



**CENTRO DE INVESTIGACIÓN Y DE
ESTUDIOS AVANZADOS DEL
INSTITUTO POLITECNICO NACIONAL**

**Diseño y construcción de un robot autónomo para
transportación intrahospitalaria de insumos en un
hospital con alto grado de contagio.**

Tesis que presenta
Ing. Sergio Delgadillo Marín

Para obtener el grado de
Maestro en Ciencias

En la especialidad de
Sistemas Autónomos de Navegación Aérea y Submarina

Asesores de Tesis:
Dr. Sergio Rosario Salazar Cruz
Dr. Antonio Osorio Cordero

CIUDAD DE MÉXICO

Abril 2022.

Resumen

Hoy en día, derivado de la pandemia por COVID-19 se ha despertado un gran interés en la implementación de soluciones tecnológicas en las diversas áreas del cuidado de la salud. Actualmente se cuenta con una gran variedad de robots que son utilizados en ambientes cerrados, el problema radica en implementar este tipo de soluciones en entornos abiertos donde interactúan con elementos dinámicos los cuales hacen que el entorno este cambiando constantemente y en consecuencia el sistema no se pueda controlar fácilmente.

En este trabajo se realizó el diseño y construcción de un prototipo robótico pensado como plataforma de desarrollo para aplicaciones que involucren interacción con humanos en ambientes hospitalarios. El tipo de robot seleccionado para este propósito fue el robot móvil con ruedas de tipo diferencial debido a que aunque esta configuración es no holónomica resulta mas fácil de construir y controlar que la alternativa omnidireccional.

Se explican los conceptos necesarios para comprender el funcionamiento del prototipo así como los elementos necesarios para hacer la selección y diseño de los componentes que lo conforman. Se obtiene el modelo matemático que describe la dinámica del prototipo y se realiza la programación del robot utilizando ROS en un sistema embebido.

Los experimentos realizados comprobaron que el robot es capaz de realizar la tarea de regulación de postura y navegación en interiores. Además, es capaz de generar el mapa de ocupación del lugar donde se encuentra y actualizarlo conforme este cambia con el tiempo.

Finalmente, con base en la experiencia y resultados obtenidos se proponen trabajos futuros para mejorar la funcionalidad del robot e integrar nuevas funciones al mismo.

Abstract

Nowadays, as a consequence of the Covid-19 pandemics, increasing interest in the implementation of technological solutions to the care health problem, has arisen. A great variety of robots are currently being employed in structured environments, however actual situations arise where robots must interact with dynamical objects that modify the environment constantly. This leads to make the control of the robots a difficult task to perform.

This work presents the design and construction of a robotic platform aiming to the development of applications involving the interaction with humans in hospital environments. The type of robot selected to do this is a wheeled mobile vehicle with wheels of the differential kind, this configuration is not holonomic however it is easier to build and control as opposed to the omnidirectional alternative.

The necessary concepts are explained in this document to fully understand the functioning of the prototype and to perform the appropriate design and selection of its components. The dynamical model describing the prototype, as well as the necessary software provided by ROS for imbedded environments are presented, in order to perform its control.

The experiments performed confirm that the robot is able to fulfill the position regulation task and indoors navigation. In addition, it is able to generate the occupation map of the place and its updating as it changes over time.

Finally, based on the experience and the results obtained, future work is proposed to improve the functionality of the robot and to add extra functions to it.

Agradecimientos

Agradezco al Consejo de Ciencia y Tecnología (CONACYT) por el apoyo económico brindado durante el transcurso de mis estudios de maestría.

A mis asesores de tesis, Dr. Sergio Salazar, Dr. Antonio Osorio y al Dr. Ricardo López, gracias por permitirme formar parte de su equipo de trabajo, por el apoyo brindado a lo largo de este tiempo y por el conocimiento compartido.

De igual forma, a mis profesores de la UMI-LAFMIA, gracias por los conocimientos y experiencias compartidas.

A mi familia, especialmente a mi madre y a mi tía por siempre apoyarme incondicionalmente, y a pesar de la distancia. Espero que este logro lo sientan como suyo y permita retribuir una pequeña parte de todo lo que ustedes han hecho por mi.

A mis compañeros, por compartir momentos de risas, corajes, desesperación, pero sobre todo; gracias por su amistad.

Sinceras gracias a todas las demás personas que en algún momento de mi vida me apoyaron y motivaron para seguir adelante, a mis familiares y amigos que me brindan su apoyo incondicional y la fuerza para seguir avanzado.

Glosario

Notación matemática

\mathbb{R} : Conjunto de números reales.

\mathbf{u} : Ley de control (vector).

τ : Par de control (vector).

\mathbf{I} : Marco de referencia inercial (fijo a la tierra).

\mathfrak{B} : Marco de referencia fijo al vehículo.

v : Velocidad lineal.

r : Radio.

ω : Velocidad angular.

A^T : Transpuesta de la matriz A .

A^+ : Pseudoinversa de la matriz A .

Terminología

Actuador: Es aquel dispositivo o subsistema que se encarga de regular la potencia de una planta. La gama de actuadores que se pueden controlar es muy extensa y variada. Entre los más habituales se encuentran los destinados a producir movimientos (motores y cilindros), los destinados a trasiego de fluidos (bombas) y los de tipo térmico.

Control: Es la manipulación indirecta de las magnitudes de un sistema denominado planta a través de otro sistema llamado sistema de control.

Error: Es la diferencia entre el valor leído o transmitido por el instrumento y el valor real de la variable medida.

Paquete: Los paquetes de ROS pueden contener nodos, conjuntos de datos, archivos de configuración o cualquier contenido que constituya módulos útiles y no relacionados con ROS. El objetivo de los paquetes es proporcionar la funcionalidad de poder consumir y reutilizar software fácilmente.

Pila: Mecanismo principal de ROS para la distribución de software. Son la recopilación de paquetes con las dependencias necesarias para en conjunto proporcionar funcionalidad.

Planta: Es una parte del equipo o máquina que se va a controlar.

Robustez: El grado en el cual un sistema o el componente pueden funcionar correctamente en la presencia de entradas inválidas o condiciones no deseadas, a los sistemas que tienen esta propiedad se les denomina sistemas tolerantes a fallas.

Sistema: Los sistemas físicos en un sentido más amplio, son una interconexión de componentes, dispositivos o subsistemas. Un sistema puede considerarse como un proceso en el cual las señales de entrada son transformadas por el sistema o provocan que éste responda de alguna forma, lo que da como resultado otras señales como salida.

Sistema en lazo abierto: Es aquel sistema en el que no hay una realimentación de la información para ajustar y mantener la salida deseada.

Sistema en lazo cerrado: Es aquel sistema en el que hay una realimentación de la información (por medio de sensores), que permiten ajustar y mantener la salida deseada.

Sistema de control: Es aquel que se encarga de gobernar la respuesta de una planta sin que el operador intervenga directamente sobre los elementos de salida.

Abreviaturas y acrónimos

COVID-19: Enfermedad por coronavirus 2019

DDMR: Robot móvil diferencial.

DOF: Grado de libertad.

EKF: Filtro de Kalman extendido.

IMU: Unidad de medición inercial.

LIDAR: Detección de luz y rango.

MATLAB: Software MATrix LABoratory, (laboratorio de matrices).

PD: Acción de control proporcional derivativa.

PID: Acción de control proporcional-integral-derivativa.

PWM: Modulación por ancho de pulso.

RC: Radio control.

ROS: Sistema Operativo Robótico.

RPM: Revoluciones por minuto.

SLAM: Localización y mapeo simultáneos.

SMC: Control por modos deslizantes.

UGV: Vehículo terrestre no tripulado.

UWB: Banda ultra ancha.

WMR: Robot móvil con ruedas.

Índice general

Glosario	v
Índice de tablas	x
Índice de figuras	xi
1. Introducción	1
1.1. Motivación	1
1.2. Objetivos	4
1.2.1. Objetivo general	4
1.2.2. Objetivos específicos	4
1.3. Hipótesis	4
1.4. Planteamiento de la solución	5
1.5. Contribuciones	6
1.6. Trabajos recientes	6
1.6.1. Control automático	6
1.6.2. Navegación	9
1.6.3. Robots en el área médica	11
1.7. Estructura de la tesis	12
2. Modelado matemático y problema de navegación	14
2.1. Robots móviles	14
2.1.1. Clasificación	15
2.1.2. Características	16

2.2. Modelado matemático	17
2.2.1. Modelo cinemático	17
2.2.2. Modelo dinámico	21
2.3. Navegación	22
3. Diseño y selección de la instrumentación del prototipo	27
3.1. Instrumentación	27
3.1.1. Criterios de selección	27
3.1.2. Sistema de navegación	28
3.1.3. Etapa de potencia.	33
3.1.4. Alimentación eléctrica	35
3.2. Diseño de la estructura	37
4. Implementación y resultados	45
4.1. Resultados experimentales	45
4.1.1. Estimación de la pose del vehículo	45
4.1.2. Control	53
4.1.3. Navegación	59
5. Conclusiones y trabajos futuros	64
5.1. Conclusiones	64
5.2. Trabajos futuros	65
A. Diagramas y especificaciones de la estructura	67
B. Especificaciones técnicas de la instrumentación seleccionada	76
Bibliografía	85

Índice de tablas

1.1. Gestión de COVID-19 utilizando soluciones robóticas.	2
3.1. Características de la unidad de medición inercial de la cámara T265.	28
3.2. Características del sensor URG-04LX-UG01.	30
3.3. Propiedades mecánicas del Poliuretano termoplástico 95A	32
3.4. Características del moto reductor Pololu 37D.	33
3.5. Consumo energético de los componentes principales del robot.	36
3.6. Especificaciones del regulador Xl4016.	36
3.7. Propiedades mecánicas del prototipo diseñado.	44
4.1. Error de estimación de la posición final del vehículo.	50
4.2. Dimensiones principales de la habitación.	53

Índice de figuras

2.1. Componentes de un sistema robótico.	14
2.2. Clasificación de los robots de acuerdo a su movilidad.	15
2.3. Configuración básica del robot diferencial.	18
2.4. Problema SLAM.	23
3.1. Cámara de seguimiento Intel RealSense T265.	29
3.2. Sensor LIDAR urg-04lx-ug01.	29
3.3. Computadora de placa única NVIDIA Jetson Nano.	31
3.4. Vistas básicas del montaje de los sensores en la base diseñada.	32
3.5. Vistas básicas del ensamble del amortiguador de vibración.	32
3.6. Moto reductor Pololu 37D con encoder.	33
3.7. Módulo puente H TB6612FNG.	34
3.8. Vistas básicas del diseño de la PCB para el control de los motores.	35
3.9. Regulador XL4016.	37
3.10. Vista isométrica del diseño del prototipo.	38
3.11. Vista superior del diseño del prototipo.	38
3.12. Vista lateral del diseño del prototipo.	39
3.13. Vista frontal del diseño del prototipo.	39
3.14. Ensamble experimental del prototipo, vista superior.	40
3.15. Ensamble experimental del prototipo, vista frontal.	40
3.16. Ensamble experimental del prototipo, vista lateral.	41
3.17. Ensamble experimental del prototipo con la cámara en la parte posterior, vista frontal.	42

3.18. Ensamble experimental del prototipo con la cámara en la parte posterior, vista posterior.	42
3.19. Diseño de los engranes de la transmisión reductora.	43
3.20. Implementación de la transmisión reductora en el robot.	43
4.1. Superficie acondicionada para pruebas experimentales.	46
4.2. Gráfica de la orientación del robot sin amortiguadores de vibración.	47
4.3. Gráfica del plano X-Y de la trayectoria del robot sin amortiguadores de vibración.	47
4.4. Gráfica de la orientación del robot con amortiguadores de vibración.	48
4.5. Gráfica del plano X-Y de la trayectoria del robot con amortiguadores de vibración.	49
4.6. Grafo de los nodos y tópicos activos para la generación del mapa de ocupación.	51
4.7. Proceso de construcción del mapa de ocupación.	52
4.8. Medidas de la habitación obtenidas mediante el algoritmo de <i>gmapping</i>	53
4.9. Descripción del sistema de control y navegación.	54
4.10. Medición de la velocidad lineal del robot.	56
4.11. Medición de la velocidad angular del robot.	56
4.12. Gráfica de la velocidad lineal del robot con un control PID.	57
4.13. Gráfica de la velocidad angular del robot con un control PID.	58
4.14. Gráfica de las velocidades lineal y angular del robot.	58
4.15. Configuración de la pila de navegación.	59
4.16. Comportamiento del modo de recuperación de la pila de navegación.	60
4.17. Gráfica de la posición del robot en la coordenada X.	60
4.18. Gráfica de la posición del robot en la coordenada Y.	61
4.19. Gráfica de la orientación del robot.	61
4.20. Interfaz RViz de ROS junto con el prototipo en tiempo real durante su desplazamiento a una pose objetivo.	63
B.1. Características eléctricas del controlador TB6612FNG.	76
B.2. Dimensiones del controlador TB6612FNG.	77

B.3. Diagrama esquemático de la tarjeta de control de motores.	78
B.4. Especificaciones de los sensores de imagen de la cámara seguimiento Intel RealSense T265.	79
B.5. Especificaciones de la unidad de medición inercial de la cámara seguimiento Intel RealSense T265.	79
B.6. Requerimientos energéticos de la cámara seguimiento Intel RealSense T265.	79
B.7. Dimensiones de la cámara de seguimiento Intel RealSense T265.	80
B.8. Diagrama esquemático de la tarjeta Arduino Nano.	81
B.9. Especificaciones técnicas de la tarjeta Arduino Nano.	82
B.10. Dimensiones del sensor Lidar URG-04LX-UG01.	83
B.11. Especificaciones técnicas del sensor Lidar URG-04LX-UG01.	84
B.12. Especificaciones técnicas de la tarjeta Nvidia Jetson Nano.	85

Capítulo 1

Introducción

1.1. Motivación

En el último año la pandemia COVID-19 ha influido en casi todos los aspectos de nuestras vidas y la atención médica no está exenta. Inicialmente se contempló a la tecnología como una herramienta que facilitaría o mejoraría los servicios médicos en un futuro, sin embargo, dada la severidad del COVID-19, implementar soluciones tecnológicas en las diversas áreas del cuidado de la salud se ha vuelto algo esencial, mediante la implementación de la tecnología, se puede mantener la distancia de seguridad entre los profesionales de la salud y los pacientes minimizando la probabilidad de contagio, asimismo se puede utilizar para generar diagnósticos mediante técnicas de inteligencia artificial, informes de resultados, realizar visitas y consultas de manera remota entre muchas otras aplicaciones.

