo que ha facilitado el desarrollo de diversas aplicaciones que sirven a sectores específicos de la población y que ayudan en sus vidas cotidianas.

En este tipo de robots se trata de integrar capacidades que en cierta forma tratan de imitar a las del ser humano (vista, tacto, oído, etc.) y a los cuales, se los ha puesto a prueba en diferentes entornos; siguiendo este principio a continuación se describen

algunos ejemplos de investigaciones relacionadas las cuales forman la base para el desarrollo del presente trabajo:

1. El proyecto realizado por (Priyandoko, Wei, & Achmad, 2018), en la facultad de Ingeniería Mecánica de la Universidad Malaysia Pahang, el cual consistió en una plataforma robótica móvil para asistencia que puede rastrear y seguir a una persona objetivo en base a los movimientos que realice.

Como se puede observar en la figura 1 el sistema robótico consta de un sistema de visión artificial, una base móvil, sensores necesarios para la navegación y un computador de procesamiento.



Figura 1. Robot de asistencia para seguir personas
Fuente: (G. Priyandoko, Choi Kah Wei & M.Hendriyawan, 2018)

2. Otro ejemplo que describe una aplicación robótica para asistencia fue el desarrollado por (Bao Xin Chen, Sahdev, & Tsotsos, 2017), en la Universidad de York, Canadá, proyecto que consistió en el desarrollo de un algoritmo para seguimiento de personas, el cual denominaron SOAB¹.

¹ De las siglas en inglés Selected Online Adaptive Boosting, método para mejorar el desempeño de algoritmos de aprendizaje supervisado.

Como se puede observar en la figura 2, el sistema final incorpora una base móvil, una cámara 3D y un computador para procesamiento; como resultado lograron mejorar el rastreo de personas en tiempo real.



Figura 2. Seguimiento de persona objetivo.

Fuente: (Bao Xin Chen, Sahdev, & Tsotsos, 2017)

3. El proyecto realizado por (López Monroy & Leal Meléndrez, 2011) en el Instituto Nacional de Astrofísica Óptica y Electrónica de Puebla-México, consistió en la implementación de algoritmos de control basado en modelos ocultos de Markov (HMM) y utilizando un sensor Kinect², con lo cual lograron desarrollar un sistema basado en un método de Machine Learning para el reconocimiento y almacenamiento de gestos.

² Sensor que captura información RGBD, utilizado en aplicaciones para percepción de movimiento humano



Figura 3. Gestos ejecutados por el usuario

Fuente: (López Monroy & Leal Meléndrez, 2011)

4. Con el propósito de continuar con la línea de investigación de plataformas robóticas que se ha desarrollado en los últimos años en la Universidad de las Fuerzas Armadas – ESPE y tomando en cuenta que, el sistema a desarrollar debe ser multiplataforma, se toma como punto de partida el caso del rediseño de la plataforma robótica SIMBA (Quinaluisa M. & Toapanta C., 2018) mostrado en la figura 4(a), la cual realiza SLAM³ y navegación reactiva y el robot de telepresencia controlado a través de internet, KRONOS (Córdova L. & Medina T., 2018), mostrado en la figura 4(b).

³ Algoritmos para realizar localización y mapeo simultaneo

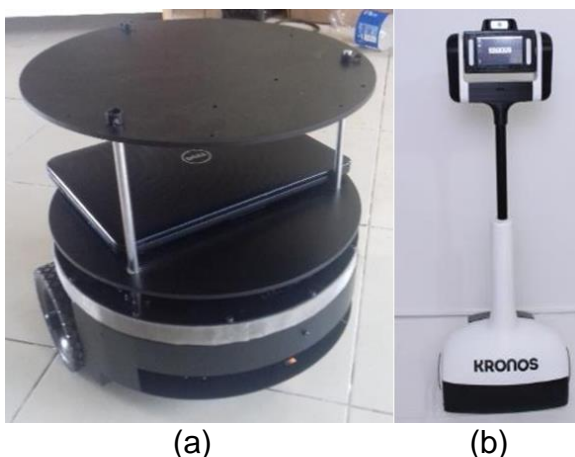


Figura 4. Robots móviles Universidad de las Fuerzas Armadas –ESPE. (a) Robot SIMBA. (b) Robot KRONOS

Fuente a) (Quinaluisa M. & Toapanta C., 2018), b) (Córdova L. & Medina T., 2018)

Analizando las consideraciones descritas anteriormente, el presente trabajo plantea una solución para integrar un sistema de visión artificial para el reconocimiento y almacenamiento de gestos en una plataforma robótica móvil con capacidad para navegación autónoma y teleoperada, basado en ROS⁴.

1.3. Justificación e importancia

Actualmente el avance de la robótica social está dirigido a desarrollar robots con mayor grado de autonomía y que sean de ayuda en tareas cotidianas para personas que lo necesiten.

Tomando en cuenta estas necesidades, este trabajo plantea una solución para integrar un sistema de reconocimiento y almacenamiento de gestos en una plataforma robótica móvil de configuración diferencial, que sea capaz de reconocer un entorno controlado (un circuito cerrado de calles a escala sin desniveles donde el robot puede

⁴ De las siglas en inglés Robot Operating System, entorno para desarrollo de aplicaciones robóticas

interactuar sin problemas). Además, la integración del sistema se fundamenta en el principio multiplataforma, para que pueda ser utilizado en diferentes aplicaciones, variando ciertos parámetros en su configuración.

Se prevé que el presente trabajo basado en ROS, sirva como estudio para futuras investigaciones y que se puede acoplar a diferentes usos en otros proyectos de robótica social.

1.4. Alcance del proyecto

El proyecto incluye el diseño y construcción de una plataforma robótica móvil de configuración diferencial, el desarrollo de software basado en ROS y la integración de un sistema de visión artificial que permita almacenar y reconocer gestos; al final se desarrollará una interfaz gráfica para la interacción entre el usuario y la plataforma robótica.

El diseño del sistema considera cuatro componentes fundamentales:

- Diseño e implementación del componente mecánico
- Diseño del componente eléctrico/electrónico
- Diseño del componente de control
- Desarrollo de software

El componente mecánico está compuesto del diseño, modelamiento matemático e implementación de un prototipo de plataforma robótica de configuración diferencial, de fácil montaje y con puntos de sujeción adecuados para los sensores y actuadores utilizados.

El componente eléctrico/electrónico está compuesto del diseño electrónico de los circuitos de control y potencia, tomando en cuenta los requerimientos de procesamiento de datos, consumo energético y características de control de los actuadores.

El componente de control se enfoca en la navegación de la plataforma robótica de forma teleoperada y autónoma, además del reconocimiento de gestos con el cuerpo humano, que permita la interacción con una persona en un entorno controlado.

El desarrollo de software está basado en ROS, ya que permite integrar los sistemas mediante nodos que intercambian información.

1.5. Objetivos

1.5.1. Objetivo General

Implementar reglas de comportamiento social en una plataforma robótica de telepresencia, través del reconocimiento de gestos con el cuerpo humano

1.5.2. Objetivos Específicos

- Diseñar y construir una plataforma robótica de configuración diferencial con soportes adecuados para cada uno de los sensores utilizados.
- Implementar el control de la plataforma robótica móvil para realizar mapeo mediante teleoperación y posterior navegación autónoma.
- Desarrollar un sistema de visión artificial, que permita identificar y almacenar gestos con el cuerpo humano.
- Realizar una interfaz gráfica de interacción entre el usuario y la plataforma robótica.

1.6. Estructura del documento

El presente documento está dividido en 5 capítulos, en los cuales se describe el desarrollo del tema.

En el capítulo 1, se hace énfasis en los precedentes que conllevaron a realizar el proyecto, se definen los objetivos y se delimita el alcance del mismo.

En el capítulo 2, se describe el estado del arte y definiciones puntuales de los sistemas robóticos relacionados al estudio.

En el capítulo 3, se realiza el diseño mecatrónico del sistema, basado en la norma VDI 2206, y el modelo para desarrollo de productos propuesto por (Ulrich & Eppinger, 2013).

En el capítulo 4, se indica las pruebas y los resultados obtenidos de los objetivos planteados en el proyecto.

Finalmente en el capítulo 5, se detallan las conclusiones y recomendaciones obtenidas del trabajo durante el proceso de diseño y pruebas realizadas.

CAPÍTULO II

ESTADO DEL ARTE

2.1. Introducción

El capítulo 2 presenta la descripción científica y tecnológica del problema. Se realiza una descripción de estudios relacionados con la visión artificial en la robótica, un análisis del uso de visión artificial para el reconocimiento de gestos corporales y la descripción del sistema de locomoción y herramientas que permiten el desarrollo de la plataforma robótica.

2.2. Introducción al uso de visión Artificial en la robótica

La tecnología que en cierta forma se encarga de suministrar sentido de la vista a un robot se denomina visión artificial, la cual, posee metodologías que se encargan de la adquisición, procesamiento y análisis de las imágenes del mundo real, con la finalidad de obtener información numérica relevante para ser procesada en diferentes aplicaciones.

Al seleccionar un sistema de visión artificial para uso en robótica, se encuentran disponibles los sistemas de visión artificial 2D y 3D. Las aplicaciones con necesidades de visión más simples, como la identificación de contornos, utilizan cámaras 2D. Para escenarios que requieran información de profundidad y orientación, se requiere un sistema de visión 3D, por ejemplo, la extracción de características del movimiento humano, que se ha convertido en un área de investigación amplia, ya que puede ser utilizado para detectar caídas, reconocer gestos, seguir personas, etc.

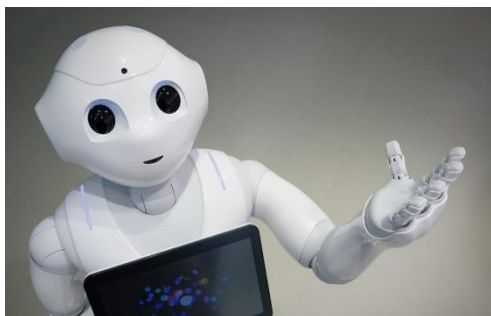


Figura 5. Robot social Pepper
Fuente: (SoftBank Robotics, 2019)

Una aplicación comercial del uso de la visión artificial en la robótica se muestra en la figura 5, un robot de nombre “Pepper”, desarrollado por (SoftBank Robotics, 2019), humanoide programable diseñado para interactuar con personas; sus múltiples sensores y cámara 3D le permiten explorar su entorno, reconocer gestos y sonidos, además, puede interpretar el estado de animo de las personas.

2.3. Machine learning aplicada a la visión robótica.

El Machine Learning o aprendizaje automático es una metodología que utiliza algoritmos que permiten a un sistema aprender por sí solo, encontrar patrones en los datos, contribuir en la toma de decisiones y mejorar predicciones.

Desde el punto de vista de los sistemas de visión por computadora, Machine Learning puede ofrecer métodos efectivos para construir sistemas de procesamiento de imágenes entrenables, reconocer objetos y enfocar la atención en un objetivo específico, gracias a que en los últimos años los métodos que utiliza se volvieron prácticos y eficientes.

En la tabla 1, se puede observar aplicaciones de visión artificial utilizada en robótica con los métodos antes mencionados.

Tabla 1.**Aplicaciones de Machine Learning y visión artificial en robots móviles**

Ítem	Aplicación	Descripción	Imagen
1	Navegación autónoma en carriles	(Bechtel, McElhiney, Kim, & Yun, 2017), utilizando un sistema de visión artificial 2D, desarrollaron algoritmos para la navegación autónoma dentro de carriles y reconocimiento de paradas establecidas	
2	Reconocimiento de objetos	(Ferguson & Law, 2019), utilizando la información de un Kinect y redes neuronales desarrollaron un algoritmo capaz de clasificar y localizar objetos para ser utilizado en robots móviles.	
3	Seguimiento de personas	(Bao Xin Chen, Sahdev, & Tsotsos, 2017), utilizando la información de una cámara 3D, desarrollaron un algoritmo para el rastreo y seguimiento de personas en tiempo real sin fallas críticas y en situaciones diversas.	

2.3.1. Técnicas de Machine Learning aplicadas a la visión artificial

Como se puede observar en la figura 6 (MathWorks, 2019), ha clasificado las técnicas de Machine Learning en dos grupos:

- Aprendizaje supervisado, el cual se utiliza cuando se tienen datos conocidos para una salida que se intenta predecir, empleando técnicas de clasificación y regresión. Las técnicas de clasificación, utilizan datos que se pueden categorizar,

etiquetar o dividir en grupos específicos. Las técnicas de regresión, trabajan con intervalos de datos que producen respuestas con números reales.

- Aprendizaje no supervisado, el cual, se utiliza cuando solo se conocen los datos de entrada y se desea realizar una clasificación que sea capaz de encontrar por sí mismo patrones o similitudes y que se utilizaran para predecir clases sobre nuevos datos.

El clustering es la técnica de aprendizaje no supervisado más común. Se emplea para el análisis de datos exploratorio, con el objeto de encontrar diferentes grupos dentro de los elementos de los datos.

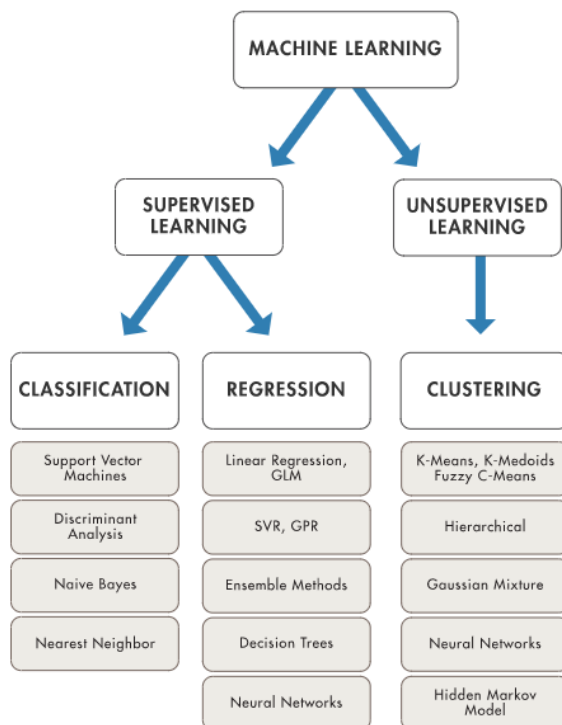


Figura 6. Técnicas de Machine Learning
Fuente: (MathWorks, 2019)

Tabla 2.

Aplicaciones con técnicas de Machine Learning para visión artificial.

Ítem	Aplicación de machine learning	Descripción	Imagen
1	Técnica de clasificación (Lenguaje supervisado)	(Cobeña Zambrano, 2018), utilizando una cámara RGBD, desarrollo un sistema de reconocimiento de gestos específicos para el control de drones, basando su estudio en la técnica de machine learning SVM (Supervised Vector Machine).	
2	Técnica de regresión (Lenguaje Supervisado)	(Strezoski, Stojanovaki, Dimitrovaki, & Madjarov, 2016), utilizaron GoogleNet, una red neuronal convolucional pre entrenada para realizar estudios de reconocimiento de gestos con la mano en imágenes.	
3	Técnica de clustering (Lenguaje no supervisado)	(López Monroy & Leal Meléndrez, 2011), realizaron un algoritmo de control basado en modelos ocultos de Markov (HMM) y utilizando un sensor Kinect para el entrenamiento y reconocimiento de gestos con la mano derecha	

Como se puede ver en la tabla 2, los tres estudios se basan en el reconocimiento de gestos utilizando diferentes técnicas de machine learning, cada uno utilizado en diferentes aplicaciones pero basadas en el mismo principio. Según (MathWorks, 2019) la selección de una técnica de Machine Learning radica en el problema de saber identificar y determinar el caso de aprendizaje que mejor se ajuste al estudio,

concluyendo que, el resultado dependerá del tamaño y tipo de datos con los que se trabaja, la información que se desea obtener y cómo se empleara dicha información.

2.4. Reglas de comportamiento social aplicadas en la robótica

(Pinter, y otros, 2015), en la patente “*Social behavior rules for a medical telepresence robot*”, realizaron una descripción de los dispositivos, sistemas y métodos para el comportamiento que debe tener un robot de telepresencia, el cuál debe incluir:

- Un sistema de navegación para movilizar al robot.
- Un sistema de control configurado para controlar el sistema de navegación en un área de trabajo.
- Un sistema de detección de objetos, configurado para detectar humanos cerca del robot.
- Reglas de comportamiento social configuradas para proporcionar instrucciones cuando se detecten personas cerca al robot.

