



TECNOLÓGICO NACIONAL DE MÉXICO

Instituto Tecnológico Superior de Teziutlán

“Desarrollo de robot para movilidad autónoma de estudiantes discapacitados del Instituto Tecnológico Superior de Teziutlán”

TESIS QUE PRESENTA:

Luis Ángel Martínez Cabrera

Como requisito parcial para obtener el título de:

MAESTRO EN SISTEMAS COMPUTACIONALES



Fracción I y II S/N, Aire Libre, Teziutlán, Puebla.
Julio de 2020

La presente tesis titulada: **Desarrollo de robot para movilidad autónoma de estudiantes discapacitados del Instituto Tecnológico Superior de Teziutlán**, fue realizada bajo la dirección del comité de asesores indicado, ha sido aprobada por el mismo y aceptada como requisito parcial para obtener el título de:

MAESTRO EN SISTEMAS COMPUTACIONALES

DIRECTOR:

M.S.C. ADRIANA PÉREZ LÓPEZ

1er. CO-DIRECTOR:

M.S.C. HÉCTOR VICENTEÑO RIVERA

2o. CO-DIRECTOR

MTRO. EMMANUEL VÁZQUEZ BENITO

Fracción I y II S/N, Aire Libre, Teziutlán, Puebla. Agosto de 2020.

AGRADECIMIENTOS

AL INSTITUTO TECNOLÓGICO SUPERIOR DE TEZIUTLÁN

Gracias al instituto tecnológico superior de Teziutlán, gracias por haberme permitido formarme y en ella, gracias a todas las personas que fueron partícipes de este proceso, ya sea de manera directa o indirecta, gracias a todos ustedes, fueron ustedes los responsables de realizar su pequeño aporte, que el día de hoy se vería reflejado en la culminación de mi paso por el instituto. Gracias a mis padres, que fueron mis mayores promotores durante este proceso, gracias a Dios, que fue mi principal apoyo y motivador para cada día continuar sin tirar la toalla.

Este es un momento muy especial que espero, perdurará en el tiempo, no solo en la mente de las personas a quienes agradecí, sino también a quienes invirtieron su tiempo para echarle una mirada a mi proyecto de tesis; a ellos asimismo les agradezco con todo mi ser.

A MIS ASESORES

Durante la realización de mi proyecto, usted ha sido mi mano derecha y quien me ha guiado en el complicado proceso. Es cierto, no ha sido nada fácil, ni mucho menos, sin embargo, gracias a su ayuda, esto ha parecido un tanto menos complicado.

El resultado de mi tesis ha sido espectacular, mejor de lo que esperaba y una gran parte del desarrollo de ese excelente trabajo se lo debo a usted, que dios la bendiga.

A MI ESPOSA

Tu ayuda ha sido fundamental, has estado conmigo incluso en los momentos más turbulentos. Este proyecto no fue fácil, pero estuviste motivándome y ayudándome hasta donde tus alcances lo permitían.

Te lo agradezco mucho amor.

A MIS PADRES

Las razones que acabo de expresar, dedico esta tesis a mis padres, doy gracias a Dios por permitirme disfrutar cada momento de mi vida al lado de mis padres, agradezco a la vida por permitirme tener tan excelentes padres, y de nuevo agradezco a Dios por hacerme el hijo más feliz de este mundo. Hoy concluye todo el esfuerzo y dedicación que demando la realización de esta tesis, pero el camino continúa por la vida, y dentro de este camino, tengo claridad y certeza de que mis padres me apoyarán y querrán lo mejor para mi hasta el último día de sus vidas.

Gracias, mamá y papá.

RESUMEN

La idea principal es desarrollar un sistema tecnológico con fusión de sensores y procesamiento paralelo aplicado a la navegación y movilidad autónoma de robots móviles, el cual pueda adaptarse a un “aparato ortopédico” que ayude a personas con capacidades diferentes en su desplazamiento. Se desarrolló un prototipo, que permite una movilidad estable y segura; implementando un middleware, un software de manipulación robótica de libre distribución, permitiendo el proceso de navegación. Se adquirió una tarjeta Raspberry pi 3B+, para ejecutar los grafos de navegación, así como sensores para que el dispositivo robótico tenga noción de su entorno, tal como los sentidos del ser humano. El proyecto que se emprendió busca una solución a través pruebas cuasi-experimentales, haciendo uso de elementos y pruebas existentes para lograr el objetivo planteado. El prototipo cuenta con una tarjeta Raspberry pi 3B+, dos motorreductores con encoder de efecto hall incremental, una tarjeta Arduino, un puente H, una batería LIPO así como un sensor laser RPLIDAR, para el mapeo en interiores y/o exteriores, se eligieron estos elementos por ser eficientes en el rendimiento. Inicialmente se desarrolló el diseño del prototipo en forma rectangular en un software CAD, (Corporation ©.-2. D., 2020) posteriormente se modificó el diseño del prototipo, adoptando la cinemática de movimiento de un robot diferencial en forma geométrica circular el cual tuvo mayor facilidad de movimiento e hizo más eficiente el cálculo de las trayectorias de navegación y procesamiento en el mapeo, el proceso de eficiencia se realizó comparándolo con un robot con sistema de control clásico.

ÍNDICE GENERAL

AGRADECIMIENTOS	3
RESUMEN.....	5
ÍNDICE GENERAL.....	6
ÍNDICE DE FIGURAS	8
ÍNDICE DE TABLAS.....	10
CARTAS DE ACEPTACIÓN.....	12
CAPÍTULO I.....	13
GENERALIDADES DEL PROYECTO	13
1.1. INTRODUCCIÓN.....	14
1.2. PLANTEAMIENTO DEL PROBLEMA	17
1.3. JUSTIFICACIÓN.....	18
1.4. HIPÓTESIS	18
1.5. OBJETIVOS.....	19
1.5.1. OBJETIVO GENERAL.....	19
1.5.2. OBJETIVOS ESPECÍFICOS	19
1.6. ALCANCES	20
1.7. LIMITACIONES	20
1.8. ESTADO DEL ARTE	21
1.8.1. Optimal Perception Planning with Informed Heuristics Constructed from Visibility Maps.....	21
1.8.2. Effective Team Coordination through Intra-Robot Replanning to Restore Team Plan Rationale	22
1.8.3. Heterogeneous Multi-Agent Planning Using Actuation Maps.....	23
1.8.4. Trajectory-Based Short-Sighted Probabilistic Planning.....	24
1.8.5. An Approach based on a Robotics Operating System for the Implementation of Integrated Intelligent House Services System	26
1.8.6. Sensor Fusion Used in Applications for Hand Rehabilitation: A Systematic Review	27
1.8.7. Detecting falling people by autonomous service robots: A ROS module integration approach	28

1.8.8. A Graph Representation Composed of Geometrical Components for Household Furniture Detection by Autonomous Mobile Robots.....	29
1.8.9. Realistic and Automatic Map Generator for Mobile Robots.....	29
CAPÍTULO II.....	31
METODOLOGÍA Y DESARROLLO.....	31
2.1. FUNDAMENTOS TEÓRICOS.....	32
2.1.1. Niveles de discapacidad.	32
2.1.2. Instituciones dedicadas a atender personas de capacidades diferentes.	35
2.1.3. Fusión de sensores y procesamiento paralelo.	37
2.1.4. Robótica.....	40
2.1.5. Middleware (ROS).....	42
2.1.5.1. Versiones de ROS.....	45
2.1.6. Raspberry	46
2.1.7. Tipos de sensores.....	53
2.1.7.1. Clasificación de los sensores	53
2.1.8. Motorreductores POLOLU	62
2.1.9. Tarjetas Arduino	67
2.1.10. Puente H y PWM.....	70
2.1.11. Comunicación serial.....	71
2.1.12. Baterías LIPO.	76
2.1.13. Dibujo Asistido por Computadora, CAD.	79
2.1.14. Sistemas de referencia vectorial.....	81
2.2. METODOLOGÍA DE LA INVESTIGACIÓN.....	87
2.2.1. TIPO DE INVESTIGACIÓN CUASI EXPERIMENTAL	87
2.2.2. DEFINICIÓN DE POBLACIÓN Y SELECCIÓN DE LA MUESTRA	90
2.3. METODOLOGÍA DE DESARROLLO	92
2.3.1. Diseñar el robot con software CAD	93
2.3.2. Control PID para la corrección de incertidumbre en la posición angular	98
2.3.3. Control de velocidad.	99
2.3.4. Tracción	100
2.3.5. Simulación ROS	107
CAPÍTULO III.....	109
IMPLEMENTACIÓN Y PRUEBAS.....	109

3.1. ANÁLISIS DE DATOS	113
3.2. SELECCIÓN DE PRUEBAS ESTADÍSTICAS.....	117
3.3. REALIZACIÓN DE ANÁLISIS.....	119
3.4. COMPROBACIÓN DE LA HIPÓTESIS	121
CAPITULO IV	123
RESULTADOS Y CONCLUSIONES.....	123
4.1 RESULTADOS	124
4.2 CONCLUSIONES.....	126
4.3 TRABAJOS FUTUROS.....	127
BIBLIOGRAFIA.....	128

ÍNDICE DE FIGURAS

Fig.2.1 Ros Multi-Communications. (YoonSeok Pyo, 2017).....	44
Fig.2.2 Versiones de ROS (http://wiki.ros.org/).....	46
Fig.2.3 Tarjeta Raspberry pi con puertos de entrada y salida. (Sánchez, 2020)	50
Fig.2.4 sistemas operativos oficiales para Raspberry pi (Sánchez, 2020).....	51
Fig.2.5 Proceso de interpretación de lecturas (Sole).....	55
Fig.2.6 Referencias estáticas de sensibilidad. (Sole).....	58
Fig.2.7 Comportamiento de histéresis con relación a la resolución. (Sole).....	59
Fig.2.8 Referencia de saturación con respecto a la linealidad. (Sole)	60
Fig.2.9 Curvas de salida de motor pololu, Simulación el Matlab.....	65
Fig.2.10 Conector para la comunicación serial bajo la norma RS-232.....	73
Fig.2.11. Señal eléctrica ideal del ASCII "H" por la interfaz RS-232.	74
Fig.2.12. Transmisión RS-232 sin interferencias en el canal de comunicación.....	75
Fig.2.13. Transmisión RS-232 con interferencias en el canal de comunicación	75
Fig.2.14. Circuitos comerciales que integran la norma RS-232.	76
Fig. 2.15. sistema de referencia en el sistema vectorial, plano 3D. a) Sistema de referencia de la mano derecha e izquierda. b) Cuadrantes del plano en 3D. c) Cubo en el plano 3D positivos. d)Cubo en cuadrantes positivos y negativos en el plano 3D. e) Cubo en los cuadrantes negativos. f) Localización de los puntos P y Q, en el sistema de coordenadas vectoriales. (Edwin J. Purcell, 2007)	83

Fig. 2.16. longitud, dirección y magnitud de un vector. (Edwin J. Purcell, 2007)	84
Fig. 2.17. Flechas de dirección, analogía para la representación de un vector.	85
Fig. 2.18. Operaciones equivalentes con vectores. (Edwin J. Purcell, 2007).	86
Fig.2.19 Vector con sentido definido (Edwin J. Purcell, 2007).....	87
Fig.2.19. Robot con sistema de control on-off. Imagen propia.	91
Fig. 2.20. simulación SLAM y ROS para la navegación autónoma. Imagen propia	92
Fig.2.21. vista del prototipo en CAD, primer diseño, descartado. Imagen propia.....	93
Fig.2.22. coordenadas de la postura del robot móvil. Se orienta en el sistema absoluto O, y el sistema de coordenadas del robot A. Imagen propia.....	94
Fig.2.23. planos en un robot diferencial rectangular. Imagen propia	95
Fig.2.24. Segundo diseño de robot diferencial. Imagen propia.....	96
Fig.2.25. Diagrama de cuerpo libre del robot móvil de direccionamiento diferencial. Imagen propia.....	96
Fig.2.26. Sistemas de coordenadas $\Sigma_0, \Sigma_1, \Sigma_2$. Imagen propia	98
Fig.2.27. Los arcos D_l y D_r son trazados por las ruedas izquierda y derecha para el cambio en el avance del robot con ángulo θ	101
Fig.2.28. Interfaz gráfica del usuario para el control en forma manual. Imagen propia	102
Fig.2.29 Sistema de control de lazo cerrado implementado en la odometría del sistema de navegación. (Ogata)	103
Fig.2.30. (a) Sistema con un controlador PID; (b)diagrama de bloques equivalente. (Ogata) ..	104
Fig.2.31. Código implementado en el microcontrolador Arduino y el sistema PID, ambos en base a la planta formulada. Imagen propia.....	105
Fig.2.32. Sistema de control para la estabilización de la planta PID. Imagen propia	106
Fig. 3.33. Metodología para el desarrollo del proyecto. Elaboración propia.	110
Fig. 3.34. Grafo de sistema de control. Elaboración propia.....	111
Fig. 3.35. Diagrama para el sistema de control autónomo. Elaboración propia	112
Fig. 3.36. Mapa, puntos de muestreo. Elaboración propia.....	113
Fig.3.37. obstáculos colocados para pruebas de navegación haciendo uso de SLAM. Imagen propia	114
Fig.3.38. Navegación del sistema de control, utilizando SLAM, fuerzas de atracción y de repulsión. Imagen propia.....	115
Fig.3.39. Mapa en 2D del algoritmo SLAM el mapa es generado por las trayectorias de ROS. Imagen propia.....	116
Fig.3.40. Histograma con relación a las muestras con respecto al eje X en el rectángulo establecido, así como las mediciones con respecto a la posición frontal del rectángulo.	118

Fig.3.41. Rectángulo establecido por el pasillo de prueba y virtualizado en SLAM. Imagen propia.....	120
Fig.3.42. Mapa en 3D del pasillo, mapa tridimensional con respecto a un eje vertical, SLAM. Imagen propia.....	121
Fig. 3.43. Prototipo robótico desarrollado con sistema de tracción diferencial en forma física. Imagen propia.....	124
Fig. 3.44. Diseño de prototipo robótico en software CAD, SolidWorks 2018. Imagen propia. .	125
Fig 3.45. Plataforma para silla de ruedas.....	127

ÍNDICE DE TABLAS

Tabla 2.1 Especificaciones generales de motor Pololu	63
Tabla 2.2 Dimensiones de motor Pololu, datos del fabricante	63
Tabla 2.3. Especificaciones de intensidad de corriente en vacío y a plena carga (Corporation ©. 2.-2., 2020)	63
Tabla 2.4 Código de colores para cables de alimentación y sensor de efecto hall motor Pololu. (Corporation ©. 2.-2., 2020)	66

DEDICATORIA (PERSONAL)

A Dios, por permitirme llegar a este momento tan especial en mi vida. Por los triunfos y los momentos difíciles que me han enseñado a valorarlo cada día más, A mi madre por ser la persona que me ha acompañado durante todo mi trayecto estudiantil y de vida, a mis hermanos quienes han velado por mí durante este arduo camino para convertirme en un profesional. A mi padre quien con sus consejos ha sabido guiarme para culminar mi carrera profesional. A mis hijos que en varias ocasiones no pude jugar con ellos por desarrollarme profesionalmente, a mi esposa que con sus palabras su amor me dio la fortaleza de salir adelante y enfrentar los problemas y a levantarme en momentos más difíciles. Gracias al equipo que formamos logramos llegar hasta el final del camino. A mis profesores, gracias por su tiempo, por su apoyo, así como por la sabiduría que me transmitieron en el desarrollo de mi formación profesional en la maestría.

CARTAS DE ACEPTACIÓN

CAPÍTULO I

GENERALIDADES DEL PROYECTO

1.1. INTRODUCCIÓN

Las últimas estadísticas presentadas por INEGI indican que en México existen alrededor de 4,527,784 personas con algún grado de discapacidad, del cual casi 2,437,397 personas presentan problemas para caminar o moverse, y 1,292,201 personas presentan problemas para ver ((INEGI), 2018). Estos padecimientos se encuentran presentes en edades de entre 60 y 84 años, en primera instancia, y de 30 a 59 años en segunda instancia, y con un tercer porcentaje de entre 15 y 29 años, en cualquiera de los datos presentados, la primera causa de discapacidad es por enfermedad, seguido de edad avanzada, y en tercera instancia padecimiento desde nacimiento.

Estudios de la OMS (Organización Mundial de la Salud), en Latinoamérica solamente un porcentaje del 2% de personas con discapacidad puede acceder a algún tipo de beneficio o proceso de rehabilitación. (Salud, 2016) En México, se estima que 33 de cada 100 personas no acceden a programas de rehabilitación, lo que ha motivado al estado, la creación de fundaciones dedicadas a la inclusión de personas con capacidades diferentes. (INEGI, La discapacidad en Mexico, 2016)

La tecnología da pasos agigantados día tras día y se ha convertido en una herramienta esencial en la vida del ser humano, su avance está ligado a la industria, la informática, robótica, medicina, y en general, a todos los aspectos que buscan una mejora en la sociedad. Los impedimentos y barreras se acortan

facilitando nuevos estudios y creación de dispositivos nuevos y cada vez más avanzados.

Por ello, es indispensable comenzar con una definición de lo que es la tecnología: Se podría definir “como la aplicación del conocimiento científico a la solución de problemas prácticos y la obtención de metas humanas”. En la actualidad el desarrollo científico se ha centrado en combatir e intentar erradicar problemas de índole social, y ayudar a personas con discapacidad a formar parte de este nuevo mundo tecnológico. Cabe recalcar que se han logrado grandes avances en este campo, como el desarrollo de aplicaciones de inclusión, pero falta aún profundizar en estudios para poder conseguir en estas personas una vida conforme a sus necesidades.

Los múltiples avances tecnológicos que se han desarrollado en la última década en el sector de la salud, específicamente en la asistencia personal y médica que han cambiado profundamente, y sin duda es una de las aplicaciones de la tecnología que más avance y beneficios ha dado a la sociedad a pesar de su corto tiempo de implementación.

El proyecto que se presenta aborda la tarea de la integración de ROS (Robot Operating System) con las plataformas de propósito general Arduino Mega ADK y Raspberry Pi.

ROS se constituye como un *middleware*, ya que se trata de un software que asiste a las aplicaciones a través del hardware conectado a él. Funciona como una capa de abstracción de software distribuida situada entre las capas de aplicaciones y las capas inferiores. El *middleware* obtiene información a través

de los sensores, la forma del entorno y la complejidad que estos pueden llegar a tener, así mismo la heterogeneidad de las redes de comunicaciones subyacentes, los sistemas operativos y lenguajes de programación, proporcionando una API (Hat, 2020) para la fácil programación y manejo de las aplicaciones distribuidas, punto clave para la consecución de este proyecto.

La existencia de plataformas de propósito general, como son Arduino y Raspberry Pi, las cuales se caracterizan por ser hardware de bajo costo, dimensiones reducidas y alto rendimiento, permiten a la amplia comunidad de desarrolladores de todo el mundo elevar la programación y control de robots a un primer nivel gracias a las facilidades que se les ofrece. Gracias a esto y a la existencia de sistemas operativos dedicados, los cuales ofrecen incontables herramientas de programación y desarrollo, estas plataformas y sistemas se están convirtiendo en una de las opciones más valoradas.

El objetivo principal que se pretende conseguir es:

1. La programación
2. El modelado para el sistema de control
3. Diseño de aplicaciones que doten ha dicho sistema de diferentes características y funcionalidades para la adquisición y envío de datos por los puertos de comunicación.
 - a. Envío de información a través del puerto serial bajo el protocolo de envío de este puerto.
 - b. Recepción de los datos en la aplicación principal de ROS, para su ejecución.

- c. Funcionalidad de realizar el escaneo del ambiente a navegar con SLAM.
- d. Retroalimentación del sistema de control para determinar el error en la posición deseada y lo que se establece en el ambiente virtual de SLAM.
- e. Control PID, para los motorreductores en relación con la velocidad angular y lineal.

Todas las características y funcionalidades deben estarse ejecutando de forma simultánea y sin complicaciones de comunicación digital, ya que si alguno de los bits se llega a modificar también lo haría la posición y la trayectoria.

1.2. PLANTEAMIENTO DEL PROBLEMA

Una de las inquietudes presentes en la sociedad y en las instituciones de los diferentes niveles educativos, es la movilidad de los estudiantes con capacidades diferentes, es un reto para este sector estudiantil con alto grado de vulnerabilidad poder llegar de un punto a otro, considerando el grado o nivel de movilidad que pueda tener. Esto ocasiona que la movilidad se complique a tal grado, que puede ocasionar que la persona que padece dicha limitación tenga problemas de integración en la sociedad y aún más en un sector productivo.

El Instituto Tecnológico Superior de Teziutlán cuenta con una población estudiantil de 2265, de los cuales 1335 son hombres y 930 son mujeres, así como 2 son considerados como alumnos con capacidades diferentes y 1 tienen problemas de movilidad, esto genera una problemática latente en los estudiantes

el poder trasladarse dentro de las instalaciones del ITST, ocasionando que dicho grupo de estudiantes con este padecimiento tenga un alto grado de dependencia en sus compañeros y docentes.

1.3. JUSTIFICACIÓN

La movilidad autónoma de personas con capacidades diferentes siempre ha sido un tema de gran interés en la sociedad, por ello asociaciones civiles, fundaciones, centros de investigación, empresarios y los diferentes niveles de gobierno, priorizan el bienestar de este sector de la sociedad la cual presenta un alto porcentaje de vulnerabilidad. La finalidad es llegar a tener un prototipo similar a una silla de ruedas el cual tenga autonomía en trasladarse de un punto a otro, e interactuar en entornos hostiles, el prototipo a desarrollar debe ser capaz de adquirir todo tipo de información a través de su sistema sensorial. Una de las múltiples aplicaciones que puede llegar a tener este sistema de navegación es la adaptación a otros aparatos ortopédicos destinados a la movilidad de personas con estas limitantes.

1.4. HIPÓTESIS

El desarrollo del prototipo robótico, que pueda adaptarse a una silla de ruedas, apoyará a los alumnos con capacidades diferentes, del Instituto Tecnológico Superior de Teziutlán, a la movilidad en interiores y exteriores de forma autónoma, evitando colisiones de hasta cinco centímetros en la precisión según la posición calculada.

1.5. OBJETIVOS

1.5.1. OBJETIVO GENERAL

Desarrollar un prototipo robótico con un sistema de fusión de sensores y procesamiento paralelo, aplicado a la movilidad autónoma de estudiantes con dificultades de movilidad en el Instituto Tecnológico Superior de Teziutlán, con la finalidad de apoyar su autonomía dentro de las instalaciones del instituto de forma autónoma.

1.5.2. OBJETIVOS ESPECÍFICOS

1. Diseñar el robot con software CAD.
2. Construir el prototipo Robot con todos los elementos mecánicos a utilizar.
3. Realizar el cableado e instalación eléctrica del robot según lo especificado por el hardware.
4. Realizar el control PI para el sistema de odometría en los motorreductores 25d con Arduino.
5. Instalar software (Debian Wheezy Raspbian) y ROS (Sistema Operativo Robótico). Sistema operativo y software de control respectivamente, para control de hardware.
6. Implementar algoritmo para el procesamiento paralelo
7. Realizar el procesamiento de las señales de entrada en la plataforma de ROS
8. Realizar la simulación con los datos de entrada en ROS
9. Aplicar pruebas de funcionamiento en las instalaciones del ITST

1.6. ALCANCES

Al implementar el robot móvil con el sistema de navegación hará más eficiente la movilidad y autonomía dentro de las instalaciones de instituto tecnológico superior de Teziutlán, el prototipo cuenta con un sistema de control, con odometría, actuadores, sensores y tarjetas de control con la implementación de algoritmos desarrollados en ROS, el cual al ser modelado con el ambiente virtual para la toma de decisiones SLAM, el prototipo robótico tiene todos los elementos de diseño eléctrico, electrónico y mecánico, los cuales son necesarios para ser adaptados a la movilidad de aparatos ortopédicos con aplicaciones de movilidad.

1.7. LIMITACIONES

Dentro de un proyecto de investigación las limitaciones son demasiadas, algunas más imprescindibles que otras. Por ello desarrollar un prototipo robótico con una movilidad autónoma conlleva múltiples retos y desafíos al ejecutar algoritmos, circuitos electrónicos digitales y analógicos, todos trabajando de forma sincronizada con los elementos mecánicos y actuadores. Todos los elementos que integran el prototipo robótico pueden tener actualizaciones que hagan más eficiente la navegación, ya que la necesidad de moverse de un punto a otro de los seres humanos sin importar su condición es muy amplia y eso hace que los algoritmos de decisión para la navegación sean más complejos. Por ello se puede decir que el límite de navegación con este sistema de control de navegación con

sensores lidar y middleware framework no podría lograr mayor precisión por los recursos mismos de hardware.

Si se pretende tener mayor eficiencia en la toma de decisión para la navegación en ambientes diseñados especialmente para seres humanos se deben implementar elementos de hardware y software, con mayor capacidad para que trabajen en conjunto a la inteligencia artificial.

1.8. ESTADO DEL ARTE

Existen trabajos que hacen uso de las técnicas que se requieren para desarrollar una solución a la problemática planteada. Por ello en el siguiente apartado se describen algunos de los más importantes según sus aplicaciones.

Manuela M. Veloso (Veloso's, 2020) es una investigadora del Departamento de Aprendizaje automático en la universidad de Carnegie Mellon, ha desarrollado una gran variedad de robots autónomos, en el área de robots de futbol y robots de servicio móvil. Sus prototipos robóticos de navegación autónoma han recorrido más de 1000km en edificios de diferentes universidades, dentro de sus líneas de investigación y desarrollo, hace uso de técnicas como filtros de Kalman, inteligencia artificial (IA), redes neuronales, visión artificial etc. Y entre sus trabajos más actuales son:

- 1.8.1. Optimal Perception Planning with Informed Heuristics Constructed from Visibility Maps.

Consiste en la planificación del movimiento para la percepción de una posición objetivo. Un robot tiene que moverse a una posición desde donde puede detectar el objetivo, mientras minimiza los recursos de movimiento y percepción. El problema de encontrar rutas para robots que ejecutan tareas de percepción se puede resolver de manera óptima mediante la búsqueda informada. En la planificación del camino de percepción, la solución cuando se considera una línea recta sin obstáculos se usa como heurística. En este trabajo, Veloso propone una técnica heurística que puede mejorar la eficiencia de búsqueda. Para reducir la expansión del nodo mediante una búsqueda más informada, hace uso de un mapa de visibilidad aproximada del robot (A-VM), que se utiliza como una representación de la capacidad de observación de un robot en un entorno determinado. En el artículo Veloso muestra cómo los puntos críticos utilizados en A-VM proporcionan información sobre la geometría del entorno, que puede utilizarse para mejorar la heurística, aumentando la eficiencia de búsqueda. Los puntos críticos permiten una mejor estimación del movimiento mínimo y el costo de percepción para objetivos en regiones no transitables que solo pueden detectarse desde más lejos. Finalmente, muestra la heurística contribuida con mejoras que domina la heurística PA * básica construida sobre la distancia euclidiana, mostrando resultados del aumento del rendimiento en términos de expansión de nodos y tiempo de cálculo.