Hoy en día China es uno de los pocos países en los que se ha logrado controlar casi en su totalidad la propagación del COVID-19, es por ello que merece la pena analizar el cómo el ministerio de salud chino ha manejado el brote de esta enfermedad. En China, se ha implementado extensamente el uso de robots, desde robots de servicio, desinfectantes y en los hospitales para recibir a los pacientes en el mostrador de recepción y monitorear sus síntomas iniciales (como gripe y fiebre) y desinfectarlos justo en el punto de entrada. En la tabla (1.1) se muestra una propuesta de soluciones robóticas en los pasos de la gestión del COVID-19 [1].

Recomendación para el manejo del COVID-19	Categoría	Prevención	Solución robótica
Contacto inicial y evaluación	Cuidados primarios y emergencias	Equipo de protección personal (EPP)	Robot enfermero, doctor robótico, robot ambulancia
Higiene de manos	Higiene personal	Esterilización	Dispensador de sanitizante robótico
Descontaminación de superficies	Higiene ambiental	Uso de desinfectantes basados en alcohol	Robot desinfectante UV para exteriores, pulverizador para exteriores
Transporte de pacientes	Transferencia ambulatoria	EPP	Vehículo autónomo, camilla autónoma
	Medidas administrativas	EPP	Robot recepcionista
	Manejo de pacientes	EPP	Robot de telemedicina, robot de elevación para mover pacientes de un lugar a otro
	Farmacia	EPP	Robot dispensador de medicinas, drones de reparto
	Servicio de alimentos	EPP	Robot cocinero, robot repartidor de comida
	Cuidados del hogar	EPP	Robots de servicio autónomos
Pruebas de laboratorio e imágenes	Toma de muestras de sangre y rayos X	EPP	Robots de limpieza y desinfección
			Robot para recolección de muestras, robot de rayos X , biopsia con robot quirúrgico
Manejo de los decesos	Medidas administrativas	EPP	Robot enfermero para elevación, vehículo autónomo para transporte al cementerio
Cuidados a largo plazo	Cuidado paliativo	EPP	Robots de entretenimiento, robot de telemedicina, robot de enfermería, rehabilitación y robots de asistencia

Tabla 1.1: Gestión de COVID-19 utilizando soluciones robóticas.

Como se puede observar, son muchas las áreas de oportunidad para la robótica en el área médica, dentro de ellas destaca el uso de los vehículos para transporte de pacientes, medicamentos, muestras, insumos, etc. Actualmente el interés por trabajar con este tipo

de vehículos ha incrementado, en especial por aquellos que son capaces de desplazarse de forma autónoma a través de entornos dinámicos.

A pesar de los avances que se han tenido, aún quedan cosas por hacer, debido a la premura de satisfacer la demanda de los hospitales aún no se logra la autonomía total, es decir, muchos de los vehículos desarrollados son teleoperados, esto trae consigo la necesidad de contar con personal calificado para operar el robot, incrementa los costos de operación y puede propiciar que un nuevo grupo de profesionales se vea expuesto al COVID-19 lo que resulta contraproducente.

El laboratorio de la UMI tiene experiencia en el desplazamiento de vehículos que siguen alguna trayectoria de manera autónoma. El antecedente más cercano es el proyecto del mini-vehículo terrestre que suministró el gobierno alemán a través del Doctor *R. Rojas* y con el que se ha trabajado.

Por consiguiente, lo anterior motiva la necesidad de diseñar, construir y controlar un robot móvil que sea capaz de desplazarse en ambientes no estructurados, con agentes dinámicos (personas) de forma autónoma, siendo capaz de interactuar con ellos si es necesario y realizar la entrega de insumos médicos.

La pertinencia del proyecto está justificada con la ocurrencia de la pandemia por COVID-19. Una situación similar podría repetirse en el futuro, pero aunque eso no ocurra, existen infinidad de enfermedades infecciosas cuyo esparcimiento podría controlarse con la ayuda de un robot como el que se propone, además de poder ser utilizado en ambientes hospitalarios estériles como lo son los hospitales de quemados o centros de tratamiento de cáncer.

1.2. Objetivos

1.2.1. Objetivo general

Diseñar, construir y controlar un vehículo autónomo que cuente con las características físicas y la instrumentación necesaria que le permitan desplazarse a los diferentes destinos dentro de un hospital con alto grado de contagio, obteniendo un prototipo, con una arquitectura que permita implementar diversas estrategias de control y navegación.

1.2.2. Objetivos específicos

- Diseñar un vehículo terrestre con las características físicas que le permitan desplazarse adecuadamente dentro de un hospital.
- Seleccionar e implementar la instrumentación eléctrica y electrónica necesaria del vehículo para que realice las tareas de navegación y control.
- Implementar un esquema de control clásico para la estabilización de la velocidad del robot.
- Implementar un algoritmo de navegación que permita el desplazamiento del vehículo de su lugar de origen a los diferentes puntos y/o lugares determinados de las áreas del hospital. Lo anterior, considerando tanto ambientes abiertos así como cerrados.
- Crear una arquitectura abierta que permita la implementación de diversos algoritmos de navegación y control en trabajos futuros.

1.3. Hipótesis

Se puede desarrollar un vehículo autónomo que en su aplicación dentro de entornos hospitalarios con alto grado de contagio pueda reducir el esparcimiento de enfermedades infecciosas disminuyendo la interacción entre los profesionales de la salud y los pacientes.

1.4. Planteamiento de la solución

En este trabajo se aborda el problema de diseñar, construir y controlar un robot que transporte insumos médicos dentro de determinadas áreas del hospital donde existe alta probabilidad de contagio. La propuesta es un vehículo autónomo que tenga una estructura que permita el transporte de diferentes tipos de insumos utilizados en los hospitales.

De lo anterior, se observa que se requiere cumplir los siguientes puntos:

1. Diseñar y construir un prototipo del cual se tendrán todos los cálculos y consideraciones necesarias para su desarrollo.
2. Seleccionar adecuadamente los componentes electrónicos (por ejemplo, sensores, actuadores, interfaz) necesarios para garantizar la correcta navegación.
3. Diseñar una plataforma que pueda ser fácilmente modificada para transportar los diferentes tipos de insumos médicos.
4. Evaluar el desempeño de la plataforma elaborada mediante la comparación de los resultados teóricos y prácticos.

A partir de los objetivos planteados y los puntos antes mencionados, se establece la siguiente metodología a seguir:

1. Estudio y selección de los componentes a utilizar en el prototipo.
2. Pruebas de los componentes a utilizar.
3. Estudio y selección de algoritmos de control y navegación propuestos en diversos trabajos e investigaciones de vehículos terrestres.
4. Diseño de la estructura considerando los requerimientos mecánicos.
5. Simulación de los algoritmos de control y navegación seleccionados.
6. Fabricación del prototipo.

7. Instrumentación del prototipo.
8. Validar los resultados teóricos con los experimentales.

1.5. Contribuciones

La principal aportación de este trabajo es desarrollar los algoritmos necesarios para realizar la navegación autónoma de un vehículo terrestre y realizar la validación de los mismos. Este trabajo proporcionará las bases necesarias para implementar diversos algoritmos de navegación autónoma en diferentes plataformas robóticas.

1.6. Trabajos recientes

En los últimos años se han desarrollado innumerables investigaciones en torno a los WDR, en lo que concierne al estudio y construcción de estos vehículos se han abordado diversos campos de investigación, entre los cuales existen trabajos en el área de desarrollo de algoritmos de control, navegación y percepción. Así también, se han desarrollado trabajos que estudian el impacto de los robots en el área médica. Algunos de los trabajos recientes en las áreas mencionadas se describen en las siguientes subsecciones.

1.6.1. Control automático

Con lo que respecta al control automático, *J García et. al. (2018)* en [2] presentan el diseño e implementación de un controlador de seguimiento conmutado robusto, en el diseño del control presentan un enfoque jerárquico donde consideran de forma independiente el diseño de la ley de control para la cinemática de la estructura mecánica, actuadores y etapa de potencia. Los resultados experimentales que se presentan demuestran que aún ante la presencia de perturbaciones paramétricas los controladores presentan un buen desempeño. El enfoque jerárquico utilizado en este trabajo tiene mucho sentido ya que como se menciona en la subsección (2.1.2), estos subsistemas tienen una influencia directa en el desempeño del robot, basado en este paradigma *M LUO et. al* presentan en [3] una revisión sobre el diagnóstico y predicción de fallas en WMRs; los resultados que se presentan

en este trabajo muestran como un enfoque basado en modelos puede ayudar a predecir falla, esto nos es especialmente útil ya que considerando los subsistemas más susceptibles a fallas podemos realizar un diseño más robusto desde la etapa de diseño preliminar y no vernos forzados a realzar rediseños en la etapa de producción, además, puede contribuir en la reducción de costos de mantenimiento.

En [4] *FG Rojas et. al. (2017)* presentan el análisis del seguimiento de la trayectoria de un DDWR considerando dos modelos de diferente grado relativo. El primer modelo considera las velocidades de las ruedas como entradas de control. En este caso, se implementa el algoritmo Super Twisting. El segundo modelo considera las tensiones del motor como entradas de control. En este caso, se implementa el Algoritmo de torsión continua. La comparación de las estrategias tanto en simulaciones como de forma experimental muestran que por un lado proporcionan rechazo de perturbaciones y, por otro lado, tienen menos efecto de cascabeleo (shattering) que el SMC de primer orden. En particular, se implementa un nuevo SMC continuo con estas propiedades denominado Modo de deslizamiento continuo.

Otro trabajo que implementa un SMC pero con un enfoque diferente es el presentado por *P Yazdjerdi et. al. (2017)* en el artículo [5], este artículo presentan el diseño de un controlador tolerante a fallas de actuador para WRMs, el robot se modela considerando las fallas en los actuadores como perturbaciones de entrada, dado que el SMC es un control robusto, su propiedad de robustez muestra en los resultados experimentales que es capaz de compensar adecuadamente las fallas de actuador. Esta robustez nos es especialmente útil ya que a pesar de que se presente pérdida de efectividad en los actuadores, el controlador será capaz de mantener operando correctamente el vehículo sin la necesidad de la detección y corrección de la falla inmediatamente.

Además de los enfoques antes mencionados, *Y Koubaa et. al. (2018)* presentan en [6] el modelado del WRM donde existen dinámicas no modeladas o inciertas, con el propósito de compensar estas dinámicas se propone un control por modos deslizantes adaptativo, este

control incorpora un término de estimador para compensar las perturbaciones externas y las incertidumbres y un término de retroalimentación para mejorar la estabilidad en lazo cerrado. La estabilidad del sistema se analiza utilizando la teoría de Lyapunov garantizando estabilidad asintótica. Además, las simulaciones presentadas confirman la robustez del esquema de control diseñado.

Como ya se ha mencionado, uno de los principales problemas en el control de seguimiento de trayectoria de los DDWRs es el deslizamiento producido por la pérdida de agarre de las ruedas contra el suelo sin embargo la fricción entre el suelo y la rueda también tiene otros efectos en el seguimiento de trayectorias ya que el suelo disipará cantidades mayores de energía motriz, especialmente en entornos exteriores sofisticados. En general, este tipo de fricción entre la rueda y el suelo se puede considerar una perturbación del par de accionamiento del motor de la rueda, en [7] *M Yue et.al. (2017)* estudia el control de un DDWR con un enfoque de seguimiento de trayectoria basado en la técnica de control por modos deslizantes terminal. Se proponen diversos comportamientos de fricción, los resultados de la simulación numérica muestran que se mejorará significativamente la adaptabilidad del terreno, precisión de movimiento y capacidad de maniobra.

Así también, *A Stefek et. al. (2020)* presentan en [8] un artículo donde se evaluó el consumo energético de los controladores para DDWRs y la precisión del seguimiento de trayectoria, esto con el fin de establecer un criterio de selección de las estrategias de control para estos vehículos y poder decidir si vale la pena sacrificar un mayor consumo energético a costa de tener un seguimiento más preciso o si de acuerdo a los requerimientos de diseño se puede permitir un margen de error. Los resultados muestran que los controladores de bajo consumo energético presentan una incertidumbre en el seguimiento de trayectoria del 5% mientras que los de alto consumo presentan un consumo energético de hasta el 200% mayor, esto nos presenta un parámetro más para considerar en la etapa de diseño, este parámetro como ya se mencionó tendrá una relación directa con la autonomía de nuestro vehículo.

Tal como se puede observar en los trabajos mencionados anteriormente, existen muchas áreas de investigación en torno a los DDWRs, de ellas destacan el control y la navegación, en estos trabajos se observa que se ha logrado un avance considerable a lo largo de los años, sin embargo, pocos trabajos consideran la unión de todas estas áreas. Lo anterior motiva a diseñar una metodología que integre todos los conocimientos que se han obtenido con el fin de realizar diseños más eficientes que se puedan adaptar fácilmente a los requisitos de diseño que se requieren para resolver las problemáticas particulares de cada proyecto.

1.6.2. Navegación

En lo que se refiere a navegación, *YX Wang et. al. (2020)* presentan en [9] la implementación de navegación SLAM en ROS, el principal aporte de su trabajo es evaluar la implementación de diferentes sensores y tecnologías de posicionamiento, como detección de luz y rango (LIDAR), unidad de medición inercial (IMU), odómetro y banda ultra ancha (UWB) y fusionar estos datos mediante el algoritmo de filtro de Kalman extendido (EKF) para proporcionar precisión en el posicionamiento del robot. Los resultados que presentan muestran que el uso del EKF proporciona una navegación factible con un error de posicionamiento de $10cm$. Para nuestro trabajo este artículo es útil ya que presenta los resultados de una implementación física y nos da una idea de los errores esperados basados en el tipo de sensor utilizado para la navegación.

Así también, *SP Thale et. al. (2020)* presentan en su artículo [10] una descripción general de las herramientas con las que cuenta ROS para realizar la simulación, implementación y visualización de datos para la navegación autónoma, este artículo nos proporciona un buen punto de partida para comprender porque del uso tan extendido de ROS en el ámbito de la investigación y desarrollo de robots.