A su vez la patente clasifica los componentes de comportamiento social en 6 grupos:

- Componente de trayectoria social: crea o modifica rutas para mantener una distancia aceptable con los humanos.
- Componente de clasificación: clasifica e identifica personas para interacciones individuales en diferentes escenarios.
- Componente para determinación de estado: determina el estado actual de un robot o persona cercana.
- Componente de reconocimiento: reconocimiento de personas, indicaciones visuales, indicaciones audibles y reconocimiento de gestos.

- Componente de gestos: gestos realizados con el cuerpo.
- Componente de personalidad: control del robot para operar de acuerdo a una personalidad asignada.

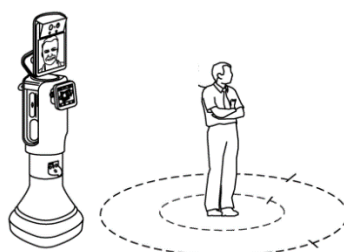


Figura 7. Comportamiento social de un robot de telepresencia

Fuente: (Pinter, y otros, 2015)

De las condiciones anteriores se puede concluir que el principal aspecto de la robótica social se reduce a resolver el problema de interpretación del comportamiento humano. Aunque las personas pueden comunicarse solo con el habla, los gestos se usan con frecuencia en la vida diaria, especialmente al explicar acciones. Por lo tanto, el reconocimiento de gestos se convierte en una tarea importante para que los robots y humanos coexistan en el mismo entorno.

2.4.1. Gestos corporales

En este contexto, la palabra gesto hace referencia a cierta orientación de la totalidad o parte del físico de una persona para transmitir información no verbal; como es el caso de los brazos y las manos con especial interés de estudio en diferentes aplicaciones, ya que, ambos son esenciales en la comunicación y acciones que se realizan en el día a día.

Según (Konar & Saha, 2018), los gestos incluyen una descripción estática y dinámica del físico del cuerpo humano; el primero se representa como una orientación fija por

parte de una o varias partes del cuerpo humano para transmitir un mensaje no verbal a otros y el segundo incluye el movimiento de una parte del cuerpo.

Independientemente de las características estáticas y dinámicas de los gestos, se los puede clasificar en dos grupos principales: gestos intrínsecos y gestos extrínsecos.

Los gestos intrínsecos hacen referencia a una intención subjetiva destinada a uno mismo y no a comunicarse con otros, como por ejemplo una persona con ansiedad que lo representa por morderse un dedo.

Los gestos extrínsecos por su parte están destinados a comunicar un mensaje hacia otras personas, e involucra gestos emocionales y gestos no emocionales (instrucciones, señales, etc.)

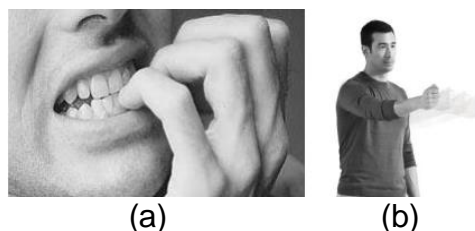


Figura 8. Gestos realizados con el cuerpo. a) Gesto intrínseco, ansiedad. b) Gesto extrínseco, señal

Fuente: a) (Sepimex, 2012). b) (Microsoft Corporation, 2014)

2.4.2. Reconocimiento de gestos mediante visión artificial

El reconocimiento de gestos hace referencia a la identificación y reconocimiento de gestos originados por cualquier tipo de movimiento corporal, permitiendo a las personas interactuar con una máquina sin tener que usar un dispositivo mecánico de por medio.

El principio básico de funcionamiento involucra una cámara que lee los movimientos del cuerpo humano y transmite esos datos a una computadora que usa los datos como entrada para procesarlos y posteriormente controlar dispositivos y aplicaciones.

2.5. Uso del sensor Kinect para reconocimiento de gestos

En 2010 Microsoft, lanza a la venta el Kinect, dispositivo que permite al usuario interactuar de manera natural mediante comandos de voz y gestos con la videoconsola Xbox 360 y posteriormente con computadoras, a través de una versión de Kinect para desarrolladores y software específico para Windows como el Microsoft Kinect SDK y OpenNI para Linux.

La tecnología del Kinect tanto en hardware como en software, ha permitido extender su uso más allá de los videojuegos, convirtiéndose en una herramienta para investigaciones basadas en visión artificial para el reconocimiento y seguimiento del movimiento humano.

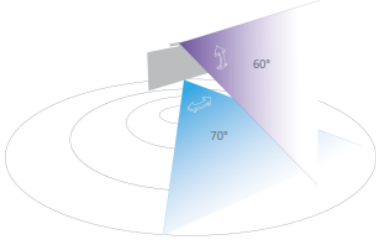


Figura 9. Detección de caídas utilizando sensor Kinect
Fuente: (Tran, Le, Hoang, & Vu, 2017)

2.5.1. Aspectos técnicos del sensor Kinect

En la tabla 3 se puede observar las especificaciones técnicas del sensor, que le permiten capturar información RGBD con una resolución de 640x480 donde cada pixel tiene información de color y de profundidad.

Tabla 3.*Especificación técnica sensor Kinect para Windows modelo 1517*

Característica	Especificación	Imagen
Ángulo de visión	70° horizontal 60° vertical	
Detección	Limitación física de 0.5m y 4.5m Punto óptimo 0.8m a 3.5m	
Resolución	640 x 480 30 fps y 1280 x 960 12 fps (solo cámara RGB)	
Entrada de audio	Array de 4 micrófonos ADC de 24bits con cancelación de eco y supresión de ruido	

Fuente: (Microsoft Corporation, 2014)

2.5.2. OpenNI y NITE

OpenNI es un framework⁵ de código abierto disponible para Linux y otras distribuciones, que permite comunicarse con los sensores disponibles en el Kinect y desarrollar aplicaciones basadas en interacción natural (gestos y movimientos corporales) mediante el middleware⁶ NITE, que permite extraer la posición y orientación de las articulaciones del esqueleto.

⁵ Entorno de trabajo, esencial para el desarrollo de diferentes programas

⁶ Software que funciona como una capa de traducción oculta y permite la comunicación entre aplicaciones

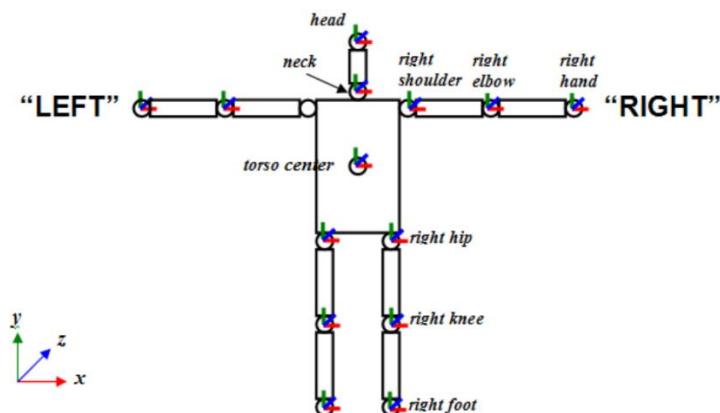


Figura 10. Articulaciones del esqueleto. NITE
Fuente: (Prime Sense Inc., 2010)

2.5.3. Proceso de reconocimiento de gestos

La figura 11, muestra el diagrama de bloques de un sistema de reconocimiento de gestos utilizando un Kinect. Inicialmente, un usuario realiza un gesto, que se detecta y registra con el Kinect. Utilizando los datos de esqueleto proporcionados por el software del sensor (OpenNI y NITE), se extraen las características relevantes al gesto que se utilizan como observaciones. Este proceso se repite hasta obtener un conjunto de datos relevante al gesto realizado. Los datos obtenidos sirven como medio de entrenamiento para una técnica de aprendizaje de máquina arbitraria.

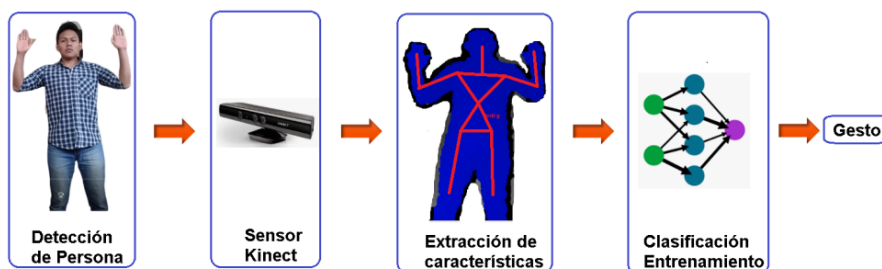


Figura 11. Proceso de reconocimiento de gestos con Kinect

2.6. Robots móviles

Un robot móvil se define como un dispositivo electromecánico, el cual posee un sistema de locomoción, capaz de desplazarse en un entorno de trabajo.

(Silva Ortigoza, García Sánchez, Barrientos Sotelo, & Molina Vilchis, 2007), describen una clasificación para los robots móviles de acuerdo al tipo de locomoción que poseen y los dividen en tres grupos, que son:

- Robots móviles tipo oruga
- Robots móviles con patas
- Robots móviles con ruedas

De la clasificación anterior se considera para este caso de estudio a los robots móviles con ruedas, debido a razones prácticas, como lo son: facilidad de construcción, pueden transportar más carga y existe mayor información y trabajos realizado con este mecanismo.

(Ollero Baturone, 2005), clasifica a los robots móviles con ruedas de acuerdo a la aplicación que este enfocado su uso en:

- Ackerman
- Triciclo clásico
- Tracción diferencial
- Skid steer
- Tracción síncrona
- Tracción omnidireccional

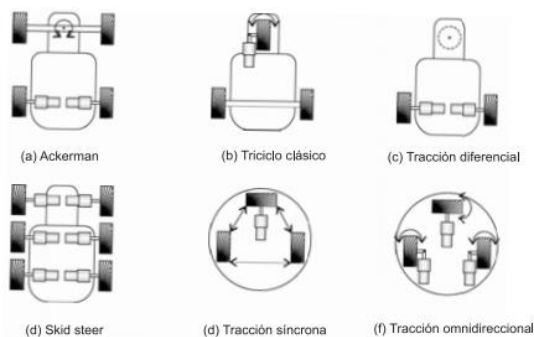


Figura 12. Clasificación de robots móviles con ruedas

Fuente: (Silva Ortigoza, García Sánchez, Barrientos Sotelo, & Molina Vilchis, 2007)

De estas configuraciones, se considera la de mayor facilidad de implementar, el robot móvil de configuración diferencial, debido a que su diseño minimiza el sistema de control a utilizar, reduce la complejidad mecánica de implementación, así como la electrónica y programación del mismo; por lo tanto se convierte en una opción barata y fiable para muchas aplicaciones.

El robot móvil de configuración diferencial, consiste de dos ruedas principales, montadas en un eje común, consiguiendo el direccionamiento controlando la velocidad relativa de las ruedas de cada lado; además, posee una o dos ruedas locas con la finalidad de dar estabilidad a la estructura mecánica como rueda o ruedas de dirección.

2.7. Navegación en robots móviles

Para (Rhoebly Dynamics, 2015-2019), la solución al problema de la navegación de robots móviles comprende diferentes componentes de hardware y software, incluyendo:

- Sensor LIDAR
- IMU(giroscopio / acelerómetro)
- Localización
- Mapeo

- Planeación de trayectoria
- Detección y evasión de obstáculos

2.7.1. Dispositivo de percepción móvil LIDAR

El LIDAR (Light Detection and Ranging), es un sensor cuyo principio básico de funcionamiento es la emisión y recepción de una señal laser infrarroja que permite medir datos de distancia; entre los modelos más comunes se encuentran los dispositivos que giran 360 grados sobre sí mismo para cubrir todo el entorno. (Ibáñez, 2017).

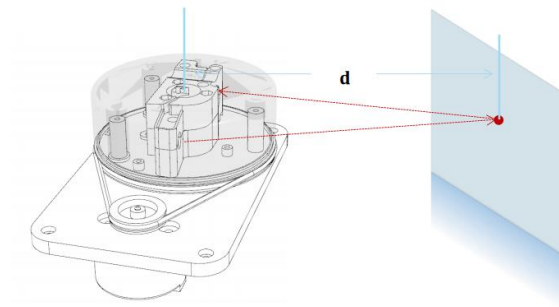


Figura 13. RPLIDAR A1 esquema de trabajo
Fuente: (Slamtec, 2013-2016)

De esta manera el procesador del LIDAR obtiene una nube de puntos del entorno, los cuales pueden ser procesados por un computador para generar una imagen en tiempo real; lo más importante de esta nube de puntos, es que para cada uno de los puntos se conoce su posición en el espacio y distancia que hay hasta él. (Slamtec, 2013-2016).

2.7.2. IMU (Inertial Measurement Unit).

El sensor IMU (Inertial Measurement Unit), por sus siglas en inglés es un módulo electrónico que recolecta datos de velocidad y aceleración lineal (Castillo, Lozano, & Dzul, 2006), usando una combinación de acelerómetros, giroscopios y a veces magnetómetros.

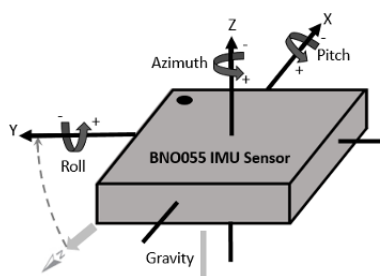


Figura 14. Descripción general de un sensor IMU
Fuente: (Mathworks, 1994-2019)

El sensor IMU es utilizado en un los robots móviles de navegación autónoma, con el objetivo de dar una buena estimación de la odometría y la postura del robot. Los valores de odometría obtenidos por los encoders en los robots móviles no son suficientes para una navegación eficiente, por lo tanto se utiliza un IMU para compensar el error de movimiento especialmente en la rotación. (Lentin, 2018).

2.7.3. Localización

La localización es un requisito esencial para la navegación de robots móviles, ya que se debe conocer la posición y orientación en cada instante de tiempo dentro del entorno con el fin alcanzar un punto de destino.

Un robot móvil equipado con sensores para controlar su movimiento (por ejemplo encoder en los motores y un sensor inercial IMU), puede calcular una estimación

relativa de su ubicación actual en relación al lugar donde empezó, si hay disponible un modelo matemático de movimiento conocido como odometría.

Sin embargo a medida que el robot navega en su entorno y sumado los errores presentes en las mediciones de los sensores, las estimaciones de ubicación del robot son menos confiables. Dichos errores pueden corregirse cuando el robot puede observar su entorno utilizando algún sensor (LIIDAR, Kinect, etc.), que pueda correlacionar la información obtenida con la información contenida en un mapa.

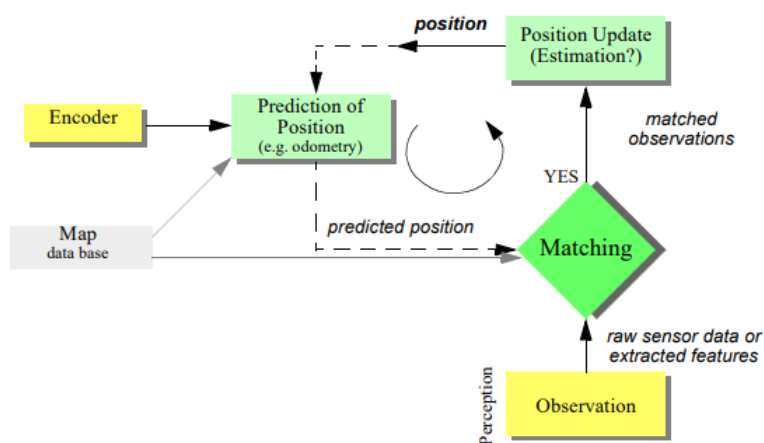


Figura 15. Esquema general de localización para robots móviles
Fuente: (Siegwart & Nourbakhsh, 2004)

2.7.4. Mapeo

El proceso de mapeo y localización simultánea o SLAM por sus siglas en inglés, resuelve el problema de obtener una estimación de un entorno o un mapa y a la vez la localización en el mismo, tomando en cuenta que el ambiente desconocido debe ser estático, conteniendo solo objetos rígidos y estacionarios.