1.8.2. Effective Team Coordination through Intra-Robot Replanning to Restore Team Plan Rationale

Consiste en múltiples vehículos submarinos autónomos (AUV) que participan en misiones oceánicas que requieren colaboración. En este trabajo, abordan el desafío de la ejecución efectiva del plan distribuido de múltiples robots: cuando ocurre una falla, un robot individual necesita decidir cómo proceder para mantener el éxito del plan de múltiples robots. Las fallas ocurren cuando los modelados de los AUV no logran el resultado esperado, debido en gran medida en este dominio a la dinámica mal modelada del océano. La replanificación intra-robot es ventajosa porque los métodos controlados tradicionalmente centralizados utilizan una comunicación satelital costosa y que consume mucho tiempo para reparar todas las fallas. Por lo tanto, Veloso presenta una representación de plan de equipo basada en la lógica que incluye explícitamente la lógica de las acciones planificadas de los robots individuales. El algoritmo de motivación racional dentro del robot que permite la selección y ejecución de re-planeación para robots individuales para superar el fracaso. La selección del algoritmo se mejora a través del aprendizaje del costo previsto de cada política de re-planificación, que en última instancia mejora el rendimiento general del equipo.

1.8.3. Heterogeneous Multi-Agent Planning Using Actuation Maps

Muchos escenarios robóticos del mundo real requieren la planificación de tareas para decidir recursos de acciones a ejecutar por robots (posiblemente heterogéneos). Un enfoque de planificación centralizado clásico que considera en el mismo espacio de búsqueda todas las combinaciones de robots y objetivos

podría conducir a soluciones ineficientes que no escalan bien. *La planificación de múltiples agentes* (MAP) proporciona un buen marco para resolver este tipo de tareas de manera eficiente. Algunas técnicas de MAP han propuesto asignar previamente objetivos a los agentes (robots) para que disminuya el esfuerzo de planificación. Sin embargo, estas técnicas no se escalan cuando aumenta el número de agentes y objetivos, como en la mayoría de los escenarios del mundo real con grandes mapas u objetivos que no pueden ser alcanzados por subconjuntos de robots. En este trabajo Veloso, propone una forma de calcular qué objetivos deben asignarse a cada agente mediante el uso de *mapas de actuación* (AM). Dado un mapa, los AM pueden determinar las regiones en las que cada agente puede actuar. Esto ayudara a aliviar el esfuerzo de las técnicas MAP al saber qué objetivos que puede abordar cada agente, así como a estimar de manera económica el costo de usar cada agente para lograr cada objetivo. Los experimentos muestran que cuando la información extraída de los AM se proporciona al planificador de múltiples agentes, la asignación de objetivos es significativamente más rápida, acelerando considerablemente el proceso de planificación. Los experimentos también muestran que este enfoque supera en gran medida la planificación centralizada clásica

1.8.4. Trajectory-Based Short-Sighted ProbabilisticPlanning

La planificación probabilística captura la incertidumbre de la ejecución del plan al modelar probabilísticamente los efectos de las acciones en el entorno y, por lo tanto, la probabilidad de alcanzar diferentes estados de un estado y una acción

determinados. Para calcular una solución para un problema de planificación probabilística, los planificadores deben gestionar la incertidumbre asociada con las diferentes rutas de un robot, desde el estado inicial a un estado objetivo. Veloso propuso, varios enfoques para gestionar la incertidumbre, por ejemplo, considerar todas las rutas una vez, realizar la determinación de acciones y muestreo. En este trabajo Veloso, introduce los *problemas estocásticos de corto recorrido* (SSP), sistemas cortos de visión, basados en la trayectoria, un enfoque novedoso para gestionar la incertidumbre de los problemas de planificación probabilística en los que los estados accesibles con baja probabilidad son sustituidos por objetivos artificiales que de forma heurística estimar su costo para alcanzar un estado objetivo. También extendemos los resultados teóricos del *Planificador probabilístico de vista corta* (SSiPP) al demostrar que SSiPP siempre termina y es asintóticamente óptimo en condiciones suficientes en la estructura de los SSP. Veloso, compara, empíricamente los SSiPP utilizando SSP miopes basados en la trayectoria con los ganadores de las competencias de planificación probabilísticas anteriores y otros planificadores de vanguardia en los problemas del mundo. SSiPP basado en la trayectoria supera a todos los competidores y es el único planificable para escalar hasta el problema número 60, un problema en el que la solución óptima contiene aproximadamente 1070 estados.

En México también existen trabajos relacionados con inteligencia artificial, redes neuronales, arquitectura de sistemas, y visión artificial, aplicados a la robótica móvil autónoma. Dichos trabajos están más al alcance, por parte del centro de investigación en inteligencia artificial ubicado en la ciudad de Xalapa Veracruz.

Una de las personas que ha hecho aporte en esta línea de investigación en México es el Dr. Antonio Marín Hernández, (Veracruzana, 2020) el cual tiene como línea de investigación, Robótica Móvil, Percepción, y acción robótica identificadores. Algunos de los trabajos del Dr. A. Marín son:

1.8.5. An Approach based on a Robotics Operating System for the Implementation of Integrated Intelligent House Services System

Este trabajo presenta un enfoque basado en el Sistema de Operación de Robótica (ROS) para integrar diferentes tipos de servicios en un Sistema de Servicios Domésticos Inteligentes (IHSS). El sistema propuesto mejora el concepto común de domótica al incorporar agentes físicos, es decir, robots de servicio autónomos. Los robots móviles pueden moverse sobre el entorno y modificarlo, pero también interactuar con los habitantes de la casa de una manera más cercana; por lo tanto, el sistema tendrá un aumento en los servicios prestados. Los enfoques IHSS se compone de dos subsistemas: a) un servidor maestro interno y b) al menos un robot de servicio móvil inteligente. Del mismo modo, a medida que el robot se comunica y controla su hardware; el servidor maestro de la casa se ocupa de todos los dispositivos integrados en el hogar, por ejemplo, luces, aire acondicionado y detectores de incendios. La comunicación entre subsistemas se realiza utilizando las herramientas proporcionadas en ROS. Se presenta un enfoque conceptual y una implementación preliminar para la validación del concepto IHSS

1.8.6. Sensor Fusion Used in Applications for Hand Rehabilitation: A Systematic Review

Las condiciones médicas y los accidentes pueden causar inmovilidad en ciertas partes del cuerpo. Para ayudar a las personas en el proceso de rehabilitación, los sensores que obtienen señales biológicas del cuerpo se han fusionado para crear tecnología de asistencia. Esta revisión sistemática se centra en presentar el estado de la técnica con respecto a la fusión de sensores utilizada en las aplicaciones para rehabilitación de manos. Se buscaron artículos en cuatro bases de datos: IEEE Xplore, Web of Science, Biblioteca Digital ACM y PubMed. Además, se emplearon los protocolos PRISMA y QualSyst para filtrar los artículos y evaluar sus cualidades. Como resultado, de los 102 artículos recuperados inicialmente, solo 29 artículos realizaron fusión de sensores para rehabilitación de manos. Específicamente, se encontraron tres áreas de aplicación: fusión de sensores para detectar movimientos de manos, fusión de sensores para aplicaciones de control de exoesqueleto y fusión de sensores en juegos serios para rehabilitación de manos. Los estudios incluidos en esta revisión midieron dos aspectos clave para fines de rehabilitación: la fuerza de la extremidad del usuario y la posición de la extremidad del usuario. Con respecto a los sensores empleados en la fusión, el sensor más utilizado fue el sensor de electromiografía seguido de los sensores de inercia. Además, los estudios incluidos en esta revisión se han centrado en los siguientes movimientos de la mano utilizados para la rehabilitación: flexión, extensión, pronación, supinación, agarre de fuerza,

radial/urinaria, mano abierta, contracciones con un solo dedo, contracciones con varios dedos, agarre con pellizco, mano en reposo y empuñadura.

1.8.7. Detecting falling people by autonomous service robots: A ROS module integration approach

En este trabajo se presenta la integración de diversos módulos para la detección de caídas de personas por un robot de servicio móvil. Esta integración se ha logrado en el middleware ROS (Robotics Operation System). La implementación propuesta se organiza sobre una arquitectura modular de tres capas: Hardware, Procesamiento y Decisión. Los módulos implementados están en la capa de procesamiento. El primer módulo utiliza una cámara RGB-D para detectar y rastrear a una persona en el entorno. Este módulo calcula características para detectar la pose caída. En el segundo módulo, se utiliza un controlador PID en una unidad de panorámica/inclinación, para rastrear a la persona con un error mínimo y un movimiento suave. Para este propósito, el centroide de la persona se encuentra en el centro de la imagen del plano. Las características principales de nuestra arquitectura son: 1) Se utiliza la segmentación en profundidad, porque se requiere información 3D para detectar la pose caída; 2) Los parámetros de control PID se ajustan utilizando un método manual y un algoritmo genético, para comparar y mejorar el rendimiento del módulo de seguimiento de personas. Una vez que se optimizó el controlador PID, la arquitectura para seguir a la persona y detectar la pose caída se prueba en tiempo real.

1.8.8. A Graph Representation Composed of Geometrical Components for Household Furniture Detection by Autonomous Mobile Robots

Este estudio propone un marco para detectar y reconocer muebles domésticos utilizando robots móviles autónomos. La metodología propuesta se basa en el análisis e integración de características geométricas extraídas sobre nubes de puntos 3D. Se construye un gráfico relacional usando aquellos presenta un modelo y reconoce cada mueble. Un conjunto de sub-gráficos correspondientes a diferentes partes. Las vistas parciales permiten hacer coincidir la percepción del robot con los modelos de muebles parciales. Un conjunto reducido de características geométricas se emplea: planos horizontales y verticales y las patas de los muebles. Estas características se caracterizan por sus propiedades, tales como: altura, planeidad y área. Se propone un método lineal para la detección de algunas características geométricas, que se basa en histogramas de puntos 3D adquiridos de una cámara RGB-D a bordo del robot. También se proponen características geométricas y gráficos.

1.8.9. Realistic and Automatic Map Generator for Mobile Robots

En muchas ciencias, como en los robots móviles, siempre es necesario determinar el rendimiento de los nuevos algoritmos. La creación de perfiles como una herramienta útil durante el proceso de desarrollo y prueba de nuevos algoritmos. El uso de simuladores de robots móviles permite verificar y probar el rendimiento y la eficiencia de los nuevos enfoques. Cuando se evalúa un

algoritmo, es muy útil contar con un gran conjunto de datos de prueba para garantizar que la metodología propuesta sea robusta para una amplia gama de escenarios posibles. En la mayoría de los conjuntos de datos públicos, el número de escenarios para probar nuevos algoritmos es limitado, los mapas son difíciles de modificar y la descripción del mapa no es estándar. Por eso, este trabajo presenta un generador de mapas realista y automático para robots móviles basado en Gazebo debido a que Gazebo es un simulador muy avanzado que se puede usar junto con ROS. Esta combinación permite la creación sencilla de prototipos y pruebas de aplicaciones de robótica, en robots simulados y reales con modificaciones mínimas.

CAPÍTULO II

METODOLOGÍA Y DESARROLLO

2.1. FUNDAMENTOS TEÓRICOS

En este apartado se tendrá la información como una búsqueda rápida o una fuente de información documental de diversa índole, que aborda los temas del trabajo a desarrollar. Los conceptos por definir facilitaran a tener un enfoque, una orientación de estudio, más efectiva que dará una idea más clara de cómo abordar las problemáticas y términos del trabajo en desarrollo.

2.1.1. Niveles de discapacidad.

La necesidad de contar con información estadística sobre la población con algún padecimiento o discapacidad o actualmente denominado capacidades diferentes, fungen una función bastante importante, ya que se debe de tener una idea concreta de cuál y qué tipos, así como los niveles de discapacidad existentes. En 1980 surge la clasificación internacional de deficiencias, discapacidades y minusvalías, esto con el fin de poder obtener una información, de tipo estadística, de todas aquellas personas que tienen algún grado de discapacidad, para poder hacer una elaboración y evaluación de las políticas, así como también de los programas que tienen una finalidad de protección a favor de este grupo de la población vulnerable.

En México, la CIDDM por sus siglas, “Clasificación internacional de las deficiencias discapacidades y minusvalías “. Es un sistema de clasificación y una forma de catalogar las consecuencias a largo plazo, de algunos defectos o

enfermedades en términos de eficiencia o discapacidad, el cual permiten un análisis médico, así como de un sistema sanitario y rehabilitación. Esta clasificación aparece en 1983. En México se llevan a cabo esfuerzos para determinar el número de personas con estas discapacidades o algún padecimiento de movilidad, El objetivo de que exista un sistema de clasificación Es para hacer una depuración De las descripciones que no corresponden a la definición de discapacidad empleada en el censo general de población y vivienda a partir del año 2000 (INEGI, Clasificación de tipo de discapacidad - Historica).

1. **las deficiencias:** se refiere al órgano o parte del cuerpo afectado por ejemplo lesiones del cerebro médula espinal extremidad u otra parte del cuerpo son ejemplos descritos como ausencia de piernas desprendimiento de retina etcétera.
2. **Las discapacidades** se refieren a la consecuencia de la deficiencia en las personas afectadas, por ejemplo, limitaciones para aprender, hablar, caminar, u otra actividad, son ejemplos, “no puede ver”, “no mueve medio cuerpo” y otros padecimientos similares a los antes descritos.

La clasificación De las discapacidades o capacidades diferentes está Clasificada bajo los siguientes criterios:

- discapacidades sensoriales y de la comunicación
 - Discapacidad para ver, oír, hablar, así como de la comunicación y la comprensión del lenguaje.
- Discapacidades motrices

- Discapacidades en extremidades inferiores tronco cuello y cabeza, código 210
 - discapacidades de las extremidades superiores, código 220
- discapacidades mentales
 - Discapacidades intelectuales como por ejemplo retraso mental, código 310
 - discapacidades conductuales y otras que involucran padecimientos mentales, código 399
- discapacidades múltiples y otras
 - Limitación para moverse o caminar y limitación para usar sus brazos y manos, esta discapacidad también puede ser localizada a través del código 401. Otra forma de limitación para moverse o caminar incluyendo la deficiencia de no oír, “sordo” o usar un aparato para oír, está representada bajo la clave 402, en caso de que tenga más de 2 discapacidades se clasificarán con la clave 403, hasta 422, dependiendo el grado de discapacidad.
- claves especiales
 - Estas responden a una necesidad o forma de contar con claves, que permitan resolver problemas de falta de precisión, a las discapacidades detectadas, son formas de discapacidad detectada, pero no sustentadas de qué tipo, e indican desconocimiento y no permiten saber si existe o no la discapacidad, este tipo de padecimientos se clasifican con claves 960.

Cómo se puede leer en el texto anterior, dentro de la clasificación de los múltiples padecimientos en capacidades diferentes o discapacidad, se hará un mayor énfasis para aportar conocimiento y/o apoyar a las personas que sufren con algún padecimiento en la limitación para moverse, o en su caso, con algún padecimiento o dificultad de nacimiento.

2.1.2. Instituciones dedicadas a atender personas de capacidades diferentes.

Existen diferentes organizaciones gubernamentales, particulares, y de grupos sin algún interés, que se dedican a promover a través de fundaciones o financiamiento gubernamental, el desarrollo de proyectos para la inclusión de personas con discapacidad en México existe la organización “juntos.org.mx”, el cual es una organización que ayuda a las personas con discapacidad, a tener un empleo y así poder lograr, que se contabilicen en el sector productivo. La organización “juntos”, promueve cursos a empresas, asociaciones civiles, y fundaciones, para promover la inclusión de personas con discapacidades, al sector productivo. Cuentan con el apoyo de diferentes empresas, Tales como:

- General Motors
- Herdes
- l'oreal
- Mabe
- oxxo
- Shell

- valeo
- ABB
- Cemento Cruz Azul
- Coca Cola
- Estafeta

Por solo mencionar algunas de las más importantes, aunque en el país existen mucho más, así como también existen instituciones educativas y centros de investigación, que se enfocan en el desarrollo de proyectos para poder impulsar la inclusión, en personas con este tipo de padecimientos o dificultades (juntos.org.mx, s.f.).

Un ejemplo, de los muchos que podemos encontrar en las diferentes instituciones superiores, e investigación, gubernamental o particular, es el Instituto politécnico nacional, El cual desarrolla proyectos, principalmente para personas que sufren alguna enfermedad o algún accidente, que hace complicada o imposible tener movimientos voluntarios. Lo que realiza esta institución, a través de su departamento de investigación, Ingeniería y Ciencias, es tomar las señales eléctricas cerebrales y decodificarlas y a través de una aplicación, para así hacer el control de una prótesis robótica (Occidental, 2019). Así como existe este proyecto de investigación, se pueden hacer referencia de múltiples, que se desarrollan día con día a nivel nacional y a nivel mundial, demostrando que el interés de la sociedad, por incluir a personas con capacidades diferentes, es sumamente amplia, de gran importancia y que día a día tiene mayor interés,

intentar dar solución a los múltiples padecimientos y dificultades que llegan a tener este gran grupo vulnerable.

2.1.3. Fusión de sensores y procesamiento paralelo.

Desde 1986 a la fecha y a través de la historia, el desarrollo tecnológico para el sistema de procesamiento de información ha ido en aumento, en un promedio del 50% por año, este porcentaje puede sonar un poco alarmante, debido a que siempre ha tenido un aumento sin precedentes y bastante significativo, los desarrolladores de software a menudo simplemente pueden esperar la próxima generación de microprocesadores, para obtener un mayor rendimiento de un programa de aplicación. Este desarrollo no rebasa más del 80% al año. Desde el 2002 la mejora del rendimiento de un solo procesador sea ralentizado, aproximadamente un 20% por año, esta diferencia es dramática, al 50% por año, el rendimiento aumentará en un casi factor del 60%, en 10 años ,mientras que el 20%, sólo aumentará en un factor del 60%, para los años actuales, la mayoría de los principales fabricantes de microprocesadores, han tomado la decisión que el camino para aumentar rápidamente el rendimiento, se orientaba hacia el paralelismo, en lugar de intentar continuar desarrollándose cada vez más rápido en procesadores monolíticos, estos fabricantes comenzaron a colocar múltiples procesadores completos en un solo circuito integrado, este cambio tiene una consecuencia muy importante para los desarrolladores de software, simplemente agrega más procesadores y mejorará magníficamente el rendimiento de la gran mayoría de los programas en serie, es decir programas que se escribieron, para

ejecutarse en un solo procesador, dichos programas desconocen la existencia de múltiples procesadores, el rendimiento de dicho programa en un sistema con múltiples procesadores, será efectivamente el mismo que su rendimiento en un único procesador, el sistema multiprocesador todo vino a plantear también una serie de preguntas con el desarrollo de este tipo de tecnología, no son los sistemas de procesador único lo suficientemente rápidos, después de todo, el 20% por año sigue siendo una mejora de rendimiento bastante significativa, los fabricantes de microprocesadores no pueden seguir desarrollando sistemas de procesador único mucho más rápidos, deben construir sistemas paralelos y con múltiples procesadores, para que así se puedan escribir programas que convierten automáticamente programas en serie, en programas paralelos, es decir programas que aprovechan la presencia de múltiples procesadores. Sin embargo, se debe tener en cuenta, que muchas de las respuestas a este tipo de tecnología es el múltiple cambio que tiene el software y el hardware, el cual se ha convertido en una carrera de adaptación, que sufre cambios consecutivamente inclusive hasta por mes (Pacheco).

Actualmente los sistemas están desarrollados a la medición. Los sensores, esta tecnología, el cual hace tareas similares a los sentidos básicos del ser humano, el cual, da un panorama a cualquier dispositivo electrónico, mecánico, que necesita de este tipo de tecnología. Estos componentes electrónicos en sus diferentes características, presentaciones y tamaños, así como sus aplicaciones, para que puedan tener una referencia mejor, respecto a lo que le rodea. La necesidad de combinar datos, que provienen de diferentes sensores, para extraer

información útil, el cual permitirá una mayor comprensión en el entorno y toma oportuna de decisiones, se le denomina sistema de fusión de datos o sistemas de fusión de sensores, a la idea de combinar datos provenientes de diversas fuentes, para obtener una descripción completa del fenómeno observado.

Ha estado presente durante mucho tiempo, la fusión de datos es un tema bastante amplio, para el cual ha tenido diferentes formas de conceptualización, o diferentes formas para explicarlo. Una de las definiciones más citadas corresponde, a la propuesta por la Junta de directores de laboratorios del grupo de fusión de datos (JDL). que la define como un proceso de múltiples niveles y fases de detección automática que tratan con la asociación, correlación, estimación, y combinación, de datos e información de una o múltiples fuentes, para lograr posiciones de refinado, estimaciones de identidad, evaluaciones completas situaciones de amenazas oportunas y su significado.

La fusión de sensores surge a partir de los años 90's, por des fortuna con aplicaciones bélicas militares, así como también en el área de la robótica, por obvias razones abarca una gran rama de las ciencias de la ingeniería, la medicina, geología, medio ambiente, geofísica, entre otros.

se pueden encontrar múltiples formas de poder implementar procesos para la fusión de datos o fusión de sensores, esto se puede dar a través de operaciones tipo aritméticas, hasta modelos con un alto grado de complejidad. Algunos investigadores se inclinan a enfocar la fusión de sensores en datos estadísticos, haciendo combinaciones ponderadas, estadística multivariada, minería de datos, y sistemas de medias aritméticas, otras investigaciones aplican enfoques

probabilísticos, aproximaciones bayesianas, modelos de espacio de Estado, métodos de máxima probabilidad, filtros de Kalman, teoría de posibilidad específicamente la teoría evidencia (Wendy Janett Guzman Gonzalez, 2014).

2.1.4. Robótica

A lo largo de toda la historia, el hombre se ha sentido fascinado por máquinas y dispositivos que hagan más fácil el trabajo, así como imitar los movimientos de los seres vivos. La palabra robot, fue usada por primera vez en el año de 1921, en la representación en el Teatro Nacional de Praga, en esta representación, los robots eran máquinas androides, fabricadas a partir de una fórmula brillante. En esta obra los robots servían a sus jefes, en este caso los humanos, el término robot posiblemente hubiera caído en desuso, si no hubiese sido por los escritores del género literario de la ciencia ficción. Algunos de los cuales retomaron la palabra e incluso el mensaje de la obra de Capek. En octubre de 1945 se publicó en la revista Galaxy de ciencia ficción, una historia en la que por primera vez se enunciaron las tres leyes de la robótica.

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

El robot como máquina, lleva un desarrollo independiente, son máquinas que hacen uso de la tecnología y la electrónica, el servo control, la mecánica y las Ciencias de la computación. La primera patente de un dispositivo robótico fue solicitada en marzo de 1954, por el inventor británico Kenguard. La robótica está experimentando en la actualidad una ampliación bastante notoria saturada en cierta medida de múltiples aplicaciones, en talleres, líneas de fabricación etc. Estos robots están tomando posiciones en aplicaciones fuera del taller, caracterizada fundamentalmente por falta de estandarización de procesos y la estructuración de su entorno.

La definición de robot puede caer en múltiples controversias, a la hora de establecer una definición formal.

La definición, más comúnmente aceptada, posiblemente sea la de la asociación de industrias robóticas, la cual se describe de la siguiente forma: un robot industrial es un manipulador multifuncional reprogramable capaz de mover materiales piezas herramientas o dispositivos especiales, según trayectorias variables programadas para realizar tareas diversas.

Esta definición, ligeramente modificada, ha sido adoptada por la organización internacional de estándares (ISO), el cual establece una segunda definición que se describe como un manipulador multifuncional reprogramable, con varios grados de libertad, capaz de manipular materiales, piezas, herramientas o dispositivos especiales, según trayectorias variables programadas para realizar tareas diversas.

La robótica se clasifica en robótica secuencial, robótica de trayectoria controlable, robótica adaptativa, y robótica tele manipulada. Un robot está formado por los siguientes elementos: estructura mecánica, transmisiones, sistemas de accionamiento, sistema sensorial, sistema de control y elementos terminales entre muchas otras áreas de especialización que se pueden adaptar a estos dispositivos de gran avance tecnológico (Antonio Barrientos, 2012).

Este trabajo se enfoca en un tipo de robot móvil denominado generalmente como robot diferencial, es un tipo de robot que se basa en el modelo cinemático, en base a un sistema de propulsión, este sistema permite que un dispositivo móvil pueda moverse dentro de un entorno determinado, uno de los sistemas más usuales que se implementa en sistemas mecánicos, similares al que se describe es a partir del uso de ruedas de tracción de tipo diferencial, ya que establece características cinemáticas del movimiento poco complejos y bastante eficientes para sistemas de navegación en entornos especialmente, diseñados para actividades humanas, por ejemplo pasillos edificios parques así como oficinas bodegas etcétera.

2.1.5. Middleware (ROS).

Ros es un sistema meta operativo de código abierto para cualquier sistema robótico. Brinda servicios para sistemas operativos, incluyendo hardware de abstracción, control de dispositivos de bajo nivel, así como también la implementación de la funcionalidad de uso común, Incluye también dentro de sus bibliotecas transmisiones de mensajes entre procesos y gestión de paquetes,

también proporciona herramientas y bibliotecas para obtener, construir, escribir y ejecutar código, en varias computadoras.

Ros incluye una capa de abstracción, de hardware similar a los sistemas operativos, pero a diferencia de estos, puede usarse para numerosas combinaciones de implementación de hardware, además es un software especialmente para robótica, es una plataforma que proporciona varios entornos de desarrollo especializados para robots, con programas de aplicaciones específicas o especiales.

En el mercado, existen diferentes sistemas operativos como por ejemplo Windows, Linux, Ubuntu, Fedora, Mac, yoséMIT, capitán, Symbian, Vicent, Android, iOS, etcétera. Pero en realidad Ros, también se le considera como un sistema operativo.

Aunque el Sistema Meta-Operativo no es un término definido en el diccionario, se podría describir como un sistema que realiza procesos tales como programación, carga, monitoreo y manejo de errores utilizando capa de virtualización entre aplicaciones y recursos informáticos distribuidos.