Por otra parte, *R Giubilato et. al. (2019)* presentan en [11] la evaluación de algoritmos de navegación SLAM visual, en este caso el sensor utilizado es una cámara estereoscópica, así como en el trabajo citado anteriormente se utiliza ROS para hacer la implementación

de software y se utiliza además una placa de computadora Jetson TX2. Para realizar el procesamiento de las imágenes utilizan OpenCV, una librería desarrollada específicamente para esta tarea. Las principales aportaciones que realizan es primeramente mostrar una comparativa de la eficiencia de los algoritmos, posteriormente evaluar el consumo de recursos de computo necesarios para obtener buenos rendimientos en los mismos y finalmente proporcionan una comparativa de los beneficios de repartir la carga de trabajo del computador entre su CPU y GPU. Este artículo proporciona un punto de partida para la selección de los algoritmos y hardware necesarios para utilizar visual SLAM.

Además, *Z Meng et. al. (2020)* presentan en [12] la obtención del modelo dinámico y cinemático de una plataforma experimental, y realizan la evaluación de diferentes algoritmos SLAM, basados en la comparación de estos algoritmos determinan que la opción más viable es utilizar GMapping como algoritmo para el mapeo láser y el algoritmo de posicionamiento adaptativo Monte Carlo. La implementación la realizan nuevamente en ROS y de acuerdo a lo reportado en sus resultados, tras realizar la optimización del algoritmo pudieron mejorar la precisión del mapeo.

Otro trabajo relevante es el presentado por *P Sankalprajan et. al. (2020)* en [13], en este artículo realizan una comparativa similar a algunos de los trabajos anteriormente citados pero en este caso comparan diferentes algoritmos SLAM tanto 2D como 3D a nivel simulación en ROS y en tiempo real con una plataforma a escala. Para validar los resultados de la simulación desarrollan un escenario virtual que se asemeja a un estacionamiento subterráneo, de acuerdo a los resultados presentados, los algoritmos presentan comportamientos similares tanto a nivel simulación como tiempo real, además los resultados arrojan que el algoritmo que mejor rendimiento tiene es el GMapping.

Con base a los resultados obtenidos en los trabajos anteriormente presentados, podemos observar que consistentemente seleccionan a ROS como el sistema operativo ideal para realizar la implementación tanto a nivel simulación como física, en lo que respecta a los algoritmos de SLAM, se muestra que los algoritmos basados en la tecnología LIDAR

y que utilizan la fusión de datos son los que presentan resultados robustos. Otra cosa que se puede observar es que en las diferentes evaluaciones e implementaciones revisadas no consideran ambientes dinámicos, esto nos motiva a continuar con el estudio de estos algoritmos de navegación contemplando este enfoque que es el más cercano a la vida real.

1.6.3. Robots en el área médica

Los robos móviles han sido ampliamente utilizados en el área médica los últimos años, teniendo un auge a partir de la pandemia del COVID-19, algunos de los últimos trabajos realizados en torno a ellos se presentan a continuación.

En 2020, *MP. Fanti et. al.* en su artículo [14] aborda la problemática de la distribución de medicamentos en hospitales donde se trata el COVID-19, para disminuir el número de interacciones entre personas y además disminuir el tamaño de las flotas de robots necesarios para la distribución propone el enfoque de distribución cruzada de medicamentos. Mediante este enfoque basado en la optimización de la distribución pretende reducir el tiempo y número de viajes necesarios sin disminuir la calidad del servicio. Mediante un estudio de caso muestra que el método propuesto permite reducir la duración total de los viajes en el proceso de distribución cruzada de medicamentos dentro del hospital.

Por otra parte, en [15] *Bin Shamsudin et. al.* presentan el diseño de un brazo robótico de 7 DOF, este brazo robótico tiene la particularidad que integra un estetoscopio electrónico para la toma de signos de un paciente con COVID-19, en su trabajo realizan la simulación del prototipo mediante ROS mostrando buenos resultados en lo que respecta al control del brazo robótico.

Otro trabajo relevante en el estudio de robots para la distribución de insumos en hospitales es el presentado por *F. Capezio.* [16], en este artículo se presenta el caso de estudio de la implementación de una plataforma robótica comercial para la entrega de medicamentos y el manejo de desechos. Los resultados de este trabajo muestran que para el diseño de

una plataforma robótica que se utilizará en un entorno civil se deben tener en consideración problemas del mundo real, es decir problemas que pueden surgir en ambientes no controlados (fuera del laboratorio) y no solo considerar el marco teórico y resultados de simulaciones. En el artículo se muestran además las consideraciones y modificaciones que se requirieron para obtener finalmente un prototipo funcional.

Otros artículos relevantes con respecto a la implementación de robots móviles para la distribución de insumos son [17], [18] y [19]. Así también, se puede consultar [1] para obtener una perspectiva más amplia con respecto a la influencia que ha tenido el COVID-19 en el desarrollo de la tecnología en la medicina.

El estudio de los trabajos anteriormente mencionados motiva a pensar en la idea de realizar la integración de uno o varios de ellos en futuros trabajos, algunas posibilidades son la integración de un brazo robótico como el presentado en [15] para extender la capacidad de nuestro robot y realizar chequeos de rutina a pacientes mientras cumple con su función de distribución de insumos, otra posibilidad es la integración de algoritmos como los presentados en [14], entre otros.

1.7. Estructura de la tesis

Este trabajo consta de 5 capítulos. En el primer capítulo se conforma por el protocolo de este trabajo, el cual consiste en la presentación de la motivación, objetivos, hipótesis, planteamiento de la solución, y las contribuciones del mismo. Además, en este capítulo se presentan los trabajos más recientes relevantes para la elaboración de esta tesis.

En el capítulo dos se presentan de forma general los robots móviles y se da una clasificación de los mismos, además, se describen de forma breve las características principales de ellos y su importancia. Posteriormente se presenta la obtención de los modelos dinámico y cinemático del robot que serán utilizados para la obtención de las leyes de control.

Finalmente, en este capítulo se presenta una descripción general del problema de navegación SLAM donde se hace referencia a la bibliografía necesaria para profundizar en el tema.

En el tercer capítulo se encuentra la información correspondiente al diseño, construcción e instrumentación del prototipo; se hace mención de los criterios de selección para cada componente utilizado en el robot y se mencionan las características más relevantes para su selección. Así también, se presenta el prototipo elaborado, destacando las características más importantes en su diseño.

En el cuarto capítulo se presentan los resultados obtenidos de la implementación práctica del prototipo diseñado. Inicialmente se presentan los resultados obtenidos en cuanto a la obtención de la pose del vehículo y la construcción del mapa de ocupación, posteriormente se presentan los resultados de la implementación del control y la pila de navegación.

Por último, en el capítulo cinco se mencionan las conclusiones de este trabajo y también se exponen los posibles trabajos futuros que se pueden desarrollar.

Capítulo 2

Modelado matemático y problema de navegación

En este capítulo inicialmente se presenta una breve descripción y clasificación de los robots, posteriormente se presenta la obtención del modelo cinemático de un robot móvil con configuración diferencial y finalmente se presenta el problema de la navegación SLAM.

2.1. Robots móviles

Una forma común de referirse a la robótica es como la ciencia que estudia la conexión inteligente entre percepción y acción. Con base en esta definición, un sistema robótico es un sistema complejo que puede ser representado funcionalmente por múltiples subsistemas como en la Figura 2.1.

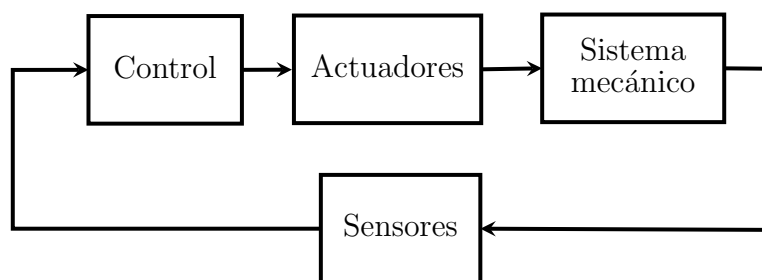


Figura 2.1: Componentes de un sistema robótico.

El componente esencial de un robot es el sistema mecánico, dotado generalmente, con

un aparato de locomoción (ruedas, orugas, patas mecánicas) y un aparato de manipulación (brazos mecánicos, efectores terminales, manos artificiales).

La capacidad de ejercer una acción, tanto de locomoción como de manipulación, es proporcionada por un sistema de actuación que acciona los componentes mecánicos del robot.

La capacidad de percepción se confía a un sistema sensorial que puede adquirir datos sobre el estado interno del sistema mecánico (como los transductores de posición), así como sobre el estado externo del entorno (como los sensores de fuerza y cámaras).

La capacidad de conectar la acción con la percepción de una manera inteligente es proporcionada por un sistema de control que puede controlar la ejecución de la acción con respecto a los objetivos establecidos por una técnica de planificación de tareas, así como de las limitaciones impuestas por el robot y el entorno [20].

2.1.1. Clasificación

De forma general los robots se pueden clasificar de acuerdo a sus atributos de movilidad en las siguientes categorías [20]

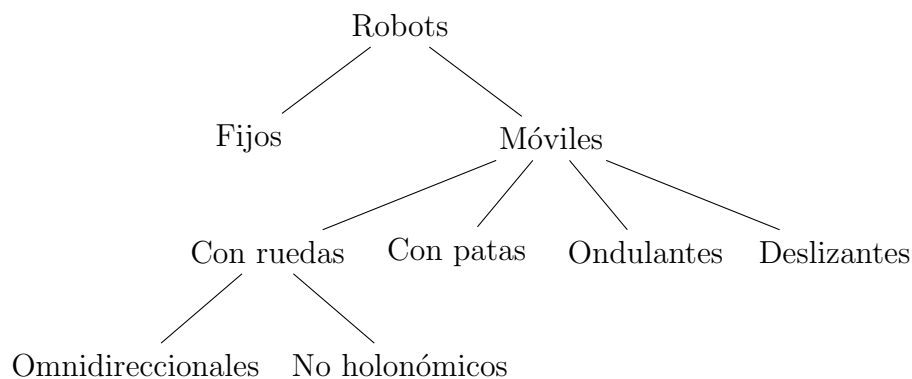


Figura 2.2: Clasificación de los robots de acuerdo a su movilidad.

donde son de particular interés los robots con ruedas que a su vez se pueden clasificar en cuatro grandes categorías acorde a la disposición de sus ruedas en la estructura del

robot como:

- Ackerman (tipo carro): Este tipo de robots cuenta con ruedas orientables y es una de las configuraciones más populares y la más usada en vehículos terrestres.
- Diferencial: Este tipo de robots elimina las ruedas orientables y en su lugar permite controlar la velocidad de las ruedas de cada lado del robot de forma independiente; si las velocidades no son iguales el vehículo girara.
- Omnidireccional: Estos vehículos cuentan con un tipo de rueda especial que permite realizar desplazamientos laterales permitiendo maniobras complejas que no podrían lograrse con las otras configuraciones mencionadas.

2.1.2. Características

En los robots móviles con ruedas se pueden identificar subsistemas básicos que son responsables del movimiento y maniobras del robot. Los siguientes subsistemas son los principales:

1. Subsistema de suministro de energía: Este sistema suministra energía a todos los componentes y subsistemas del vehículo. Consiste en una fuente de energía (baterías, celdas) y redes de distribución de energía (cables, interruptores y circuitos de seguridad).
2. Subsistema de conducción: Este sistema genera una fuerza de empuje para que WMR avance de acuerdo con los comandos especificados por la computadora host. Consta de motores, reductores, amplificadores, tarjeta de control y codificadores.
3. Subsistema de dirección: El subsistema de dirección, si está equipado en un WMR, utiliza actuadores para dirigir las ruedas y así orienta su ángulo de rumbo.
4. Suspensiones: Este subsistema soporta y equilibra el WMR y reduce la vibración durante el funcionamiento. Incluye neumáticos de goma y suspensiones.
5. Comunicación: Los WMR necesitan comunicación inalámbrica para intercambiar datos y comandos con el centro de control y / u otros robots.

6. Sensores: Los sensores se utilizan para medir los estados de WMR, como la posición, la velocidad o el ángulo de dirección, así como la distancia relativa a otros objetos o robots, etc.

Todos estos subsistemas están estrechamente acoplados y deben estar bien coordinados para lograr que el WMR pueda realizar tareas de navegación de forma correcta.

2.2. Modelado matemático

Tener ruedas orientables como la configuración Ackerman implica un diseño mecánicamente complejo; por su parte, una configuración omnidireccional implica además de un diseño complejo una mayor complejidad en el control del mismo, costos mayores y no suele ser adecuada para vehículos de carga o aquellos que se desplazan en superficies lisas como lo son las que se suelen encontrar en los hospitales. Es por lo anterior que este trabajo se centra en una configuración del vehículo de tipo diferencial, el DDMR más simple cuenta únicamente con dos ruedas motrices y una rueda que puede ser tipo castor o de bola, en configuraciones más grandes se puede contar con una rueda adicional para proporcionar estabilidad al vehículo.

2.2.1. Modelo cinemático

Dado que cada uno de los motores acciona una de las ruedas del robot las entradas cinemáticas reales que impulsan al robot y afectan su velocidad y dirección de movimiento son las velocidades de las dos ruedas motrices. Tomando esto en consideración se pueden escribir las ecuaciones cinemáticas del movimiento del DDWR como en [21], para esto se definen dos marcos de referencia, uno $\mathbf{I} = \{x_1, x_2\}$ fijo a la tierra que llamaremos marco inercial y un marco $\mathfrak{B} = \{x_{r1}, x_{r2}\}$ fijo al robot. Ahora, se asume que no existe ningún deslizamiento entre las ruedas y el piso, la consecuencia directa de esta condición implica que no existen componentes de velocidad a lo largo del eje x_{r2} .