(Velásquez, Chávez, & Córdova, 2015), describen los pasos que garantizan un mapeo y estimación de la posición lo más preciso posible. Primero calcular la posición

mediante odometría, luego realizar una muestra del entorno para refinar una primera estimación de la pose del robot y estimar el mapa alrededor del robot. Se procede a introducir un paso de control o movimiento y se calcula por odometría la nueva ubicación del robot, finalmente se debe poder observar nuevamente la muestra que se vio en el primer paso, con el fin de actualizar y refinar la pose y mapa del robot en su entorno.

El mapa generado tiene la forma de una cuadrícula de ocupación (occupancy grid), donde se identifica las regiones ocupadas y libres de un entorno, mientras que los sensores a bordo del robot móvil miden la distancia a la región ocupada más cercana en una dirección específica. Ya que la información contenida en los sensores posee ruido es necesario estimar además de su localización, la incertidumbre asociada a la ubicación, y es aquí donde nuevamente la confiabilidad de la estimación de ubicación se ve afectada por condiciones externas; una solución a este problema es la incorporación de filtros como un filtro de Kalman o un filtro de partículas.

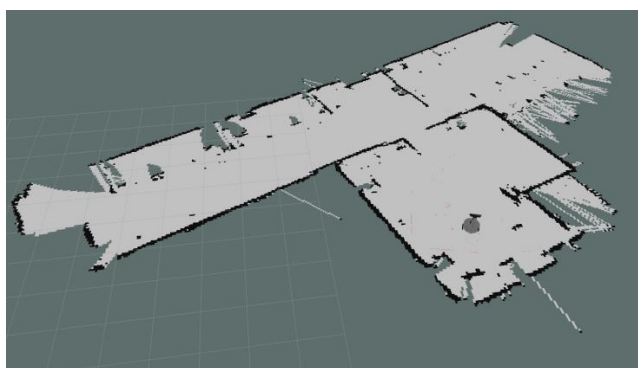


Figura 16. Mapa de ocupación

2.7.5. Planeación de trayectoria

Se refiere a la identificación de una trayectoria que hará que el robot alcance un objetivo una vez sea ejecutada. Según (Siegwart & Nourbakhsh, 2004) la planificación de trayectorias es una competencia estratégica de resolución de problemas, lo cual, significa que el robot debe decidir qué hacer a largo plazo para lograr su objetivo.

El principio de funcionamiento de evasión de obstáculos pretende cambiar la trayectoria del robot durante la navegación cuando este encuentra un obstáculo estático o dinámico, a medida que se acerca a su objetivo.

Un método de planeación de trayectorias que utiliza una cuadrícula de ocupación divide el área en celdas y las asigna como ocupadas o libres, además, una de las celdas se marca como posición del robot y otra como destino. Encontrar la trayectoria se basa en encontrar la línea más corta que no cruce ninguna de las celdas ocupadas.

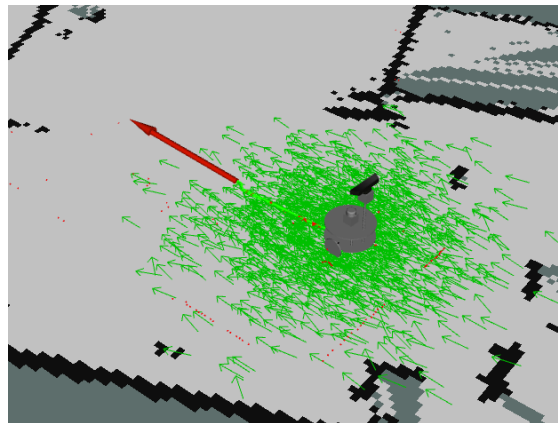


Figura 17. Planeación de trayectorias

CAPÍTULO III

DISEÑO MECATRÓNICO

3.1. Introducción

El capítulo 3 presenta la metodología de diseño del sistema robótico.

Primero, haciendo uso del “*modelo V*”, disponible dentro de norma para el diseño de sistemas mecatrónicos, VDI 2206, indicado en la figura 18, se propuso dividir el proyecto en etapas o subsistemas que puedan ser analizadas por separados y luego integrarse en un solo sistema o producto final.

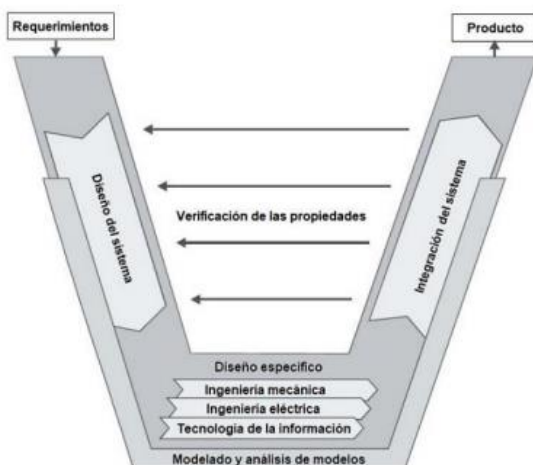


Figura 18. Norma VDI 2206 Modelo V para diseño de sistemas mecatrónicos

Fuente: (German Association of Engineers, 2004)

A continuación, para cada uno de los subsistemas, se utilizó el modelo de diseño y desarrollo de productos propuesto por (Ulrich & Eppinger, 2013), indicado en la figura 19, con la finalidad de facilitar el estudio y mejorar el proceso de desarrollo.

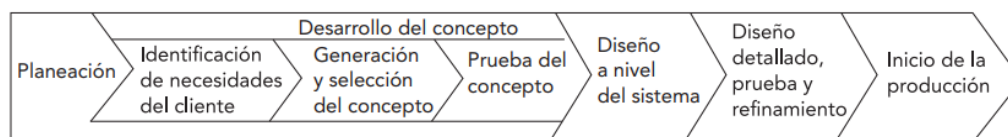


Figura 19. Proceso de desarrollo de productos
Fuente: (Ulrich & Eppinger, 2013)

3.2. Diseño de subsistemas

Para el diseño de la plataforma se identifican 3 subsistemas, mecánico, electrónico y uno de control. El subsistema mecánico y electrónico se analizó de manera simultánea, ya que, para el diseño de la plataforma robótica es necesario identificar los elementos que los conforman, característica, medidas y pesos en la mayoría de casos.

3.3. Subsistema Mecánico

La identificación de requerimientos generales que debe cumplir el subsistema mecánico para desarrollar una plataforma robótica de bajo costo y fácil ensamblaje, se presentan en la tabla 4.

Tabla 4
Identificación de requerimientos. Subsistema Mecánico

ítem	Requerimiento
1	Navegación en ambiente controlado (circuito de calles a escala donde el robot pueda navegar sin problemas)
2	Peso máximo del robot 12kg
3	Chasis de diseño simple y bajo costo
4	Modular, para fácil ensamblaje
5	Puede cargar objetos sobre ella máximo 2kg
6	Materiales utilizados de fácil acceso
7	Diseño estético
8	Velocidad máxima de funcionamiento 0.5m/s
9	Espacio adecuado para cada sensor
10	Soportes adecuados para correcto funcionamiento de los sensores

Una vez identificados los requerimientos que debe cumplir el subsistema mecánico, estos deben ser asociados con una variable que permitan su evaluación, para poder determinar el grado de cumplimientos de los objetivos.

Las métricas establecidas para los requerimientos planteados se presentan en la tabla 5. Además, se identifica el grado de importancia de las métricas asociadas al diseño del subsistema mecánico, las cuales, son tomadas en consideración para el desarrollo de la plataforma, midiendo la importancia de la función en una escala del 1 al 5, siendo 5 el grado de mayor importancia.

Tabla 5.
Listado de métricas. Subsistema mecánico

Ítem	Requerimiento	Métrica	Importancia	Unidades
1	3,4,6,7,10	Tiempo de fabricación	2	horas
2	3,4,9,10	Tiempo de ensamble	3	horas
3	2,5	Masa total	5	kg
4	2,5	Carga Útil	3	kg
5	1,4,9,10	Diámetro máximo	2	mm
6	1,2,5,8	Velocidad máxima	4	m/s
7	1,2,5	Aceleración máxima	4	m/s ²

3.3.1. Componentes del subsistema mecánico

El diseño del subsistema mecánico incluye una serie de etapas que se describen en la figura 20, y las cuales se detallan a continuación.

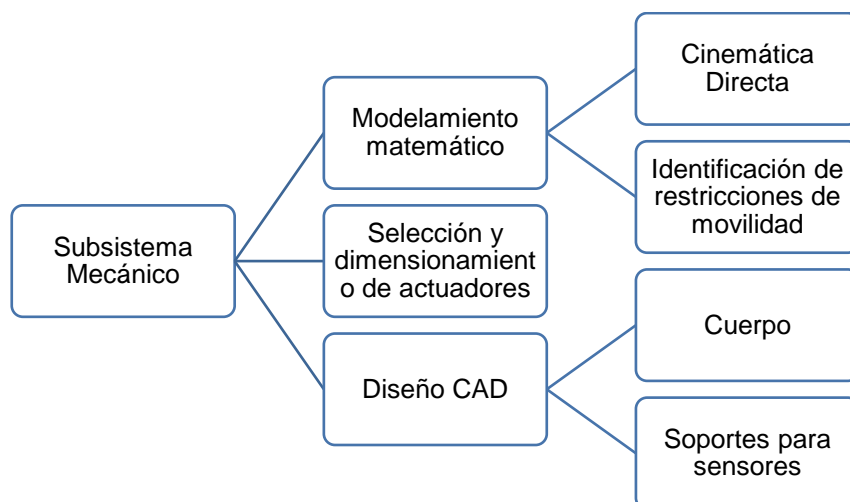


Figura 20. Componentes del subsistema mecánico

3.3.2. Modelo matemático de un robot móvil de configuración diferencial.

Para describir el movimiento de un robot móvil de configuración diferencial, se considera su localización en un plano horizontal, ya que su movimiento está restringido a la navegación por terrenos planos, por lo tanto, se necesitan de dos coordenadas en un plano (x, y) y un ángulo (θ) para describir su ubicación.

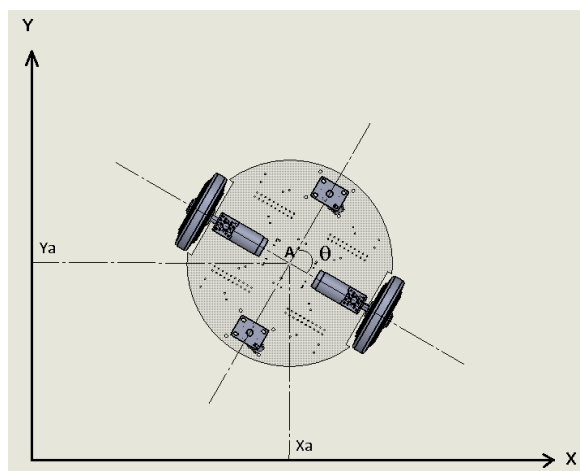


Figura 21. Sistema global de coordenadas de un robot móvil de configuración diferencial.

Tomando como punto de partida el análisis de movimiento de una rueda en un plano, cuando la rueda completa una vuelta y asumiendo que la superficie es plana y uniforme como se muestra en la figura 22(a), se cubre una distancia igual a $2\pi r$, donde r es el radio de la rueda. Por lo tanto la velocidad lineal y angular de la rueda se puede escribir como:

$$v_{rueda} = \frac{L}{t} = \frac{2\pi r_{rueda}}{t} \quad (1)$$

$$W_{rueda} = \frac{2\pi}{t} \quad (2)$$

$$v_{rueda} = W_{rueda} r_{rueda} \quad (3)$$

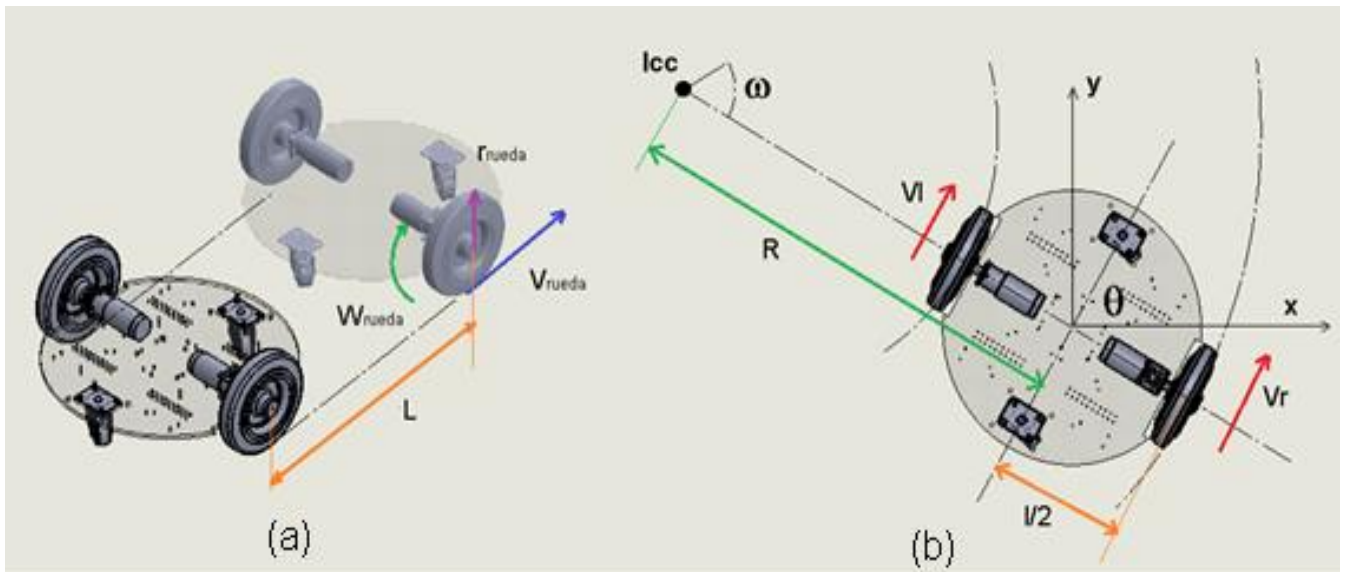


Figura 22. Diagrama detalle sistema de coordenadas robot diferencial. (a) Análisis de rueda en el plano. (b) Centro Instantáneo de giro

De manera generalizada la ecuación 3 describe el movimiento de una rueda en un plano dado, ecuación que se toma como punto de partida para desarrollar el análisis cinemático diferencial.

Aplicando de manera generalizada la ecuación 3 en el plano descrito en la figura 22(b), en ambas ruedas, se obtienen las siguientes ecuaciones:

$$Vr = \omega r = \omega(R + l/2) \quad (4)$$

$$Vl = \omega r = \omega(R - l/2) \quad (5)$$

Despejando y operando las ecuaciones 4 y 5 para encontrar R y ω , se obtiene las siguientes ecuaciones:

$$R = \frac{l(Vr + Vl)}{2(Vr - Vl)} \quad (6)$$

$$\omega = \frac{Vr - Vl}{l} \quad (7)$$

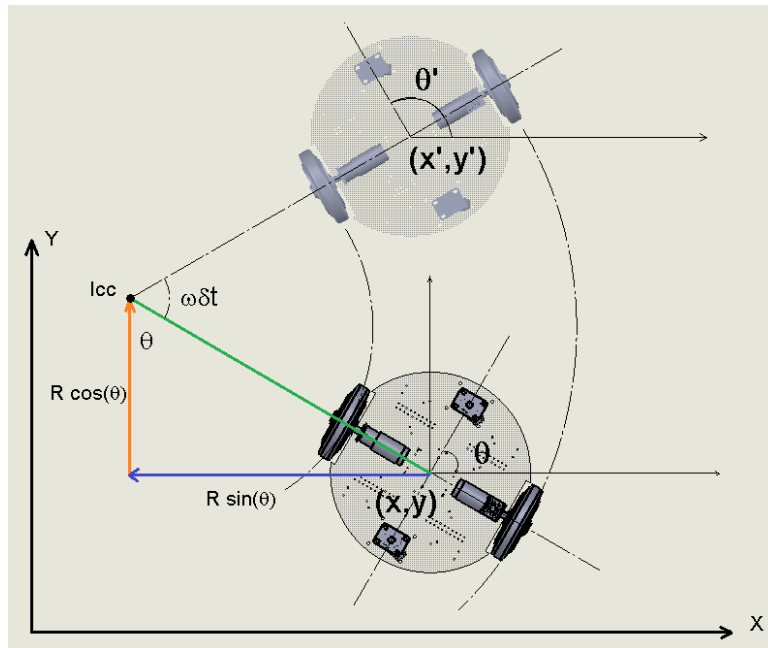


Figura 23. Rotación del robot alrededor de l_{cc} en un tiempo δt .