Como se muestra en la siguiente figura, la comunicación de datos ROS, no solo es compatible con un sistema, pero también por múltiples sistemas operativos, hardware y programas, lo que lo hace altamente Adecuado para el desarrollo de robots donde se combinan varios hardware.

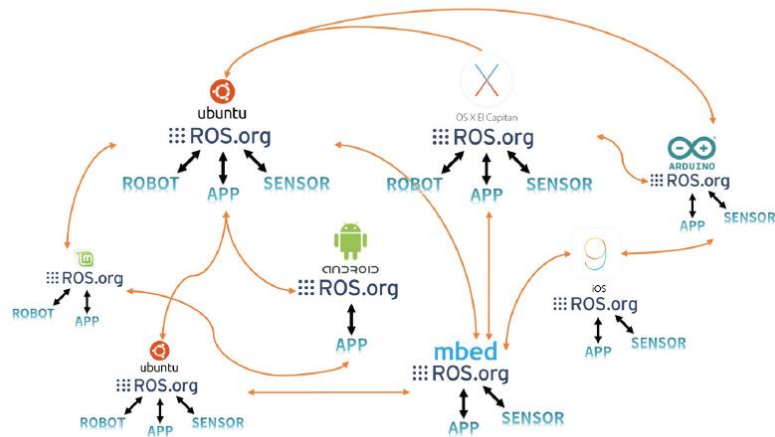


Fig.2.1 Ros Multi-Communications. (YoonSeok Pyo, 2017)

Una de las preguntas más frecuentes a lo largo de los años, es el comparativo con otras plataformas de software, (OpenRTM, OPRoS, Player, YARP, Orocos, CARMEN, Orca, MOOS, Microsoft Robotics Studio).

ROS se centra en maximizar la reutilización del código en la investigación y el desarrollo de la robótica, en lugar de orientándose hacia la llamada plataforma de software de robot, middleware y framework.

Para apoyar esto, ROS tiene las siguientes características.

- **Proceso distribuido:** se programa en forma de unidades mínimas de ejecutables procesos (nodos), y cada proceso se ejecuta de forma independiente e intercambia datos sistemáticamente.
- **Gestión de paquetes:** múltiples procesos que tienen el mismo propósito se gestionan como un paquete para que sea fácil de usar y desarrollar, así como conveniente para compartir, modificar y volver a distribuir.

- **Depósito público:** cada paquete se hace público al público preferido del desarrollador repositorio (por ejemplo, GitHub) y especifica su licencia.
- **API:** cuando se desarrolla un programa que usa ROS, ROS está diseñado para simplemente llamar a una API e insértelo fácilmente en el código que se está utilizando. En el código fuente introducido en cada capítulo, se podrá ver la programación ROS, no es muy diferente de C ++ y Python.
- **Admite varios lenguajes de programación:** el programa, ROS proporciona una biblioteca cliente para soportar varios lenguajes de programación. La biblioteca se puede importar en programación lenguajes que son populares en el campo de la robótica, como Python, C ++ y Lisp, así como lenguajes como JAVA, C #, Lua y Ruby. En otras palabras, puede desarrollar un programa ROS utilizando un lenguaje de programación preferido.

Estas características de ROS han permitido a los usuarios establecer un entorno donde está posible colaborar en el desarrollo de software de robótica a nivel global. Reutilizando un código en La investigación y el desarrollo de la robótica se están volviendo más comunes, que es el objetivo final de ROS. (ROS.org, s.f.)

2.1.5.1. Versiones de ROS

En 2007, Willow Garage tuvo éxito en la investigación del marco de software del robot que comenzó con Switchyard en el Laboratorio de inteligencia artificial de la Universidad de Stanford y continuó el desarrollo bajo el nombre del sistema operativo del robot (ROS). Con la sexta versión de lanzamiento oficial "ROS

Groovy Galapagos", Willow Garage intentó penetrar en el mercado de robots de servicio comercial en 2013, pero terminó dividiéndose en varias start-ups, y finalmente se entregó a Open Source Robotics Fundación (OSRF). Desde entonces, se lanzaron cuatro versiones más y, a partir de mayo de 2017, OSRF cambió su nombre a Open Robotics y ha estado desarrollando, operando y administrando ROS Más recientemente, la undécima versión de ROS, ROS Lunar Loggerhead, se lanzó el 23 de mayo, 2017. ROS etiqueta la primera letra de cada nombre de lanzamiento en orden alfabético y usa una tortuga como su símbolo

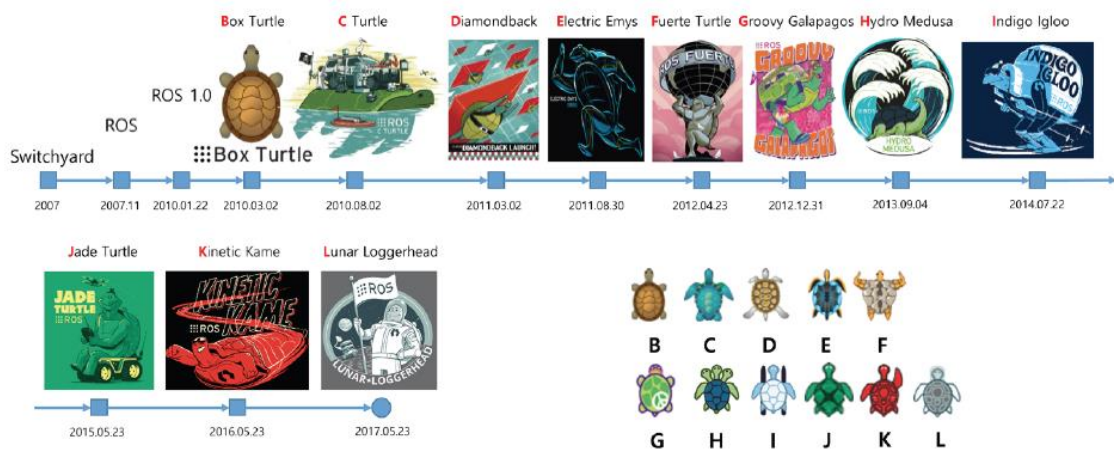


Fig.2.2 Versiones de ROS (<http://wiki.ros.org/>)

2.1.6. Raspberry

Raspberry es un ordenador o una forma de computadora con una placa de control única, está diseñada para ser una computadora de bajo costo, desarrollada en el Reino Unido por la Fundación Raspberry Pi, esta placa se desarrolló especialmente para estimular la enseñanza de la computación e informática en

las escuelas de los países con bajo desarrollo económico, Raspberry inicialmente empezó como hardware libre, simplemente con derechos de marca que son expuestos en los FAQs donde explican que solamente disponen de los derechos de distribución y venta con dos empresas y algunos contratos para tener distribuidores oficiales de las tarjetas raspberry Pi, por lo que se puede establecer que es un producto con un registro pero de uso libre, de esta forma la fundación raspberry puede tener el control de la plataforma permitiendo el uso de su sistema a nivel educativo, como particular en forma gratuita, aunque también ha tenido varios repuntes al ser utilizado en el sector empresarial, aunque para poder hacer utilizado en este sector debe existir una consulta o un derecho con la Fundación.

El software si es completamente Open Source, siendo este un sistema operativo oficial con una versión singular, con adaptación de debían, denominada bajo el nombre registrado de raspbian, aunque también se pueden hacer uso de otros sistemas operativos, incluyendo también una versión para Windows 10 IoT. en todas las presentaciones de esta tarjeta incluyen un procesador broadcom con una memoria RAM, Puertos USB, una gpu, y en las primeras versiones de las tarjetas se incluía un puerto de vídeo RCA, pero en estas últimas generaciones la tarjeta incluye un puerto de vídeo HDMI, así como también una tarjeta de tipo Ethernet, lo que hace particularmente especial esta tarjeta, es un puerto gpio, con 40 pines, para poder hacer uso de entradas y salidas digitales y analógicas, aunque todas son a un nivel de voltaje de 3.3, en las últimas versiones también incluye un conector para una cámara de tipo VGA, tal vez una desventaja que puede presentar esta tarjeta o equipo de cómputo es que en ninguna de sus

versiones o presentaciones incluye una memoria interna tipo rom, ya que se le tiene que agregar una tarjeta tipo micro SD en las últimas versiones.

El primer modelo que salió en el año 2012 fue la raspberry Pi 1 modelo A, crecía de puerto Ethernet, por lo que se tenía que hacer una adaptación de tipo wifi, por alguno de sus puertos USB, poseía solamente 26 conectores para su puerto gpio incluyendo un vídeo HDMI y también un vídeo de tipo RCA, con una salida de audio de 3.5 mm, un puerto USB y un conector para tarjeta micro SD.

Inicialmente se agregó a este modelo un procesador broadcom bcm 2835 que corría a una velocidad de 700Mhz, teniendo así 256 megabytes de memoria RAM y una tarjeta gráfica broadcom vídeo core IV, de cuarta generación. En todas las versiones de la tarjeta raspberry se requiere una fuente de alimentación externa de 5 voltios y dos amperios como mínimo.

En el año 2012 raspberry lanza un modelo como una variante del modelo A, el cual incluyo varias mejoras, agregando el doble de memoria RAM, pasando de 256 megabytes a 512 megabytes, una mejora que también agregó en este nuevo modelo fue un puerto usb más y un conector, Ethernet RJ 45, el cual hizo que esta tarjeta tuviera una mayor aceptación para los cibernautas, no incluyo variaciones en su procesador gráfico, ni agregó más pines a su puerto gpio, este nuevo modelo lo denominó como raspberry Pi modelo B y B+ pasando así de hacer uso de una tarjeta SD a una tarjeta micro SD.

En el año 2014 se lanza el primer modelo raspberry Pi2 modelo P, lo particular en esta tarjeta es que ya no incluye el mismo procesador utilizado en las 3

versiones anteriores, se reemplaza por uno de la misma marca, pero con mayor capacidad, pasa de hacer uso de un solo núcleo a hacer uso de 4 núcleos que corren a 700Mhz y 900Mhz cada uno, también se hace una mejoría significativa en el uso de la memoria RAM pasando de 512 megabytes a un gigabyte, aunque esta memoria está compartida con el núcleo gráfico y se incluye en esta versión un gpio de 40 pines y agregando 4 puertos USB.

En el 2016 se lanza la tarjeta raspberry Pi 3 modelo B el cual hace una actualización en su procesador, aún de la compañía broadcom que pasa de trabajar de una velocidad de 900Mhz a 1.20Ghz, Manteniendo 1Gb de memoria RAM, haciendo su mayor actualización el agregar una tarjeta wifi y un módulo Bluetooth, eliminando de esta forma una gran cantidad de adaptadores.

Aparte de los modelos normales la fundación raspberry ha desarrollado una serie de placas con aplicaciones especiales, denominadas raspberry Pi Zero, la característica principal de estos modelos, son el tamaño, ya que son más pequeños, obviamente tienen menos componentes, así como menores capacidades de hardware, a pesar de tener algunas carencias lo que hace que estas tarjetas tipo cero sean atractivas para los consumidores, es su precio, mucho menor que las tarjetas normales.

Pi zero, fue el primer modelo lanzado en el año 2015, es un 40% más potente que el primer modelo de raspberry, funciona a 1Ghz, manteniendo dos núcleos trabajando a 512 megabytes de RAM, debido a su tamaño sustituye el puerto HDMI por un mini HDMI, Sustituye el conector usb, de tamaño estándar, agregando dos micro USB, 1 de alimentación y 1 de datos, usando una tarjeta

micro SD como sistema de almacenamiento principal, la tarjeta sucesora de la versión Pi Zero es la Pi cero W, por el acrónimo de Wireless, se puede decir que es lo único novedoso en la versión que la de sus antecesores.

La instalación del software o sistema operativo principal se descarga directamente desde los servidores de raspberry Pi, el cual consta de un sistema de booteo, para poder hacer el montaje del sistema operativo inicialmente se debe contar con una memoria con una capacidad mínima de 8 gigas y una máxima de 16 gigas así como un lector de tarjetas y haber descargado la última versión del sistema operativo raspbian, La memoria micro SD, debe de contener el sistema operativo raspbian, así como también debe de tener conectadas los puertos de entrada y salida Como por ejemplo teclado mouse y pantalla.

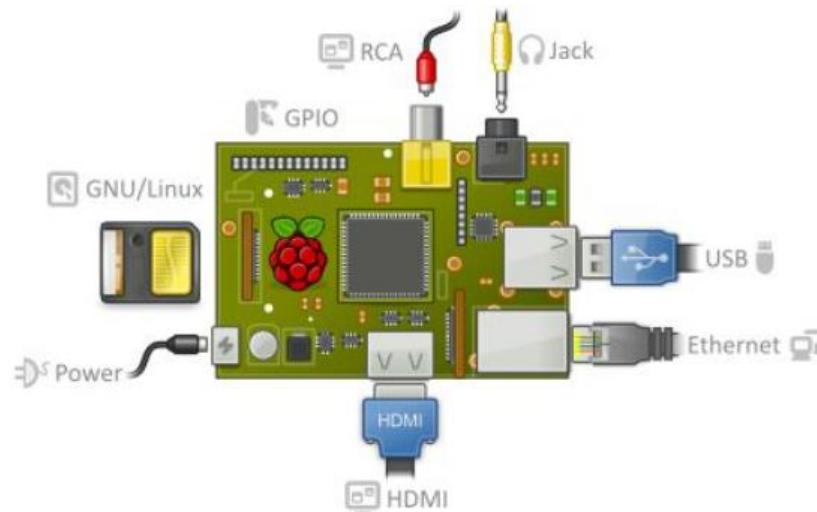


Fig.2.3 Tarjeta Raspberry pi con puertos de entrada y salida. (Sánchez, 2020)

La tarjeta Raspberry Pi usa mayormente sistemas operativos de tipo Linux, así como también distribuciones derivadas de debían, optimizadas para el hardware de esta tarjeta.

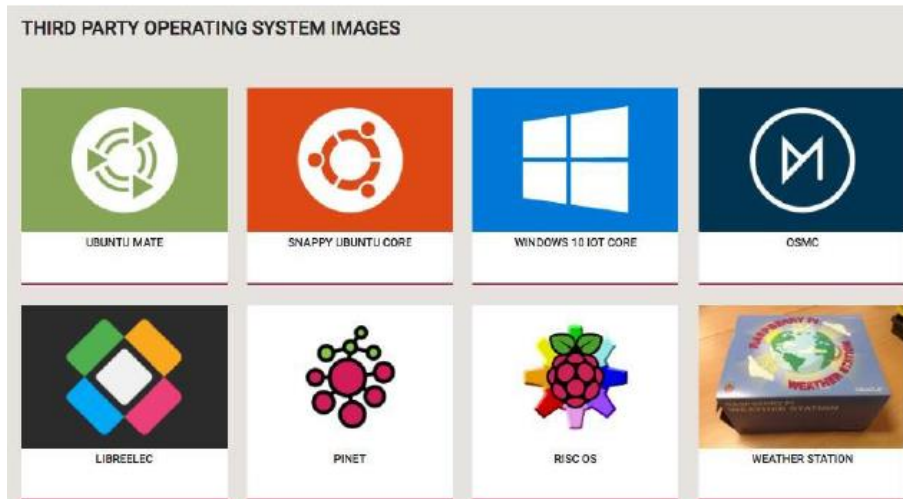


Fig.2.4 sistemas operativos oficiales para Raspberry pi (Sánchez, 2020)

La tarjeta raspberry ha tenido múltiples aplicaciones en diferentes ámbitos, desde un ordenador de mesa, con aplicaciones en pequeños proyectos hasta hacer que la raspberry funcione como un servidor de contenidos de tipo multimedia, también ha tenido aplicaciones bastante sobresalientes como por ejemplo crear un centro de medios para un salón, o aplicaciones de investigación mayormente ambiciosas como por ejemplo impresoras 3D o sistemas de escaneo, en 3 dimensiones el cual han llegado a conectar hasta 40 tarjetas Raspberry con 40 cámaras, para poder hacer un escaneo de un modelo de tipo tridimensional, otro proyecto que también llamó mucho la atención fue el de adaptar una antena especial para recepción Gps, a través de globos aerostáticos especiales para investigación, también existe un proyecto que se desarrolló para unir una tarjeta Raspberry con una pantalla táctil, así también con un módulo gsm, que permite realizar llamadas telefónicas, tal vez no sea un competidor directo para las compañías de teléfonos pero desarrollar proyectos de este tipo se evitan tener

restricciones de hardware y software que los fabricantes interponen sus productos.

El proyecto Air Pi, ha sido un desarrollo tecnológico que ha permitido la gestión y uso de diferentes tipos de sensores aplicados a la investigación meteorológica, este tipo de aplicaciones ha generado mucho interés para poder hacer el monitoreo de temperatura, humedad, presión de aire, niveles de luz, radiación ultravioleta, niveles de monóxido de carbono, dióxido de nitrógeno, etcétera, todos estos datos son usados al área de investigación meteorológica o simplemente son compartidos en la red, todo este tipo de proyectos se hacen posible por su puerto de conexión GPIO, el cual ofrece entradas y salidas digitales analógicas, puertos de comunicación serial, aplicando así protocolos de comunicación estándares y especiales en el ramo de la informática, la tarjeta Raspberry Pi se ha utilizado también para desarrollo de consolas de videojuegos, aplicaciones como emisoras de radio, al igual que se han creado proyectos de computación paralela desarrollando clúster, con más de 16 Raspberry Pi conectados bajo un mismo fin, una de las áreas de investigación de mayor interés particular es la aplicación de la tarjeta al desarrollo de la robótica, uno de los proyectos más sobresalientes de la tarjeta Raspberry es el desarrollo del robot PiCar V.

En sí como podemos observar o cómo podemos apreciar las alternativas de aplicaciones para este tipo de tarjetas no se deben limitar a su tamaño a sus capacidades sino más que nada a la capacidad de poder desarrollar aplicar e implementar esta tarjeta al desarrollo de proyectos de innovación o de

investigación, si bien para muchos la tarjeta Raspberry no es un computador con un gran éxito en el mercado, sí ha logrado ser una de las placas de desarrollo más importantes en el mercado para temas de investigación. (Sánchez, 2020)

2.1.7. Tipos de sensores

Se hace referencia al dispositivo que proporciona una respuesta (normalmente mediante la generación de una señal eléctrica) frente a estímulos o señales físicas o químicas.

Transductor: hace referencia al dispositivo que convierte una señal de una forma de energía en otra señal de naturaleza diferente. La conversión puede ser de una señal física o química en una señal eléctrica (transductor de entrada) o viceversa (transductor de salida o actuador), o incluso puede no involucrar señales eléctricas (por ejemplo, un bimetálico convierte cambios de temperatura en cambios de curvatura del dispositivo). Al realizar una medida se toma energía del medio donde se mide, y éste se perturba. Por ello, la energía tomada debe ser mínima y la señal de salida (resultado de la transformación de la energía tomada del medio) es pequeña. El dominio eléctrico de las señales de salida permite la utilización de una gran cantidad de recursos electrónicos para su tratamiento (Cis, amplificadores, filtros, etc.), transmisión o almacenamiento.

2.1.7.1. Clasificación de los sensores

- Según aporte de energía
 - Moduladores: precisan una fuente externa de alimentación.

- Generadores: toman únicamente la energía del medio donde miden.
- Según la señal de salida
- Analógicos: la salida varía de forma continua. Normalmente la información está en la amplitud. Cuando la información está en la frecuencia se denominan “cuasi-digitales”.
 - Digitales: la salida varía en pasos discretos.
- Según el modo de funcionamiento
 - Deflexión: la magnitud medida genera un efecto físico (deflexión).
 - Comparación: se intenta mantener nula la deflexión mediante la aplicación de un efecto opuesto al generado por la magnitud medida.
- Según la relación entrada-salida: orden cero, 1erorden, 2º orden
- Según el principio físico: resistivo, capacitivo, inductivo, termoeléctrico, piezoeléctrico, ...
- Según la magnitud medida: temperatura, presión, aceleración, pH ...

CARACTERÍSTICAS GENERALES

De diseño

- Magnitud medida

Naturaleza: magnitud que es detectada por el sensor (ej. Presión, aceleración, etc). A veces, la magnitud medida es usada para calcular otras magnitudes. Idealmente el sensor solo responde a una magnitud de

entrada (la salida es solo función de una magnitud) pero es frecuente que la salida se vea afectada por otras magnitudes (selectividad a una magnitud).

Rango: límites superior e inferior de la magnitud medida.

Span de entrada o fondo de escala de entrada (input full scaleFS): diferencia algebraica entre los límites superior e inferior de la magnitud medida.

- Características eléctricas

Hacen referencia a las interfaces eléctricas del sensor considerado éste como una «caja negra».

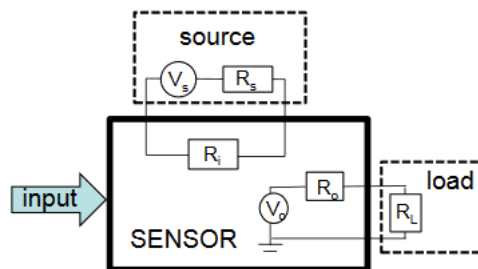


Fig.2.5 Proceso de interpretación de lecturas (Sole)

- Características eléctricas

Salida: magnitud eléctrica producida por el sensor función de la magnitud medida. Generalmente la salida es una función continua de la entrada (salida analógica) en la que la información va en la amplitud del voltaje o la corriente, o en cambios en la resistencia, la capacidad o la inducción magnética. La información puede ir en las variaciones de frecuencia, el periodo o la anchura de pulsos. En los sensores tipo switch la salida

presenta solo dos valores posibles (todo-nada). La salida digital presenta incrementos discretos codificados (ej. código binario).

Puntos finales: valores de salida para los límites inferior y superior del rango de entrada del sensor.

Span de salida (output span) o fondo de escala de salida (output full scaleFSO): diferencia algebraica entre las salidas eléctricas medidas cuando se aplican los valores máximo y mínimo de la magnitud de entrada.

Excitación o alimentación: señal eléctrica externa que suministra la potencia necesaria para activar el funcionamiento del sensor.

Generalmente se especifica como un rango de tensión o voltaje. Otras veces se indica la máxima potencia aplicable, limitada para evitar un autocalentamiento excesivo. A veces también se indica la frecuencia y la estabilidad de la fuente de alimentación. Una fuente externa es necesaria en los sensores moduladores, pero no en los sensores generadores.

Muchos sensores moduladores son ratiométricos porque la salida es proporcional a la señal de alimentación. Por ejemplo, el voltaje de salida (caída de tensión en un sensor resistivo) es el doble si se alimenta con una fuente de corriente que proporciona el doble de corriente.

Impedancia de entrada: impedancia que presenta el sensor a la fuente de alimentación. Es importante que haya un buen acoplo con la impedancia de la fuente.

Impedancia de salida: impedancia medida en los terminales de salida del sensor. Es importante que haya un buen acoplo con la impedancia de

carga (impedancia de entrada del circuito al que se conecta la salida del sensor).

- **Características mecánicas**

Definen las interfaces físicas del sensor, normalmente se indica con dibujos esquemáticos todas las dimensiones y las localizaciones de todas las conexiones mecánicas, eléctricas y de fluidos, incluyendo cualquier agujero de montaje. Encapsulados o sellados especiales deben especificarse por la norma industrial que cumplen.

Identificación, mediante la nomenclatura apropiada se señalan las características más relevantes del modelo.

De Prestaciones

- **Características estáticas**

Describen las prestaciones del sensor en condiciones ambientales normales (temperatura $25^{\circ}\text{C} \pm 10^{\circ}\text{C}$, humedad relativa $< 90\%$, presión barométrica entre 88 y 108 kPa, en ausencia de vibraciones) cuando la entrada cambia muy lentamente. Sensibilidad: es la relación entre el cambio en la salida y el cambio en la entrada. Determina la pendiente de la función de transferencia o de la curva de calibración. Offset, Zero o null: es el valor de la salida para entrada cero.

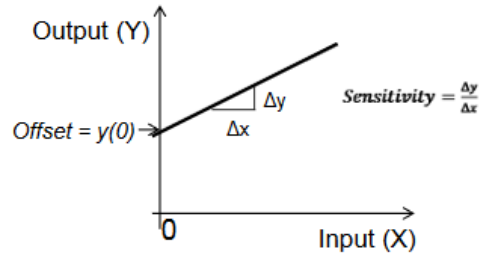


Fig.2.6 Referencias estáticas de sensibilidad. (Sole)

Resolución: es el mínimo cambio en la entrada que puede ser detectado a la salida.

La mayoría de los sensores analógicos tienen una resolución infinitesimal. En los sensores digitales, la resolución es finita y está limitada por el número de bits. Umbral: mínimo valor de la entrada que es detectado a la salida.

Repetitividad: diferencia en la salida cuando se aplican los mismos valores de entrada y en las mismas condiciones.

Reproducibilidad: hace referencia a la capacidad de obtener la misma salida cuando se aplican los mismos valores de entrada en «aparentemente» las mismas condiciones, pero por diferentes personas o en diferentes laboratorios.

Histéresis: máxima diferencia en la salida cuando los valores de la entrada se aproximan de forma creciente y luego decreciente.

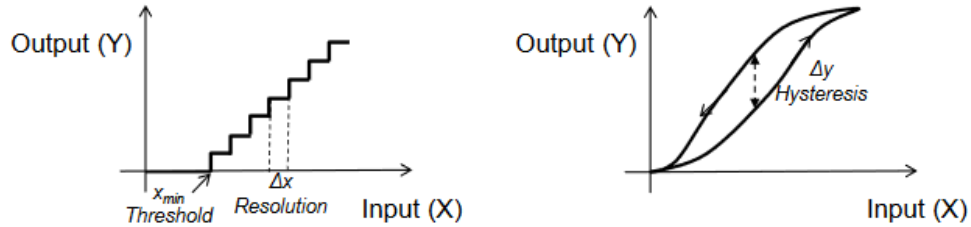


Fig.2.7 Comportamiento de histéresis con relación a la resolución. (Sole)

Linealidad: es la proximidad de la curva de calibración del sensor a una línea recta.

Debe siempre indicarse a qué recta nos referimos.

Theoretical-slope linearity: línea recta entre los puntos finales teóricos.

Terminal linearity: línea recta entre los puntos finales teóricos cuando estos van del 0 al 100% del rango y el fondo de escala de salida.

Independent linearity o best straight line: línea recta intermedia entre dos líneas rectas paralelas que envuelven todos los valores de salida en la curva de calibración.

Least-squares linearity: línea recta obtenida de minimizar el cuadrado de los residuos (desviaciones de la salida real respecto a la línea recta calculada).

Point based linearity: línea recta que pasa por un punto determinado.

La falta de linealidad puede ser debida a la existencia de saturación o zonas muertas.