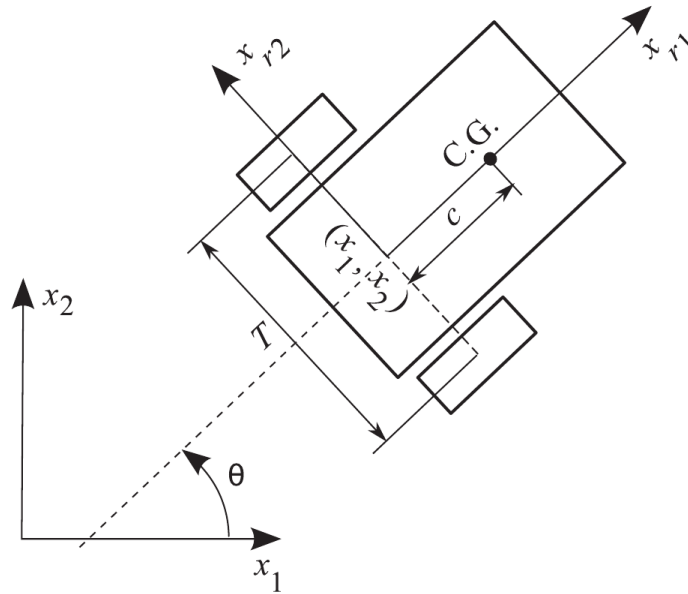


Figura 2.3: Configuración básica del robot diferencial.

Ahora considérese una línea perpendicular al eje de la rueda y pasa por el punto (x_1, x_2) como referencia de orientación para el robot. El ángulo que forma esta línea con el eje x_{r1} positivo, θ , representa la orientación del robot. Las tres variables que definen la configuración geométrica del robot en un momento dado son

$$\mathbf{q} = \begin{bmatrix} x_1 \\ x_2 \\ \theta \end{bmatrix} \quad (2.1)$$

Suponga que el punto (x_1, x_2) del robot se mueve con una velocidad lineal de v , para esto la velocidad de las ruedas debe ser igual, por lo tanto, se puede definir la velocidad lineal del robot como el promedio de las velocidades de las ruedas, la cual es proporcional al radio (r) de las ruedas

$$v = r \frac{v_l + v_r}{2} \quad (2.2)$$

donde v_l y v_r son las velocidades lineales de las ruedas izquierda y derecha del robot respectivamente. Para que el robot tenga un movimiento de rotación sobre su centro de masa, las velocidades de sus ruedas deben tener la misma magnitud pero diferente signo, por lo tanto se puede definir una velocidad angular ω como la diferencia de las velocidades

de sus ruedas sobre la distancia que hay entre ellas, al igual que la velocidad v , la velocidad angular es proporcional al radio de las ruedas del robot, con lo que la velocidad angular queda definida como

$$\omega = r \frac{v_r - v_l}{T} \quad (2.3)$$

Ahora, en consecuencia del supuesto de que no hay deslizamiento en las ruedas, se pueden escribir las componentes de velocidad del punto (x_1, x_2) en el marco inercial como

$$\begin{aligned} \dot{x}_1 &= v \cos \theta \\ \dot{x}_2 &= v \sin \theta \end{aligned} \quad (2.4)$$

Además, la tasa de cambio de la orientación del robot es

$$\dot{\theta} = \omega \quad (2.5)$$

Combinando las ecuaciones (2.4) y (2.5) dan como resultado las ecuaciones cinemáticas de movimiento del robot, que se pueden escribir en forma matricial de la siguiente forma:

$$\dot{\mathbf{q}} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \mathbf{u} \quad (2.6)$$

Donde $\mathbf{u} = [v \ \omega]^T$ es el vector de entrada. Si se mueve el marco de referencia del cuerpo una distancia (c) al C.G. se puede reescribir (2.6) como

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & -c \sin \theta \\ \sin \theta & c \cos \theta \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (2.7)$$

Otro aspecto cinemático presente en el robot, es la presencia de restricciones de tipo no holónomicas. En este caso, la restricción no holónomica la satisface la ecuación siguiente

$$-\dot{x}_1 \sin \theta + \dot{x}_2 \cos \theta = 0 \quad (2.8)$$

Para controlar el desplazamiento del robot se deben controlar las señales de entrada de la planta, por lo tanto se debe buscar una representación matemática en la que se puedan determinar las velocidades ω_l, ω_r en función de las velocidades v, ω deseadas, para esto se obtiene la cinemática inversa del robot.

Para determinar la expresión matricial de la cinemática inversa, primero se reescribe 2.7 en función de las velocidades angulares de las ruedas como

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{r}{2} \cos \theta & \frac{r}{2} \cos \theta \\ \frac{r}{2} \sin \theta & \frac{r}{2} \sin \theta \\ r/T & -r/T \end{bmatrix} \begin{bmatrix} \omega_r \\ \omega_l \end{bmatrix} \quad (2.9)$$

donde ω_l y ω_r son las velocidades angulares de las ruedas izquierda y derecha respectivamente. Posteriormente se deben despejar las variables de interés como

$$\begin{bmatrix} \omega_r \\ \omega_l \end{bmatrix} = \begin{bmatrix} \frac{r}{2} \cos \theta & \frac{r}{2} \cos \theta \\ \frac{r}{2} \sin \theta & \frac{r}{2} \sin \theta \\ r/T & -r/T \end{bmatrix}^{-1} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{\theta} \end{bmatrix} \quad (2.10)$$

debido a que la matriz no es cuadrada, no se puede obtener su inversa. Para resolver este problema se procede a determinar la pseudoinversa de dicha matriz, la pseudoinversa se puede obtener mediante la fórmula de Moore-Penrose

$$A^+ = (A^T A)^{-1} A^T \quad (2.11)$$

donde A^T y A^+ son la transpuesta y la pseudoinversa de A respectivamente, definiendo A como

$$A = \begin{bmatrix} \frac{r}{2} \cos \theta & \frac{r}{2} \cos \theta \\ \frac{r}{2} \sin \theta & \frac{r}{2} \sin \theta \\ r/T & -r/T \end{bmatrix} \quad (2.12)$$

resolviendo la ecuación se obtiene la pseudoinversa

$$A^+ = \begin{bmatrix} \frac{\cos \theta}{r} & \frac{\sin \theta}{r} & \frac{T}{r} \\ \frac{\cos \theta}{r} & \frac{\sin \theta}{r} & -\frac{T}{r} \end{bmatrix} \quad (2.13)$$

con lo que la cinemática inversa del robot queda dada como

$$\begin{bmatrix} \omega_r \\ \omega_l \end{bmatrix} = \begin{bmatrix} \frac{\cos \theta}{r} & \frac{\sin \theta}{r} & \frac{T}{r} \\ \frac{\cos \theta}{r} & \frac{\sin \theta}{r} & -\frac{T}{r} \end{bmatrix} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{\theta} \end{bmatrix} \quad (2.14)$$

2.2.2. Modelo dinámico

Para obtener el modelo dinámico utilizamos el procedimiento descrito en [22], donde utilizan la ecuación de Rosenberg [23]. Esta ecuación permite obtener el modelo dinámico de robots móviles con restricciones no holónomicas.

$$\frac{d}{dt} \left(\frac{\partial K}{\partial \dot{q}_i} \right) - \frac{\partial K}{\partial q_i} = Q_i - a_{1i} \lambda_1, \quad i = 1, \dots, 3 \quad (2.15)$$

donde K es la energía cinética, Q_i son los pares de entrada del sistema, a_{1i} son los elementos de la matriz de restricciones no holónomicas y λ_1 es un multiplicador de Lagrange. La energía cinética del robot móvil se obtiene de la siguiente ecuación

$$K = \frac{1}{2} m v_{cm}^T v_{cm} + \frac{1}{2} \dot{\theta}^2 I \quad (2.16)$$

donde

$$v_{cm} = \begin{bmatrix} \dot{x}_0 - c \sin(\theta - \gamma) \dot{\theta} \\ \dot{y}_0 - c \cos(\theta - \gamma) \dot{\theta} \\ 0 \end{bmatrix} \quad (2.17)$$

En este caso, v_{cm} es la velocidad del centro de masa la cual queda expresada en términos de $q = [x_0 \ y_0 \ \theta]^T$, por lo que al sustituir (2.17) en (2.16) se obtiene el modelo dinámico

$$M(q)\ddot{q} + V_m(q, \dot{q})\dot{q} = B(q)\tau - A^T(q)\lambda \quad (2.18)$$

con:

$$M = \begin{bmatrix} m & 0 & -mc \sin(\theta + \gamma) \\ 0 & m & mc \cos(\theta + \gamma) \\ -mc \sin(\theta + \gamma) & mc \cos(\theta + \gamma) & mc^2 + I \end{bmatrix} \quad (2.19)$$

$$V_m = \begin{bmatrix} 0 & 0 & -mc \cos(\theta + \gamma) \dot{\theta} \\ 0 & 0 & -mc \sin(\theta + \gamma) \dot{\theta} \\ 0 & 0 & 0 \end{bmatrix} \quad (2.20)$$

$$B = \frac{1}{r} \begin{bmatrix} \cos \theta & \cos \theta \\ \sin \theta & \sin \theta \\ T - c \sin(\gamma) & -T - c \sin \gamma \end{bmatrix} \quad (2.21)$$

$$A = \frac{1}{r} [-\sin \theta \cos \theta - c \cos \gamma] \quad (2.22)$$

$$\tau = \begin{bmatrix} \tau_l \\ \tau_r \end{bmatrix} \quad (2.23)$$

donde τ_l y τ_r son los pares de entrada de las ruedas, m es la masa del robot e I es el momento de inercia en $P_c m$ alrededor del eje Z_m .

2.3. Navegación

En esta subsección se presentan los conceptos básicos relacionados con la navegación y en particular con la navegación SLAM. Para profundizar más en estos temas se sugiere consultar [24, 25, 26, 27, 28], etc.

El problema de la navegación SLAM surge cuando el robot no tiene acceso a un mapa del medio ambiente o de su propia pose. La localización y el mapeo simultáneos es el proceso de construcción del mapa de un entorno y al mismo tiempo la generación de una estimación de la ubicación del robot.

Considere un robot móvil que se mueve a través de un entorno tomando observaciones relativas de una cantidad de puntos de referencia desconocidos usando un sensor ubicado en el robot como se muestra en la Figura 2.4. En un instante de tiempo k , se definen las siguientes cantidades:

- \mathbf{x}_k : El vector de estado que describe la ubicación y orientación del vehículo.
- \mathbf{u}_k : El vector de control, aplicado en tiempo $k - 1$ para conducir el vehículo a un estado x_k en el tiempo k .
- \mathbf{m}_i : Un vector que describe la ubicación del i -ésimo hito cuya verdadera ubicación se supone invariante en el tiempo.
- \mathbf{z}_{ik} : Una observación tomada desde el vehículo de la ubicación del i -ésimo hito en el momento k . Cuando hay múltiples observaciones de hitos en cualquier momento o cuando el hito específico no es relevante para la discusión, la observación se escribirá simplemente como \mathbf{z}_k .

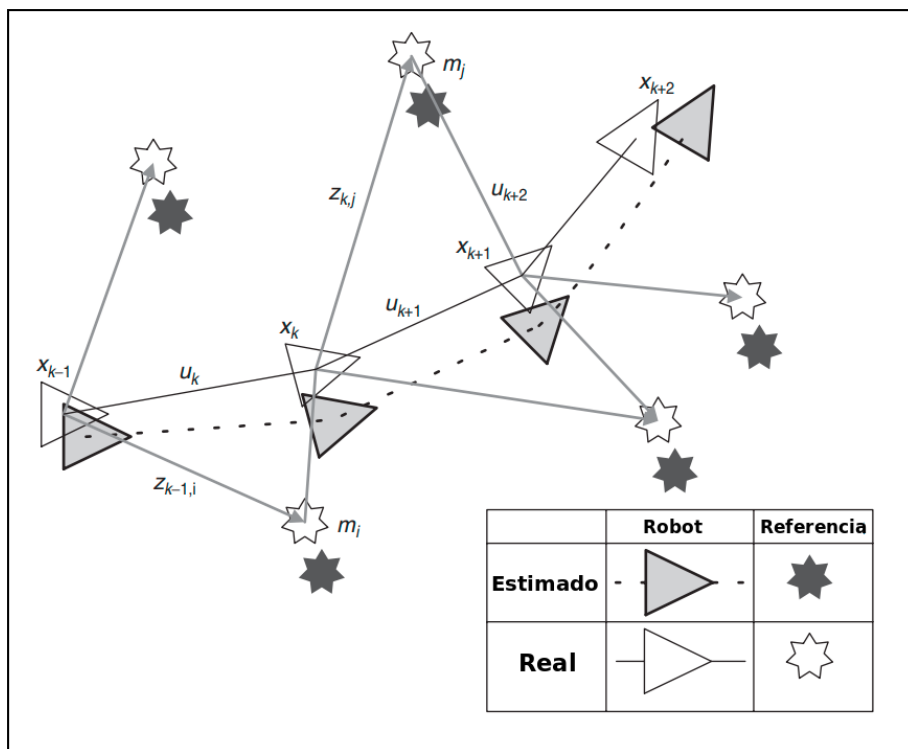


Figura 2.4: Problema SLAM.

Además, también se definen los siguientes conjuntos:

- $\mathbf{X}_{0:k} = \{\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_k\} = \{\mathbf{X}_{0:k-1}, \mathbf{x}_k\}$: Es la historia de ubicaciones del vehículo.
- $\mathbf{U}_{0:k} = \{\mathbf{u}_0, \mathbf{u}_1, \dots, \mathbf{u}_k\} = \{\mathbf{U}_{0:k-1}, \mathbf{u}_k\}$: Es la historia de las entradas de control.
- $\mathbf{m} = \{\mathbf{m}_1, \mathbf{m}_2, \dots, \mathbf{m}_n\}$: Es el conjunto de todos los puntos de referencia
- $\mathbf{Z}_{0:k} = \{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_k\} = \{\mathbf{Z}_{0:k-1}, \mathbf{z}_k\}$: Es el conjunto de todas las observaciones de los puntos referencia.