Suponiendo que el robot asume una rotación alrededor de l_{cc} , como se muestra en la figura 23, con una velocidad angular ω durante δt segundos, da como resultado que la orientación cambie conforme:

$$\theta' = \omega\delta t + \theta \quad (8)$$

$$Icc = [Icc_x, Icc_y] = [x - R\sin(\theta), y + R\cos(\theta)] \quad (9)$$

Dada una posición (x, y) y orientación inicial θ , se puede calcular una nueva posición (x', y') y orientación θ' utilizando la matriz de rotación:

$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} \cos(\omega\delta t) & -\sin(\omega\delta t) & 0 \\ \sin(\omega\delta t) & \cos(\omega\delta t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x - Icc_x \\ y - Icc_y \\ \theta \end{bmatrix} + \begin{bmatrix} Icc_x \\ Icc_y \\ \omega\delta t \end{bmatrix} \quad (10)$$

Ya que un robot de configuración diferencial es un sistema no holonómico (tienen restricciones que no pueden ser integradas), su movimiento se reduce a un movimiento en línea recta ($v = vl = vr$), ecuación 11 y una rotación en el eje de las ruedas ($v = -vl = vr$), ecuación 12.

$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} x + v\cos(\theta)\delta t \\ y + v\sin(\theta)\delta t \\ \theta \end{bmatrix} \quad (11)$$

$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} x \\ y \\ \theta + 2v\delta t/l \end{bmatrix} \quad (12)$$

3.3.3. Selección y dimensionamiento de actuadores

La selección de actuadores para una plataforma de configuración diferencial es una de las principales decisiones a tomar en cuenta ya que estos determinan las condiciones de operación del sistema.

Las principales consideraciones a tomar en cuenta son:

- Obtener una estimación de la masa total o mejor aún conocer la masa total real, con lo cual se puede determinar el torque necesario para mover la plataforma.

- Determinar la velocidad máxima de traslación de la plataforma.
- Determinar el voltaje de operación para los actuadores. Generalmente a mayor voltaje el actuador genera mayor velocidad.
- Precisión de los actuadores.
- Seleccionar el diámetro y material de las ruedas; esta información junto con la velocidad de traslación son importante para determinar el torque del motor.

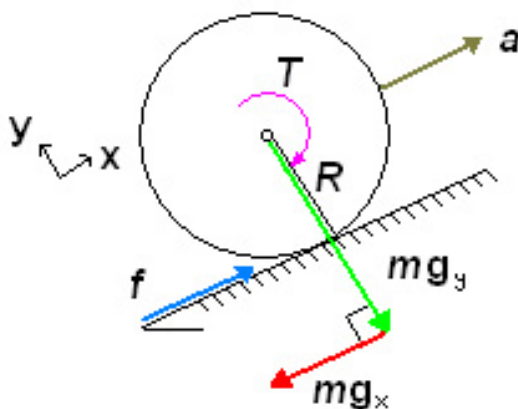
En la tabla 6, se describe la masa total estimada (kg), para el desarrollo de la plataforma robótica, además se agrega un factor de seguridad del 10% de la masa total con el fin de evitar esfuerzos en los motores seleccionados.

Tabla 6.
Cálculo de masa total de la plataforma robótica

Ítem	Elemento	Cantidad	Kg
1	Kinect	1	0.74
2	Lidar	1	0.17
3	NUC	1	0.67
4	Baterías	2	4.2
5	Cables	1	0.2
6	Electrónica	1	0.5
7	Soportes	7	1.2
8	Base	2	1.2
9	Carga útil		2
		SUMATORIA	10.88
		FACTOR 10%	1.088
		TOTAL	11.968

Tabla 7.*Consideraciones de diseño de la plataforma robótica*

Ítem	Consideración	Descripción	Valor
1	Masa total del robot	Valor calculado	12 [kg]
2	Numero de motores	Robot de configuración diferencial	2 [unidades]
3	Diámetro de las ruedas	Rueda estándar de 6 in para carga	0.1524 [m]
4	Velocidad del robot	Velocidad máxima de traslación	0.5 [m/s]
5	Aceleración	Aceleración máxima del robot. Tomada en cuenta según criterio de selección de motores para robots móviles de interiores.	0.2 [m/s ²]
6	Inclinación máxima	Máxima inclinación que la plataforma robótica puede navegar	15 [grados]
7	Eficiencia del motor	Considerando un escenario crítico para un motor DC con caja reductora planetaria (60%-85% de eficiencia)	60 %

**Figura 24.** Diagrama de cuerpo libre de una rueda en plano inclinado

Fuente: (RobotShop Community, 2014)

A partir de las consideraciones de diseño presentadas en la tabla 7 y el diagrama de cuerpo libre de una rueda en un plano inclinado, mostrada en la figura 24, a continuación, se presentan las ecuaciones necesarias para encontrar el torque necesario para mover la plataforma robótica:

$$\sum F_x = ma$$

$$f - mg\sin(\alpha) = ma \quad (13)$$

$$T = fR \quad (14)$$

Operando las ecuaciones 13 y 14, se obtiene la formula general para el torque total del sistema:

$$T_{Total} = mR(a + g\sin(\alpha)) \quad (15)$$

Tomando en cuenta el número de motores a utilizar (n) y la eficiencia de los mismos (Ef), la ecuación final del torque de un motor sería la mostrada en la ecuación 16.

$$T_{motor} = \left(\frac{100}{Ef}\right) \left(\frac{1}{n}\right) mR(a + g\sin(\alpha)) \quad (16)$$

Reemplazando los datos conocidos para la plataforma robótica presentados en la tabla 6 en la ecuación anterior, se puede obtener el torque del motor de la siguiente forma:

$$T_{motor} = \left(\frac{100}{60}\right) \left(\frac{1}{2}\right) (12[kg])(0.0762[m])(0.2[m/s^2] + 9.81[m/s^2]\sin(15^\circ))$$

$$\mathbf{T_{motor} = 2.08Nm}$$

$$\mathbf{T_{motor} = 295.42oz - in}$$

Conocida la velocidad máxima de traslación de la plataforma robótica y el diámetro de las ruedas a utilizar, se puede calcular las RPM que describen el movimiento de la plataforma robótica, de acuerdo a la ecuación 17.

$$RPM = \frac{60V_{max}}{\pi D} \quad (17)$$

$$RPM = \frac{60 (0.5[m/s])}{0.1542[m] \pi}$$

$$RPM = 61.93 \text{ rev/min}$$

Una vez conocido el torque necesario para mover la plataforma robótica y tomando en cuenta los requerimientos de diseño para el subsistema mecánico a continuación se detallan los criterios para la selección de los actuadores con sus respectivas ponderaciones, presentados en la tabla 8 y los conceptos de selección, presentados en la tabla 9.

Tabla 8.

Criterio de diseño para la selección de actuadores

	Costo	Disponibilidad	Tamaño	$\sum +1$	Ponderación
Costo	--	0.8	0.5	2.3	0.383
Disponibilidad	0.2	--	0.2	1.4	0.233
Tamaño	0.5	0.8	--	2.3	0.383
			TOTAL	6	1

Tabla 9.
Conceptos para la selección de actuadores

Concepto	Descripción	Características	Imagen
1	Motor DC con caja reductora planetaria y encoder	<ul style="list-style-type: none"> -Simple de controlar -Excelente torque a bajas velocidades -eficiencia entre 60% al 85% -Bajo costo -Bajo consumo de corriente -Para aplicaciones de mediana precisión -Tamaño reducido -Mayor cantidad de fabricantes -Variedad de reductores -Voltaje de funcionamiento 12Vdc 	
2	Servomotor	<ul style="list-style-type: none"> -Excelente torque a altas velocidades -Eficiencia 80-90% -Alto costo -Consume alta corriente -Para aplicaciones de alta precisión -Limitación de movimiento -Voltaje de funcionamiento 12Vdc 	
3	Motor a pasos	<ul style="list-style-type: none"> -No requiere caja reductora -alto torque con velocidades reducidas -pobre desempeño sobre variación de cargas -Consume alta corriente -Costoso -Para aplicaciones de alta precisión -pesado y de mayor tamaño que un motor dc y servomotor -Voltaje de funcionamiento 12Vdc 	

Fuente: Elaboración Propia

Tabla 10.
Matriz de decisión para la selección de actuadores

	Concepto	Costo			Disponibilidad			Tamaño			TOTAL
		peso	criterio	T	peso	criterio	T	peso	criterio	T	
motor	Concepto 1	0.383	8	3.064	0.233	8	1.864	0.383	8	3.064	7.992
servo	Concepto 2	0.383	5	1.915	0.233	6	1.398	0.383	8	3.064	6.377
pasos	Concepto 3	0.383	6	2.298	0.233	7	1.631	0.383	5	1.915	5.844

Una vez analizados los tres conceptos propuestos, se eligen los motores DC con caja reductora planetaria y encoder, que cumplan con el requisito de torque mayor a 295.42 oz-in y RPM mayor a 61.93 rev/min; como solución, es la que mejor se adapta al proyecto.

3.3.4. Diseño CAD

El diseño de la estructura parte de una forma circular y tomando en cuenta la mejor distribución de pesos, se opta por utilizar dos ruedas de apoyo (ruedas locas) para mantener el eje de los motores en el centro de la plataforma robótica, lo cual facilita el diseño.

3.3.4.1. Diseño de la base

Para el diseño de la base de la plataforma robótica se toma en consideración la ubicación y distribución de pesos de cada uno de los elementos que conforman el sistema y mediante un análisis de esfuerzos estático, se puede seleccionar el material con el cual se procede a fabricar la base.

El material seleccionado para la fabricación de la estructura de la plataforma robótica es MDF⁷ de 6mm, material de bajo costo y que cumple con las especificaciones de diseño, ya que, la resistencia real del material es mucho mayor que la resistencia requerida, como se muestra en la tabla 11, además, genera un desplazamiento máximo de 0.07mm, como se indica en la figura 26.

⁷ Tablero de fibra de densidad media.

Con el fin de optimizar la precisión de los resultados, se utilizó el método adaptativo h y se consideró al MDF como material ortotrópico⁸, además, se definió el número de iteraciones en 4, con lo cual se obtuvo un modelo de convergencia hacia un único valor.

Tabla 11.
Análisis estático. Base de la plataforma robótica

MDF 6mm espesor		
Propiedad Mecánica	Valor Propiedad Mecánica	Esfuerzos Von Mises (Simulación)
Resistencia última a tracción (sut)	18MPa	3.45Mpa
Límite elástico (sy)	17.5Mpa	

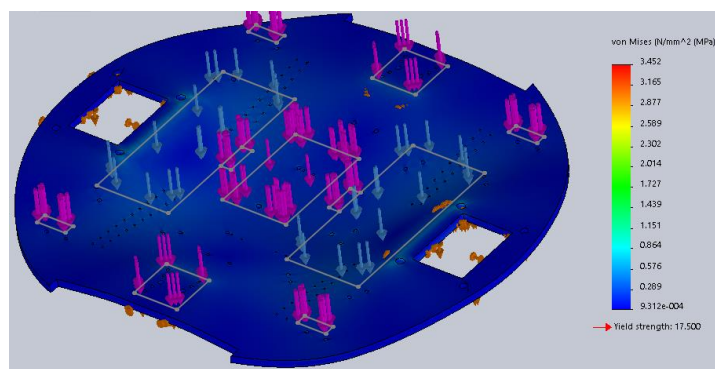


Figura 25. Cálculo esfuerzos Von Mises

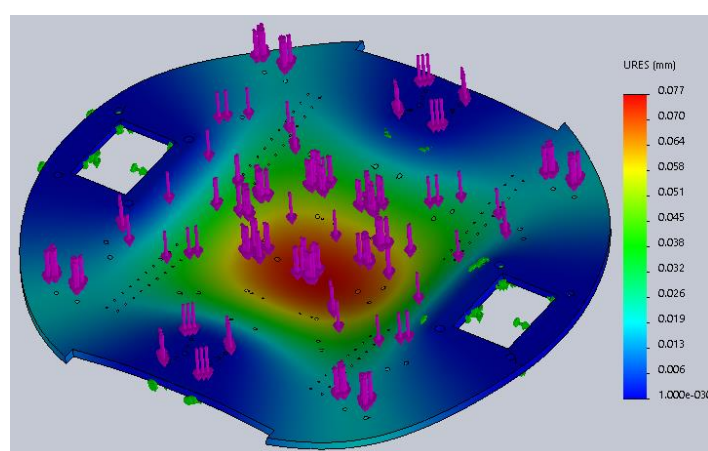


Figura 26. Cálculo de desplazamientos

⁸ Material con propiedades mecánicas únicas en tres direcciones perpendiculares entre sí, entre ellos la madera.

3.3.4.2. Diseño del sistema de tracción diferencial

El sistema de tracción diferencial propuesto consta de 5 elementos, especificados en la tabla 12 e indicados en la figura 27, de los cuales, el acople de aluminio y separador sirven de unión con la base de la plataforma y los demás elementos transmiten el movimiento generado por el motor.

Tabla 12.
Sistema de tracción diferencial

Ítem	Elemento
1	Acople en U de aluminio
2	Separador para motor
3	Motor DC con reductor planetario
4	Eje de unión
5	Rueda solida de caucho

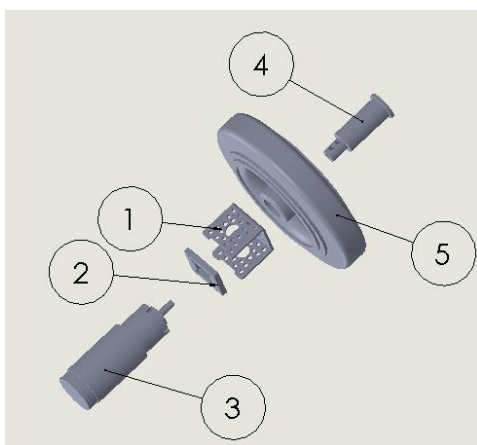


Figura 27. Acople de sistema de tracción diferencial

3.3.4.3. Distribución de elementos en la plataforma robótica

Cada uno de los elementos de la plataforma robótica ha sido colocado de manera distribuida, tomando como referencia el centro de la plataforma, lo cual facilita el diseño.

El origen del sistema de coordenadas del sensor IMU MPU6050 y del sensor RPLidar A1, coinciden con el centro de la plataforma robótica, por razones prácticas, es la mejor alternativa para su posterior simulación y programación.

El sensor Kinect se encuentra ubicado a una altura de 0.85m desde el suelo, altura desde la cual se puede detectar el cuerpo humano completo.

A continuación se realiza una descripción de los componentes de la plataforma robótica, mostrados en la figura 28 y se indica el número de elementos que la conforman en tabla 13.

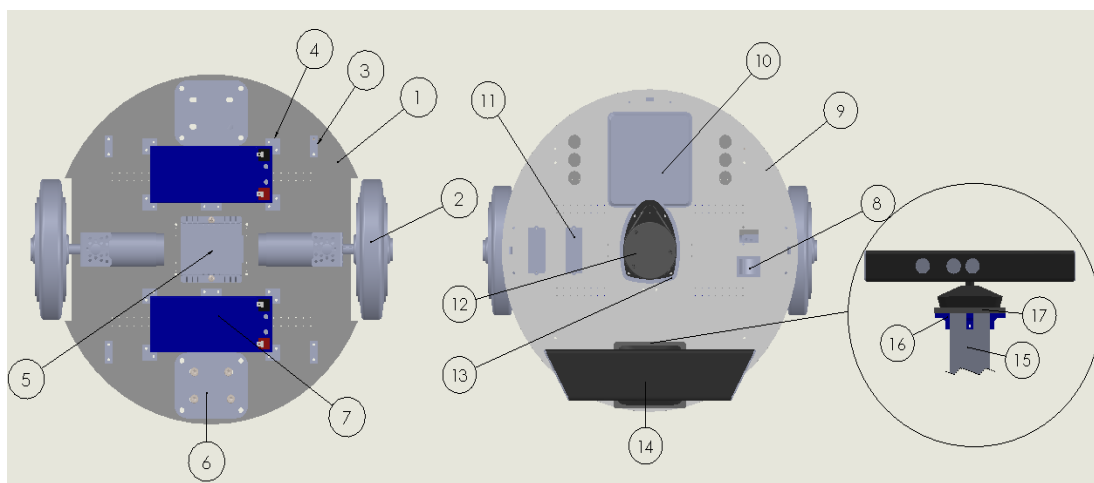


Figura 28. Distribución de componentes plataforma robótica

Tabla 13.