Cuando la curva de calibración es inherentemente no lineal, se habla de conformidad (conformance) respecto a otro tipo de curva matemática.

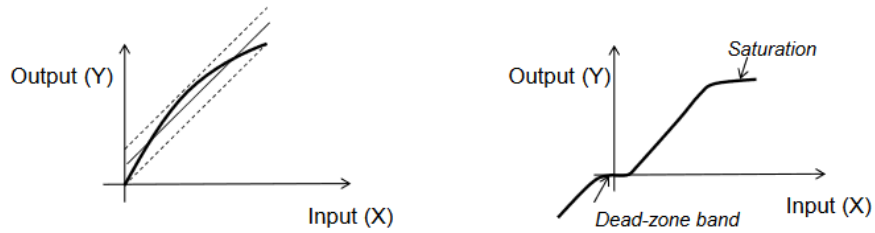


Fig.2.8 Referencia de saturación con respecto a la linealidad. (Sole)

- Características dinámicas.

Describen la respuesta del sensor a variaciones de la entrada en el tiempo, en condiciones ambientales normales.

Cuando la relación entrada-salida puede describirse a través de una ecuación diferencial lineal de coeficientes constantes, las características dinámicas pueden estudiarse a través de dicha ecuación (orden del sistema).

Respuesta en frecuencia: especifica la respuesta de un sensor ante entradas periódicas (típicamente sinusoidales).

Rango de frecuencia en el que para entradas sinusoidales la amplitud de la relación salida/entrada es constante dentro de un determinado margen de error dinámico.

Desfase, a una frecuencia dada: define el retraso, expresado en grados o rad, de la señal de salida respecto a una señal sinusoidal de entrada.

Respuesta transitoria: hace referencia a la respuesta del sensor ante entradas en salto.

Tiempo de respuesta: tiempo requerido para que la salida alcance un porcentaje especificado de su valor final (típicamente 95% o 98%).

Tiempo de subida: tiempo requerido para que la salida pase de un pequeño porcentaje a un gran porcentaje del valor final (típicamente 10% al 90%).

En un sistema de orden cero, el sensor responde instantáneamente y no es necesario especificar características dinámicas.

Un sistema de primer orden se caracteriza por f_c y τ (ambos están relacionados).

La frecuencia de corte (cuttofffrequency) es el parámetro $f_c = 1/(2\pi\tau)$ de la función de transferencia.

A esta frecuencia el error dinámico es del -30% (la amplitud de la salida se reduce en -3dB o $1/\sqrt{2}$). Para un error dinámico inferior al 5% el rango de frecuencia va de dc a $0.1f_c$.

La constante de tiempo (time constant) es el parámetro τ de la función de transferencia. Corresponde a un tiempo de respuesta del 63%.

Un sistema de segundo orden se caracteriza por, el factor de amortiguamiento (damping ratio) es el parámetro ζ de la función de transferencia y es la relación entre el amortiguamiento real y el amortiguamiento crítico del sistema. Cuando la respuesta del sensor lo más rápida posible sin overshoot, la respuesta está críticamente amortiguada (critically damped). Cuando hay overshoot la respuesta es subamortiguada (underdamped) y cuando es más lenta es sobreamortiguada (overdamped).

La Frecuencia natural (natural frequency) es el parámetro ω_n de la función de transferencia. A esta frecuencia el desfase entre la salida y la entrada es de -90° . En un sistema subamortiguado con $\zeta < 0.7$, existe una frecuencia de resonancia (resonant frequency) en la que la amplitud de salida de señales sinusoidales es máxima. Generalmente el rango de frecuencias de funcionamiento se selecciona suficientemente más bajo (al menos un 60%) de la frecuencia de resonancia, aunque algunos sensores tienen su punto de funcionamiento en la frecuencia de resonancia (mucha sensibilidad en un estrecho ancho de banda).

En respuestas transitorias oscilatorias (subamortiguadas), generalmente se habla de tiempo de establecimiento (settling time) en el que la amplitud de la oscilación es suficientemente pequeña, en lugar de tiempo de respuesta.

2.1.8. Motorreductores POLOLU

Este motorreductor consiste en un motor de corriente continua cepillado de 12 V de alta potencia combinado con una caja de engranajes cilíndricos de metal 34.014: 1, y tiene un codificador de cuadratura de 48 CPR integrado en el eje del motor, que proporciona 1632.67 recuentos por revolución del eje de salida de la caja de cambios.

El motorreductor es cilíndrico, con un diámetro de poco menos de 25 mm, y el eje de salida en forma de D tiene un diámetro de 4 mm y se extiende 12,5 mm desde la placa frontal de la caja de cambios.

Especificaciones clave

Voltaje sin carga rendimiento y extrapolación de pérdida.

Tabla 2.1 Especificaciones generales de motor Pololu (Corporation ©. 2.–2., 2020)

Voltaje	Rendimiento sin carga	Extrapolación
12v	290 RPM, 300 mA	8.5 kg·cm (120 oz·in), 5.6 A

Especificaciones técnicas

Dimensiones

Tabla 2.2 Dimensiones de motor Pololu, datos del fabricante (Corporation ©. 2.–2., 2020)

Tamaño	25D x 67L mm1
Peso	101 g
Diámetro del eje	4 mm

Especificaciones generales

Tabla 2.3. Especificaciones de intensidad de corriente en vacío y a plena carga (Corporation ©. 2.–2., 2020)

Relación de transmisión	34.014: 1
Velocidad sin carga @ 12V	290 rpm

Corriente sin carga @ 12V	300 mA
Corriente de bloqueo @ 12V	5600 mA
Par de parada @ 12V	120 oz · in
Velocidad sin carga a 6V	145 rpm
Corriente de bloqueo a 6 V	2800 mA
Par de parada a 6V	60 oz · in
Tipo de motor	pérdida de 5.6A @ 12V (HP 12V)
Longitud de plomo	8 in
Codificadores	Y

La resistencia eléctrica del motor se puede aproximar dividiendo el voltaje nominal por la corriente de bloqueo (al voltaje nominal). La constante de fuerza electromotriz (K_e) se puede aproximar dividiendo el voltaje nominal por la velocidad de funcionamiento libre (al voltaje nominal). Para aproximar la constante de par motor (K_t), puede dividir el par de bloqueo por la corriente de bloqueo.

Para casi cualquier motor de CC, las curvas de corriente, velocidad, potencia y eficiencia en función del par se verán como las del gráfico a continuación (suponiendo que el voltaje y la temperatura del motor sean constantes).

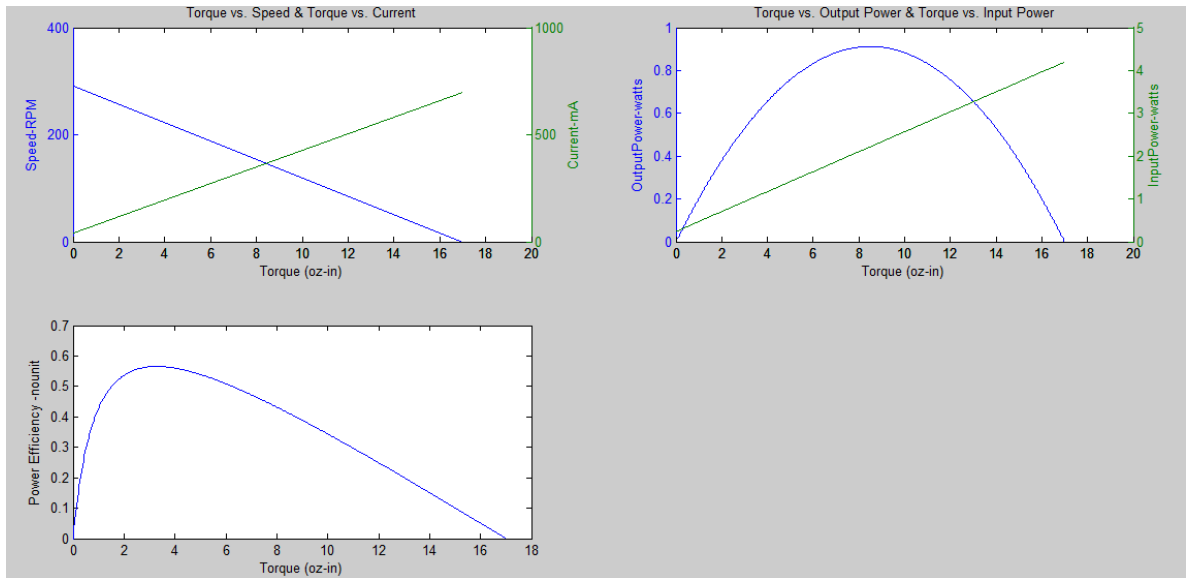


Fig.2.9 Curvas de salida de motor pololu, Simulación el Matlab. (Corporation ©. 2.-2., 2020)

La curva de potencia de salida es el producto de la velocidad y el par, lo que da como resultado una parábola invertida con su pico al 50% del par de parada.

La curva de eficiencia es la potencia de salida dividida por la potencia de entrada, donde la potencia de entrada es la corriente por el voltaje. El voltaje es constante, por lo que puede dividir la curva de potencia de salida por la línea de corriente para obtener la forma general de la curva de eficiencia, que a su vez le permite identificar el par, la velocidad y la corriente que corresponden a la eficiencia máxima.

Existen muchos programas que se pueden usar para generar estas curvas. Por ejemplo, si se tiene acceso a MATLAB, se puede usar un script MATLAB, para generar estos gráficos de motor a partir de las especificaciones que especifican

para cada motorreductor. Para descargar el script pulse la siguiente liga:

<https://www.mathworks.com/matlabcentral/fileexchange/54695-polulu-motor-plot-generator>

Uso del codificador

Se usa un codificador de efecto Hall de dos canales para detectar la rotación de un disco magnético en una protuberancia trasera del eje del motor. El codificador de cuadratura proporciona una resolución de 48 conteos por revolución del eje del motor al contar ambos bordes de ambos canales. Para calcular los conteos por revolución de la salida de la caja de engranajes, multiplique la relación de engranajes por 48. El motor / codificador tiene seis cables codificados por color de 8 "(20 cm) terminados en un encabezado hembra de 1 × 6 con un paso de 0.1".

Tabla 2.4 Código de colores para cables de alimentación y sensor de efecto hall motor Pololu. (Corporation ©. 2.-2., 2020)

Color	Función
Rojo	Potencia del motor (se conecta a un terminal del motor)
Negro	potencia del motor (se conecta al otro terminal del motor)
Verde	codificador GND
Azul	codificador Vcc (3.5 V a 20 V)
Amarillo	salida del codificador A
Blanco	salida del codificador B

El sensor Hall requiere un voltaje de entrada, V_{cc} , entre 3.5 y 20 V y consume un máximo de 10 mA. Las salidas A y B son ondas cuadradas de 0 V a V_{cc} aproximadamente 90° fuera de fase. La frecuencia de las transiciones le indica la velocidad del motor, y el orden de las transiciones le indica la dirección. (Corporation ©. 2.–2., 2020)

2.1.9. Tarjetas Arduino

Arduino es una de las plataformas más importantes basadas en hardware libre, hace uso de un microcontrolador de la marca AVR, montado en una placa PCB el cual incluye todos los elementos principales para poder hacer más fácil el uso de esta tarjeta. Arduino nació siendo un hardware completamente libre, dispone de un entorno de desarrollo IDE, el cual está basada en el lenguaje C++, todas estas ventajas hacen que las placas Arduino sean muy económicas y bastante accesibles a cualquiera que quiera emprender proyectos de desarrollo, en el área de la electrónica la robótica y la investigación. Estas placas Arduino hacen uso de un programa con un simple cable USB, sin la necesidad de programadores especiales como sus competidores principales, esta facilidad de programación es posible, gracias a un bootloader que se carga en un microcontrolador, permitiendo así crear una interfaz completamente fácil y en tiempo real.

La interfaz de Arduino es compatible con múltiples sistemas operativos como por ejemplo Windows Linux y Mac.

El proyecto data del año 2005 bajo al desarrollo de algunos estudiantes del Instituto de diseño interactivo IVREA, en Italia, inicialmente se utilizó un lenguaje

tipo Basic a través de un microcontrolador pic, obviamente los costos en aquellos años eran bastante elevados, por arriba de los €100, con el paso del tiempo un estudiante de origen colombiano, desarrolla una tarjeta electrónica con un lenguaje de programación y una plataforma de desarrollo, los estudiantes italianos se basaron en el trabajo del estudiante colombiano y así juntos desarrollaron una plataforma de hardware y software libre pequeña y económica llamándola Arduino.

Estas tarjetas Arduino con el paso del tiempo, tuvieron una aceptación fuera de serie, y en poco tiempo, desplazaron a sus principales competidores como por ejemplo adk, BasicStamp, microcontroladores pic, fue tanto el gusto de esta tarjeta de desarrollo que inclusive Google colaboró para desarrollar una placa capaz de comunicarse con dispositivos móviles, bajo el sistema operativo, Android.

Una vez iniciada la producción de la primera versión, de esta tarjeta, se tuvo una meta inicial de no superar los €30, para su disposición en los mercados, al igual que teniendo las facilidades iniciales de que sea compatible con múltiples sistemas operativos y que fuese con sistema de programación plug and play. En el 2011 en la feria maker se presenta al público la primera placa Arduino de 32 bits teniendo un rotundo éxito y logrando colaboraciones con empresas como Intel y así poder desarrollar la placa Galileo.

Arduino tuvo un éxito rotundo y como cualquier proyecto exitoso, aparecieron clones con arquitecturas compatibles a los Arduino originales, por ejemplo, pingüino, basado en el pic18F teniendo compatibilidad con hardware y software.

Modelos de Arduino disponibles en el mercado.

- Arduino Uno
- Arduino Leonardo
- Arduino Due
- Arduino Yún
- Arduino Tre (En Desarrollo)
- Arduino Zero (En venta en la tienda de EEUU)
- Arduino Micro
- Arduino Esplora
- Arduino Mega ADK
- Arduino Ethernet
- Arduino Mega 2560
- Arduino Robot
- Arduino Mini
- Arduino Nano LilyPad
- Arduino Simple LilyPad
- Arduino SimpleSnap LilyPad
- Arduino LilyPad
- Arduino USB
- Arduino Pro Mini
- Arduino Fio
- Arduino Pro

2.1.10. Puente H y PWM

Se trata de un circuito electrónico (intersección de dos o más componentes que poseen una trayectoria cerrada) que permite a un motor eléctrico de corriente directa, cambiar de sentido. Estos sentidos a los que puede cambiar consisten en girar, ir al ritmo de las manecillas del reloj o viceversa, así como ir en ambos sentidos.

Entre sus presentaciones pueden hallarse en forma de circuitos integrados, en donde sus componentes, entiéndase transmisores y resistencia se encuentran sobre una lámina de material semiconductor; pero también pueden elaborarse partiendo de componentes discretos. Por lo general, son utilizados como circuitos fundamentales dentro de la robótica, pero también pueden encontrarse en convertidores de potencia.

Por lo general, la manera más común y efectiva de construir un puente H consiste en emplear interruptores de estado sólido los cuales conmutan (sustituyen) el paso de electricidad cuando una pequeña corriente es aplicada en sus terminales de control.

Estos interruptores se acompañan de diodos que le brindan a la corriente la posibilidad de circular en sentido inverso al previsto cuando se conmuta la tensión. Esto ocurre porque el motor posee bobinas que durante breves periodos de tiempo ejercerán oposición cuando varíe la corriente.

Se le denomina puente H porque su representación gráfica tiene forma de H, donde los interruptores mecánicos se localizan a los lados y el motor se ubica en el centro.

Este circuito se puede elaborar utilizando transistores o partiendo de 4 interruptores mecánicos. Estos últimos se configuran de tal manera que cuando la corriente atraviesa a dos de ellos, se genera un giro positivo. Por su parte, la inversión del giro del motor ocurre cuando los interruptores contrarios se cierran ante la inversión del voltaje.

El puente H se puede usar para frenar de manera brusca, siendo una función más allá de inversión del motor. Este frenado de emergencia se logra al hacer un corto entre las bornas del motor. También, al desconectar el motor de la fuente que lo alimenta el motor puede frenar bajo su propia inercia.

2.1.11. Comunicación serial.

La norma RS-232C fue definida por la Asociación de Industrias en Electrónica, EIA (Electronic Industries Association), en 1969 en lo que es la definición del conector y del cable. Esta norma es idéntica a la norma V.24 definida por el UIT (International Telecommunication Union).

También se ha de destacar que, aunque la norma RS-232C fue creada para comunicar un ordenador con un módem, hoy en día este estándar se utiliza también a nivel industrial para comunicar punto a punto diferentes componentes

y equipos, como reguladores, autómatas programables, lectores de códigos de barras, variadores de frecuencia, etc.

La revisión que se realizó en 1987 a la norma RS-232C fue creada como RS-232D, donde se introducen nuevas señales destinadas a la especificación de determinadas condiciones de prueba; dos de esas señales utilizan las conexiones 18 y 25 que en el antiguo estándar (RS-232C) estaban sin asignación.

El estándar RS-232C cubre cuatro áreas, que son:

- Características mecánicas de la interfaz.
- Paso de señales eléctricas por la interfaz.
- Función de cada señal y subconjunto de señales para ciertas aplicaciones.

Características mecánicas de la interfaz

De lo que se ocupa la sección mecánica es definir:

- La asignación de señales a los 25 contactos del conector.
- Qué parte del equipo contiene el conector hembra (el DCE u ordenador).
Por tanto, el conector macho para el DTE o módem.
- La longitud máxima recomendada del cable (15 metros).
- La máxima capacidad del cable (2500 pF).

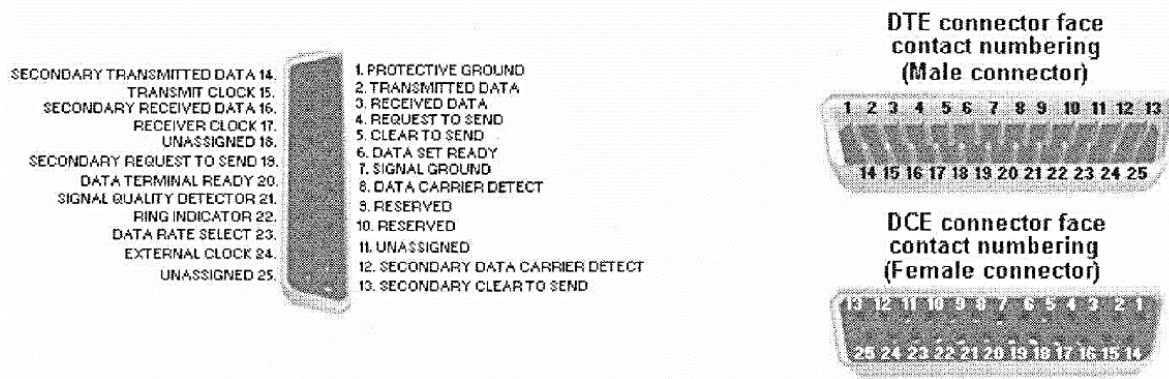


Fig.2.10 Conector para la comunicación serial bajo la norma RS-232 (Vicente Guerrero, 2015)

Paso de señales eléctricas por la interfaz

El estándar RS-232 fue creado para realizar comunicaciones serie punto a punto a velocidades ≤ 20 kbps. Lo que realmente está limitado es la longitud del pulso, siendo esta longitud inversamente proporcional a la velocidad, por lo que podemos aumentar la velocidad si disminuimos la longitud del cable o bien aumentar la longitud del cable si disminuimos la velocidad de transmisión.

Sólo el 4% del tiempo de cada periodo de bit es el que se debe utilizar para realizar una transición (paso de -3 a +3 voltios, o viceversa).

La capacidad del conductor se comporta como un freno a los cambios de tensión.

Si la velocidad es muy rápida, la capacidad del conductor puede provocar lecturas erróneas de la información transmitida. La norma establece que la velocidad máxima es de 20 kbps y que la capacidad del cable no debe superar

los 2.500 pF, como los cables que se suelen utilizar tienen una capacidad de entre 130 y 170 pF por metro de longitud.

El siguiente paso es representar la señal eléctrica ideal sin ninguna interferencia, es decir, que si se quiere enviar un nivel alto ("1" lógico) la tensión a enviar será de -25 voltios; y si por el contrario es un bit de nivel bajo ("0" lógico), la tensión deberá ser de + 25 v.

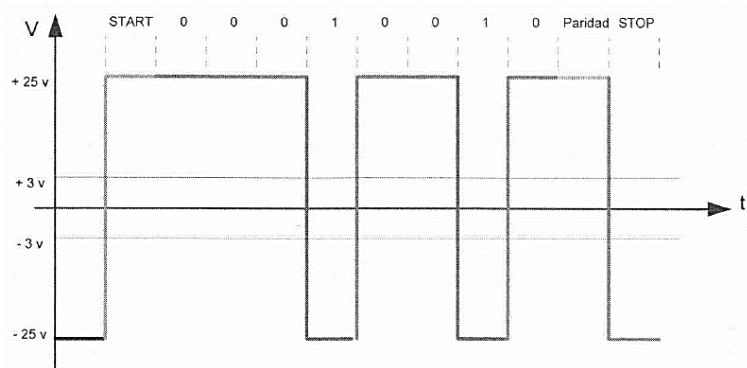


Fig.2.11. Señal eléctrica ideal del ASCII "H" por la interfaz RS-232. (Vicente Guerrero, 2015)

Tipo de transmisión

En la norma RS-232 se utiliza el tipo de transmisión simple, también llamado Single- Ended, que se caracteriza por:

- Utilizar un único hilo o cable, referenciado a una masa común (no balanceado o equilibrado).
- El dato se obtiene de la señal que llegue por ese hilo.
- Se emplea para distancias cortas y bajas velocidades de transmisión.

A continuación, se muestra cómo afecta el posible ruido, que se pueda incorporar al canal de comunicación, a los equipos que como la norma RS-232 son del tipo de transmisión simple.

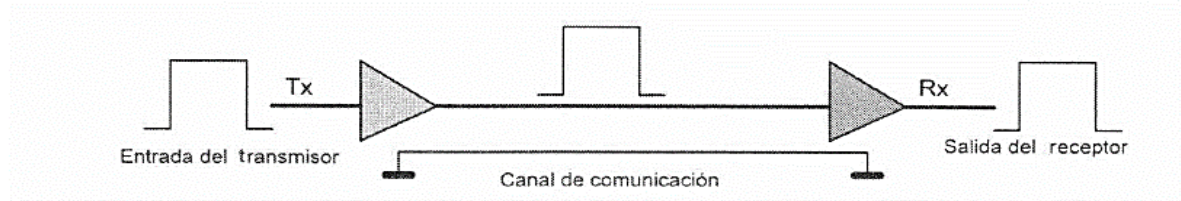


Fig.2.12. Transmisión RS-232 sin interferencias en el canal de comunicación.

(Vicente Guerrero, 2015)

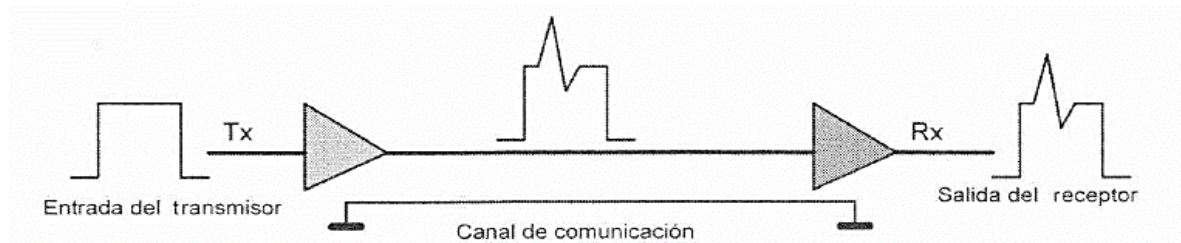


Fig.2.13. Transmisión RS-232 con interferencias en el canal de comunicación

(Vicente Guerrero, 2015)

Lo que nos indica que es un sistema bastante vulnerable a las posibles interferencias y, por tanto, se limita a comunicaciones de cortas distancias.

(Vicente Guerrero, 2015)

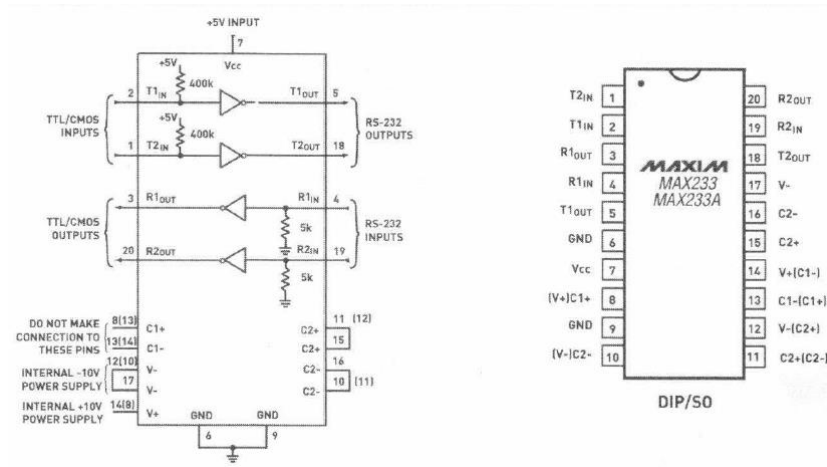


Fig.2.14. Circuitos comerciales que integran la norma RS-232. (Vicente Guerrero, 2015)

2.1.12. Baterías LIPO.

Las baterías LiPo (abreviatura de Litio y polímero), son baterías recargables usadas en aplicaciones que demandan consumo de corriente considerable en poco espacio y peso, por ejemplo, sistemas de radiocontrol, especialmente los aviones, helicópteros y multicopteros.

La vida útil de la batería LiPo es estimada en 3 años o unas 500 cargas completas. Al comparar las baterías LIPO con las baterías de NiCd/NiHm, se pueden resaltar algunas características:

- Las baterías LiPo son ligeras y son fabricadas en diversas formas y tamaños.
- Las baterías LiPo tienen una mejor relación tamaño eficiencia

- Las baterías LiPo tiene una alta tasa de descarga alta logrando alimentar sistemas más exigentes.

También existen algunos problemas con las baterías LiPo

- Problemas de seguridad a causa de sus componentes, estas baterías corren riesgo de incendiarse o explotar.
- Estas baterías requieren un cuidado único y adecuado en los procesos de carga, descarga y almacenamiento para no afectar su vida útil.
- Las baterías LiPo pueden presentar un inflamamiento debido al aumento de la temperatura, el cual se puede presentar por someterla a cargas que demanden corrientes superiores a la corriente máxima de descarga o al usar una corriente de carga superior a la corriente nominal de la batería.