En forma probabilística, el problema del SLAM requiere que la distribución de probabilidad

$$P(\mathbf{x}_k, \mathbf{m} | \mathbf{Z}_{0:k}, \mathbf{U}_{0:k}, \mathbf{x}_0) \quad (2.24)$$

sea calculada para todos los tiempos k . Esta distribución de probabilidad describe la densidad posterior conjunta de las ubicaciones de los puntos de referencia y el estado del vehículo (en el momento k) dadas las observaciones registradas y las entradas de control hasta el momento k inclusive, junto con el estado inicial del vehículo. En general, una solución recursiva al problema de SLAM es deseable. Comenzando con una estimación para la distribución $P(\mathbf{x}_{k-1}, \mathbf{m} | \mathbf{Z}_{0:k-1}, \mathbf{U}_{0:k-1})$ en el tiempo $k-1$, la articulación posterior, siguiendo un control \mathbf{u}_k y una observación \mathbf{z}_k , se calcula usando el teorema de Bayes. Este cálculo requiere que se definan un modelo de transición de estado y un modelo de observación que describan el efecto de la entrada de control y la observación, respectivamente.

El modelo de observación describe la probabilidad de realizar una observación \mathbf{z}_k cuando se conocen la ubicación del vehículo y las ubicaciones de los puntos de referencia y generalmente se describe de la forma

$$P(\mathbf{z}_k | \mathbf{x}_k, \mathbf{m}) \quad (2.25)$$

Es razonable suponer que una vez que se definen la ubicación y el mapa del vehículo, las observaciones son condicionalmente independientes dado el mapa y el estado actual del

vehículo.

El modelo de movimiento para el vehículo se puede describir en términos de una distribución de probabilidad en las transiciones de estado en la forma

$$P(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{u}_k) \quad (2.26)$$

Es decir, se supone que la transición de estado es un proceso de Markov en el que el siguiente estado \mathbf{x}_k depende solo del estado inmediatamente anterior \mathbf{x}_{k-1} y del control aplicado \mathbf{u}_k y es independiente tanto de las observaciones como del mapa.

El algoritmo SLAM ahora se implementa en una forma estándar recursiva (secuencial) de dos pasos: corrección (actualización de medición) y predicción (actualización de tiempo) de la forma:

Actualización del tiempo

$$P(\mathbf{x}_k, \mathbf{m} | \mathbf{Z}_{0:k-1}, \mathbf{U}_{0:k}, \mathbf{x}_0) = \int P(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{u}_k) \times P(\mathbf{x}_{k-1}, \mathbf{m} | \mathbf{Z}_{0:k-1}, \mathbf{U}_{0:k-1}, \mathbf{x}_0) d\mathbf{x}_{k-1} \quad (2.27)$$

Actualización de la medición

$$P(\mathbf{x}_k, \mathbf{m} | \mathbf{Z}_{0:k}, \mathbf{U}_{0:k}, \mathbf{x}_0) = \frac{P(\mathbf{z}_k | \mathbf{x}_k, \mathbf{m}) P(\mathbf{x}_k, \mathbf{m} | \mathbf{Z}_{0:k-1}, \mathbf{U}_{0:k}, \mathbf{x}_0)}{P(\mathbf{z}_k | \mathbf{Z}_{0:k-1}, \mathbf{U}_{0:k})} \quad (2.28)$$

Las ecuaciones (2.27) y (2.28) proporcionan un procedimiento recursivo para calcular la articulación posterior $P(\mathbf{x}_k, \mathbf{m} | \mathbf{Z}_{0:k}, \mathbf{U}_{0:k}, \mathbf{x}_0)$ para el estado del robot x_k y el mapa \mathbf{m} en un momento k basado en todas las observaciones $\mathbf{Z}_{0:k}$ todas las entradas de control $\mathbf{U}_{0:k}$ hasta el tiempo k incluido. La recursividad es una función del modelo del vehículo $P(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{u}_k)$ y un modelo de la observación $P(\mathbf{z}_k | \mathbf{x}_k, \mathbf{m})$.

Vale la pena señalar que el problema de construcción de mapas se puede formular calculando la densidad condicional $P(\mathbf{m} | \mathbf{X}_{0:k}, \mathbf{Z}_{0:k}, \mathbf{U}_{0:k})$. Esto supone que la ubicación del vehículo \mathbf{x}_k es conocida (o al menos determinista) en todo momento, sujeto al co-

nocimiento de la ubicación inicial. A continuación, se construye un mapa \mathbf{m} fusionando observaciones de diferentes ubicaciones. A la inversa, el problema de localización se puede formular calculando la distribución de probabilidad $P(\mathbf{x}_k | \mathbf{Z}_{0:k}, \mathbf{U}_{0:k}, \mathbf{m})$. Esto supone que las ubicaciones de los puntos de referencia se conocen con certeza y el objetivo es calcular una estimación de la ubicación del vehículo con respecto a estos puntos de referencia.

Solución al problema del SLAM

Los problemas de la navegación y el mapeo autónomos requieren algoritmos SLAM en tiempo real rápidos, robustos y precisos. Actualmente el desarrollo de la navegación SLAM ha alcanzado un considerable estado de madurez, los algoritmos basados en la tecnología LIDAR son una de las formas más efectivas y robustas.

Las soluciones al problema del SLAM probabilístico implican encontrar una representación apropiada tanto para el modelo de observación (2.25) como para el modelo de movimiento (2.26) que permita el cálculo eficiente y consistente de las distribuciones anterior y posterior en (2.28) y (2.27). De lejos, la representación más común tiene la forma de un modelo de espacio de estado con ruido gaussiano aditivo, lo que lleva al uso del filtro de Kalman extendido (EKF) para resolver el problema SLAM. Una representación alternativa importante es describir el modelo de movimiento del vehículo en (2.26) como un conjunto de muestras de una distribución de probabilidad no gaussiana más general. Esto conduce al uso de el filtro de partículas Rao-Blackwellized, o el algoritmo FastSLAM, para resolver el problema de SLAM. Mientras que EKF-SLAM y FastSLAM son los dos métodos de solución más importantes, se han propuesto alternativas más nuevas, que ofrecen mucho potencial.

Capítulo 3

Diseño y selección de la instrumentación del prototipo

Derivado de las limitaciones causadas por la crisis actual de la pandemia del COVID-19 se diseñó un prototipo con el propósito de realizar pruebas para validar los resultados teóricos sin la necesidad de estar en el laboratorio. La instrumentación fue seleccionada considerando que si las condiciones son propicias ésta podrá ser trasladada a un prototipo del laboratorio de la unidad académica.

3.1. Instrumentación

3.1.1. Criterios de selección

Los criterios considerados para la construcción del prototipo son los siguientes:

- Selección de los sensores para las tareas de mapeo y navegación considerando que el vehículo se desplazará en interiores.
- Seleccionar una unidad de procesamiento con la capacidad de ejecutar ROS y con las interfaces necesarias para la conexión de los sensores seleccionados.
- Considerar un peso aproximado del vehículo de $3.5Kg$.

- Selección de los motores con la capacidad de mover el peso del vehículo.
- Selección del sistema de alimentación contemplando los requerimientos energéticos de los motores y la instrumentación electrónica del vehículo para obtener una autonomía de 30 minutos.

3.1.2. Sistema de navegación

Para afrontar el problema de localización en la navegación SLAM típicamente se utiliza un sistema basado en la odometría de rueda o codificadores ópticos y los sistemas inerciales. El principal problema de estos métodos es la acumulación del error, es por esto que desde hace tiempo se han estudiado y desarrollado métodos basados en la fusión de odometría inercial con odometría visual y odometría de rueda.

Es por lo anterior que se eligió utilizar la *cámara de seguimiento Intel RealSense T265* la cual se muestra en la Figura (3.1). Esta cámara incluye dos sensores de lente ojo de pez, una IMU y una VPU. Así también, esta cámara es capaz de ejecutar algoritmos V-SLAM directamente en su VPU, lo que permite una latencia muy baja, es capaz de estimar la pose del vehículo hasta una frecuencia de 200Hz, las características de la unidad de la IMU integrada en la cámara se muestran en la tabla (3.1), las especificaciones técnicas de la cámara pueden consultarse con más detalle en el apéndice B o en [29].

Parámetro	Valor
Grados de libertad	6
Rango del acelerómetro	$\pm 4g$
Frecuencia del acelerómetro	62.5Hz
Rango del giroscopio	± 2000 Deg/s
Frecuencia del giroscopio	200Hz

Tabla 3.1: Características de la unidad de medición inercial de la cámara T265.



Figura 3.1: Cámara de seguimiento Intel RealSense T265.

Por otra parte, para afrontar el problema de mapeo se eligió el sensor LIDAR *URG-04LX-UG01* mostrado en la Figura (3.2) cuyas características se muestran en la tabla (3.2), las especificaciones se pueden consultar con más detalle en el apéndice B. Se seleccionó este sensor considerando que el vehículo se desplazará en interiores, el rango de distancia de medición del sensor es de $5.6m$ lo cual es adecuado ya que típicamente las habitaciones no sobrepasan los $3m$ de longitud, el rango del sensor permite además detectar objetos en corredores con un margen adecuado para que el algoritmo de navegación pueda reaccionar con tiempo suficiente ante objetos dinámicos.



Figura 3.2: Sensor LIDAR urg-04lx-ug01.

Precisión	60 a 1,000mm : +-30mm, 1,000 a 4,095mm: +-3 %
Interfaz	USB 2.0
Luminosidad	6.000 - 10.000 Lux
Peso	160 g
Voltaje	5V@0.5A
Resolución angular	0.352 ^o
Rango	240 ^o
Ambiente	Interiores
Distancia	5,6 m
Frecuencia de escaneo	10 Hz

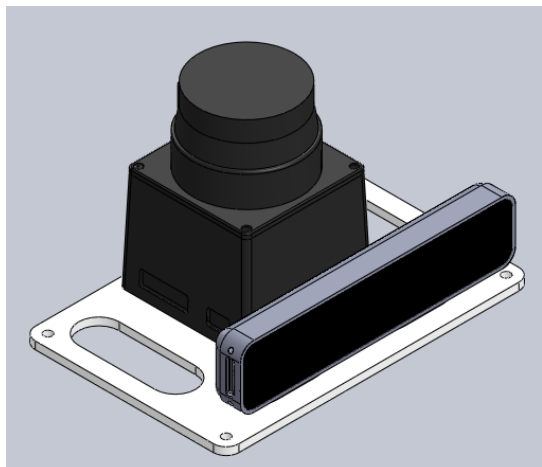
Tabla 3.2: Características del sensor URG-04LX-UG01.

Considerando que el robot deberá ser capaz de navegar de forma autónoma se determinó utilizar una computadora de placa única (SBC). La computadora seleccionada es la *NVIDIA Jetson Nano* mostrada en la imagen (3.3), se eligió utilizar esta computadora ya que a diferencia de las demás existentes en el mercado, ésta cuenta con una GPU capaz de ejecutar algoritmos de visión artificial con un mejor desempeño que las alternativas sin una GPU dedicada, además esta SBC cuenta con puertos USB 3.0 compatibles con la cámara T265 seleccionada y el sensor LIDAR. La tarjeta se puede alimentar por una fuente de voltaje de 5v y la potencia nominal es de 10W. Las características de esta computadora se muestran con mayor detalle en el apéndice B o en [30].

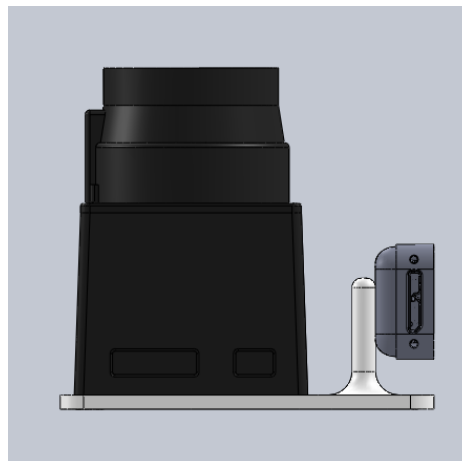


Figura 3.3: Computadora de placa única NVIDIA Jetson Nano.

Con el propósito de montar la cámara y el sensor LIDAR de forma que no sea necesario calibrar o ajustar parámetros dependiendo de la plataforma, se diseñó la base mostrada en la Figura (3.4) y ésta se imprimió en 3D. Además, tomando en cuenta que las mediciones de la unidad inercial integrada en la cámara son muy susceptibles a las vibraciones mecánicas, también se diseñó un amortiguador de vibraciones similar a los que se implementan en los estabilizadores de las cámaras en las UAV, estos amortiguadores se imprimieron en 3D utilizando poliuretano termoplástico con dureza 95A cuyas propiedades mecánicas se muestran en la tabla (3.3). El montaje de los amortiguadores junto con las bases se muestra en la Figura (3.4).



(a) Vista isométrica.



(b) Vista Lateral.

Figura 3.4: Vistas básicas del montaje de los sensores en la base diseñada.

Propiedades mecánicas	Valor típico	Método de ensayo
Módulo de elasticidad a la tracción	26,0 MPa	ASTM D638
Esfuerzo de tracción a la deformación	8,6 MPa	ASTM D638
Dureza	95 (Shore A)	ASTM D2240
	46 (Shore D)	Durómetro
Resistencia a la flexión	4,3 MPa	ISO 179

Tabla 3.3: Propiedades mecánicas del Poliuretano termoplástico 95A

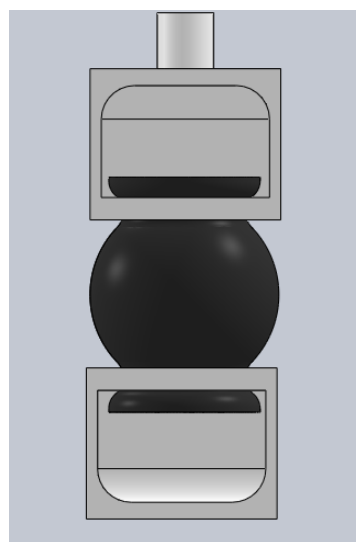
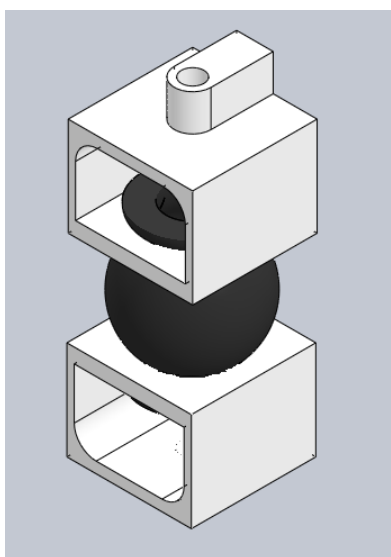


Figura 3.5: Vistas básicas del ensamble del amortiguador de vibración.