Descripción de componentes plataforma robótica

Ítem	Descripción	Elementos
1	Base de la plataforma robótica	1
2	Sistema de tracción diferencial	2
3	Poste	6
4	Soporte de batería	8
5	Electrónica	1
6	Rueda loca	2
7	Batería	2
8	Interruptor	1
9	Base superior	2
10	Computador Intel NUC	1
11	Indicador de nivel de batería	2
12	Sensor RPLidar A1	1
13	Soporte Sensor RPLidar	1
14	Sensor Kinect	1
15	Soporte de elevación sensor Kinect	1
16	Soporte de base sensor Kinect	3
17	Base sensor Kinect	1

En la figura 29, se muestra el modelo final del prototipo robótico construido.

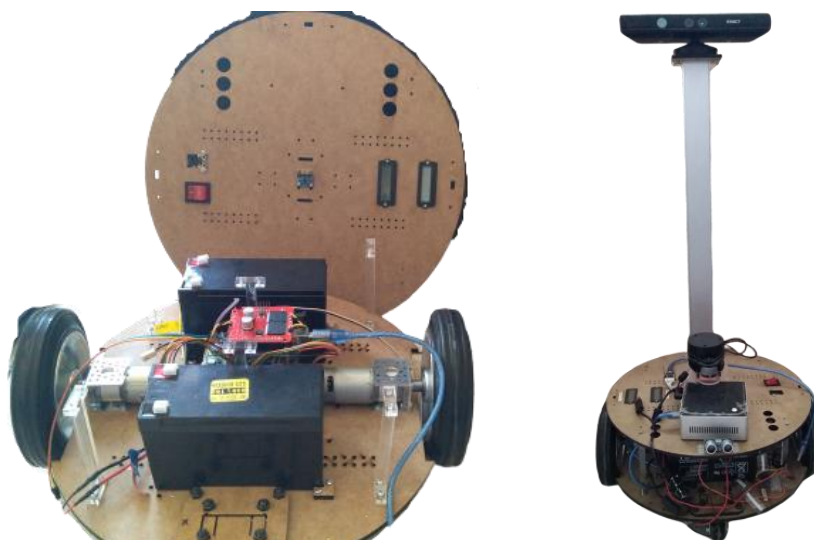


Figura 29. Modelo final del prototipo robótico

3.4. Subsistema Electrónico

La identificación de requerimientos generales que debe cumplir el subsistema electrónico para desarrollar una plataforma robótica de bajo costo y que cumpla con las características de procesamiento de datos, se presentan en la tabla 14.

Tabla 14.
Identificación de requerimientos. Subsistema Electrónico

ítem	Requerimiento
1	Alimentación energética con baterías
2	Cableado de fácil conexión
3	Bajo costo
4	Compatible con sistema ROS Ubuntu 16.04
5	Indicador de nivel de batería
6	Regulación de tensión
7	Protección contra cortocircuitos
8	Modular

Una vez identificados los requerimientos que debe cumplir el subsistema electrónico, estos deben ser asociados con una variable que permitan su evaluación, para poder

determinar el grado de cumplimiento de los objetivos. Las métricas establecidas para los requerimientos planteados se presentan en la tabla 15. Además, se identifica el grado de importancia de las métricas asociadas al diseño del subsistema electrónico, las cuales, son tomadas en consideración para el desarrollo de la plataforma, midiendo la importancia de la función en una escala del 1 al 5, siendo 5 el grado de mayor importancia.

Tabla 15.

Listado de métricas. Subsistema electrónico.

Ítem	Requerimiento	Métrica	Importancia	Unidades
1	2,8	Mantenimiento fácil	3	Subjetivo
2	2,8	Modularidad	4	Subjetivo
3	3	Costo reducido	3	Dólares
4	4	Compatibilidad con sistema operativo	4	Subjetivo
5	1,2,6,7	Autonomía energética	5	Ah

3.4.1. Componentes del subsistema electrónico

El diseño del subsistema electrónico incluye una serie de etapas que se describen en la figura 30, y las cuales se detallan a continuación.

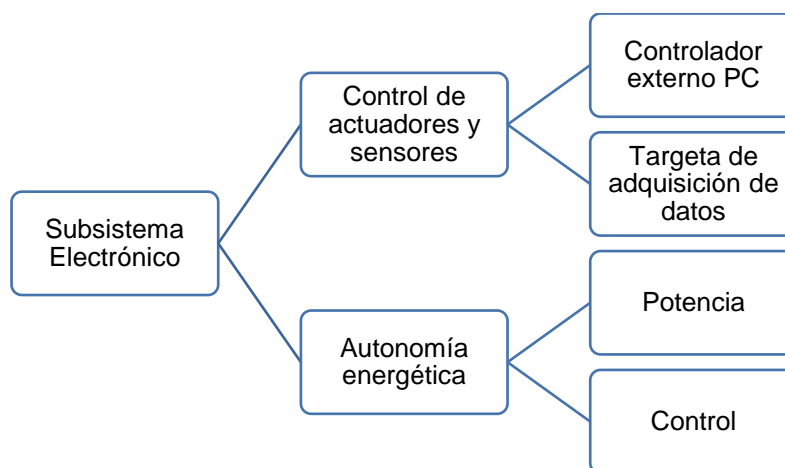


Figura 30. Componentes del subsistema electrónico

3.4.2. Control de actuadores y sensores

A partir de los requerimientos de diseño para el subsistema electrónico a continuación se detalla el criterio para la selección del controlador PC con sus respectivas ponderaciones, presentados en la tabla 16 y los conceptos de selección, presentados en la tabla 17.

Tabla 16.
Criterio para la selección del controlador PC

	Costo	Disponibilidad	Procesamiento	Tamaño	E+1	Ponderación
Costo	--	0.8	0.8	0.3	2.9	0.29
Disponibilidad	0.2	--	0.3	0.5	2	0.2
Procesamiento	0.2	0.7	--	0.3	2.2	0.22
Tamaño	0.7	0.5	0.7	--	2.9	0.29
			TOTAL		10	1

Tabla 17.
Conceptos para la selección del controlador PC




Concepto	Descripción	Características	Imagen
1	Jetson NANO developer kit	<ul style="list-style-type: none"> -GPU 128 CUDA Core Maxwell -CPU quad-core ARM A57 -Memoria 4GB LPDDR4 - 70 x 45 mm -Alimentación Micro USB 5v 2^a -Peso 0.14Kg 	
2	Intel NUC D54250WYKH	<ul style="list-style-type: none"> -Procesador Intel Core i5 -4Gb RAM -Intel HD Graphics 5000 -Tamaño 101.6mmx101.6mm -Alimentación 12-19v, 65w -Peso 0.67Kg 	
3	Computador personal Sony Vaio E series	<ul style="list-style-type: none"> -Procesador Intel Core i5 -6Gb RAM -Intel HD Graphics 4000 -Tamaño 372.1mmx253.74mm -Alimentación 19.5v 4.7A -Peso 2.7kg 	

Tabla 18.*Matriz de decisión para la selección del controlador PC*

Concepto	Costo			Disponibilidad			Procesamiento			Tamaño			TOTAL
	P	Cr	T	P	Cr	T	P	Cr	T	P	Cr	T	
Concepto 1	0.29	7	2.03	0.2	2	0.4	0.22	7	1.54	0.29	8	2.32	6.29
Concepto 2	0.29	3	0.87	0.2	9	1.8	0.22	7	1.54	0.29	8	2.32	6.53
Concepto 3	0.29	2	0.58	0.2	9	1.8	0.22	8	1.76	0.29	5	1.45	5.59

Una vez analizados los tres conceptos propuestos, se elige el controlador PC Intel NUC; como solución, es la que mejor se adapta al proyecto.

Para el proceso de adquisición de datos se hace uso de una placa Arduino Mega, debido a que el costo reducido, tamaño, disponibilidad en el mercado, procesamiento y compatibilidad lógica con sensores y drivers disponibles, hace que sea la opción más viable para el proyecto.

Tomando en cuenta las características del controlador PC y la placa de adquisición de datos y en base a la disponibilidad de componentes, se seleccionan los elementos electrónicos presentados en la tabla 19.

Tabla 19.
Elementos del subsistema electrónico

Componente	Especificación	Detalle
Arduino Mega	<ul style="list-style-type: none"> -Microcontrolador: ATmega 2560 -Voltaje Lógico: 5V -Voltaje de entrada 5-12V -Pines digitales E/S: 54 (15 PWM) -Pines analógicos S: 16 -Corriente por PIN E/S: 40mA 	
Monster Motor Shield	<ul style="list-style-type: none"> -Controlador dual -Voltaje lógico: 5V -Voltaje de motor: 5.5 – 16V -Corriente continua: 14A por canal -Protección térmica y sobrevoltaje. 	
Sensor IMU	<ul style="list-style-type: none"> -Chip: MPU6050 -Voltaje lógico 3.3 y 5V -Protocolo de comunicación: I2C -Sensores embebidos: acelerómetro, giroscopio, temperatura 	
Encoder	<ul style="list-style-type: none"> -Tipo: cuadratura -Sensor: Magnético (Efecto Hall) -Ciclos por revolución: 322 -Voltaje lógico: 2.4-26V -Salida amplitud de pulso: proporcional a la entrada lógica 	
RPLidar A1	<ul style="list-style-type: none"> -Rango de Medida: 0.15-12m -Rango angular: 360 grados -Frecuencia de escaneo: 5.5Hz -Voltaje lógico: 5V -Soporte para Linux 	
Kinect for Windows V1 modelo 1517	<ul style="list-style-type: none"> -Rango de trabajo: 0.8-3.5m -Angulo de visión: 70° horizontal y 60° vertical -Resolución: 640x480 píxeles -Voltaje lógico: 5v -Voltaje externo: 12v, 1.2A 	

3.4.3. Autonomía Energética

El consumo energético de los motores se realiza en base a los cálculos analizados en el diseño mecánico para selección de actuadores, analizando el escenario más

crítico en el cuál se obtuvo un torque igual a 295.42 oz-in o 2.08 Nm y una velocidad máxima de traslación igual a 0.5 m/s para motores DC de 12V con caja reductora planetaria y ruedas estándar de 0.0762m de radio. A partir de la fórmula presentada en la ecuación 18.

$$I = \frac{V_{max} T_{motor}}{R V_{dc}} \quad (18)$$

$$I = \frac{0.5[m/s]2.08[Nm]}{0.0762[m] 12 [Nm/As]}$$

$$I = 1.13[A]$$

La autonomía energética de la plataforma consta de dos partes una de potencia y otra de control, las cuales se detallan en la tabla 20 y 21, respectivamente; tomando en cuenta que no se debe descargar las baterías por completo se opta por considerar un factor de seguridad del 10% del consumo total de corriente calculado.

Tabla 20.
Consumo Energético de potencia

Potencia		
Elemento	Cantidad	Consumo energético[A]
Motor DC	2	2.26
Kinect	1	1.2
	SUMA	3.46
	FACTOR 10%	0.336
	TOTAL	3.796

Tabla 21.
Consumo Energético de control

Control		
Elemento	Cantidad	Consumo energético [A]
NUC	1	5.4
Arduino	1	0.2
LIDAR	1	0.6
	SUMA	6.2
	FACTOR 10%	0.62
	TOTAL	6.82

Tomando en cuenta los datos obtenidos del consumo energético de potencia y control y en base a la disponibilidad de componentes, se seleccionan los elementos electrónicos presentados en la tabla 22.

Tabla 22.
Elementos de consumo energético

Componente	Especificación	Detalle
Batería recargable	-Voltaje de salida : 12Vdc -Autonomía : 7ah -Peso: 2.2 Kg	
Convertidor DC-DC	-Voltaje de entrada: 10-32 Vdc -Voltaje de salida: 12Vdc -Corriente máxima: 6A	
Indicador de nivel de batería	-Baterías soportadas: plomo ácido 12v/24v/36v/48v/60v Indicador en porcentaje y voltios	

3.4.4. Diagrama de conexiones

El sistema electrónico consta de dos partes, una de potencia que alimenta a los motores y al Kinect y otra que se encarga de la parte de control alimentada con tensión regulada encargada de suministrar de energía al computador, que a su vez, alimenta a la tarjeta de adquisición de datos Arduino Mega, los sensores RPLidar A1 y la parte lógica del Kinect. Por otra parte, la tarjeta de adquisición de datos Arduino Mega suministra voltaje lógico y procesa los datos de los encoders y el IMU.

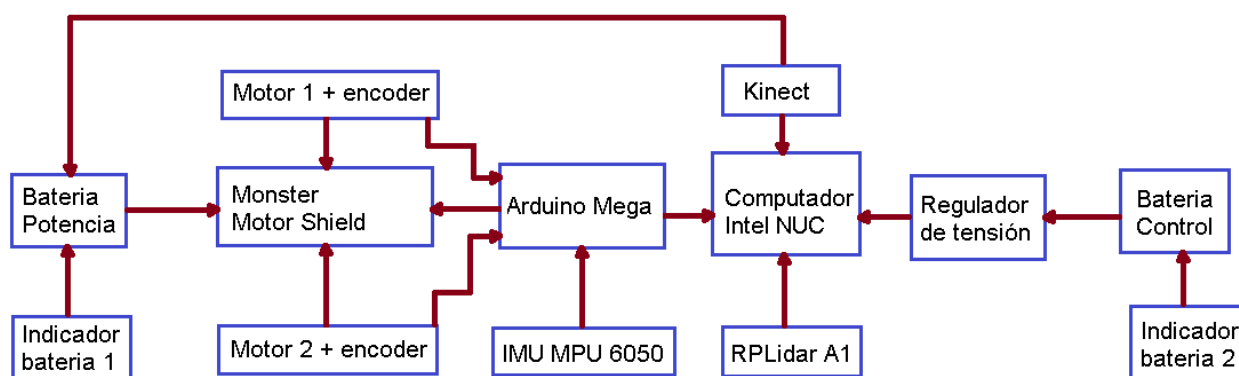


Figura 31. Diagrama de bloques subsistema electrónico.

3.5. Subsistema de Control

La identificación de requerimientos generales que debe cumplir el subsistema de Control para desarrollar un sistema robótico que cumpla los criterios de teleoperación, navegación autónoma e integración de un sistema de visión artificial, se presentan en la tabla 23.

Tabla 23.*Identificación de requerimientos. Subsistema de Control*

ítem	Requerimiento
1	Navegación en ambiente controlado (circuito de calles a escala donde el robot pueda navegar sin problemas)
2	velocidad máxima de funcionamiento 0.5m/s
3	Modelo virtual para fácil interpretación del sistema
4	Interpretación del sistema de gestos corporales
5	Interfaz humano-máquina

Una vez identificados los requerimientos que debe cumplir el subsistema de control, estos deben ser asociados con una variable que permitan su evaluación, para poder determinar el grado de cumplimientos de los objetivos.

Las métricas establecidas para los requerimientos planteados se presentan en la tabla 24. Además, se identifica el grado de importancia de las métricas asociadas al diseño del subsistema de control, las cuales, son tomadas en consideración para la implementación del sistema robótico, midiendo la importancia de la función en una escala del 1 al 5, siendo 5 el grado de mayor importancia.

Tabla 24.*Listado de métricas. Subsistema de Control*

Ítem	Requerimiento	Métrica	Importancia	Unidades
1	1	Modelo robótico	3	Subjetivo
2	1,2	Velocidad máxima	3	m/s
3	3,4,5	Comprensión de la interfaz robótica	4	Subjetivo

3.5.1. Componentes del subsistema de control

El diseño del subsistema de control incluye una serie de etapas que se describen en la figura 32, y las cuales se detallan a continuación.