Las baterías LiPo se componen de celdas de 3,7V las cuales al estar totalmente cargadas llegan a los 4,2V.

Las celdas en las baterías LiPo se conectan en serie encontrándose comercialmente baterías de diversos voltajes:

- $V = 1$ celda (1S)
- $7.4 V = 2$ celdas x 3.7 voltaje (2S)
- $11.1 V = 3$ celdas x 3.7 voltaje (3S)
- $14.8 V = 4$ celdas x 3.7 voltaje (4S)
- $18.5 V = 5$ celdas x 3.7 voltaje (5S)
- $22.2 V = 6$ celdas x 3.7 voltaje (6S)

Para aumentar la corriente las baterías se pueden conectar en paralelo, esto se indica mediante un número seguido de una "P". Ejemplo 3S2P indica 2 baterías conectadas en serie de tres celdas.

Capacidad de las baterías LiPo

La capacidad indica cuanta corriente puede suministrar la batería y se mide en miliamperios (mAh). Es una manera de indicar la cantidad de carga medida en miliamperios que puede alimentar la batería durante 1 hora antes que se descargue completamente. Por ejemplo, una batería LiPo de 1000 mAh sería completamente descargada en una hora con una demanda de 1000 miliamperios. Si ésta misma batería tenía una demanda de 500miliamperios tomaría 2 horas para descargarla. Tasa de descarga o "C" La tasa de descarga es la rapidez con que una batería puede ser descargada de forma segura. Cuanto más rápido los iones fluyen del ánodo al cátodo indicará la velocidad de la descarga. Una batería con una tasa de descarga de 10C se descarga a un ritmo de 10 veces la capacidad de la batería, 15C = 15 veces, 20C = 20 veces, y así sucesivamente.

Resistencia interna

La resistencia interna de una celda de LiPo típicamente es de 2 a 6 miliohms (0.002 a 0.006ohms), esta resistencia aumenta con la edad de la batería disminuyendo la tasa de descarga de la batería.

Cuidados para tener en cuenta en el uso de las baterías LiPo

La corriente de carga no debe ser superior a la corriente de carga nominal de la batería, el voltaje de cada celda no debe dejarse caer a un valor inferior a 3V.

Cuando las baterías se van a almacenar se recomienda que el voltaje de cada celda sea de 3.8V, esto lo garantiza la función de carga STORE presente en algunos cargadores comerciales. Las celdas de la batería deben tener el mismo valor de voltaje, de no ser así es recomendable balancear las celdas antes de iniciar el proceso de carga. (Electronics, 2014)

2.1.13. Dibujo Asistido por Computadora, CAD.

Los dibujos técnicos se crean utilizando una gran variedad de instrumentos que van desde herramientas tradicionales como lápices con pases y escuadras hasta la computadora las herramientas de dibujos se emplean para realizar dibujos y modelos legibles y exactos los instrumentos de dibujo tradicionales siguen siendo importantes en particular para dibujar croquis sin embargo en la actualidad la computadora puede utilizarse para satisfacer la mayor parte de los requerimientos de dibujo y modelado.

Elio el diseño asistido por computadora sí incluye hardware software y dispositivos periféricos relacionados con ellos y el equipo tradicional empleado normalmente por ingenieros y técnicos para realizar dibujos y modelos técnicos.

las herramientas utilizadas para comunicar de manera gráfica ideas técnicas se han ido transformando con el paso del tiempo del mismo modo que el lenguaje de las gráficas lo ha hecho en un conjunto sofisticado de estándares convencionales las herramientas se utilizan para producir 3 tipos básicos de dibujos, Croquis a mano libre dibujos de instrumentos y dibujos y modelos por computadora las herramientas han evolucionado desde lápices escuadra

escalímetros y compases hasta sistemas de diseño o dibujo asistido por computadora CAD.

El CAD es un software de computadora que junto con el hardware relacionado con él complementa o reemplaza las herramientas tradicionales manuales en la creación de modelos y dibujos técnicos.

Puesto que muchas industrias todavía no integran de manera completa el CAD, en sus oficinas, es necesario aprender los métodos de diseño tanto tradicionales como por computadora. Por otra parte, las herramientas tradicionales se utilizan para elaborar croquis o bosquejos, uno de los métodos más eficientes disponibles para representar ideas de diseño con rapidez.

Las herramientas tradicionales continuarán siendo útiles en la elaboración de croquis y el trabajo de presentación poco detallado, sin embargo, un buen software de CAD puede crear virtualmente cualquier tipo de dibujo técnico.

Los comandos de dibujo de círculos reemplazarán el compás, los comandos para el trazo de líneas, la regla T y las escuadras y los comandos de edición, los compases de división y la máscara de borrado.

Un sistema de CAD está formado por dispositivos de hardware empleados en combinación con software específico, el hardware de un sistema de CAD, consiste en dispositivos físicos empleados para poder dar soporte al software CAD. Existen muchos fabricantes de hardware y tipos de dispositivos de hardware, los cuales pueden utilizarse para crear, guardar o imprimir dibujos y modelos

técnicos. No es raro encontrar en la industria un sistema de CAD, con dispositivos para entrada, almacenamiento, y salidas múltiples. (Bertoline)

El software SOLIDWORKS

El software CAD SOLIDWORKS es una aplicación de automatización de diseño mecánico que les permite a los diseñadores croquizar ideas con rapidez, experimentar con operaciones cotas, y producir modelos dibujos detallados. Este documento analiza los conceptos y la terminología utilizados en toda la aplicación SolidWorks. Permite al usuario familiarizarse con las funciones que se emplean habitualmente en SOLIDWORKS.

SOLIDWORKS emplea un procedimiento de diseño en 3D. Al diseñar una pieza, desde el croquis inicial hasta el resultado final, está creando un modelo en 3D. A partir de este modelo, puede crear dibujos en 2D o componentes de relaciones de posición que consten de piezas o sub-ensamblajes para crear ensamblajes en 3D. También puede crear dibujos en 2D a partir de los ensamblajes en 3D. Cuando diseñe un modelo con SOLIDWORKS, puede visualizarlo en tres dimensiones para ver su aspecto una vez fabricado.

2.1.14. Sistemas de referencia vectorial.

Coordenadas cartesianas en el espacio tridimensional

Hemos llegado a una transición importante en nuestro estudio del cálculo. Hasta ahora, hemos viajado a través de este ámbito amplio y llano conocido como plano

euclidiano, o espacio bidimensional. Hemos aplicado los conceptos del cálculo a funciones de una sola variable, cuyas gráficas se pueden trazar en el plano. Vamos a estudiar el cálculo de variables en tres dimensiones. Nuevamente exploraremos todas las ideas familiares (como límite, derivada, integral) desde una perspectiva superior.

Para comenzar, considere tres ejes de coordenadas mutuamente perpendiculares (los ejes x , y , z) con sus ceros en un punto común O , llamado el *origen*. Aunque estas líneas se pueden orientar de cualquier forma deseada, seguiremos la costumbre de pensar en los ejes y y z como si estuvieran en el plano del papel, con sus direcciones positivas hacia la derecha y hacia arriba, respectivamente. Entonces, el eje x es perpendicular al papel y suponemos que su extremo positivo apunta hacia nosotros, formando así un **sistema de mano derecha**. Le llamamos así porque, si los dedos de la mano derecha se doblan de modo que se curve desde el eje x positivo hacia el eje y positivo, el pulgar apunta en la dirección del eje z positivo figura 2.15, inciso a.

Los tres ejes determinan tres planos: yz , xz , y xy , que dividen al espacio en ocho octantes (figura 2) A cada punto P del espacio le corresponde una terna ordenada de números (x, y, z) , sus **coordenadas cartesianas**, que miden sus distancias dirigidas a los tres planos, figura 2.15, inciso c.

La ubicación de puntos en el primer octante (el octante donde las tres coordenadas son positivas) es relevantemente fácil. En la figura 2.15 en los incisos d y e, se ilustra algo más difícil, localizando dos puntos de otros octantes, los puntos $P(2, -3, 4)$ y $Q(-3, 2, -5)$.

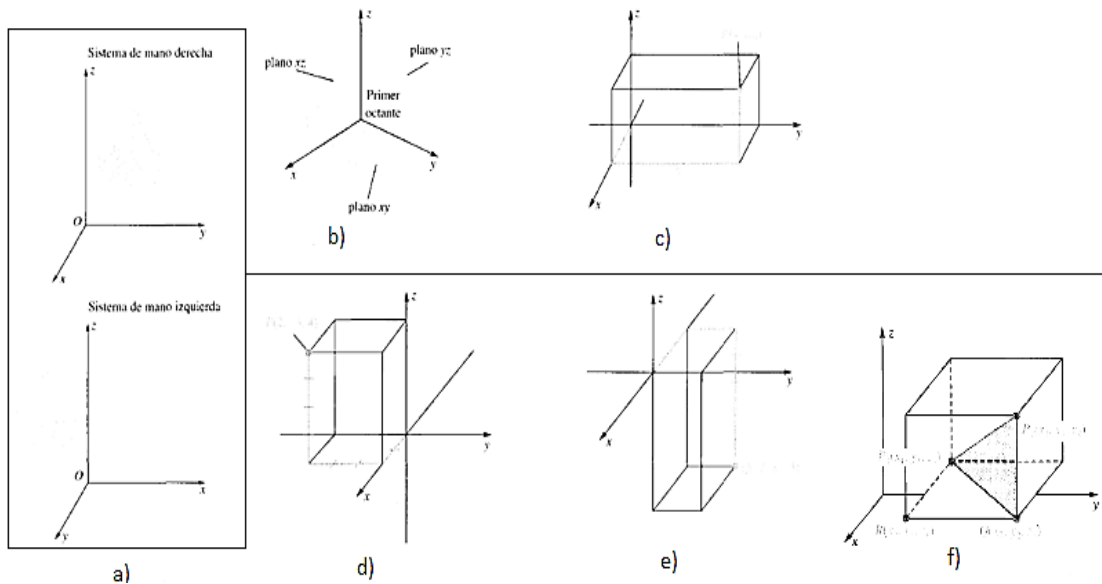


Fig. 2.15. sistema de referencia en el sistema vectorial, plano 3D. a) Sistema de referencia de la mano derecha e izquierda. b) Cuadrantes del plano en 3D. c) Cubo en el plano 3D positivos. d)Cubo en cuadrantes positivos y negativos en el plano 3D. e) Cubo en los cuadrantes negativos. f) Localización de los puntos P y Q, en el sistema de coordenadas vectoriales. (Edwin J. Purcell, 2007)

La fórmula de la distancia. Considere dos puntos $P_1 (x_1, y_1, z_1)$ y $P_2 (x_2, y_2, z_2)$ en el espacio tridimensional ($x_1 \neq x_2, y_1 \neq y_2, z_1 \neq z_2$). Ellos determinan un **paralelepípedo** (es decir, una caja rectangular) con P_1 y P_2 como vértices opuestos y con aristas paralelas a los ejes de coordenadas (figura 6). Los triángulos P_1QP_2 y P_1RQ son triángulos rectángulos y, por el tema de Pitágoras.

$$|P_1P_2|^2 = |P_1Q|^2 + |QP_2|^2$$

y

$$|P_1Q|^2 = |P_1R|^2 + |RQ|^2$$

Así,

$$\begin{aligned} |P_1P_2|^2 &= |P_1R|^2 + |RQ|^2 + |QP_2|^2 \\ &= (x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2 \end{aligned}$$

Esto da la **fórmula de la distancia** en el espacio tridimensional, que es correcta, aunque algunas de las coordenadas sean idénticas.

$$|P_1P_2| = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

Vectores

Muchas cantidades que aparecen en la ciencia (por ejemplo, longitud, masa, volumen y carga eléctrica) pueden identificarse con un solo número. Estas cantidades (y los números que los miden) se denominan **escalares**. Otras cantidades, tales como velocidad, fuerza, torca y desplazamiento, requieren una magnitud y una dirección para especificarlos por completo. A tales cantidades les llamamos **vectores** y los representamos mediante flechas (segmentos de recta dirigidos). La longitud de la flecha representa la **magnitud**, o longitud, del vector; su dirección es la **dirección** del vector. El vector en la figura 2.16 tiene longitud de 2.3 unidades 30° al noreste (o 30° respecto al eje x positivo).

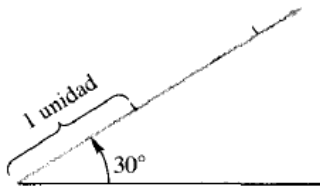


Fig. 2.16. longitud, dirección y magnitud de un vector. (Edwin J. Purcell, 2007)

Las flechas que dibujamos, como las que se disparan desde un arco, tienen dos extremos. Aquél en donde va la punta (el punto inicial), denominado **cola**, y la punta (el punto terminal) denominado la **cabeza** o punta figura 2.17 inciso a. Dos vectores se consideran **equivalentes** si tienen la misma magnitud y dirección, figura 2.17 inciso b. Por medio de letras en negritas como **u** y **v** simbolizaremos a los vectores. Como esto es difícil hacerlo en la escritura normal, usted podrá emplear \vec{u} y \vec{v} . La magnitud, o longitud, de un vector **u** se simboliza con $\|\mathbf{u}\|$.

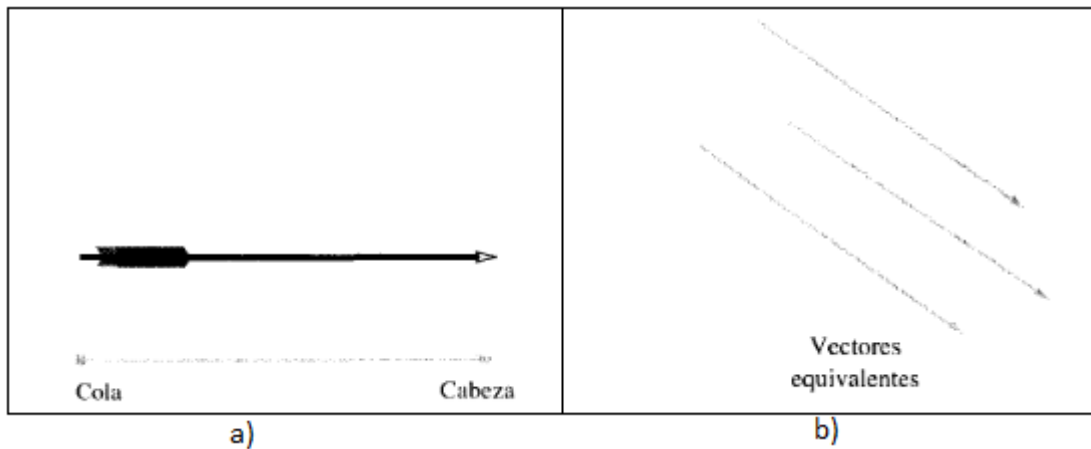


Fig. 2.17. Flechas de dirección, analogía para la representación de un vector.
(Edwin J. Purcell, 2007)

En general, consideramos a los vectores en tres dimensiones, esto es, sus puntos inicial y final son puntos en el espacio tridimensional. Sin embargo, existen muchas aplicaciones en donde los vectores se encuentran en el plano xy . El contexto del problema indicará si los vectores son de dos o de tres dimensiones.

Operaciones con vectores. Para determinar la **suma**, o **resultante**, de **u** y **v**, mueva **v**, sin cambiar su magnitud ni su dirección, hasta que su punto inicial coincida con el punto final de **u**. Entonces **u + v** es el vector que conecta la cola de **u** con la cabeza **v**. Este método (llamado *ley del triángulo*) se ilustra en la parte izquierda de la figura 2.18.

Como una forma alternativa para determinar **u + v**, mueva **v** de modo que su punto inicial coincida con el de **u**. Luego **u + v** es el vector con ese punto inicial común y que coincide con la diagonal del paralelogramo que tiene a **u** y a **v** como lados. Este método (denominado *ley del paralelogramo*) se ilustra en la parte derecha de figura 2.18.

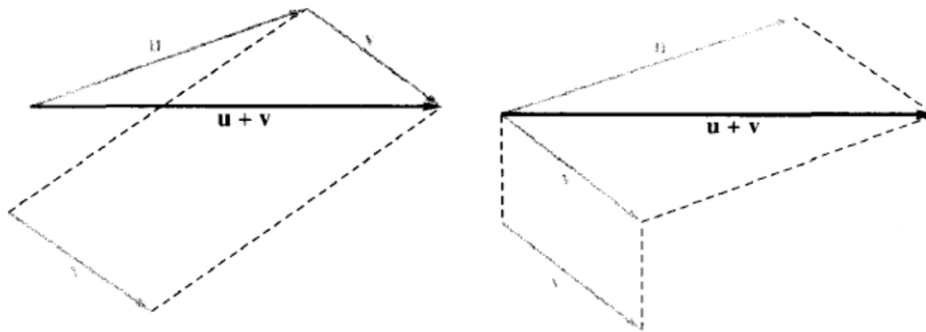


Fig. 2.18. Operaciones equivalentes con vectores. (Edwin J. Purcell, 2007).

Estos dos métodos son formas equivalentes de definir lo que queremos decir con la suma de dos vectores. Debe convencerse de que la suma de vectores es conmutativa y asociativa; esto es

$$\mathbf{u} + \mathbf{v} = \mathbf{v} + \mathbf{u}$$

$$(\mathbf{u} + \mathbf{v}) + \mathbf{w} = \mathbf{u} + (\mathbf{v} + \mathbf{w})$$

Si \mathbf{u} es un vector, entonces $3\mathbf{u}$ es el vector con la misma dirección que \mathbf{u} pero tres veces más largo; $-2\mathbf{u}$ es el doble de largo pero dirigido en sentido opuesto (figura 2.19) En general, $c\mathbf{u}$, denominada **multiplicación por escalar** de \mathbf{u} , tiene magnitud $|c|$ veces la de \mathbf{u} , y está dirigido en la misma dirección o en sentido opuesto, dependiendo de si c es positiva o negativa. En particular, $(-1)\mathbf{u}$ (por lo regular se escribe $-\mathbf{u}$) tiene la misma longitud que \mathbf{u} , pero en dirección opuesta. Se denomina el **negativo** de \mathbf{u} , ya que cuando lo sumamos a \mathbf{u} , el resultado es un vector que no es más que un punto. Este último vector (el único vector sin una dirección bien definida) se denomina **vector cero** (o vector nulo) y se denota con $\mathbf{0}$. Es el elemento identidad para la suma; esto es, $\mathbf{u} + \mathbf{0} = \mathbf{0} + \mathbf{u} = \mathbf{u}$. por último, la resta se define por medio de $\mathbf{u} - \mathbf{v} = \mathbf{u} + (-\mathbf{v})$. (Varberg)

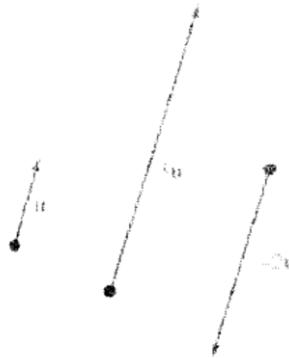


Fig.2.19 Vector con sentido definido (Edwin J. Purcell, 2007)

2.2. METODOLOGÍA DE LA INVESTIGACIÓN

2.2.1. TIPO DE INVESTIGACIÓN CUASI EXPERIMENTAL

Es la investigación que se apega al objetivo del proyecto, ya que los resultados de la investigación se enfocan a la situación en el cual no es posible el control y la manipulación absolutos de las variables.

1. **Definición y planteamiento del problema:** El sistema de control y navegación que se está desarrollando va a ser sometido a pruebas experimentales internas y externas en instalaciones de ITST. En el cual múltiples factores estarán interfiriendo con el desempeño de navegación del dispositivo, por ejemplo, luz solar, la forma no lineal de las instalaciones, los pasillos, las escaleras, etc. Esto hace que se cumpla una característica de este método experimental.
2. **Formulación de la Hipótesis:** Por medio de los sistemas de control implementados para el desarrollo de este prototipo, se espera que se llegue a tener una alta precisión en la navegación autónoma, así como para poder hacer un mapa y un trazo para la trayectoria y punto final del sistema de navegación. Se espera que este sistema de control el cual será implementado en una problemática real tenga los resultados que se desean obtener formulados y estipulados en esta hipótesis.
3. **Recogida y análisis de datos:** los datos que se obtuvieron a través de las pruebas que se realizaron. Estipula que existe aún un camino largo de recorrer para poder establecer conclusiones concretas con respecto a este sistema de navegación y así poder impulsar líneas de investigación y aplicaciones reales que den solución a problemáticas presentes en la sociedad y en el sector productivo a pesar de que sólo la simulación se

realizó en forma empírica y solo logrando una pequeña prueba de navegación autónoma se sabe que los datos simulados pueden llegar a ser muy similares a los datos reales que un sistema de navegación autónoma puede llegar a encontrar En un ambiente hostil.

4. **Confrontación de los datos de la Hipótesis:** En la mayoría de las veces, al realizar una investigación, Un tanto más desarrollada, siempre se espera que los resultados sean favorables, que los resultados satisfagan, todo el trabajo que se invierte en el desarrollo de algún proyecto de investigación. En esta ocasión Muchas de las pruebas que se llevaron a cabo no concordaron y no arrojaron resultados esperados, o favorables, a pesar de haber logrado pruebas que sí hayan arrojado resultados esperados, es ahí donde se llega a la conclusión De que aún existe camino por recorrer para así poder resolver la hipótesis planteada en un porcentaje altamente favorable, pero en base al trabajo realizado se puede llegar a predecir un resultado esperado.
5. **Conclusiones y generalización de los resultados:** Dentro de las pruebas realizadas hay datos que evidentemente no respaldan la hipótesis planteada, pero después de hacer modificaciones, ajustes y simulaciones, en las diferentes plataformas antes mencionadas, poco a poco se está logrando qué esta hipótesis puede ser alcanzada y en un tiempo futuro puede ser comprobada, se espera que este sistema, que se encuentra en vías de desarrollo, pueda aportar mucho a las investigaciones que se están desarrollando de forma paralela, todos con el objetivo de dar una solución, a problemas que nos podemos encontrar en el sector productivo,

o en la sociedad misma; se espera que este sistema de control, pueda dar un aporte bastante significativo a la comunidad científica y en especial para las personas a quienes está dirigida, a las personas que tienen dificultades de movimiento, a las personas que día a día luchan para poder integrarse en una sociedad, altamente cambiante.

6. **Nuevas predicciones:** Así como se desarrollan nuevas técnicas para la movilidad de dispositivos o robots móviles, esta línea de investigación también tendrá un desarrollo futuro, el cual deberá perfeccionar técnicas que deberá adaptar así como nuevas formas de navegación, nuevos algoritmos nuevo hardware, nuevas plataformas, así como las nuevas tecnologías como por ejemplo, inteligencia artificial, sistemas de procesamiento paralelo mucho más avanzado, esto ayudará no solo a comprobar la hipótesis en una forma meramente científica, y en múltiples aspectos, Haciendo que surjan más planteamientos, más posibles soluciones, a problemáticas que se llegaran a presentar a pesar del perfeccionamiento que se desarrolla día con día.

2.2.2. DEFINICIÓN DE POBLACIÓN Y SELECCIÓN DE LA MUESTRA

La población en el cual se aplicará el proyecto a desarrollar serán las instalaciones del ITST, el cual cumple con los requisitos de tener infraestructura amplia con múltiples secciones que incluyen rampas, escaleras, pasillos, etc.

Se realizará un mapeo en una sección del instituto, esta actividad se debe realizar por lo menos 3 veces, en el cual se deben comparar las incertidumbres y

posteriormente compararlas con lo planteado en la hipótesis, si hay mejora en la navegación disminuyendo errores.

Dentro de las instalaciones específicamente en el edificio multifuncional de talleres y laboratorios del ITST, se realizaron las primeras pruebas de mapeo, inicialmente se realizó con un robot que cuenta con un sistema de navegación con control clásico on-off, Fig. 2.19. Para la toma de decisiones en la navegación autónoma.

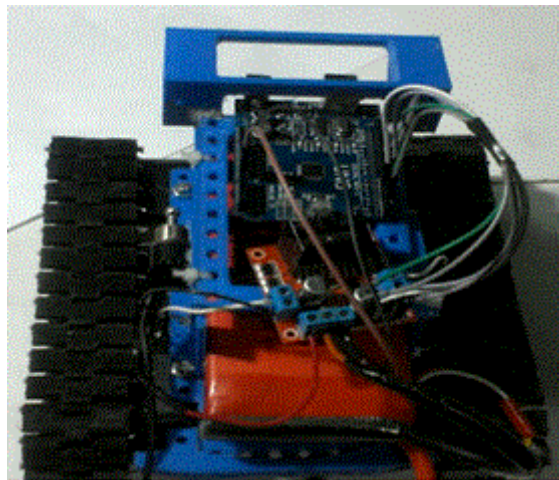


Fig.2.19. Robot con sistema de control on-off. Imagen propia.

El robot que se muestra en la imagen anterior sirvió de apoyo para poder establecer el comparativo entre la navegación con sistemas de mapeo, fusión de sensores y un sistema de control clásico, on-off, similar al que se utiliza para resolver laberintos.

Posteriormente se implementó el sistema de control con ROS y SLAM obteniendo los siguientes resultados de mapeo.

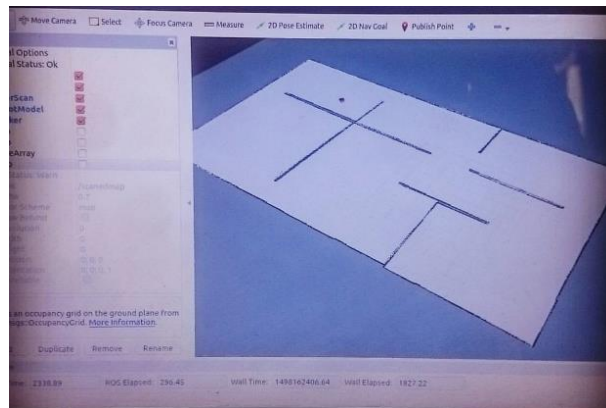
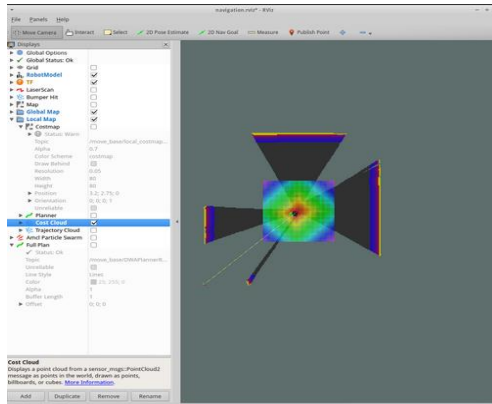


Fig. 2.20. simulación SLAM y ROS para la navegación autónoma. Imagen propia

En la figura 2.20 se aprecia como se va estableciendo el sistema de navegación con SLAM y ROS al momento de realizar el mapeo en el edificio de talleres y laboratorios del instituto tecnológico superior de Teziutlán.