3.1.3. Etapa de potencia.

Inicialmente se seleccionó el motor *Pololu 37D* con una reducción metálica. Considerando el peso del vehículo y un diámetro de las ruedas de 12cm se determinó que el torque máximo del motor necesario es de $42\text{kg} \cdot \text{cm}$, por lo tanto la reducción seleccionada fue de 131:1 con lo cual se obtiene un torque máximo de $45\text{kg} \cdot \text{cm}$. Adicionalmente, este motor cuenta con dos variantes, con o sin codificadores ópticos, se optó por la versión con ellos para poder realizar las pruebas de estimación de la pose del vehículo con y sin fusionar los datos de la cámara de seguimiento con la pose calculada a partir de la odometría de los motores. En la tabla (3.4) se observan las características del motor seleccionado y en la Figura (3.6) se muestra el mismo.



Figura 3.6: Moto reductor Pololu 37D con encoder.

Voltaje nominal	12v
Corriente máxima	5A
Corriente sin carga	0.2A
Reducción	131:1
Velocidad sin carga	76 RPM
Torque máximo	$45 \text{ kg} \cdot \text{cm}$

Tabla 3.4: Características del moto reductor Pololu 37D.

Tomando en consideración las características del motor seleccionado, se eligió el módulo puente-H *TB6612FNG* mostrado en la Figura (3.7). El módulo elegido nos permite controlar dos motores en ambas direcciones al mismo tiempo, y soporta un voltaje máximo de alimentación para los motores de $15v$, la corriente pico por canal es de $3.2A$, aunque la corriente máxima del puente H es menor que la corriente máxima del motor, de acuerdo con las gráficas mostradas en la hoja de datos del motor, éste alcanza una eficiencia máxima a $2.5A$ y la corriente necesaria para proporcionar el torque máximo necesario para el robot es de $1.5A$.

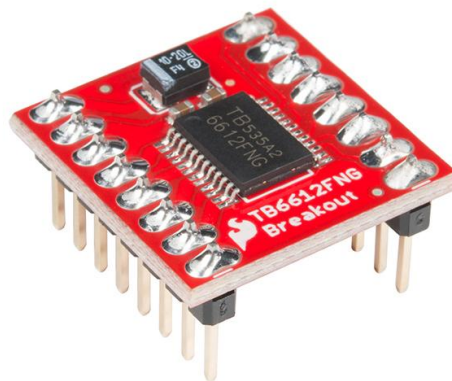
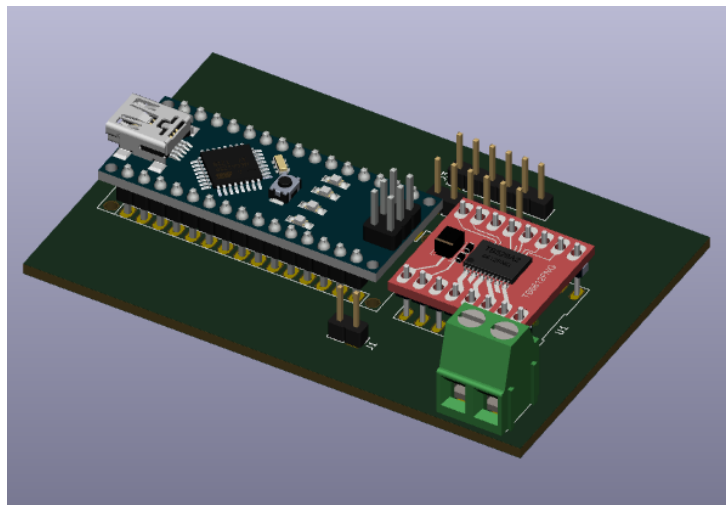
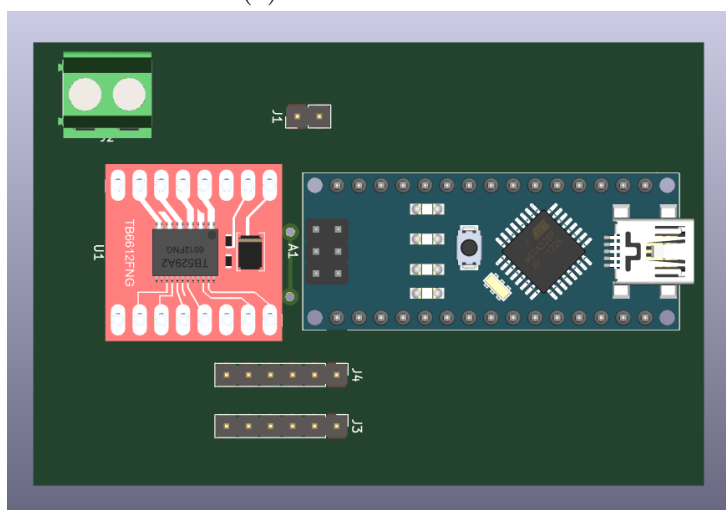


Figura 3.7: Módulo puente H TB6612FNG.

Para poder obtener la lectura de los codificadores ópticos de los motores y poder enviar las señales de control al puente H se eligió utilizar una tarjeta Arduino Nano. Se seleccionó este microcontrolador ya que es compatible con la librería ROS Serial, la cual permite la comunicación con la Jetson Nano como con cualquier nodo de ROS. Para realizar las conexiones entre la Arduino, el puente H, los motores y la fuente de alimentación se diseñó una placa de circuito impreso, el diagrama esquemático de la misma se puede consultar en el apéndice B, en la Figura (3.8) se muestran la vista isométrica y superior del mojado de la placa diseñada.



(a) Vista isométrica.



(b) Vista superior.

Figura 3.8: Vistas básicas del diseño de la PCB para el control de los motores.

3.1.4. Alimentación eléctrica

Derivado de los componentes seleccionados se requiere de dos fuentes de alimentación para la operación del robot, 12V y 5V. Para satisfacer los requerimientos de alimentación se eligió utilizar baterías tipo Li-PO de tres celdas en conjunto con un módulo regulador de voltaje, las principales ventajas de este tipo de baterías es que entregan un mayor voltaje por celda, tienen una capacidad de descarga elevada y una densidad de carga elevada. Para seleccionar la capacidad de la batería se determinó el consumo total de energía del vehículo, en la tabla (3.5) se muestran los consumos de energía de los componentes principales, de

esta tabla se puede determinar que la potencia nominal del robot es de $50.095W$ con lo que el consumo energético total es de $4.175A$, para obtener una autonomía media de al menos media hora se requiere de una batería de $2.0875Ah$, por lo tanto se eligió utilizar una batería de $2200mAh$ que nos dará una autonomía de 32 minutos.

	Voltaje	Corriente	Potencia
Jetson Nano	5v	2A	10W
Arduino Nano	5v	0.019A	0.095W
Motores	12	3A	36W
URG-04LX-UG01	5v	0.5A	2.5W
Cámara T265	5v	0.3A	1.5W

Tabla 3.5: Consumo energético de los componentes principales del robot.

De la tabla (3.5) también se sabe que el consumo nominal de corriente de los dispositivos alimentados con 5v es de $2.819A$, sin embargo de la hoja de datos de la Jetson Nano se sabe que ésta puede tener un consumo de hasta $4A$, por lo tanto el consumo máximo es de $4.819A$, tomando esto en consideración se eligió el módulo regulador de voltaje *XL4016* para suministrar los 5v a estos dispositivos, la corriente de salida de este módulo es de $8A$ máximo lo cual satisface los requerimientos. En la tabla (3.6) se muestran las especificaciones del regulador seleccionado y en la Figura (3.9) se muestra una imagen del mismo.

Voltaje de entrada	8v-40v
Voltaje de salida	1.25v-36v
Corriente de salida	8A
Eficiencia de la conversión	94 %

Tabla 3.6: Especificaciones del regulador XL4016.



Figura 3.9: Regulador X14016.

3.2. Diseño de la estructura

Las vistas básicas del modelo del prototipo diseñado para este trabajo se muestra en las Figuras de (3.10) a (3.12). Para diseñar esta estructura se consideró un diseño simétrico en el cual se pueden observar dos componentes principales, el ensamble de la transmisión y las tapas del mismo que unen la estructura y permiten el montaje de la instrumentación.

El ensamble de la transmisión consiste de dos placas de aluminio aleación 1060 donde se montan los motores y los baleros del eje de la llanta, ambas placas son unidas mediante espaciadores de bronce, se eligió el aluminio debido a la facilidad de maquinado, disponibilidad y el bajo peso del mismo.

Para unir las transmisiones de cada rueda se diseñaron una tapa superior y una inferior, en estas placas además de colocar barrenos para atornillar la transmisión se le colocaron barrenos para realizar el montaje de la instrumentación y las ruedas locas. El material considerado para las tapas es acrílico de 5mm, se seleccionó este material por su disponibilidad y debido a que este material se puede trabajar con corte láser permitiendo que las piezas se manufacturen rápidamente.

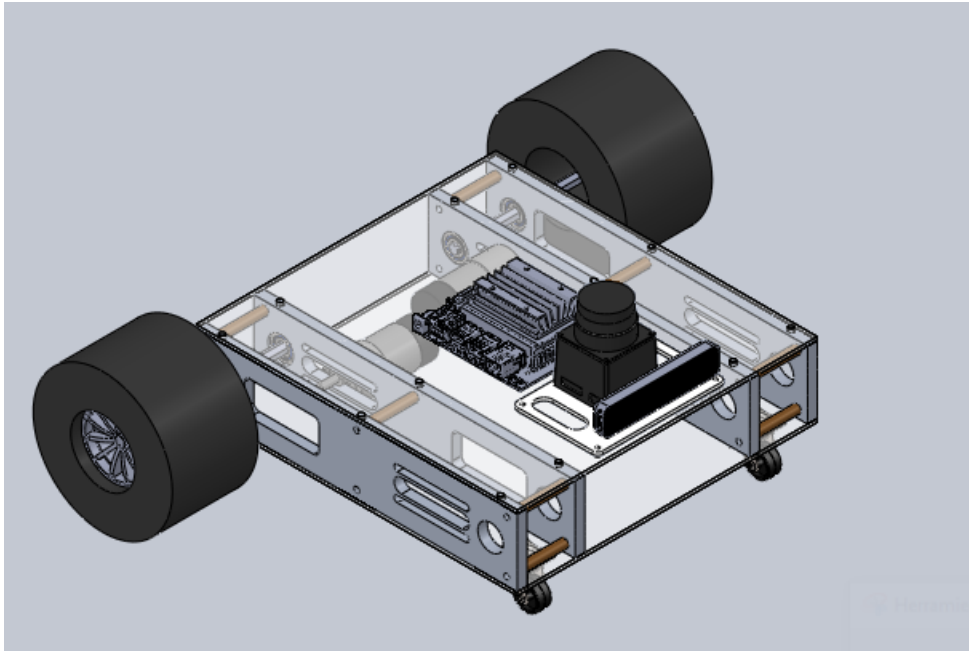


Figura 3.10: Vista isométrica del diseño del prototipo.

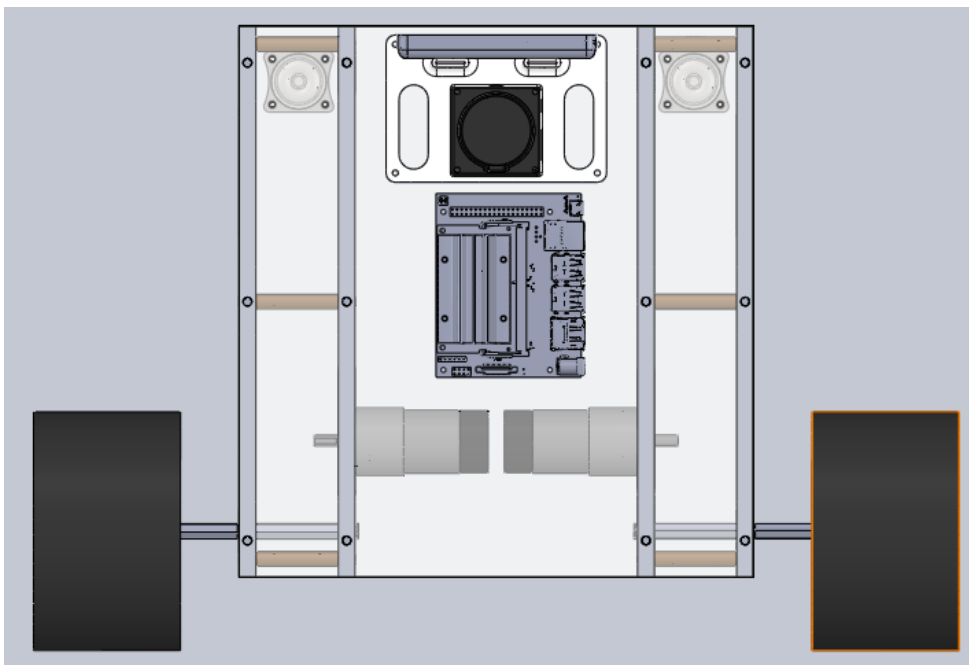


Figura 3.11: Vista superior del diseño del prototipo.

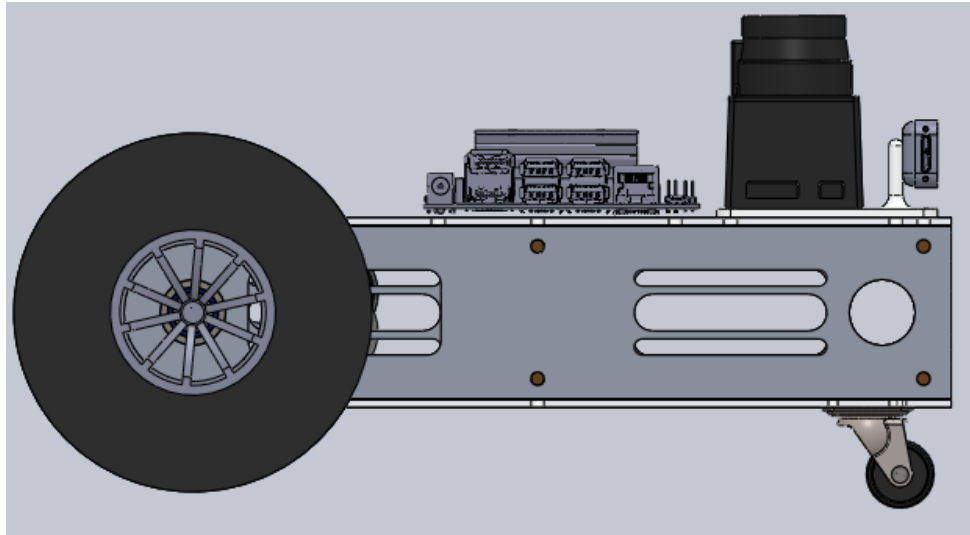


Figura 3.12: Vista lateral del diseño del prototipo.