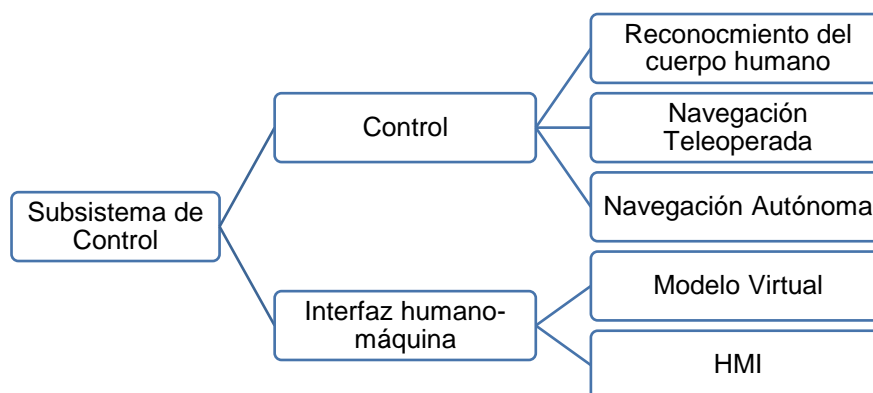


Figura 32. Componentes del subsistema de control

3.5.1.1. Reconocimiento de las juntas del cuerpo humano

El reconocimiento de características del cuerpo humano se realiza en forma de transformadas (tf), mediante el paquete de ROS “*skeleton_markers*” (Goebel, 2014), el cual utiliza la información de la cámara de profundidad del Kinect para rastrear y mostrar las juntas del cuerpo humano.

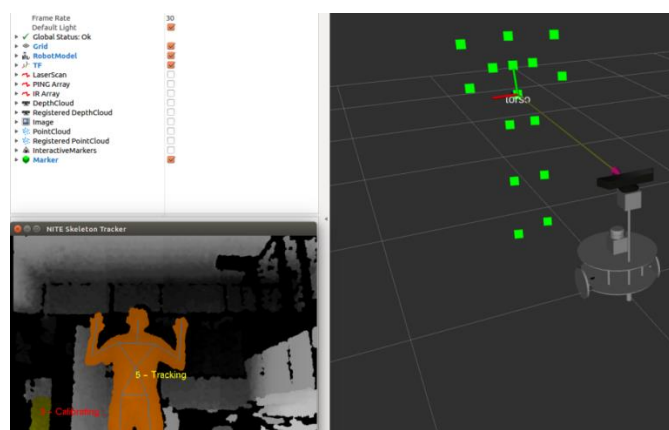


Figura 33. Pose Psi para detección de juntas del cuerpo humano

Como se indica en la figura 33, El proceso de rastreo de las juntas del cuerpo humano empieza una vez que la persona enfrente del Kinect realiza una pose conocida como “Psi”, y el proceso de calibración termine. Las juntas del cuerpo humano se

muestran en forma de un esqueleto, las cuales, serán reconocidas siempre y cuando estén dentro del área de visión del sensor Kinect.

3.5.1.2. Navegación de la plataforma robótica

El principio básico de funcionamiento de la plataforma robótica está basada en el paquete *"differential_drive"* desarrollado por (Stephan, 2012) y el proyecto realizado por (Lentin, 2018), el propósito de este paquete, basado en ROS, es proveer una interfaz para el control de robots con navegación diferencial. El diagrama de este paquete se muestra en la figura 34.

El nodo *"virtual_joystick"*, comanda los movimientos de robot usando un teclado.

El nodo *"twist_to_motors"*, traduce los comandos de teclado en información de velocidad lineal y angular para dos motores.

Los tópicos *"lwheel_vtarget"* y *"rwheel_vtarget"*, contienen la información de velocidad de la rueda izquierda y derecha respectivamente para ser utilizada por un PID.

El nodo *"pid_velocity"*, es un controlador PID para la velocidad de las ruedas.

Los tópicos *"lwheel"* y *"rwheel"*, publican la información de velocidad controlada, para ser interpretada por una tarjeta de adquisición de datos.

El nodo *"diff_tf"* es el encargado de generar la odometría del robot y genera una relación de transformadas (tf) entre la odometría del robot y la base del mismo.

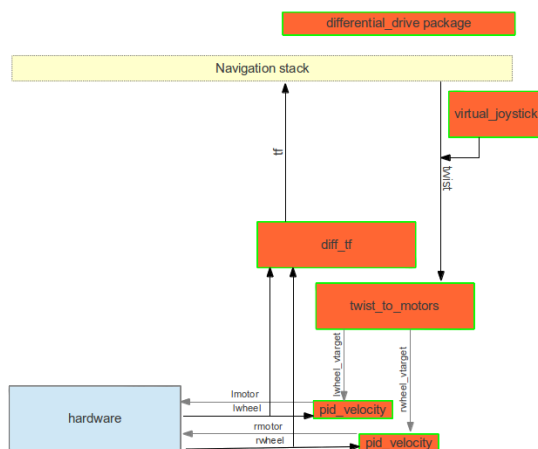


Figura 34. Nodos del paquete differential_drive ROS
Fuente: (Stephan, 2012)

3.5.1.3. Navegación Autónoma

El paquete de ROS “*move_base*”, desarrollado por (Marder-Eppstein, 2013), para plataformas móviles, puede generar una trayectoria para navegación autónoma mientras el robot evita obstáculos, siempre y cuando se de una estimación o pose inicial del robot en un mapa, lo cual se logra con el sistema de localización AMCL⁹, incorporado en el mismo.

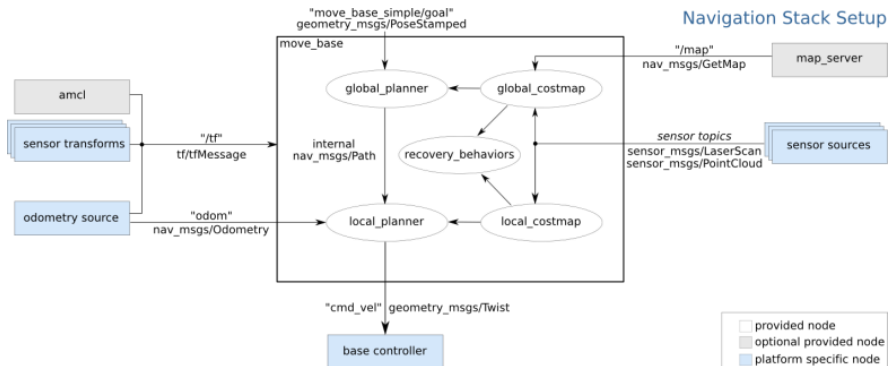


Figura 35. Descripción del paquete ROS move_base
Fuente: (Marder-Eppstein, 2013)

⁹ De las siglas Adaptive Monte Carlo Localization, sistema probabilístico de localización para robots en un mapa 2D.

El nodo “*move_base*”, incorpora algoritmos de planificación para realizar las tareas de navegación (“*global_planner*”, “*local_planner*”) y algoritmos que interpretan la información contenida en los sensores y en un mapa para reconocer obstáculos y los límites de navegación (“*global_costmap*”, “*local_costmap*”).

3.5.1.4. Modelo virtual de la plataforma robótica

ROS proporciona herramientas de descripción y modelamiento de robots, en un formato denominado URDF¹⁰, un paquete con documentos en formato XML¹¹, en los cuales se puede representar la descripción cinemática y dinámica, la representación visual y el modelo de colisión de las partes de un robot.

La descripción de las partes del robot consta de dos elementos, un eslabón (link) y un conjunto de uniones (joints), que conectan los eslabones.

Los eslabones (links), describen un cuerpo rígido con parámetros de inercia, forma visual y forma de colisión; mientras que las uniones (joints), especifican el tipo de unión entre links, los límites y propiedades dinámicas.

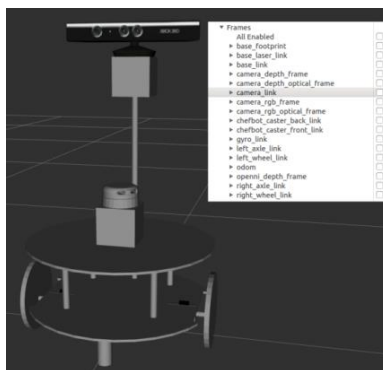


Figura 36. Modelo virtual de la plataforma robótica

¹⁰ De las siglas en inglés Unified Robot Description Format, paquete de archivos que describen el modelo virtual de un robot

¹¹ Lenguaje que define formatos aceptados por grupos para intercambiar información

La figura 36 indica el modelo virtual del robot y la figura 37 indica cada uno de los elementos visuales que lo conforman.

Para el desarrollo de los elementos virtuales que conforman el robot, se utilizó, como base el proyecto desarrollado por (Lentin, 2018) y el software de diseño Blender versión 2.79b, el cual permite exportar los archivos en formato DAE¹², el cual es un formato de archivo soportado por URDF.

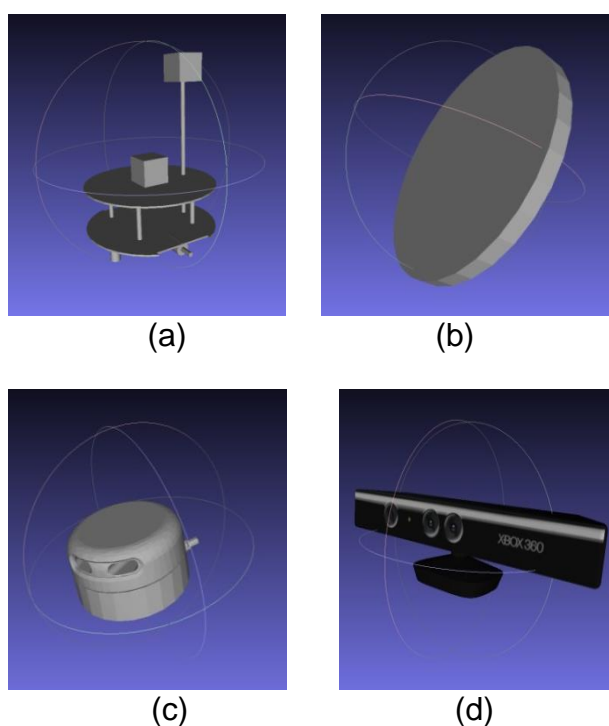


Figura 37. Modelo virtual del robot. (a) Base principal. (b) Rueda. (c) Sensor LIDAR. (d) Sensor Kinect.

3.5.1.5. Desarrollo de la interfaz gráfica de interacción humano-máquina

Para el desarrollo de la interfaz gráfica, se utilizó el software Qt Designer versión 4.8.7, para incorporar una ventana de navegación con botones que activan las funcionalidades del sistema robótico. Además, se utilizó la interfaz gráfica de ROS,

¹² Formato de archivo de intercambio 3D, utilizado por múltiples programas gráficos.

Rviz, la cual permite visualizar las funcionalidades del robot en un entorno 3D. Las diferentes ventanas de la interfaz gráfica se indican en la figura 38.

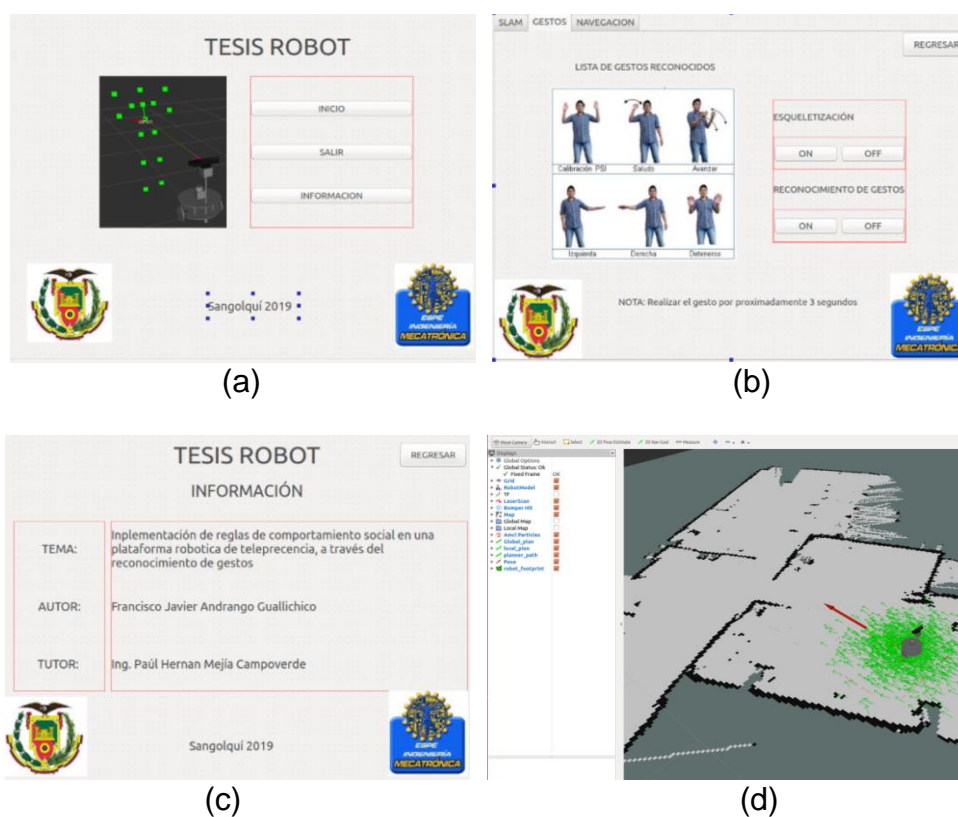


Figura 38. Interfaz gráfica de interacción humano-máquina. (a) Ventana principal. (b) Venta de opciones. (c) Ventana de información. (d) Rviz - ROS.

CAPÍTULO IV

PRUEBAS Y RESULTADOS

4.1. Gestos Corporales

Para el desarrollo de este estudio se hizo uso de gestos extrínsecos no emocionales, 2 instrucciones (saludo y avance) y 3 señales (izquierda, derecha y detenerse), cada uno mostrado en la figura 39.

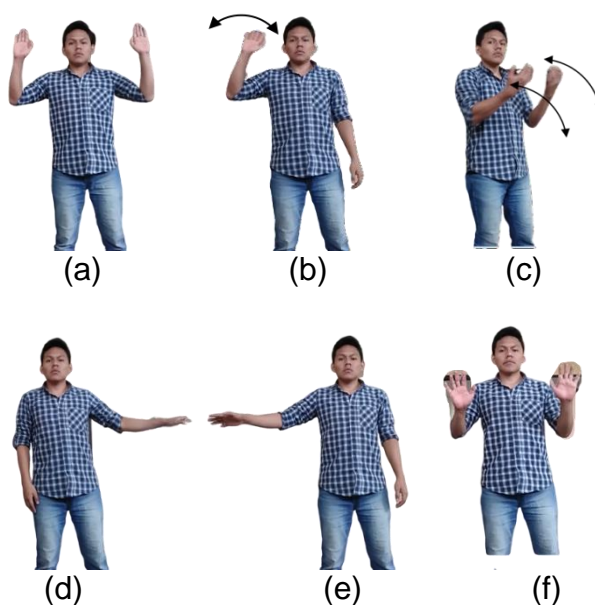


Figura 39. Gestos realizados con el cuerpo. (a) Calibración “PSI”. (b) Saludo. (c) Avanzar. (d) Izquierda. (e) Derecha. (f) Detenerse

Los gestos corporales propuestos utilizan la información de los dos brazos para realizar la acción pertinente, por lo tanto, para el proceso de recolección de datos se utilizó la información de orientación y profundidad de la mano y el codo en ambos casos.

4.1.1. Seguimiento de esqueleto

El sensor Kinect, se encuentra ubicado en la plataforma robótica a una distancia de 0.86m desde el suelo; altura desde la cual se puede reconocer a la persona sin problemas.