2.3. METODOLOGÍA DE DESARROLLO

El diseño del prototipo a desarrollar se describe en el siguiente apartado, eligiendo la forma más eficiente para obtener los mejores resultados.

2.3.1. Diseñar el robot con software CAD

Se realizó el diseño del robot en el software CAD SolidWorks. Inicialmente se implementó un modelo de forma rectangular; encontrando desventajas en un diseño como el que se muestra a continuación.

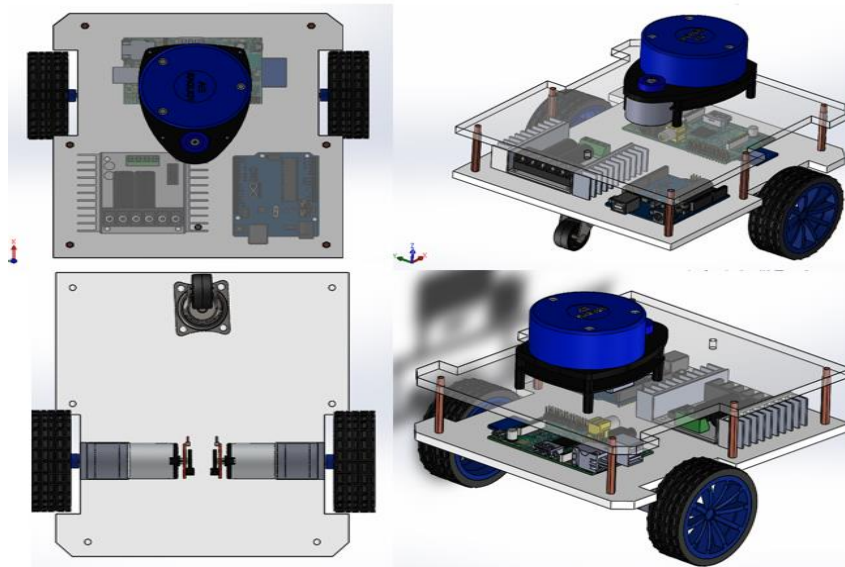


Fig.2.21. vista del prototipo en CAD, primer diseño, descartado. Imagen propia.

La forma rectangular no es factible por los vectores en los sistemas de rotación tal como se muestran en la siguiente imagen.

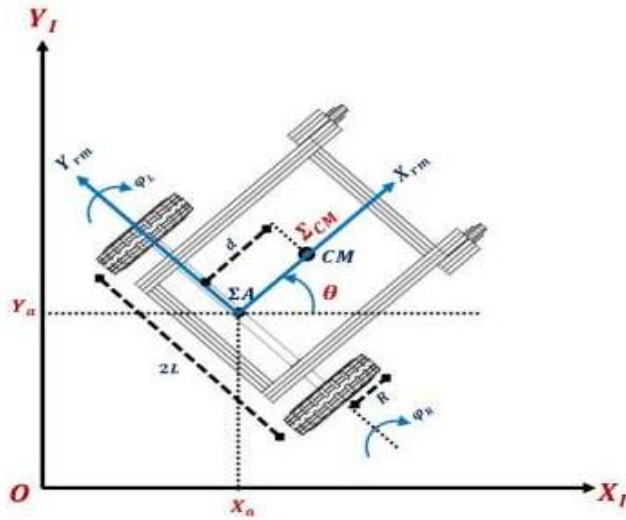


Fig.2.22. coordenadas de la postura del robot móvil. Se orienta en el sistema absoluto O , y el sistema de coordenadas del robot A . Imagen propia.

Como se puede observar en la imagen anterior la forma rectangular tiene diferentes complicaciones al ejecutar el diseño de un robot móvil diferencial.

- La anchura del robot no es el límite en los cálculos de navegación, hay que tomar en cuenta el vector Y_{rm} , positivo y negativo trazan una circunferencia para conocer los límites correspondientes. Los cuales son independientes respecto a su eje de rotación.
- La longitud del robot X_{rm} , de igual forma se debe trazar su circunferencia para saber los límites de posición, considerando que toman un nuevo plano independiente de Y_{rm} . Tal como se muestra en la siguiente imagen.

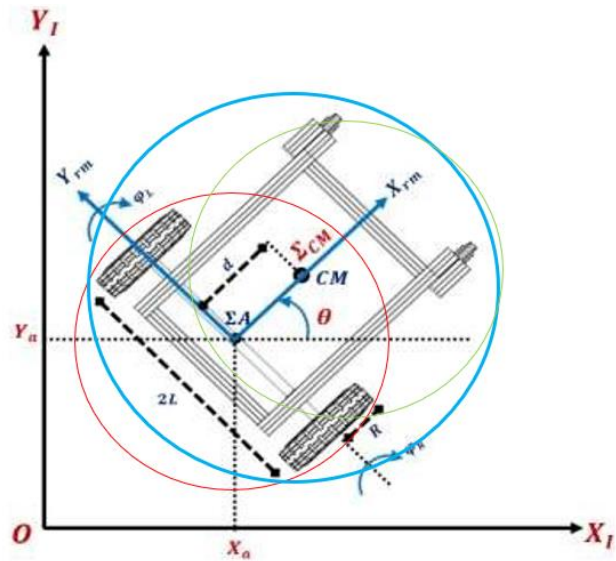


Fig.2.23. planos en un robot diferencial rectangular. Imagen propia

Como se ilustra en la imagen 2.23, la circunferencia más grande es el límite real del robot diferencial rectangular, esto ocasiona mayores complicaciones al realizar los cálculos cinemáticos. En el diseño se deben considerar tres planos de movilidad, como consecuencia ocasiona el uso de recurso en procesamiento de datos y evidentemente Hardware. (Arturo Getsemani Roa-Borbolla, 2018)

Por ello se realizó el diseño de un segundo prototipo considerando lo expuesto anteriormente.

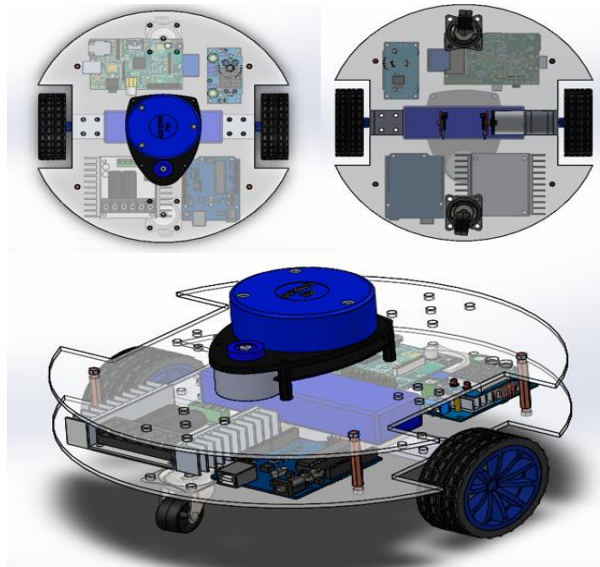


Fig.2.24. Segundo diseño de robot diferencial. Imagen propia.

La movilidad que puede llegar a tener un robot diferencial circular es más eficiente y simple al momento de efectuar los cálculos cinemáticos, además, solo se toma como referencia un solo plano, no tres como el rectangular, teniendo una libertad como se muestra a continuación.

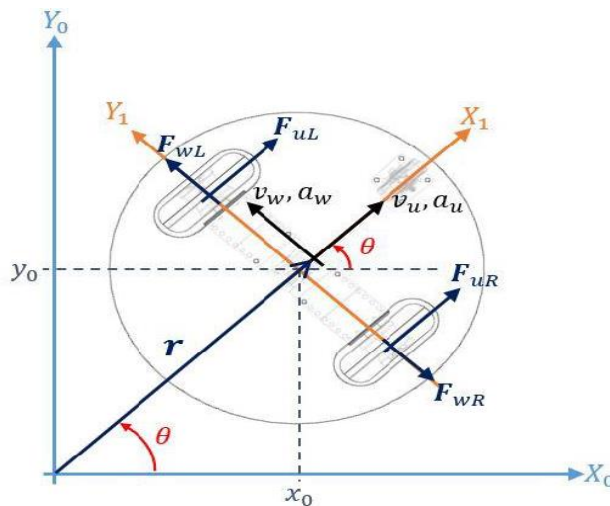


Fig.2.25. Diagrama de cuerpo libre del robot móvil de direccionamiento diferencial. Imagen propia

El modelado cinemático es el estudio de movimiento de sistemas mecánicos sin considerar las fuerzas que afectan dicho movimiento. Para el robot móvil de direccionamiento diferencial, tiene un principal modelado cinemático el cual, es representar la velocidad del robot en función a las velocidades de las ruedas conjuntamente a los parámetros geométricos del robot.

Un robot móvil con ruedas es un dispositivo que debe moverse sobre una superficie mediante la acción de ruedas montadas en él.

- El robot móvil se mueve sobre una superficie plana horizontal, es decir la energía potencial es constante.
- Los ejes de referencia son perpendiculares al suelo.
- No existen elementos flexibles en la estructura del robot, incluyendo las ruedas.
- El contacto entre cada rueda y el suelo se reduce a un solo punto.

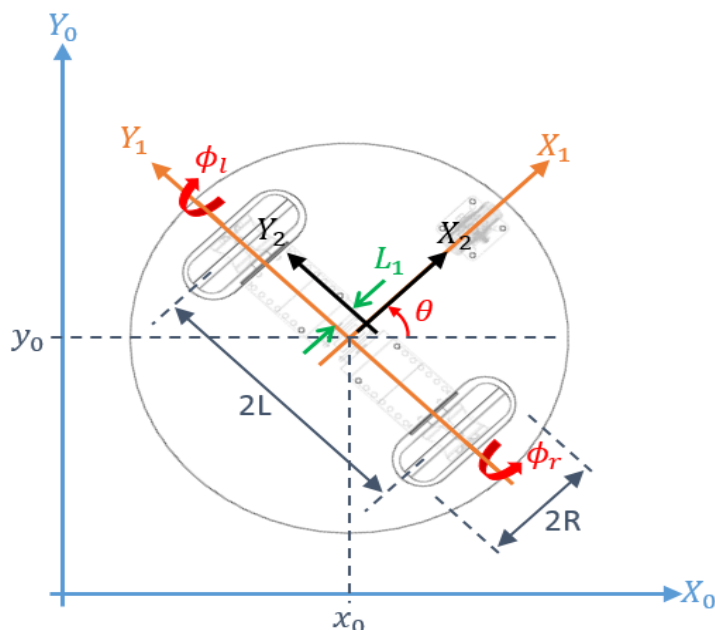


Fig.2.26. Sistemas de coordenadas Σ_0 , Σ_1 , Σ_2 . Imagen propia

El sistema de coordenadas $\Sigma_0(X_0, Y_0, Z_0)$, $\Sigma_1(X_1, Y_1, Z_1)$ y $\Sigma_2(X_2, Y_2, Z_2)$ permiten describir el desplazamiento y la orientación del robot móvil.

$2L$ es la distancia de separación entre las dos ruedas de tracción, $2R$ es el diámetro de las ruedas, L_1 es la distancia del centro de masa al eje de rotación de las ruedas y ϕ es el ángulo que describe la orientación del móvil.

2.3.2. Control PID para la corrección de incertidumbre en la posición angular

En esta parte de la investigación se llevó a cabo el diseño del controlador PID, para la corrección de incertidumbre del sistema de odometría.

Los robots móviles en comparación con los robots fijos, abren un abanico de posibilidades en la realización de tareas, sean de modo automático o bien pueden ser manipulados en forma remota, los dispositivos que están diseñados para movilidad autónoma se mueven en ambientes cambiantes llenos de obstáculos que son aleatorios y en muchas ocasiones inesperados, esto hace una gran diferencia con respecto a los sistemas de control fijos ya que un sistema de control implementado en dispositivos móviles o mecanismos móviles, deben tomar una decisión por cada movimiento o nueva posición, tomando en cuenta su orientación y su sistema de coordenadas, el cual le van a estar indicando en cada momento y en cada movimiento, cómo debe reconocer constantemente el

ambiente donde se mueve para poder adaptarse. (Leonardo Enrique Solaque Guzmán, 2014)

Para que un sistema móvil como el que se está desarrollando se necesita tener un alto grado de sensores y tener cierto grado de autonomía, aplicando algoritmos que tengan la capacidad de poder resolver problemas de movilidad.

Para realizar un proyecto que involucre un control, para un dispositivo móvil, se necesita invertir una cantidad considerable de tiempo dinero y esfuerzo; construir o desarrollar un proyecto en estas características, no suele ser sencillo, implica invertir varias horas de trabajo y se necesitan diferentes tipos de herramientas tanto para el área mecánica electrónica y de sistemas computacionales.

2.3.3. Control de velocidad.

Para lograr que el robot pueda avanzar de manera lineal es necesario ajustar la velocidad de las llantas. Por ejemplo, para que el robot pueda avanzar en línea recta es necesario que ambas ruedas giren a la misma velocidad. Para esto se utilizó un control de lazo cerrado que controla la velocidad de cada llanta en forma independiente. Para poder ajustar las variables de control se utilizaron dos salidas PWM (Pulse Width Modulator) del sistema Arduino hacia el controlador de motores (puente H). Tomando en cuenta los beneficios del concepto de modularidad. (Sites.google.com, 2020) Se utilizó para la etapa de potencia un Motor Shield especial para Arduino (Adafruit), el cual tiene cuatro puentes H integrados en dos circuitos electrónicos L293D. Para el fácil manejo de estos circuitos se utilizó una librería de descarga libre, (AFMotor.h) ofrecida por el

fabricante del Motor Shield especial para Arduino. La función `setSpeed()` permite manejar por argumentos, la velocidad desde cero hasta 255, siendo éste último el valor máximo de la velocidad que los motores pueden alcanzar.

2.3.4. Tracción

El movimiento del robot está basado en una tracción diferencial, la cual es una configuración muy común para sistemas utilizados en interiores, debido a que permite girar sobre su propio eje. De esta manera, el sistema puede moverse en espacios congestionados con cierta facilidad. La tracción diferencial utiliza dos ruedas controladas individualmente con una rueda loca como tercer punto de apoyo. Para la tracción de las llantas, se utilizaron motores de corriente directa con transmisión Pololu.

La velocidad y la posición del vehículo se pueden obtener por medio de encoders (Pololu Corporation) los cuales vienen en conjunto con los motores antes descritos, especial para este tipo de aplicaciones. Los encoders tienen la capacidad de dar 48 cuentas por revolución (Pérez Arreguín Jorge Israel, 2011).

El desplazamiento del robot "D", tal como se ilustra en la siguiente figura 7, a lo largo de la trayectoria está determinado por la siguiente ecuación:

$$D = DI + Dr/2 \text{ Donde:}$$

D = Desplazamiento del robot

DI = Desplazamiento de la rueda izquierda

Dr = Desplazamiento de la rueda derecha

De forma similar, la velocidad V del robot está dada por la siguiente ecuación:

$$V = V_l + V_r/2 \text{ Donde:}$$

V = Desplazamiento del robot

V_l = Desplazamiento de la rueda izquierda

V_r = Desplazamiento de la rueda derecha

Así mismo, el ángulo θ se puede obtener mediante la siguiente formula (figura 7):

$$\theta = D_l - D_r / d$$

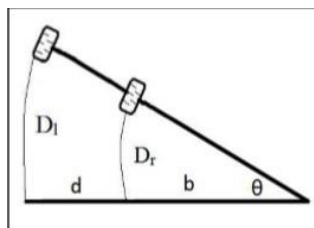


Fig.2.27. Los arcos D_l y D_r son trazados por las ruedas izquierda y derecha para el cambio en el avance del robot con ángulo θ . (Pérez Arreguín Jorge Israel, 2011)

Se puede observar un cambio en la forma de cómo se orienta en base a sus desplazamientos referente a sus ruedas, la variable en el denominador representa una fuente de error significativa debido a las rotaciones internas y las incertidumbres que están referenciadas al punto inicial. Los errores suelen estar asociados a desplazamientos aleatorios inesperados debido a la forma irregular de la superficie, así como los diámetros de las ruedas y cualquier otro elemento que interfiera con los puntos de referencia de cada decisión que el dispositivo móvil realiza, todos estos errores hacen que el sistema sufra un desvío en la

trayectoria deseada, cada error hace que la incertidumbre incremente complicando más el cálculo de las trayectorias a seguir.

Para el fácil e intuitivo control del sistema de control, se diseñó en Visual Basic una interfaz gráfica (GUI) con varias opciones de movimiento, Fig.8.



Fig.2.28. Interfaz gráfica del usuario para el control en forma manual. Imagen propia

El usuario puede manipular el robot presionando con el ratón los botones o también tiene la opción de manipular el robot por medio del cursor o del teclado de la computadora. La interfaz envía información vía puerto serial y solo puede ser usada en forma manual, esta aplicación no es para manipulación directa del robot, sólo se utiliza para poder establecer movimientos de prueba, así como desplazamientos circulares, de rotación y traslación, para poder tener una mayor idea de las incertidumbres o los errores que puede llegar a tener en la trayectoria o curso referente a un punto de referencia.

El sistema de control de odometría se encuentra estructurado bajo el siguiente esquema:

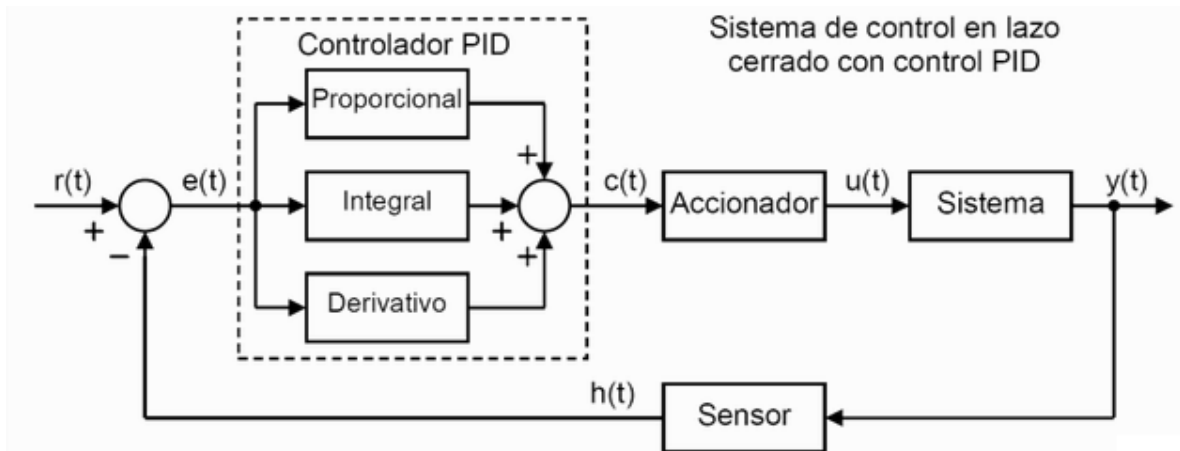


Fig.2.29 Sistema de control de lazo cerrado implementado en la odometría del sistema de navegación. (Ogata)

En la figura anterior fig. 9, podemos observar un sistema de control de lazo cerrado el cual cuenta con un sistema proporcional integral derivativo que van a enviar los cálculos de salida referente a los datos de entrada que le van a proporcionar los sensores, El accionador enviara una respuesta al sistema de manipulación a través del puente h para que posteriormente exista una respuesta de error, que en este caso sería el sensor, el cual estará haciendo una retroalimentación al sistema de entrada en base al control de lazo cerrado, esto hace que el sistema de odometría tenga una mayor eficiencia y pueda así establecer una estabilidad mayor en el movimiento rotacional de las ruedas y reducir la incertidumbre al momento de hacer el trazo de las trayectorias a seguir.

En la siguiente figura, se puede ver con mayor claridad cómo se establece o se formulan los valores de la planta del sistema de control de lazo cerrado, el cual van a dar paso a la programación para el microcontrolador Arduino, esta planta hará más fácil y eficiente la forma o la manera de visualizar la estabilidad del

sistema de control en el cual podemos observar una entrada de referencia siendo la rotación lineal y angular de las ruedas, el cual van a entrar a los datos del controlador PID, los cuales van a ser un ajuste y estabilizar la señal lo más que se pueda, bajo las capacidades del microcontrolador, posteriormente existirá una perturbación, el cual será también estabilizada por la planta, para poder tener datos de salida. El ruido, el cuál sería el valor de error referente al valor ideal que se requiere o que se necesita, en la rotación de las ruedas, será nuevamente retroalimentada al sistema de entrada de referencia, para qué nuevamente el controlador haga una comparación respecto al valor ideal y el valor real que obtienen los sensores bajo los encoder y así poder tener una estabilidad y poder hacer un balance de entre lo real y lo ideal, estabilizando en una forma mayormente uniforme la rotación de las ruedas.

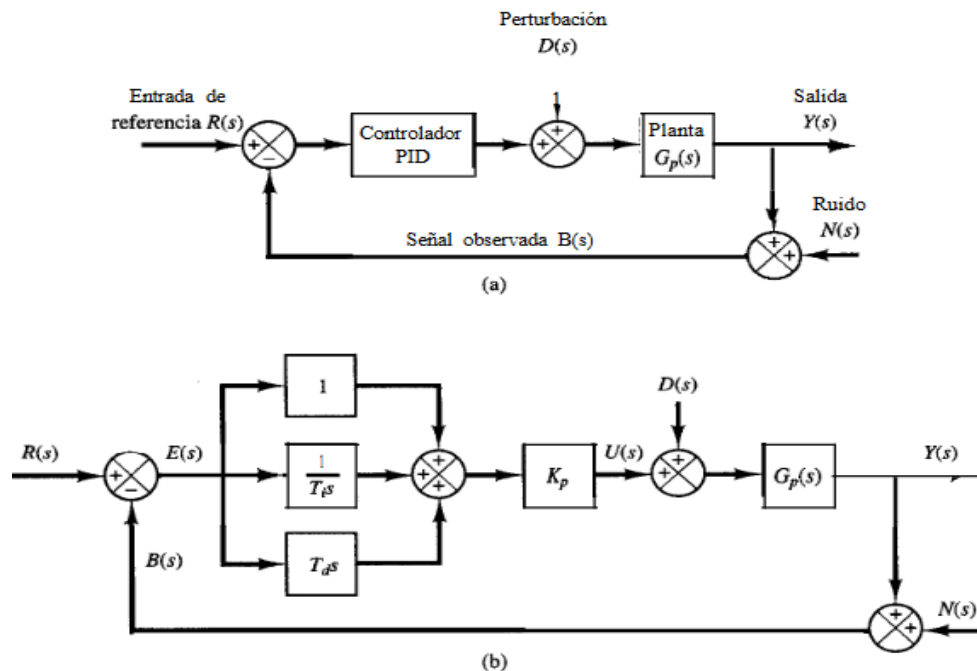
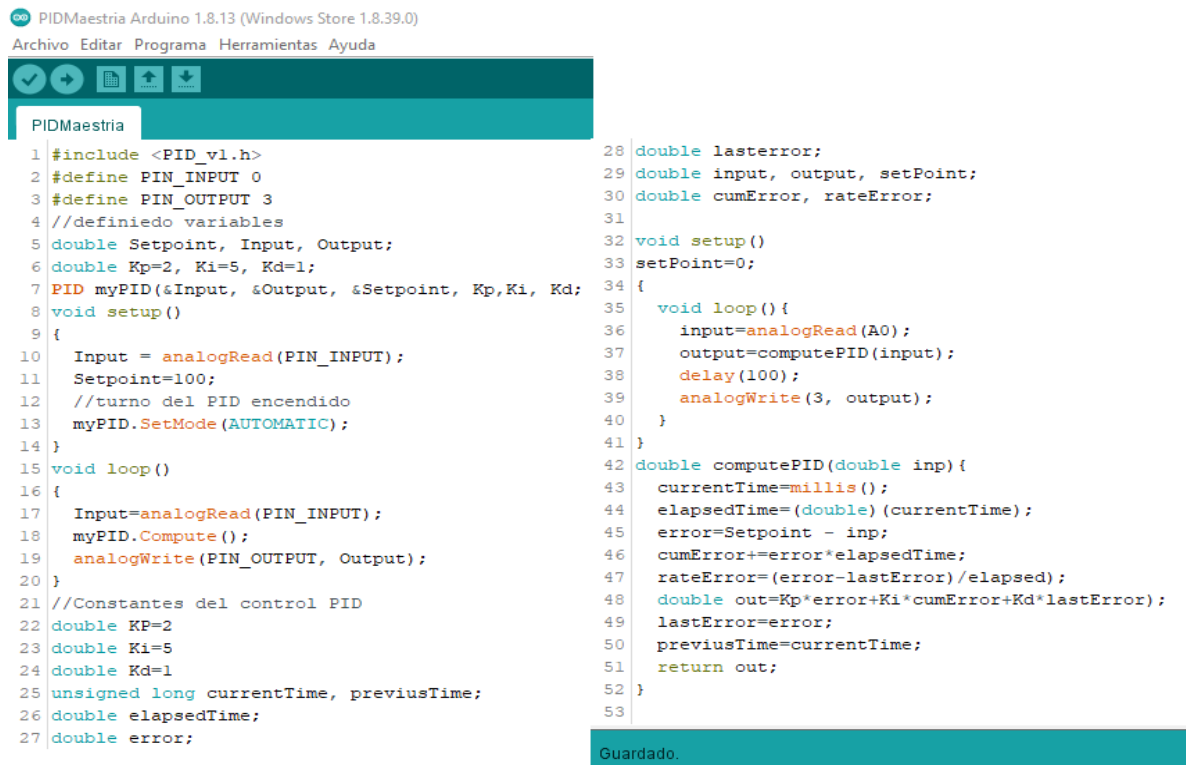


Fig.2.30. (a) Sistema con un controlador PID; (b)diagrama de bloques equivalente. (Ogata)

El código implementado en el sistema de control es el siguiente:

Este código, Incluye los datos de la planta para la estabilización del sistema y así poder minimizar la incertidumbre entre los sensores, el tamaño de las ruedas, las rotaciones, y las trayectorias, que se traza en la aplicación que se describió con anterioridad.



```
PIDMaestria
1 #include <PID_v1.h>
2 #define PIN_INPUT 0
3 #define PIN_OUTPUT 3
4 //definiedo variables
5 double Setpoint, Input, Output;
6 double Kp=2, Ki=5, Kd=1;
7 PID myPID(&Input, &Output, &Setpoint, Kp,Ki, Kd;
8 void setup()
9 {
10   Input = analogRead(PIN_INPUT);
11   Setpoint=100;
12   //turno del PID encendido
13   myPID.SetMode(AUTOMATIC);
14 }
15 void loop()
16 {
17   Input=analogRead(PIN_INPUT);
18   myPID.Compute();
19   analogWrite(PIN_OUTPUT, Output);
20 }
21 //Constantes del control PID
22 double KP=2
23 double Ki=5
24 double Kd=1
25 unsigned long currentTime, previusTime;
26 double elapsedTime;
27 double error;
28 double lasterror;
29 double input, output, setPoint;
30 double cumError, rateError;
31
32 void setup()
33 setPoint=0;
34 {
35   void loop(){
36     input=analogRead(A0);
37     output=computePID(input);
38     delay(100);
39     analogWrite(3, output);
40   }
41 }
42 double computePID(double inp){
43   currentTime=millis();
44   elapsedTime=(double)(currentTime);
45   error=Setpoint - inp;
46   cumError+=error*elapsedTime;
47   rateError=(error-lastError)/elapsed;
48   double out=Kp*error+Ki*cumError+Kd*lastError);
49   lastError=error;
50   previusTime=currentTime;
51   return out;
52 }
53
```

Guardado.