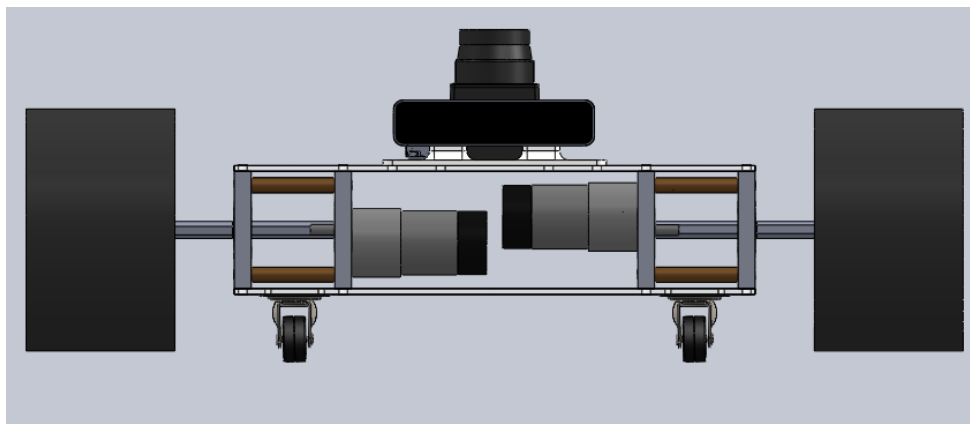


Figura 3.13: Vista frontal del diseño del prototipo.

Mediante los resultados de la simulación de movimiento en SolidWorks se determinaron el centro de masa, peso y los momentos de inercia principales del prototipo, estos resultados mostraron que el diseño se encontraba dentro de los requerimientos por lo que se procedió a la construcción del prototipo, en las Figuras (3.14, 3.15) y (3.16) se muestran la vista superior, frontal y lateral de prototipo ensamblado e instrumentado respectivamente.

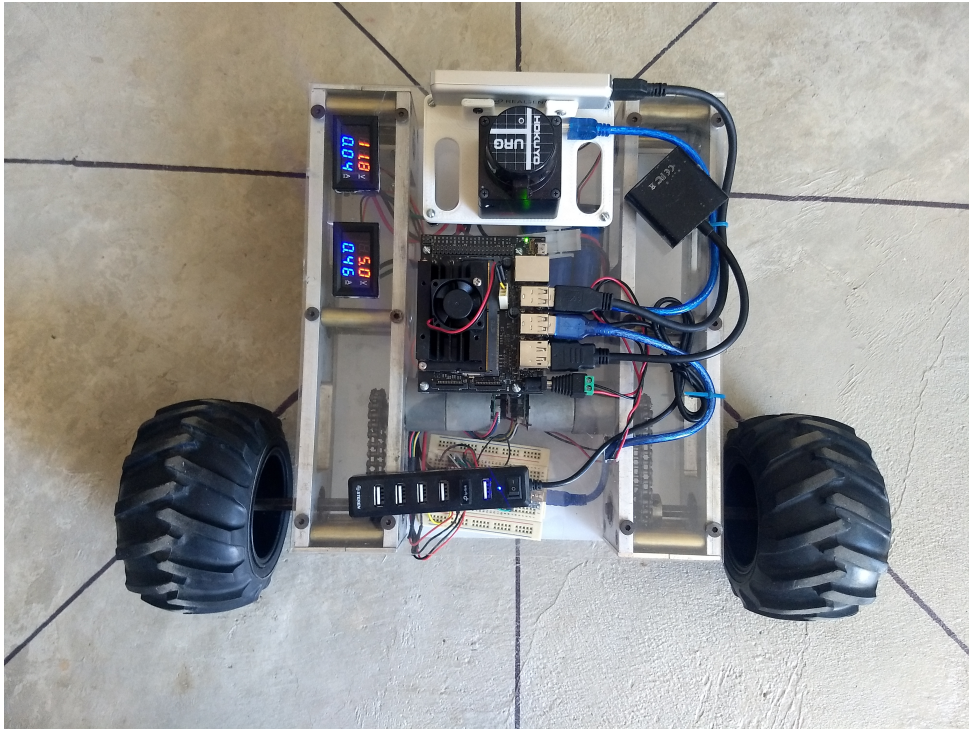


Figura 3.14: Ensamble experimental del prototipo, vista superior.

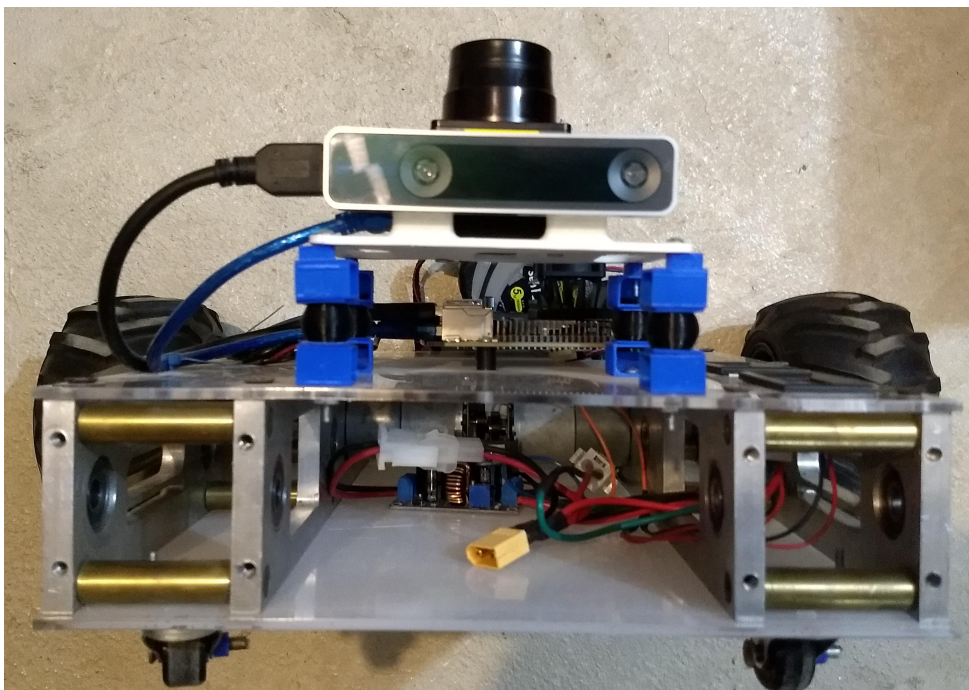


Figura 3.15: Ensamble experimental del prototipo, vista frontal.



Figura 3.16: Ensamblaje experimental del prototipo, vista lateral.

Posteriormente a realizar las primeras pruebas de control y navegación se observó que al estar colocada la cámara en la parte frontal del robot se transmitía demasiada vibración de las ruedas castor por lo que se decidió re-posicionar la cámara en la parte posterior del robot, la cámara se orientó hacia la parte posterior del mismo y se alineó con el eje de rotación del robot con lo que también se eliminó la necesidad de realizar las transformaciones de coordenadas para obtener la velocidad lineal del robot mediante el modelo cinemático inverso lo cual contribuye a reducir el esfuerzo computacional y los errores de cálculos. En la Figura (3.17) y (3.18) se muestra el prototipo en donde ya se reorientó la cámara quedando colocada en la parte posterior y también orientada en el mismo sentido.



Figura 3.17: Ensamble experimental del prototipo con la cámara en la parte posterior, vista frontal.

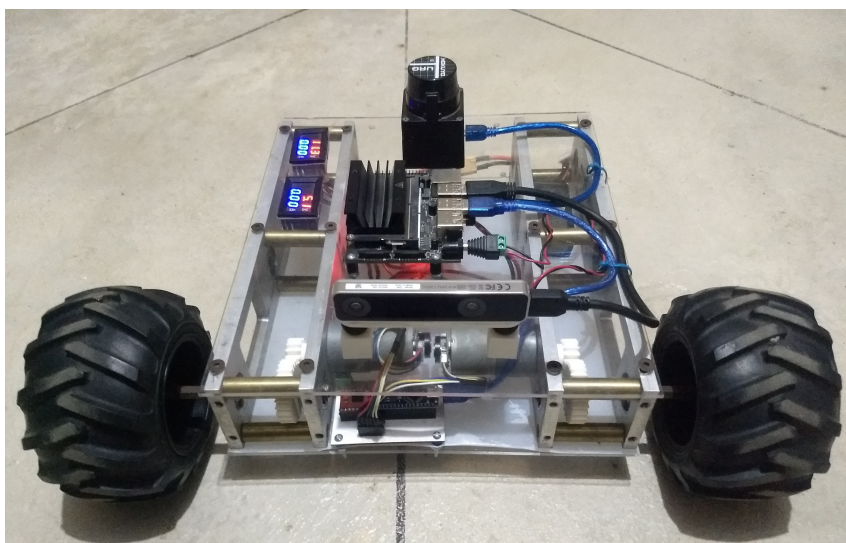


Figura 3.18: Ensamble experimental del prototipo con la cámara en la parte posterior, vista posterior.

Además, las pruebas mostraron que la cámara de seguimiento proporciona una estimación adecuada de la odometría mientras las velocidades lineales se mantengan dentro de un rango de 0.1 a 0.7 m/s y las velocidades angulares se mantengan en el rango de 0.2 a 1.0 rad/s. De acuerdo a las pruebas realizadas en el prototipo se determinó que para

mantenerse dentro de ese rango en necesario limitar el intervalo de control de los motores en un 29% y un 23% para la velocidad lineal y angular respectivamente lo que reduciría significativamente el desempeño del control por lo que se decidió implementar una transmisión reductora a los motores. Para implementar la transmisión se diseñaron dos engranes en Solidwoks, se seleccionó un número de dientes de 24 y 14 respectivamente con lo que resulta una relación de transmisión de 1:1.785, dando como resultado una trasmisión final de 1:234.375 con lo cual se mejora la controlabilidad del sistema. En la Figura (3.19) se muestran los engranes diseñados y en la Figura 3.20 se muestra la implementación de los mismos en el robot.

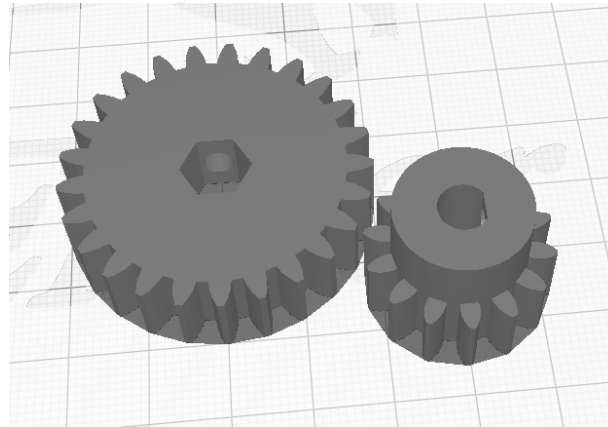


Figura 3.19: Diseño de los engranes de la transmisión reductora.



Figura 3.20: Implementación de la transmisión reductora en el robot.

De acuerdo con el análisis realizado al ensamble del prototipo en SolidWorks, los momentos de inercia principales del robot y el centro de masa están dados como en la tabla (3.7) y la masa del robot es de 3426.1g.

Eje	Centro de masa (m)	Momentos de inercia ($g \cdot m^2$)
X	-0.3130	37.6798
Y	0	83.2074
Z	0.7536	13.8107

Tabla 3.7: Propiedades mecánicas del prototipo diseñado.

Capítulo 4

Implementación y resultados

En este capítulo se presentan los resultados obtenidos de la implementación práctica del prototipo diseñado. Inicialmente se presentan los resultados obtenidos en cuanto a la obtención de la pose del vehículo y la construcción del mapa de ocupación, posteriormente se presentarán los resultados de la implementación de control y la pila de navegación.

4.1. Resultados experimentales

4.1.1. Estimación de la pose del vehículo

Las implementaciones experimentales para la obtención de la pose del vehículo se realizaron en una superficie de concreto de $2 \times 2m$ sin obstáculos, además se marcó la superficie para tener un marco de referencia. El área utilizada junto con el prototipo se muestran en la Figura (4.1)

Con el objetivo de evaluar la estimación de la pose del vehículo se realizaron diversas pruebas, la primera consistió en conducir el vehículo describiendo una trayectoria circular constante para evaluar la estimación de la orientación, inicialmente sin amortiguadores de vibración, posteriormente con los amortiguadores diseñados y finalmente se condujo el robot en una línea recta para evaluar la estimación de la posición final del vehículo.

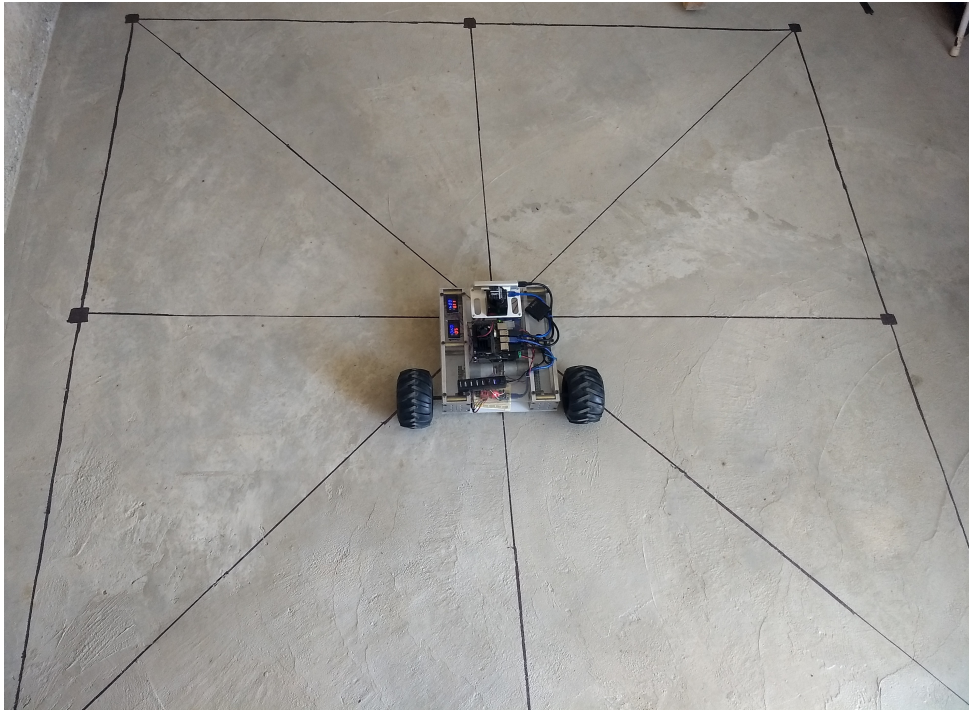


Figura 4.1: Superficie acondicionada para pruebas experimentales.