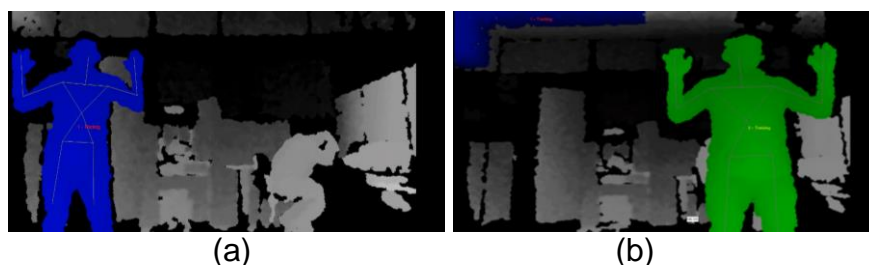


Figura 40. Determinación del límite para detección de esqueleto. (a) Límite de escena derecha. (b) Límite de escena derecha

Con el fin de delimitar la escena para la detección del esqueleto, como se indica en la figura 40, se realizaron varias pruebas de calibración a diferentes distancias, las cuales permitieron determinar el área óptima para reconocimiento de personas; los resultados se muestran en la tabla 25.

Tabla 25.
Distancia de detección usuario-robot

Ítem	Distancia	Escena		Rango Óptimo para reconocimiento
		Izquierda	Derecha	
1	0.5m	-	-	
2	1.50m	0.55m	0.55m	
3	2.50m	1.10m	1.10m	
4	3.50m	1.40m	1.40m	

4.1.2. Base de datos de gestos realizados

Como se puede observar en la figura 41, la información de cada gesto realizado se guardó en un archivo con formato CSV¹³, el cual contiene una matriz de 12x35, donde las filas representan la información de orientación (x, y) y profundidad (z) del gesto y las columnas representan la cantidad de datos registrados cada 0.1s, con un tiempo total de 3.5s.

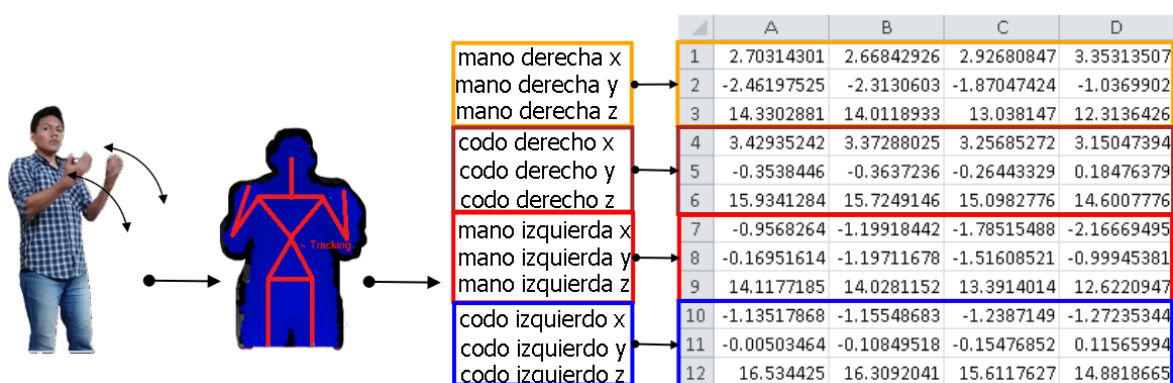


Figura 41. Base de datos de los gestos realizados

La recolección de información se realizó con 3 usuarios de diferentes estaturas, los cuales, realizaron cada uno de los gestos indicados en la tabla 26; proceso que se repitió durante 15 veces, con lo cual se obtuvo una base de datos de 45 archivos para cada gesto.

Tabla 26.
Base de datos de los gestos

Ítem	Usuarios		Numero de gestos realizados				
	Estatura (m)	Saludo	Avanzar	Izquierda	Derecha	Detenerse	Pausa
1	1.67	15	15	15	15	15	15
2	1.72	15	15	15	15	15	15
3	1.83	15	15	15	15	15	15
SUBTOTAL		45	45	45	45	45	45

¹³ Documento de formato abierto para representar datos en forma de tabla.

4.1.3. Red Neuronal Pre alimentada

Para entrenar la base de datos recolectada, se utilizó el paquete para ROS “*neuro_gesture_kinect*”, desarrollado por (Parhar, 2015), el cual utiliza una red neuronal pre alimentada y un algoritmo de retro propagación para el entrenamiento de la red.

El algoritmo original fue modificado para reconocer las juntas de los 2 brazos.

4.1.4. Predicción de Gestos

Con el propósito de evaluar el desempeño de la Red Neuronal, se realizó una matriz de confusión, a partir del conteo de aciertos y errores en el proceso de predicción de los gestos.

El proceso se realizó con 40 secuencias de observaciones para cada gesto registrado, como se indica en la figura 42, analizando un escenario donde la persona se mueve dentro del campo de visión del robot a medida que realiza los gestos.

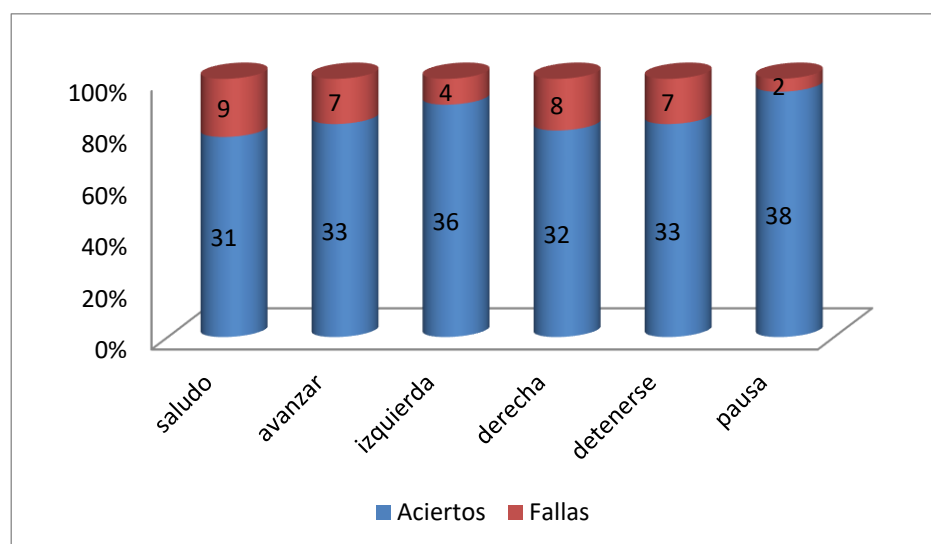


Figura 42. Total de aciertos y fallas en las observaciones

Los resultados de la matriz de confusión se muestran en la tabla 27 y en la tabla 28 se muestra el desempeño de la Red Neuronal basada en la matriz de confusión.

Tabla 27.
Matriz de confusión

		Predicciones					
		Saludo	Avanzar	Izquierda	Derecha	Detenerse	Pausa
Observaciones	Saludo	31	0	0	9	0	0
	Avanzar	0	33	2	1	4	0
	Izquierda	0	0	36	1	0	3
	Derecha	6	0	1	32	0	1
	Detenerse	4	0	0	1	33	2
	Pausa	0	0	1	1	0	38

Tabla 28.
Desempeño de la Red Neuronal

Desempeño	Porcentaje (%)
Exactitud	84.58
Razón de verdaderos positivos, Saludo	77.5
Razón de verdaderos positivos, Avanzar	82.5
Razón de verdaderos positivos, Izquierda	90
Razón de verdaderos positivos, Derecha	80
Razón de verdaderos positivos, Detenerse	82.5
Razón de verdaderos positivos, Pausa	95

La exactitud de los datos se representa por la proporción del número de predicciones que fueron correctas y el total de las predicciones.

La razón de verdaderos positivos, es la proporción de casos positivos correctamente identificados.

La experimentación indica que la predicción de los gestos cumple con una exactitud de 84.58% cuando una persona realiza los gestos.

Los gesto de saludo y derecha, tienen el porcentaje de predicción más bajo con un 77.5%, 80%, respectivamente, debido a la naturaleza de los gestos, estos tienden a confundirse entre sí.

En la experimentación se incluyó un gesto de pausa, que reconoce a una persona parada sin movimiento en sus brazos, tiene una predicción del 95%, ya que no guarda ninguna relación con las otras observaciones.

4.2. SLAM

El algoritmo seleccionado para el desarrollo de SLAM, es Gmapping; según (Gonçalo dos Santos Martins, 2014), dicho algoritmo es apropiado para utilizarse con los más bajos requerimientos computacionales y con buenos resultados, por lo cual es utilizado en procesadores con memoria RAM desde 1GB.

4.2.1. Elaboración de mapas

Para configurar los parámetros del algoritmo Gmapping, se utilizó el Laboratorio de Mecatrónica y Sistemas Dinámicos de la Universidad de las Fuerzas Armadas-ESPE, como entorno de prueba.



Figura 43. Laboratorio de Mecatrónica y Sistemas Dinámicos

La figura 43, muestra el proceso de mapeo que utiliza la información del sensor RPLidar A1, la odometría y el control del robot mediante teleoperación; las rejillas marcadas de color negro muestran los obstáculos estáticos detectados y las zonas libres se muestran en color gris.

A medida que el robot navega por el entorno se registran los datos que corresponden al mapa del lugar.

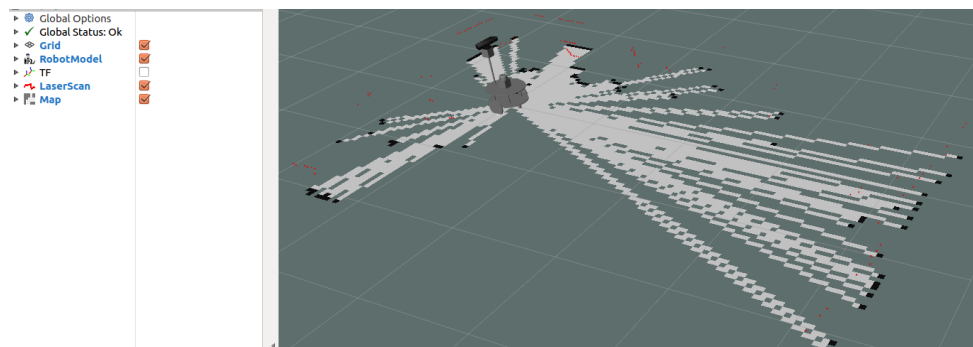


Figura 44. Proceso de mapeo con Gmapping

La figura 45 muestra los resultados de los mapas realizados del entorno; a medida que se redujo el número de muestras (partículas) es más visible la detección de zonas libre marcadas en color blanco y las ocupadas, que corresponden a las paredes e inmuebles, en color negro; además, se requiere menor tiempo de navegación para obtener el resultado.

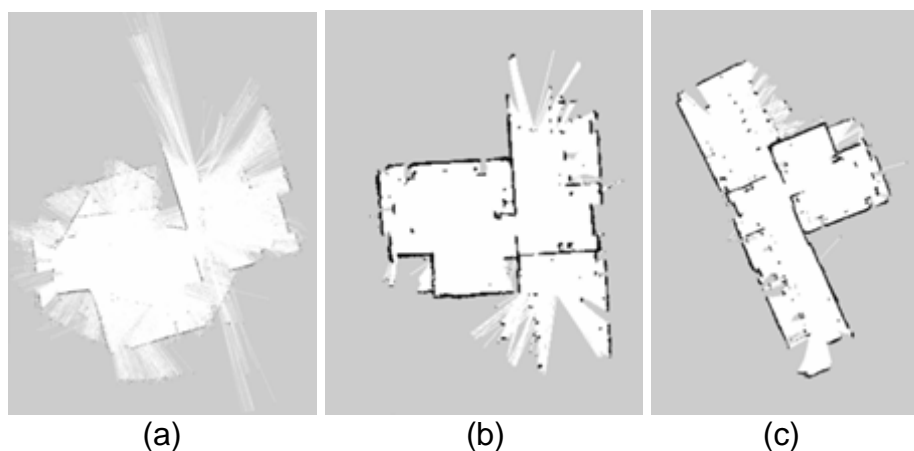


Figura 45. Configuración parámetros Gmapping ROS. (a) Mapa con 80 partículas (b) Mapa con 30 partículas (c) Mapa con 8 partículas

Tabla 29.*Configuración de parámetros para mapeo*

Ítem	Numero de Partículas	Velocidad
1	80	0.5 m/s
2	30	0.35 m/s
3	8	0.26 m/s

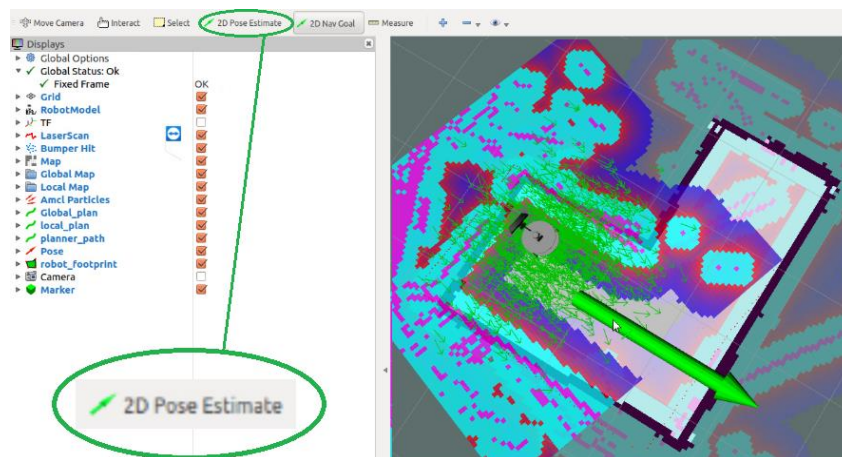
Como entorno de pruebas para posteriores pruebas de navegación autónoma, se optó por desarrollar un circuito de calle cerrado donde el robot pueda navegar, como se muestra en la figura 46(a) y su correspondiente mapa indicado en la figura 46(b).



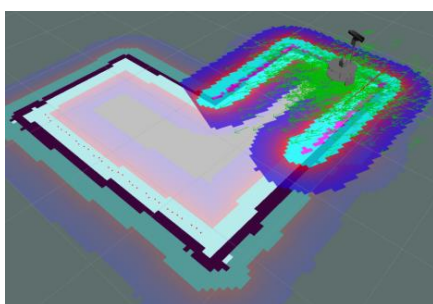
Figura 46. Circuito de calle cerrado, como medio de pruebas. (a) Entorno real.
(b) Mapa generado.

4.3. Navegación Autónoma

El primer paso a realizar antes de ejecutar las tareas de navegación autónoma, es la ubicación del modelo de robot en el mapa, para ello se hace uso de la herramienta gráfica Rviz de ROS y usando el botón “2D Pose Estimate”, como se observa en la figura 47(a), se ubica al robot en un punto de partida.



(a)



(b)



(c)

Figura 47. Estimación de pose inicial. (a) Estimación de Pose. (b) Localización en el entorno virtual. (c) Localización real.

Una vez estimada la posición inicial del robot en el mapa se debe usar el botón “2D Nav Goal”, para ordenarle al robot dirigirse a una posición indicada.

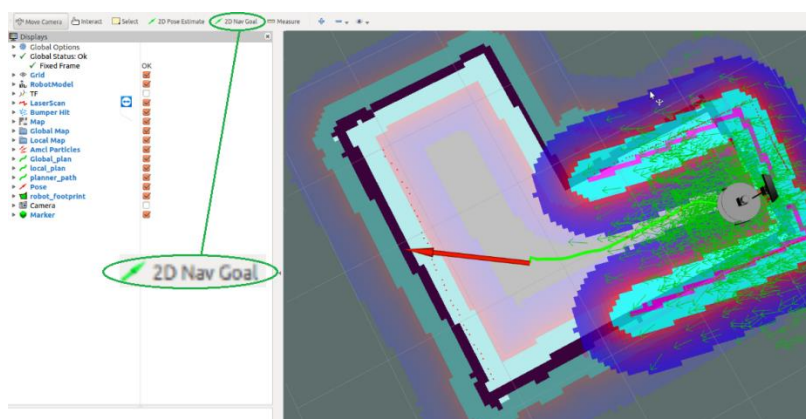
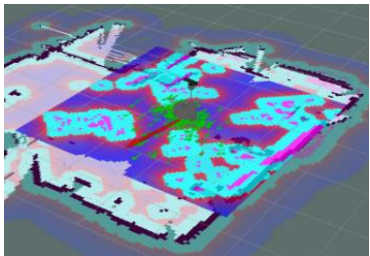

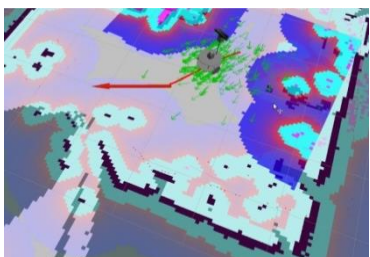

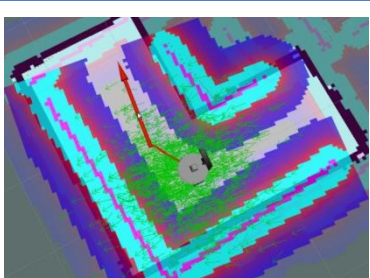
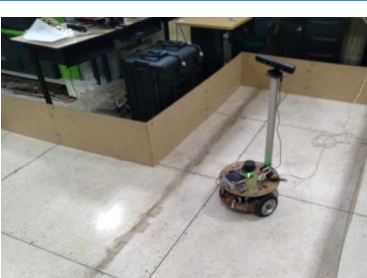
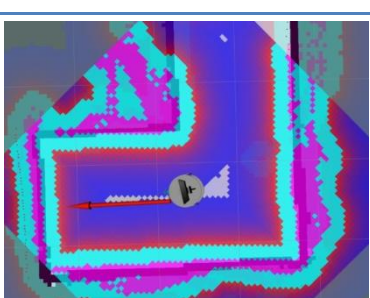



Figura 48. Estimación de trayectoria

En la tabla 30 se indica los experimentos de navegación que llevan al robot de un lugar a otro; en la columna izquierda se observa la ubicación del robot en el mapa previamente creado. En la columna derecha se indica la posición real del robot. La trayectoria que el robot sigue se indica en color verde. El punto de llegada y la orientación esta representada por la flecha roja.