Fig.2.31. Código implementado en el microcontrolador Arduino y el sistema PID, ambos en base a la planta formulada. Imagen propia

El código que se muestra en la figura 11, inicialmente se definen las variables que son utilizadas en la planta, que se describió en la figura 10, se establecen los parámetros para Kp, Ki, Kd, las cuales son las variables y los parámetros que le van a dar estabilidad al sistema de control de motores, posteriormente se establece los parámetros para la

lectura de los datos analógicos del sensor, los pines en donde se va a conectar, para que estos valores enviados por el sensor sean utilizados por la programación. Sí hace uso de la librería PID_v1.h, dentro de esta librería se establecen los parámetros ya desarrollados dentro de la Comunidad de Arduino para implementar en controles o plantas PID.

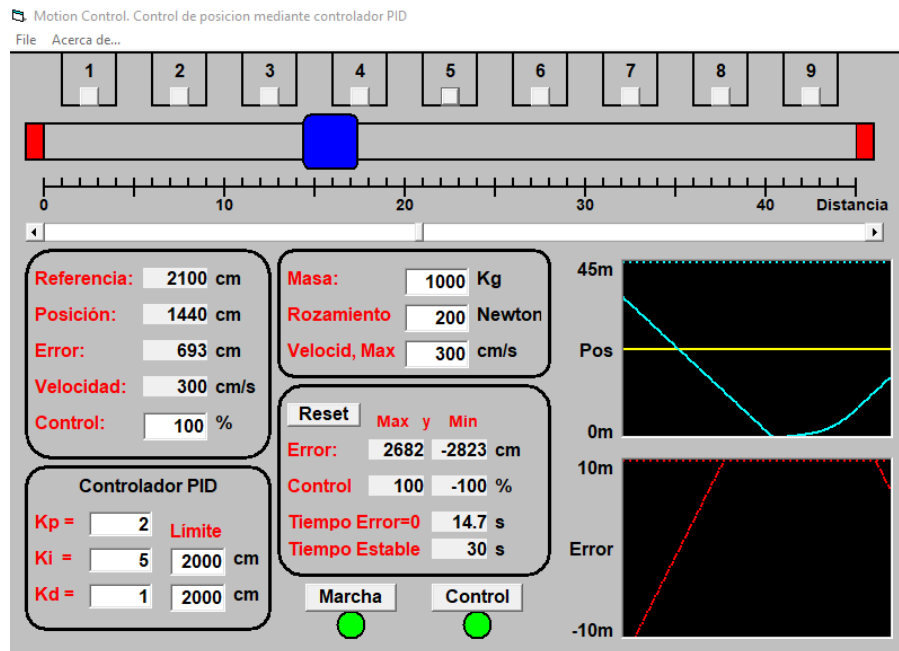


Fig.2.32. Sistema de control para la estabilización de la planta PID. Imagen propia

En la figura 2.32, podemos apreciar un sistema de control, el cual, es un simulador que ayuda a tener una idea y al mismo tiempo, realizar pruebas para el funcionamiento de un sistema de control de lazo cerrado, esta aplicación ayuda a visualizar cuál es el comportamiento de estabilización del sistema de control implementado, así como también, se pueden ir introduciendo valores e ir analizando el comportamiento de cada uno de los datos que se van ingresando, tales como la referencia, la posición, el error, la velocidad, el porcentaje de

control, los parámetros del controlador PID , el tiempo de retraso o eficiencia del microcontrolador, el torque, la fuerza, la velocidad con relación al diámetro de las llantas; es una herramienta de muchísima utilidad que facilita la visualización y el comportamiento que se requiere implementar.

Como se puede apreciar en los gráficos de las figuras anteriores, se debe incrementar la acción proporcional y así poder bajar el estado del error, esto sería la diferencia entre lo que se desea y lo que se obtiene en base a la trayectoria, esto con un fin de poder incrementar la velocidad de respuesta. Cuando el sistema es inestable el controlador incrementará la constante derivativa k_d , para poder conseguir una estabilidad más uniforme, en caso de que el error sea mayor, se debe incrementar la constante K_i , hasta que se pueda minimizar el error a una variable deseada, con estas modificaciones, el controlador PID puede mantener sistemas de trayectoria estable, siempre y cuando se mantengan dentro de los parámetros establecidos.

2.3.5. Simulación ROS

Como se mencionó anteriormente Ros, es una aplicación para plataformas libres para el desarrollo de robótica móvil, el cual puede trabajar con el sistema de navegación SLAM, este sistema de navegación o localización trabaja en espacios bidimensionales y tridimensionales que son esenciales para conocer la navegación del robot, se trabajarán con paquete TAF(Archivos para aplicaciones usados en video juegos basados en textos ideales para generar gráficos y ambientes virtuales), el cual son paquetes de gran importancia para la

navegación del prototipo móvil. Primeramente se debe construir un mapa en Ros y posteriormente establecer los parámetros de configuración, así los algoritmos podrán ser trasladados a SLAM, para poder obtener un mapa de alta calidad, posteriormente se hace la navegación del robot usando un mapa construido con el algoritmo, esta demostración experimental se realiza utilizando un simulador gratuito de la plataforma Ros Turtle, (es un simulador gráfico en forma de tortuga, especializado para trabajar robots diferenciales.) que es el más famoso simulador de robótica móvil y a la vez, también es una plataforma de enseñanza para la comunidad ROS.

CAPÍTULO III

IMPLEMENTACIÓN Y PRUEBAS

Las pruebas pudieron ser posibles a los trabajos de investigación, Realistic and Automatic Map Generator for Mobile Robots. (Arturo Getsemani Roa-Borbolla, 2018) y el trabajo, Robot Móvil de Tracción Diferencial con Plataforma de Control Modular para Investigación y Desarrollo Ágil de Proyectos. (Jorge Israel Pérez Arreguin, 2011), entre otros que están involucrados en el desarrollo del proyecto.

En el siguiente apartado se detalla cómo se realiza la implementación del sistema para la navegación autónoma.

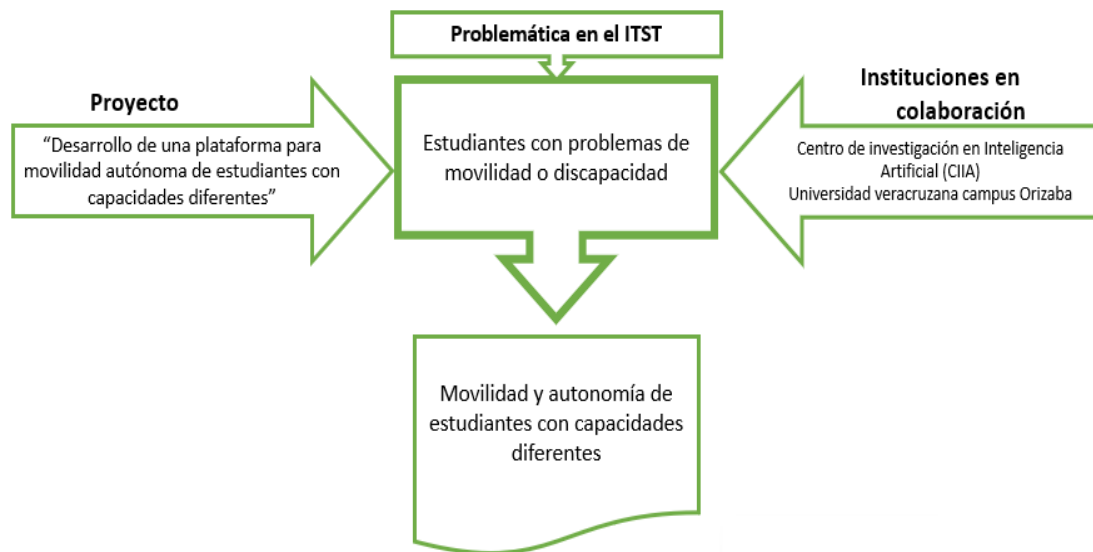


Fig. 3.33. Metodología para el desarrollo del proyecto. Elaboración propia.

En la figura 3.33, se establece la implementación que se tiene para el desarrollo del proyecto, ya que en los once años que se tiene como trabajador en el ITST, siempre se ha observado una población estudiantil con la problemática antes mencionada, por ello en una de las líneas de investigación de este instituto, “Automatización Industrial” adscrita a la división de ingeniería Mecatrónica, se

estableció el desarrollo de dicho proyecto, en colaboración con el CIIA de U. Veracruzana de la ciudad de Xalapa y de la U. veracruzana campus Orizaba, quienes han aportado asesoramiento con una estadía realizada en los meses de octubre y noviembre del 2019, logrando desarrollar el algoritmo en ROS para el sistema de navegación autónoma de la plataforma, tal como se ilustra en la Fig. 3.34.

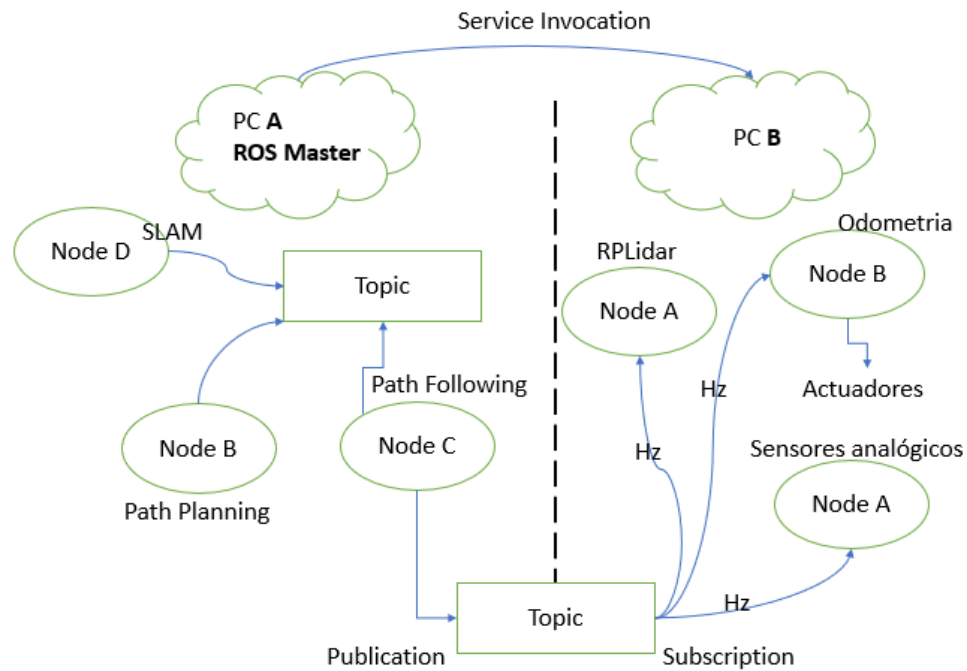


Fig. 3.34. Grafo de sistema de control. Elaboración propia.

Los robots que son controlados por ROS, funcionan como un Grafo, el cual facilita el desarrollo de las aplicaciones implementadas. Como se observa en la imagen de la Figura 3.34; el nodo A, se encarga de recolectar las distancias que los sensores analógicos y digitales, enviándolos a la PCA, el nodo B cuenta con un PID para el posicionamiento de los motores, los cuales cuentan con encoder de efecto hall para estabilizar el sistema de control, el sensor RPLidar envía sus

coordenadas al nodo A, por medio del protocolo de comunicación serial efectuando la transferencia de información a la pc A, en el cual se efectúa la localización y mapeo simultáneos (SLAM), la planificación de ruta y el cálculo de trayectorias o camino a seguir.

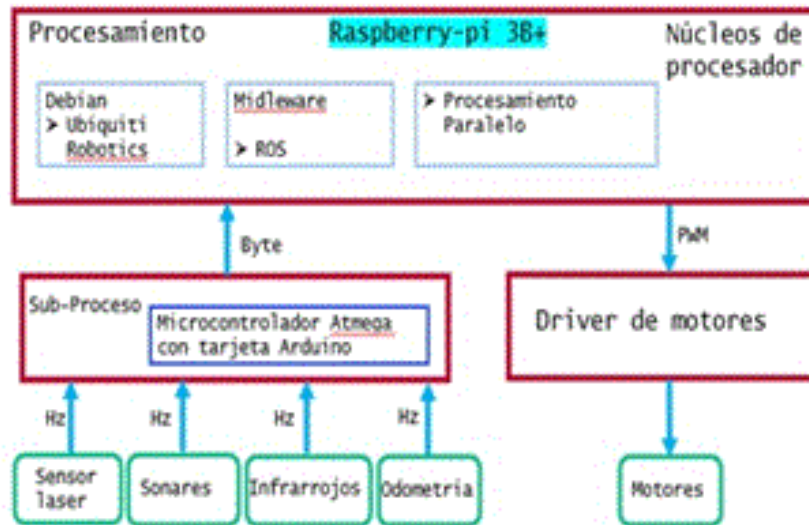


Fig. 3.35. Diagrama para el sistema de control autónomo. Elaboración propia

El sistema de control tiende a inclinarse y tener mayor confianza en la señal del sensor con mayor fidelidad de señal, por ello el sistema presenta menor incertidumbre en la estabilización de los sistemas de control, tal como se muestra en la figura 3.36.

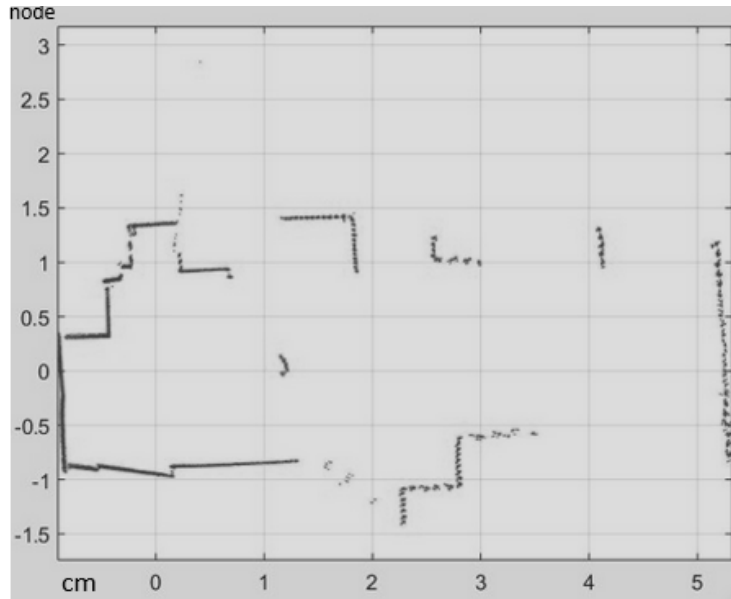


Fig. 3.36. Mapa, puntos de muestreo. Elaboración propia

La frecuencia de cada sensor establece los parámetros para realizar el procesamiento de la señal, por ello, antes de que sea procesada la información en la tarjeta Raspberry pi, debe ser acondicionada por un microcontrolador Arduino, quien realiza el trabajo de procesar los tiempos de cada sensor y decidir qué señal se dejara procesar en el siguiente nivel.

3.1. ANÁLISIS DE DATOS

En las pruebas realizadas para la obtención de las pruebas se establece que se tomaron posiciones deseadas, donde a través de un área. el controlador impulsa al robot a través de las posiciones que se requieren, al robot se le colocaron algunos obstáculos en los cuales fueron reconocidos a través del sensor láser, en las figuras anteriores se muestran cómo se comportan las fuerzas de atracción

para cada posición y para cada obstáculo, qué se le interpuso en la trayectoria de las posiciones establecidas.

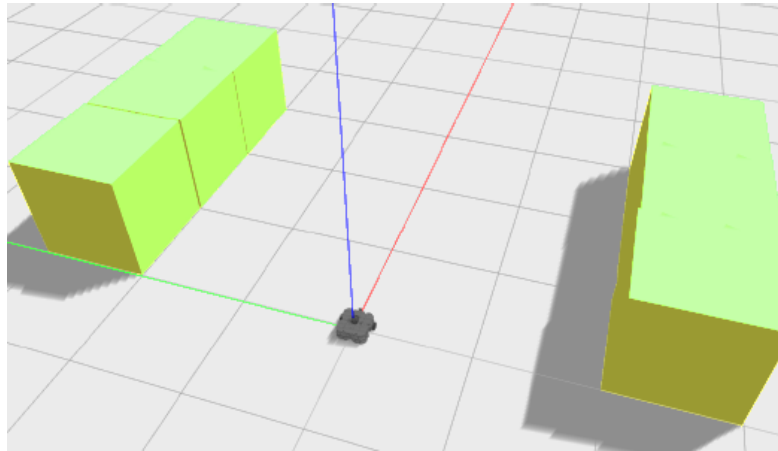


Fig.3.37. obstáculos colocados para pruebas de navegación haciendo uso de SLAM. Imagen propia

Para poder hacer pruebas de navegación se usó el algoritmo SLAM, en este simulador se creó un escenario figura 3.37, en el cual se colocó el robot en medio de obstáculos estos fueron colocados de tal forma que se puedan simular las paredes de algún edificio o alguna infraestructura en la figura anterior se puede observar el escenario establecido los obstáculos se colocaron alrededor de 3 metros por cada lado el robot tuvo que desplazarse por este espacio llegando a una posición establecida sin salirse de las longitudes laterales establecidas, para poder efectuar estas pruebas se aplicó el paquete de SLAM llamado gmapping establecido por ROS, el paquete antes mencionado permite crear mapas con dos dimensiones a partir de los datos del sensor láser y de los datos enviados por el encoder u odometría del robot móvil qué es controlado a través del PID, el algoritmo empieza a crear diferentes puntos donde se establecen los datos en la

odometría del robot y los del sensor láser esta comunicación se hace a través de los nodos de comunicación entre el PID y el algoritmo que se encuentra funcionando a través de ROS, tal como se ilustra en la siguiente imagen.

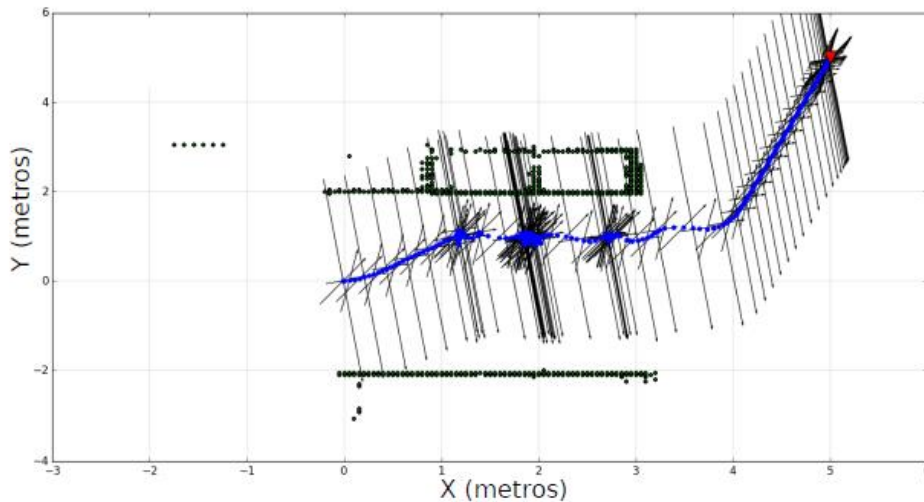


Fig.3.38. Navegación del sistema de control, utilizando SLAM, fuerzas de atracción y de repulsión. Imagen propia.

Para que el robot puede tener un desplazamiento con las de las dimensiones establecidas en el mapa por SLAM, se puede llegar a la conclusión de que el recorrido fue bueno debido a que el algoritmo gmapping, redujo en cierta parte la cantidad de lecturas que envía el sensor láser hacia el mapa uniforme establecido junto a los obstáculos para establecer una posición correcta.

Como parte del proceso el sensor laser localiza los puntos de los obstáculos en el ambiente establecido, estas posiciones establecidas por el algoritmo localizan la posición deseada en el punto $x=3$ y $y=3$ m, inicialmente la trayectoria va desde $x=0$, $y=0$, hasta la posición deseada, el algoritmo SLAM se encarga de disminuir las mediciones del sensor laser, posteriormente añadirá datos de la odometría

enviada por los sensores encoder de efecto hall. En la siguiente figura 3.39, se puede ver cómo fue avanzando el robot estableciendo puntos de dirección por fuera de cada obstáculo.

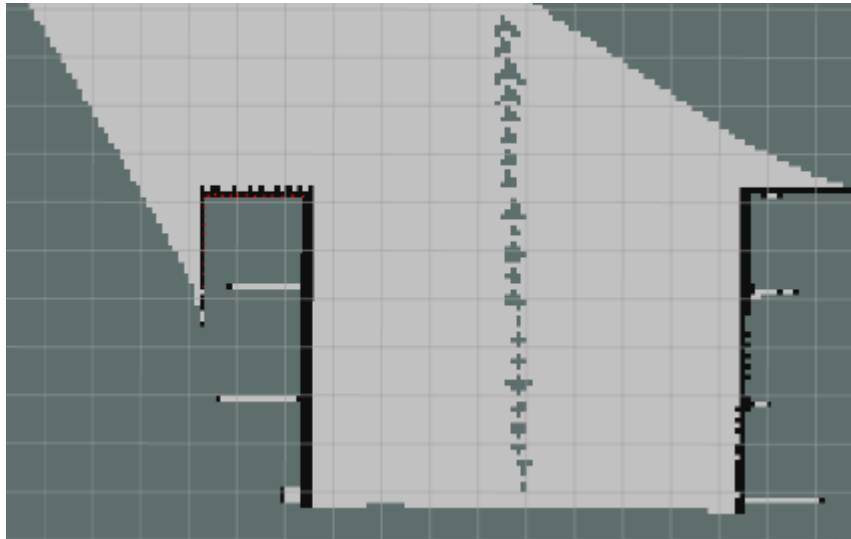


Fig.3.39. Mapa en 2D del algoritmo SLAM el mapa es generado por las trayectorias de ROS. Imagen propia.

La magnitud de los puntos va acorde a la referencia con la trayectoria del robot, esto quiere decir que mientras el robot esté más cerca del punto establecido, el tamaño de la referencia será mucho mayor, por lógica la fuerza establecida de repulsión. De igual manera también incrementará estas fuerzas y aportarán al robot referencias para poder evadir los obstáculos y así tener pocas colisiones, en la figura anterior se puede ver que el mapa que se generó está establecido por 3 colores principales los cuales dan a entender un espacio libre en el cual el robot tiene un movimiento sin interferencia o sin que se interpongan obstáculos, obviamente hacen falta más puntos o más espacios de recorrido para generar un mapa con mayor precisión, pero con los datos que se pudieron obtener se

podieron establecer los puntos de referencia y se pudieron comprobar las trayectorias que se requerían establecer como puntos de llegada, sin tener complicaciones en los recorridos, a pesar de que se colocaron algunos obstáculos el robot, pudiendo generar una trayectoria autónoma con la información que le brindo el algoritmo SLAM.

3.2. SELECCIÓN DE PRUEBAS ESTADÍSTICAS

En la prueba realizada se estableció un entorno tridimensional con las medidas de los pasillos del edificio de talleres y laboratorios, esto con el fin de poder probar el algoritmo tridimensional, en esta representación para la construcción del algoritmo se tiene un espacio vacío en cual solo se conoce la forma de este, a través de puntos de referencia vectorial, se eligió una forma rectangular ya que se pueden tener sus dimensiones a través de mediciones reales obtenidas con un flexómetro, así se pudo establecer un punto medio del rectángulos diseñado, esto con el fin de poder colocar la forma redonda del dispositivo, esta prueba tiene gran importancia ya que proporciona una perspectiva de cómo se comporta el algoritmo y así mismo permite obtener el valor de la incertidumbre que hay en las mediciones establecidas por el sistema de odometría del robot, en si es comparar los valores de las distancias recorridas del robot, comparándolas con las distancias medidas con el flexómetro.

El diseño circular del robot partiendo del principio diferencial permite al sensor establecer lecturas de mediciones más eficientes, incluyendo las zonas que

tienen ángulos de 90°, se inicia con una posición central en el rectángulo $X=0$ y $Y=0$, para así comprobar que tan cerca se pueden distinguir las paredes del rectángulo con relación a las paredes de edificio de forma real.

Inicialmente se analizó la posición del rectángulo en $X=-1.44\text{m}$, tomando muestras de los puntos de referencia con promedio de diferentes lecturas que van de $X=-439$ como promedio. La desviación estándar es de aproximadamente 0.08mts , esto refleja una histéresis de demasiado pequeña con respecto al promedio que es de -439m , con estas conclusiones se llega que se pudo llegar a tener un error de $E=18.4\text{m}$, y un error relativo de $E_R = 21.5\%$ el cual se puede deducir que es mínimo y aceptable.