Tabla 30.
Pruebas de navegación

Mapa	Real	Observación
		<p>Mapa de laboratorio. Entorno de pruebas de complicado reconocimiento, debido a los obstáculos presentes en el mapa. De las 3 pruebas realizadas solo una genero una trayectoria hacia el punto de destino. Distancia recorrida 1.2m</p>
		<p>Mapa de laboratorio. En este caso se generó una trayectoria en un área más amplia Distancia recorrida de 1.5m sin obstáculos presentes en su camino. Otros ensayos realizados presentaron pérdida total de la trayectoria.</p>
		<p>Circuito de calle cerrado. Entorno de prueba de simple reconocimiento, sin obstáculos. Se intentó generar una trayectoria relativamente grande para la navegación, sin embargo, el robot rotó varias veces en el área de escaneo sin lograr generar una ruta a su destino.</p>
		<p>Circuito de calle cerrado. Se realizaron varias pruebas con trayectos más cortos hasta alcanzar Un punto más lejano en el mapa.</p>

Al momento de realizar los experimentos presentados en la tabla 32, se encontraron ciertas dificultades, entre las cuales se mencionan:

- La velocidad varía en diferentes tramos del trayecto afectando al correcto desplazamiento del robot, debido a irregularidades en el piso y suciedad en ciertos momentos el robot patina o la velocidad es insuficiente para mover la plataforma.
- El robot puede abandonar el objetivo aunque cuenta con una herramienta de recuperación que lo hace rotar 360 grados, en ocasiones se agotarán todos los recursos disponibles para recuperar información y simplemente el robot se detendrá.
- El objetivo principal de realizar un circuito cerrado de calle, se debe principalmente a que el robot puede perder la trayectoria por completo si el área de navegación es muy compleja.

CAPÍTULO V

CONCLUSIONES, RECOMENDACIONES Y TRABAJOS FUTUROS

5.1. Introducción

El capítulo 5 presenta las conclusiones y recomendaciones obtenidas al momento de realizar esta investigación, además, se presentan propuestas de trabajos de titulación que se podrían implementar con el sistema robótico desarrollado.

5.2. Conclusiones

- Se diseñó y construyó una plataforma robótica de configuración diferencial de forma circular, con un diámetro de 0.406m y una altura de 0.85m.

Para lograr una mejor distribución de masas, la cual afecta directamente al equilibrio del robot, se opta por usar dos ruedas de apoyo o ruedas locas y mantener el eje de rotación en el centro de la plataforma.

- El sistema robótico desarrollado se integró con ROS, convirtiéndose en un sistema modular, que puede servir para realizar pruebas experimentales de robótica social que involucren tareas de navegación y visión artificial, con la posibilidad de expandir sus recursos de hardware y software para mejorar su desempeño.
- Se implementó un sistema de navegación con mando manual, para realizar mapeo utilizando Gmapping; al final de las pruebas la velocidad óptima de navegación fue de 0.26m/s y con el fin de reducir el coste computacional, el número de partículas procesadas para realizar un mapa se redujo a 8.

- Se implementó un sistema de navegación autónoma, utilizando el algoritmo AMCL para localizar al robot en un mapa conocido y el paquete *“move_base”*, que ayuda al robot a realizar tareas de navegación y evasión de obstáculos a medida que se acerca a un punto de destino.
- Un problema encontrado durante la navegación autónoma, ocurre cuando se intenta generar trayectorias muy largas y el entorno de navegación es muy complejo. La solución a este problema fue realizar un circuito de calle cerrado, y generar rutas cortas de navegación hasta alcanzar el objetivo final que se encuentra relativamente lejos.
- Se implementó un sistema de visión artificial para el reconocimiento de gestos realizados con los dos brazos, utilizando una base de datos de 45 observaciones por gesto con tres usuarios de prueba. Para la predicción de los gestos se utilizó una red neuronal pre alimentada y un algoritmo de retro propagación para el entrenamiento, resultando en una predicción de gestos del 84.58%, por lo cual, se concluye que el método utilizado para la predicción cumple su objetivo.
- Los gestos de saludo y derecha tienen el porcentaje más bajo de predicción con 77.5% y 80% respectivamente, con una tendencia a confundirse entre ellos, ya que ambos se realizan con el mismo brazo; a pesar de esto los porcentajes de predicción siguen siendo altos, en conclusión, si se desea mejorar los resultados se puede utilizar una base de datos mayor.
- Se desarrolló un sistema gráfico para la interacción hombre-máquina, que involucra el modelo virtual del robot y un modelo de esqueleto de la persona

reconocida; además, se incorporó una ventana de navegación con botones que activan las funcionalidades del sistema robótico dividido en 4 partes: una subventana de inicio que es una presentación del robot, una subventana de opciones que permite navegar entre las funcionalidades del robot y una subventana de información que indica el autor y el tema de tesis.

5.3. Recomendaciones

- Mejorar la navegación del robot agregando sensores de seguridad para detección de obstáculos pequeños y desniveles en el piso.
- Verificar compatibilidad de computadores diseñados para tareas de machine learning (Jetson Nano, Raspberry 4, etc.) y el Kinect, con el fin de optimizar recursos energéticos y de procesamiento.
- Por el momento el sistema de visión artificial predice 6 gestos (saludo, avanzar, izquierda, derecha, pausa). Se recomienda ampliar el número de gestos y la base de datos de los mismos con el fin de obtener un modelo más robusto que sirva en otras tareas aplicadas en robótica social.
- Incorporar una pantalla en el sistema robótico, con el fin de mejorar la interacción humano-máquina.

5.4. Trabajos futuros

- **Optimización e implementación de algoritmos en una plataforma de telepresencia completa.**

El sistema robótico desarrollado cumple con las características básicas para un robot de telepresencia al tener un sistema de navegación autónomo limitado a un

medio controlado y un sistema de visión artificial para reconocer gestos, por lo cual, se propone el desarrollo de una plataforma robótica robusta de telepresencia, que incluya sensores de protección, para ser probada en ambientes que demanden un mayor control.

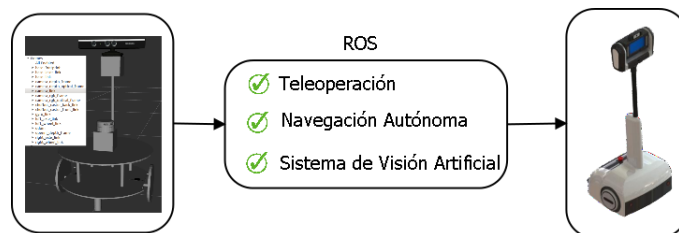


Figura 49. Propuesta: Plataforma de telepresencia robusta

- **Interpretación de gestos registrados.**

El siguiente paso, después del reconocimiento de gestos, es la interpretación de los mismos por parte del sistema robótico. Por lo cual, se propone la implementación de algoritmos de control que permitan a la plataforma robótica seguir a una persona objetivo y detenerse cuando esta realizase los gestos correspondientes, en una escena con obstáculos estáticos y dinámicos.

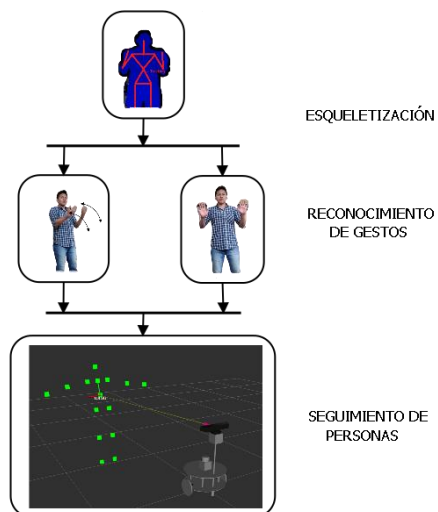


Figura 50. Propuesta: Sistema para seguimiento de Personas.

BIBLIOGRAFÍA

- Bao Xin Chen, Sahdev, R., & Tsotsos, J. (2017, mayo 17-19). Person following robot using selected Online Ada-Boosting with stereo camera .
- Bechtel, M., McElhiney, E., Kim, M., & Yun, H. (2017, Diciembre). DeepPicar: A Low-cost Deep Neural Network-based Autonomous Car.
- Bereciartua, A. (2016). Desarrollo de algoritmos de procesamiento de imagen avanzado para interpretación de imágenes médicas. Bilbao.
- Campana, R. (2011). People Detection and Tracking with Kinect for Mobile Platforms. Italia.
- Castillo, P., Lozano, R., & Dzul, A. (2006). Modelling and Control of Mini-Flying Machines. Springer.
- Cobeña Zambrano, G. (2018). Reconocimiento de gestos corporales basado en svm y sensor rgbd para el control de micro vehículos aéreos multirrotor. Sangolquí, Ecuador.
- Córdova L., F. E., & Medina T., S. A. (2018). Diseño y construcción del prototipo de un robot móvil para telepresencia controlado a través de internet. Sangolquí, Ecuador.
- Dent, S. (2014, Diciembre 31). Engadget. Retrieved from <https://www.engadget.com/>
- Días Alabart, S. (2018). Robots y responsabilidad social. Madrid: Reus.
- Ferguson, M., & Law, K. (2019, Enero). A 2D-3D Object Detection System for Updating Building Information Models with Mobile Robots.

- Ferrer, R. F. (2014). Implementación de algoritmo slam basado en sensor láser hokuyo 04lx - ug01. Tesis, Escuela Técnica Superior Ingenieros Industriales Valencia, Valencia.
- Gallardo López, D. (1999, Mayo). Aplicación del muestreo bayesiano en robots móviles: estrategias para localización y estimación de mapas del entorno.
- German Association of Engineers. (2004). Design methodology for mechatronic systems.
- Goebel, P. (2014). skeleton_markers. Retrieved from http://wiki.ros.org/skeleton_markers
- Gonçalo dos Santos Martins. (2014). A Cooperative SLAM Framework with Efficient Information Sharing over Mobile Ad Hoc Networks. Universidad de Coimbra.
- Ibáñez. (2017, Septiembre 28). Qué es un LIDAR, y cómo funciona el sensor más caro de los coches autónomos. Retrieved from www.motorpasion.com
- Jorquera Guillen, P., & Espinoza Olguín, D. (2015, Junio). Reconocimiento Facial.
- Konar, A., & Saha, S. (2018). Gesture Recognition. Principles, Techniques and Applications.
- Lentin, J. (2018). Learning Robotics using Python. Packt Publishing.
- López Monroy, A. P., & Leal Meléndrez, J. A. (2011). Reconocimiento de gestos basados en modelos ocultos de markov utilizando el kinect. Puebla, México.
- López Monroy, A., & Leal Meléndrez, J. (2011). Reconocimineto de Gestos basados en Modelos Ocultos de Marcov utilizando el Kinect. Puebla, México.

- Lopez, R. (2017, junio sábado). Raul E. Lopez Briega Matemáticas, análisis de datos y python. Retrieved from www.relopezbriega.github.io/blog/2017/06/10/boosting-en-machine-learning-con-python/
- Marder-Eppstein, E. (2013, Enero 11). move_base. Retrieved from http://wiki.ros.org/move_base
- Martín del Brío, B., & Sanz, A. (2001). Redes Neuronales y Sistemas Difusos. Zaragoza: Alfaomega.
- Mathworks. (1994-2019). BNO055 IMU Sensor. Retrieved from www.mathworks.com/help/supportpkg/arduino/ref/bno055imusensor.html
- MathWorks. (2019). La.mathworks.com. Retrieved from <https://la.mathworks.com/discovery/machine-learning.html>
- Microsoft Corporation. (2014). Kinect for Windows. Human Interface Guidelines.
- Ollero Baturone, A. (2005). Robótica, manipuladores y robots móviles. Marcombo.
- Parhar, T. (2015, Julio 08). Neuro Gesture- For Kinect. Retrieved from http://wiki.ros.org/TanvirParhar/neuro_gesture_kinect
- Pinter, M., Lai, F., Sanchez, D., Ballantyne, J., Roe, D., Wang, Y., . . . Wong, C. (2015). Patent No. 9174342. Estados Unidos.
- Pokharna, H. (2016, Julio 26). <https://medium.com/>. Retrieved from <https://medium.com/technologymadeeasy/for-dummies-the-introduction-to-neural-networks-we-all-need-c50f6012d5eb>
- Prime Sense Inc. (2010). Prime Sensor Nite 1.3 Algorithms notes.
- Priyandoko, G., Wei, C., & Achmad, M. S. (2018, Junio). Human Following on ROS framework a mobile robot.

- Quinaluisa M., E. F., & Toapanta C., J. A. (2018). Implementación de un sistema de navegación autónomo basado en SLAM y navegación reactiva. Sangolquí, Ecuador.
- Rhoebly Dynamics. (2015-2019). Rhoebly Dynamics,. Retrieved from www.rhoebly.com
- RobotShop Community. (2014, 02 01). RobotShop. Retrieved from <https://www.robotshop.com/community/blog/show/drive-motor-sizing-tutorial>
- ROS.org. (2019, febrero 4). gmapping. Retrieved from <http://wiki.ros.org/gmapping>
- Sepimex. (2012, Abril 16). Blog Pepimex Psicología de alto nivel. Retrieved from <https://sepimex.wordpress.com/2012/04/16/que-es-la-ansiedad/>
- Siegwart, R., & Nourbakhsh, I. R. (2004). Introduction to Autonomous Mobile Robots. Londres, Inglaterra: Brandford Books.
- Silva Ortigoza, R., García Sánchez, J., Barrientos Sotelo, V., & Molina Vilchis, M. (2007). Una panorámica de los robots móviles. Retrieved from <https://www.redalyc.org/articulo.oa?id=78460301>
- Slamtec. (2013-2016). RPLIDAR A1 Low Cost 360 Degree Laser Range Scanner Introduction and Datasheet. Retrieved from www.slamtec.com/en/lidar/a1
- SoftBank Robotics. (2019). Softbank robotics. Retrieved from <https://www.softbankrobotics.com/us/pepper>
- Stephan, J. (2012). differential_drive. Retrieved from http://wiki.ros.org/differential_drive
- Strezoski, G., Stojanovaki, D., Dimitrovaki, I., & Madjarov, G. (2016, Septiembre). Hand gesture recognition using deep convolutional neural networks.
- Tran, T.-H., Le, T.-L., Hoang, V.-N., & Vu, H. (2017). Continuous detection of human fall using multimodal features from Kinect sensors in scalable environment.

Ulrich, K., & Eppinger, S. (2013). Diseño y desarrollo de productos. Mexico: Mc Graw Hill.

Velásquez, C., Chávez, J., & Córdova, E. (2015). Implementación de sistema de navegación autónomo en robot móvil experimental para reconstrucción y exploración de entornos desconocidos. Bogotá, Colombia.

ANEXOS