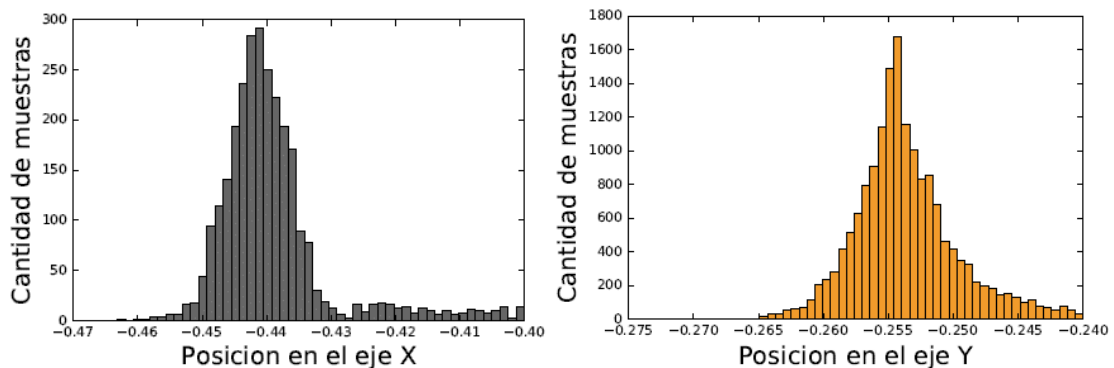


Fig.3.40. Histograma con relación a las muestras con respecto al eje X en el rectángulo establecido, así como las mediciones con respecto a la posición frontal del rectángulo.

Los histogramas reflejan una distribución normal, por la forma de campana peculiar, y establece una forma simétrica con respecto a la media establecida, esto indica que las posiciones que el robot pudo calcular son correctas y que están dentro de los parámetros de histéresis, de igual forma se puede ver una

mejor distribución en las gráficas mostradas por la cantidad de muestras obtenidas, esto se logró por los datos y los pulsos que los encoder pudieron enviar al sistemas de control, permitiendo una distribución uniforme y fácil de interpretar.

3.3. REALIZACIÓN DE ANÁLISIS

El robot se estableció en el pasillo de talleres y laboratorios para que se pueda obtener un mapa en tres dimensiones, tal como se muestra en la siguiente figura, se intentó colocar en robot en una posición al centro del rectángulo formado por las paredes con las líneas de referencia en color rojo, de igual forma se puede observar las dimensiones de la zona de pruebas. La prueba tuvo una duración de 40 segundos en donde los encoder enviaron los pulsos del sensor de efecto hall al controlador PID, para que posteriormente la información fuese empaquetada y enviada vía puerto serial al nodo que controla ROS y ser recibido por el ambiente virtual SLAM, este proceso fue realizado por la tarjeta Raspberry pi 3B+. z

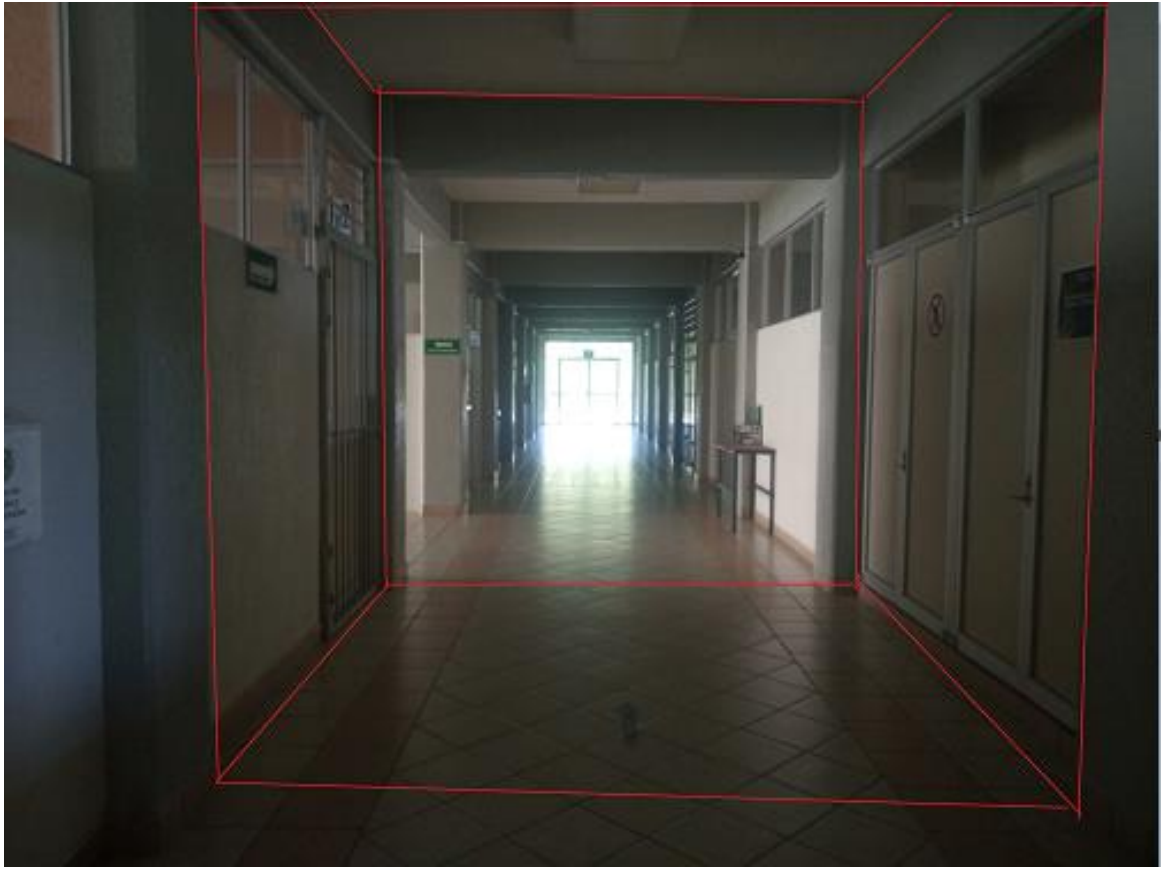


Fig.3.41. Rectángulo establecido por el pasillo de prueba y virtualizado en SLAM. Imagen propia.

El ambiente tiene una forma rectangular simétrica en el cual existe una proporcionalidad entre las paredes y las columnas. Esta prueba se realizó inicialmente con un robot de control clásico On-Off, y posteriormente con el sistema de mapeo con el sensor laser y el sistema de navegación SLAM y ROS.



Fig.3.42. Mapa en 3D del pasillo, mapa tridimensional con respecto a un eje vertical, SLAM. Imagen propia.

El robot fue controlado por medio de la tarjeta Raspberry pi 3B+, el cual al recibir los parámetros del controlador PID, en tiempo real el robot o más bien el sistema de control inicia la construcción de un ambiente virtual, cuando el ángulo está en 0°, los parámetros de velocidad angular y velocidad lineal se encuentran en cero, así también el sensor laser se encuentra en un punto de referencia horizontal, en esta posición el sensor envía las mediciones gmapping a SLAM, esta transferencia de información ocurre en tiempo real ya que el sistema de control necesita saber cuál es la posición de forma detallada y con poca histéresis, este proceso ocurre en tiempos muy cortos ya que cada movimiento que realizan los motores en un punto de referencia vectorial diferente, estos datos deben ser procesados por la tarjeta Raspberry pi 3B+ para luego mostrarlos en el ambiente virtual.

3.4. COMPROBACIÓN DE LA HIPÓTESIS

El trabajo realizado es el esfuerzo inicial en la solución de una problemática latente en la sociedad, el cual pretende resolver el movimiento autónomo, basado el sistema de mapeo, fusión de sensores y sistemas de control PID, todos en conjunto para poder establecer un objetivo, el cual puede ser una solución viable y latente para la aplicación de este sistema.

una vez estableciendo, una solución a la problemática planteada se tuvo que trabajar en ello aplicando un método el cual consistía en el sistema de navegación

usando sensores tipo Ylidar, el cual ayuda mucho al sistema, por su sistema de captación de información casi perfecta en rangos de cero a 360°, esto hace o hizo que el sistema pueda tener una funcionalidad aceptable, arrojando resultados favorables en las pruebas que se realizaron dentro del ITST.

Como se puede apreciar En la figura 3.41, en donde se llevó a cabo el monitoreo de la navegación, y en la figura 3.42, en el cual se pudo hacer una lectura más concreta, de lo que el sensor y los sistemas de control trabajando de forma unánime pudieron arrojar, un resultado de 5 cm en las pruebas, figura 3.36 y 3.39, que se llevaron a cabo a través de los nodos A y B, que se establece en el grafo de la figura 3.34, comprobando así que el sistema implementado sí puede dar una solución viable, segura y efectiva, para ser implementado en la silla de ruedas y en algunos otros dispositivos que requieran de un sistema similar.

Este resultado, no pudo ser logrado sin el apoyo del Centro de Investigación en inteligencia artificial, así como de los asesores asignados, sin el apoyo que brindaron a este trabajo, no hubiera sido posible, ya que trabajar con este tipo de tecnología es relativamente complicada, ya que solamente es usada en centros de investigación de alto nivel, por ello se tiene una gran satisfacción, el haber obtenido un resultado favorable en la comprobación de la hipótesis establecida inicialmente.

CAPITULO IV

RESULTADOS Y CONCLUSIONES

4.1 RESULTADOS

El trabajo realizado es el esfuerzo que se le dio a una problemática que se encuentra presente en la sociedad y en las instituciones educativas en lo particular en el instituto tecnológico superior de Teziutlán.

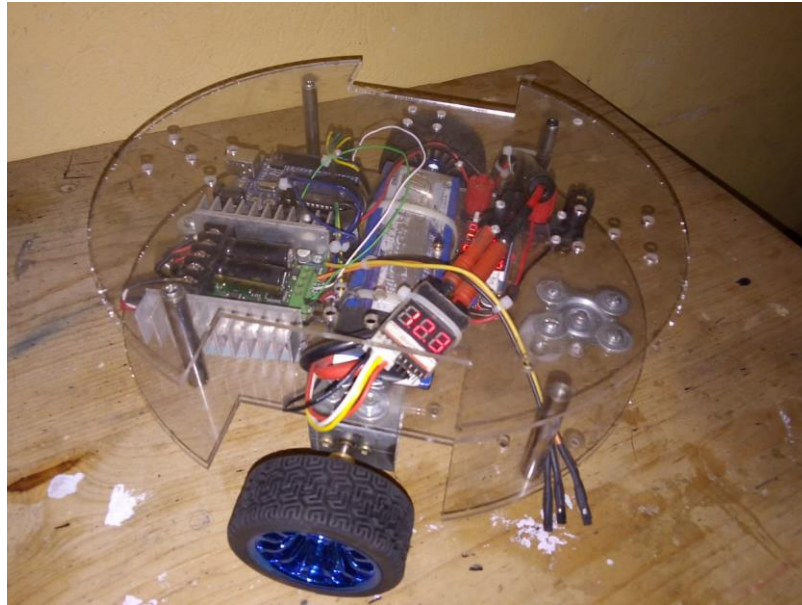


Fig. 3.43. Prototipo robótico desarrollado con sistema de tracción diferencial en forma física. Imagen propia.

Se trata de presentar un robot que tenga una navegación segura fig.3.43, para establecer la ruta más corta, después de realizar mapeos en las instalaciones de algún edificio, este sistema de navegación se aplicó en un robot de accionamiento diferencial. El control de movimiento se basa en control PID a través de sensores encoder de efecto hall, demostrando el rendimiento del robot en un ambiente virtual y sincronía en los sistemas de control, todos trabajando para establecer

una ruta estable, de esta forma también se demuestra la capacidad del robot, con el sistema de mapeo, navegando de forma estable a pesar de tener un entorno desconocido. El sensor de mapeo lidar envía la información del entorno para que el robot pueda tomar una decisión de los movimientos que va a realizar hacia el punto de llegada establecido, evitando obstáculos a través de las decisiones programadas en los grafos.

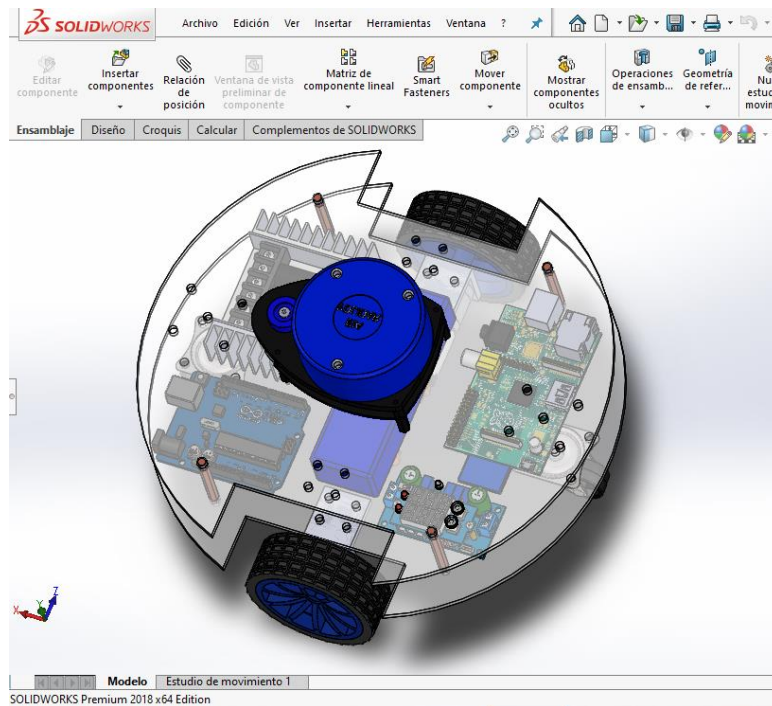


Fig. 3.44. Diseño de prototipo robótico en software CAD, SolidWorks 2018.

Imagen propia.

Se diseñó un chasis con tracción diferencial fig.3.44, en forma redonda para así hacer más fácil la construcción del mapa tridimensional en ROS, a través del sensor lidar. Lo anterior sirvió para realizar el mapeo tridimensional en un cubo similar a los pasillos del edificio de talleres y laboratorios, de esta forma se pudo

concluir que las mediciones más concretas se logran a través de las paredes de los edificios tal como se pudo apreciar en el edificio de talleres del itst.

Todo lo mencionado anteriormente es información que sirvió para el algoritmo aplicado en SLAM, ya que con lo que se pudo apreciar en SLAM, se llega a la conclusión de que el robot tiene la información necesaria para ir avanzando en ambientes desconocidos, concluyendo así, que el prototipo robótico desarrollado puede modelar rutas con errores de posición de hasta 5cm, con la ayuda del sensor lidar.

4.2 CONCLUSIONES

El trabajo presentado es solo la etapa inicial de lo que se pretende llegar en un futuro, el tener una plataforma que traslade sillas de ruedas de forma autónoma, eficiente y segura.

El uso de herramientas nuevas como ROS, son primordiales en el desarrollo de proyectos de investigación en la rama de la robótica móvil y arquitectura de sistemas.

El proyecto está estructurado en tres etapas, los cuales serán el desarrollo del sistema de control con un robot de pruebas, posteriormente se implementará en un prototipo de tamaño y diseño real y por último se realizarán pruebas con personas que tengan dificultades de movimiento. De igual forma se ampliará el mapeo de la zona de navegación dentro de las instalaciones del ITST.

4.3 TRABAJOS FUTUROS

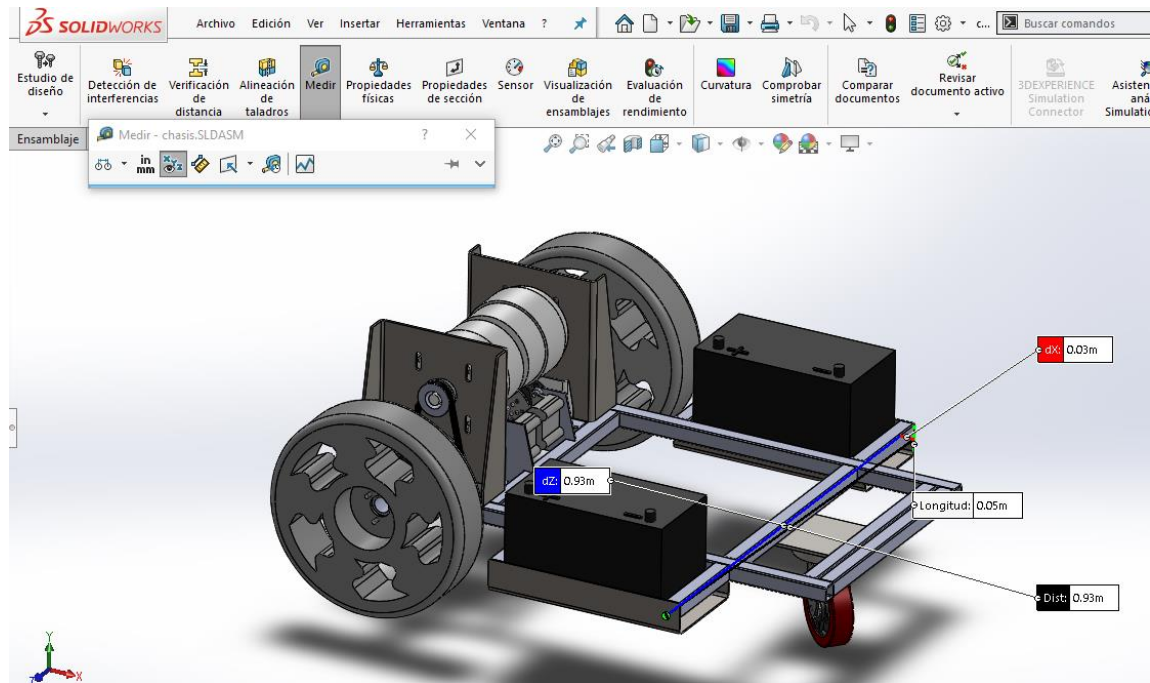


Fig 3.45. Plataforma para silla de ruedas.

En la figura 3.46 se puede ver el diseño de la plataforma en el software SolidWorks, el cual se colocara una silla para un usuario, en la plataforma se integraron elementos como motores con capacidad de 60kg cada uno, asi como baterias y ruedas que soporten la demanda de corriente y torque. Los motores son comerciales similares a los que se implementan en las motocicletas electricas. El prototipo robotico desarrollado en este proyecto de investigacion sera implementado en esta plataforma como continuacion en una segunda etapa.

BIBLIOGRAFIA

- (INEGI), I. N. (2018). Ficha temática Personas con discapacidad. *Conapred*, 6.
- Antonio Barrientos, L. F. (2012). *Fundamentos de Robotica*. Madrid. España: McGRAW-HILL/INTERAMERICANA DE ESPAÑA, S.A. de C.V.
- Arturo Getsemani Roa-Borbolla, A. M.-H.-B.-R.-F. (2018). Realistic and Automatic Map Generator for Mobile Robots. *IEEE*.
- Bertoline, W. M. (s.f.). *Dibujo en Ingenieria y Comunicacion Grafica*. . McGrawHill.
- Corporation, ©. 2.-2. (2020). *Pololu Robotics & Electronics* . Obtenido de 12V High-Power (HP) 25D mm Gearmotors: <https://www.pololu.com/category/186/12v-high-power-hp-25d-mm-gearmotors>
- Corporation, ©.-2. D. (2020). *SolidWorks* . Obtenido de SolidWorks : https://www.solidworks.es/sw/purchase/requestdemo.htm?mktid=7614&gclid=Cj0KCQjwvIT5BRCqARIsAAwwD-ScrdNkQyX0of8E3CgOTH_VTFPCgyu6jJCxgDDRnwrEluzkm7VDqboaAtnSEALw_wcB&gclid=aw.ds
- cPanel, I. (2018). *Sistemas Master Magazine*. Obtenido de <https://sistemas.com/proceso-paralelo.php>
- Daniel Eduardo Hernández Sánchez, J. R. (2017). Diseño, construcción y modelo dinámico de un robot móvil de tracción diferencial aplicado al seguimiento de trayectorias. *MEMORIAS DEL XXIII CONGRESO INTERNACIONAL ANUAL DE LA SOMIM*, 7.
- Dr. Roberto Hernández Sampieri, D. C. (2014). *Metodologia de la Investigacion* . Edificio Punta Santa Fe Prolongación Paseo de la Reforma 1015, Torre A Delegación Álvaro Obregón C.P. 01376, México D.F.: McGrawHill.
- Edwin J. Purcell, D. V. (2007). *Calculo* . Mexico: PEARSON EDUCACION.
- Electronics, D. (2014). *Dynamo Electronics* . Obtenido de Dynamo Electronics : https://www.dynamoelectronics.com/descargas/Baterias_Lipo.pdf
- Felipe W. Trevisan, M. M. (2012). Trajectory-Based Short-Sighted ProbabilisticPlanning. *Carnegie Mellon University - Pittsburgh, PA*, 9.
- Hat, R. (2020). *Qué es una API*. Obtenido de Qué son las API y para qué sirven : <https://www.redhat.com/es/topics/api/what-are-application-programming-interfaces>
- Inc, D. (5 de diciembre de 2019). *ABC Tecnologia* . Obtenido de <https://www.abc.es/tecnologia/informatica-hardware/20130716/abci-raspberry-como-201307151936.html>
- INEGI. (2016). La discapacidad en Mexico. *INEGI*, 368.

- INEGI. (s.f.). Clasificación de tipo de discapacidad - Histórica. *INEGI*, 54.
- Irving Herrera-Luna, H. V.-F.-H. (2019). Sensor Fusion Used in Applications for Hand Rehabilitation: A Systematic Review. *IEEE*, 11.
- JHONNY A. VALENCIA V. ALEJANDRO MONTOYA O. LUIS HERNANDO RIOS, M. S. (2009). MODELO CINEMÁTICO DE UN ROBOT MÓVIL TIPO DIFERENCIAL Y NAVEGACIÓN A PARTIR DE LA ESTIMACIÓN ODOMÉTRICA. *Scientia et Technica Año XV*, 6.
- Jorge Israel Pérez Arreguín, S. T. (2011). Robot Móvil de Tracción Diferencial con Plataforma de Control Modular para Investigación y Desarrollo Ágil de Proyectos. *10º Congreso Nacional de Mecatrónica*, 3.
- juntos.org.mx. (s.f.). *Juntos*. Obtenido de Juntos: <https://www.juntos.org.mx/nuestra-mision/>
- Leonardo Enrique Solaque Guzmán, M. A. (2014). SEGUIMIENTO DE TRAYECTORIAS CON UN ROBOT MÓVIL DE CONFIGURACIÓN DIFERENCIAL. *Ing. USBMed, Vol. 5, No. 1*, 9.
- M. Kryuchcov Castro-Antonio, G. C.-A.-L.-H.-F.-R. (2019). An Approach based on a Robotics Operation System for the Implementation of Integrated Intelligent House Services System. *IEEE*.
- M.M. Veloso, F. T. (2012). Trajectory-based short-sighted probabilistic planning. *ResearchGate*.
- middleware, R. (Mayo de 2017). *Robotics middleware*. Obtenido de https://en.wikipedia.org/wiki/Robotics_middleware#ROS
- N S Widodo, S. A. (2018). Robot Operating System (ROS) Compatible Low Cost Rotating Light Detection and Ranging (Lidar) Design. *IOP Conference Series: Materials Science and Engineering*, 6.
- Occidental, E. (28 de Abril de 2019). Lanzarán plataforma para promover empleo. *Lanzarán plataforma para promover empleo*, pág. 1.
- Ogata, K. (s.f.). *Ingeniería de control moderna*. Minnesota: Pearson Education.
- Oscar Alonso-Ramírez 1, A. M.-H.-F.-H.-R. (2018). A Graph Representation Composed of Geometrical Components for Household Furniture Detection by Autonomous Mobile Robots. *applied sciences MDPI*, 19.
- Pacheco, P. (s.f.). *An Introduction Parallel Programming*. San Francisco USA: Morgan Kaufmann.
- Pérez Arreguín Jorge Israel, T. A. (2011). Robot Móvil de Tracción Diferencial con Plataforma de Control Modular para Investigación y Desarrollo Ágil de Proyectos. *10º Congreso Nacional de Mecatrónica*, 6.
- Philip Cooksey, M. V. (2018). Effective Team Coordination through Intra-Robot Replanning to Restore Team Plan Rationale. *Robotics Institute, Carnegie Mellon University*, 6.
- ROS.org. (s.f.). *ROS.org*. Obtenido de <http://wiki.ros.org/melodic/Installation/Ubuntu>

- Rospy. (08 de Noviembre de 2017). *Open Source Robotics Foundation* . Obtenido de <http://wiki.ros.org/rospsy>
- Ruiz, H. M. (2012). *Metodologia de la investigacion* . Mexico DF: CENGAGE Learning.
- Salud, O. M. (2016). Informe Mundial de la Discapacidad . *Discapacidad y Rehabilitacion*, 27.
- Sánchez, I. C. (2020). Raspberry Pi & Arduino, los artesanos del siglo XXI y el movimiento maker. *Asociacion colombiana de ingenieros de sistemas* , (pág. 89). Colombia.
- Sergio Hernandez-Mendez, C. M.-M.-H.-F. (2017). Detecting falling people by autonomous service robots: A ROS module integration approach. *IEEE*, 2.
- Sites.google.com. (2020). *Lenguajes de programacion T3 P1*. Obtenido de Modularidad : <https://sites.google.com/site/lenguajedeprogramacionit3p1/iv-desarrollo-de-modularidad/unidad-iv-desarrollo-de-modularidad>
- Sole, A. C. (s.f.). *Intrumentacion Industrial* . Alfaomega.
- Tiago Pereira, A. M. (2017). Optimal Perception Planning with Informed Heuristics Constructed from Visibility Maps. *Journal of Intelligent & Robotic Systems*, 24.
- Tiago Pereira, N. L. (2018). Heterogeneous multi-agent planning using actuation maps. *IEEE*, 3.
- UPIITA, I. B. (2019). *ROBOTS MÓVILES DE RUEDAS: GENERALIDADES*. Obtenido de ROBOTS MÓVILES DE RUEDAS: GENERALIDADES: <http://www.boletin.upiita.ipn.mx/index.php/ciencia/593-cyt-numero-45/1081-robots-moviles-de-ruedas-generalidades>
- Varberg, P. (s.f.). *Calculo* . Pearson.
- Veloso's, M. (October de 2020). *Manuela Veloso's Publications*. Obtenido de Manuela Veloso's Publications: <http://www.cs.cmu.edu/~mmv/Veloso.html>
- Veracruzana, U. (2020). *Artificial Intelligence Research Center*. Obtenido de Universidad Veracruzana : <https://www.uv.mx/anmarin/>
- Vicente Guerrero, R. L. (2015). Comunicaciones Industriales . En R. L. Vicente Guerrero, *Comunicaciones Industriales* (págs. 17-26). Alfaomega, mrcombo .
- Wendy Janett Guzman Gonzalez, J. R. (2014). Fusion de datos en redes de sensores: una revision del estado del arte. *Revista Politecnica, ISSN 1900-2351*, 12.
- YoonSeok Pyo, H. C. (2017). *ROS Robot Programming*. GeumCheon-gu, Seoul, Republic of Korea: ROBOTIS Co., Ltd.