Los resultados de la primera prueba se muestran en las Figuras (4.2) y (4.3). Como se puede observar en la Figura (4.2) la estimación de la orientación presenta discontinuidades en $\pm\pi$, esto se debe a que la estimación de la orientación se realiza mediante cuaterniones que posteriormente se convierten a radianes, obteniendo de esta manera la orientación absoluta del vehículo con respecto al marco de referencia fijo en la tierra. Además, en esta grafica se puede observar que la estimación de la orientación parece ser estable sin embargo presenta mucho ruido de baja magnitud y alta frecuencia, consistente con el ruido comúnmente presente en los acelerómetros. Por otra parte, en (4.3) se observa el plano $X - Y$ de la trayectoria seguida por el vehículo durante la prueba, se observa que la trayectoria presenta una forma ovalada y no presenta un comportamiento uniforme ya que muestra muchos saltos de posición durante todo el recorrido. Los resultados de la prueba muestran que la estimación tanto de la orientación como de la posición del vehículo en el plano presentan un desempeño muy pobre.

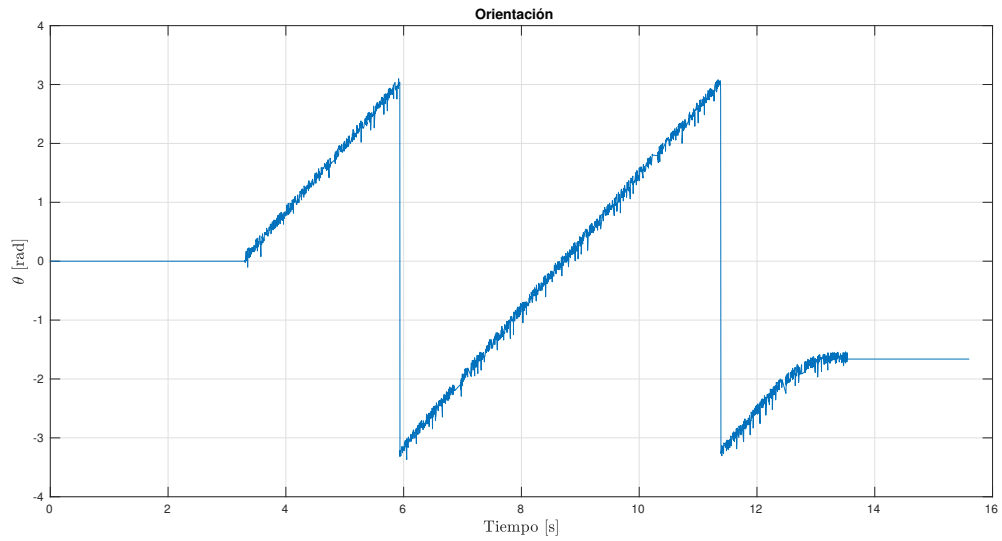


Figura 4.2: Gráfica de la orientación del robot sin amortiguadores de vibración.

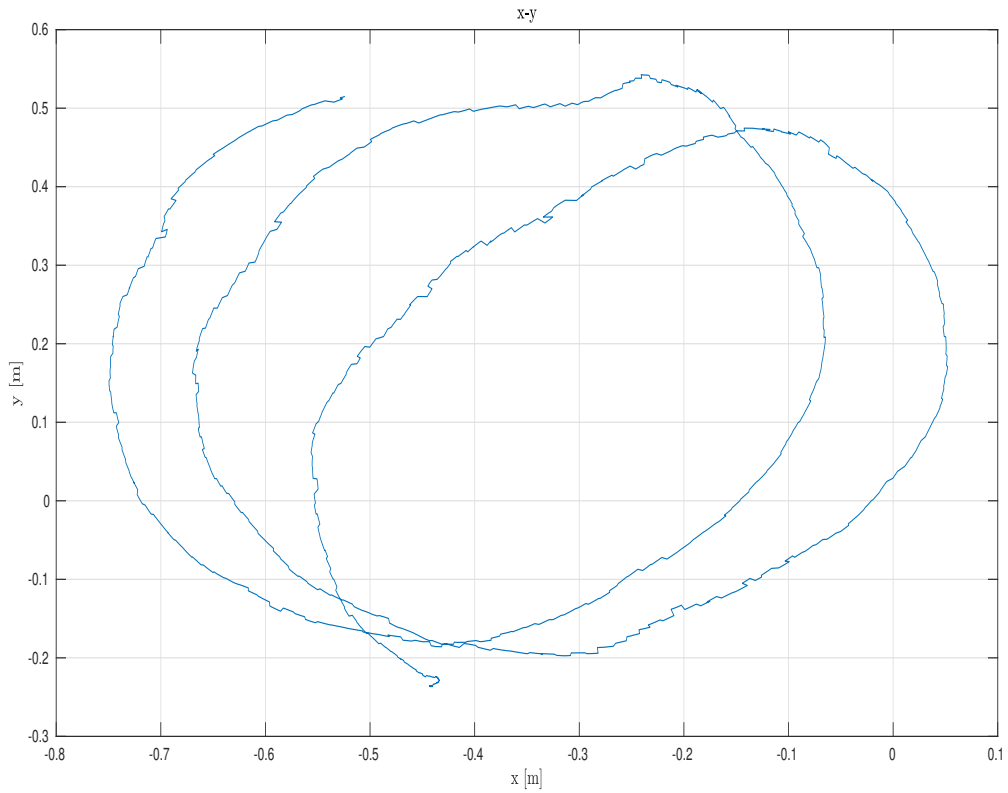


Figura 4.3: Gráfica del plano X-Y de la trayectoria del robot sin amortiguadores de vibración.

En las Figuras (4.4) y (4.5) se muestran los resultados obtenidos de la prueba realizada con los amortiguadores de vibración colocados en la base de los sensores. En la Figura (4.4) se muestra la gráfica de la orientación del robot, en esta gráfica se puede observar que el ruido que presentaba la estimación de la orientación ha desaparecido prácticamente en su totalidad con la adición de los amortiguadores. En la Figura (4.5) se muestra el plano $X - Y$ de la trayectoria descrita por el robot, esta vez se observa que el robot siguió una trayectoria circular casi perfecta donde solo se aprecia un salto en la estimación de la posición en las coordenadas $(-0.25, -0.1)$ aproximadamente. En esta prueba se observó que el desempeño de la cámara para estimar la pose del vehículo se ve afectada drásticamente por el efecto de las vibraciones mecánicas, sin embargo se observó que basta con la adición de los amortiguadores de vibración diseñados para mejorar su desempeño.

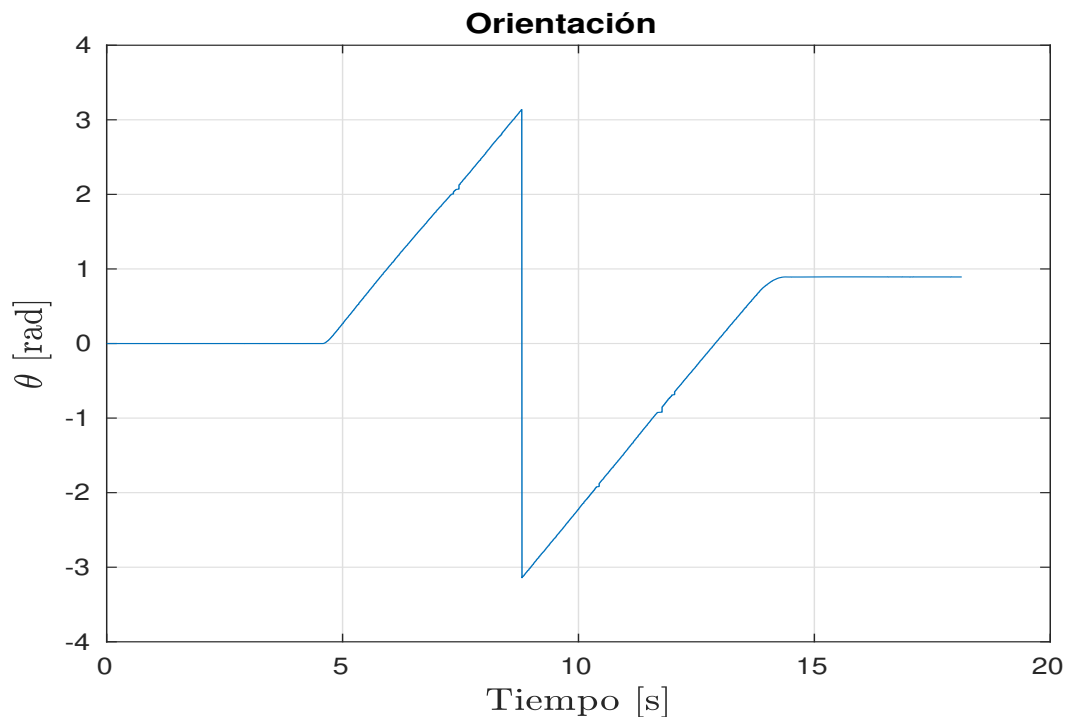


Figura 4.4: Gráfica de la orientación del robot con amortiguadores de vibración.

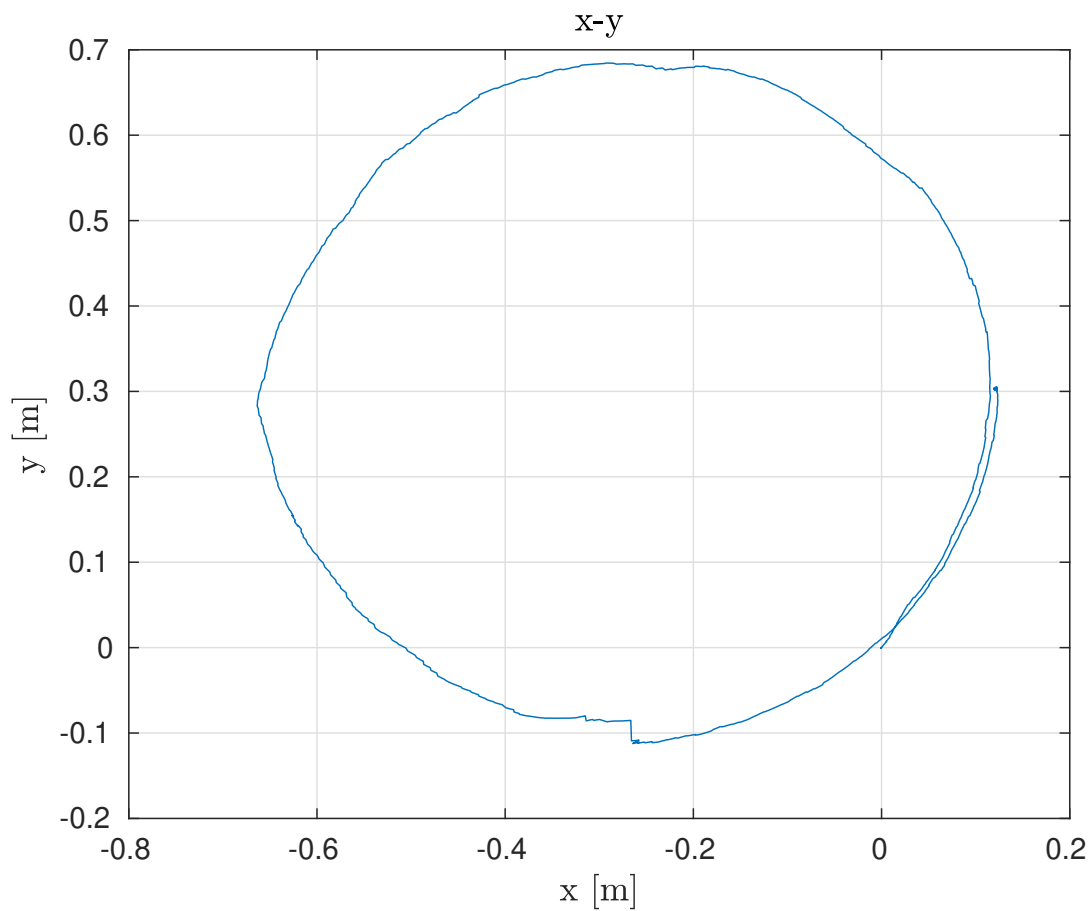


Figura 4.5: Gráfica del plano X-Y de la trayectoria del robot con amortiguadores de vibración.

Para evaluar el error de estimación de la posición final del prototipo, se colocó el vehículo en una posición conocida y se condujo en línea recta, posteriormente se determinó la posición final real mediante un flexómetro y la estimación de la cámara. Estos datos se registraron en una tabla junto con los errores de estimación, los resultados obtenidos tras realizar la prueba 10 veces se muestran en la tabla (4.1).

	Posición estimada (m)		Posición real (m)		Error de estimación		$\sqrt{\tilde{x}^2 + \tilde{y}^2}$
	x	y	x	y	\tilde{x}	\tilde{y}	
1	2.686	-0.086	2.69	-0.08	-0.002	-0.007	0.008
2	2.368	0.040	2.45	0.08	-0.083	-0.036	0.090
3	2.487	0.087	2.53	0.07	-0.048	0.019	0.052
4	2.581	0.028	2.49	0.01	0.088	0.019	0.090
5	2.499	0.072	2.60	0.12	-0.097	-0.048	0.109
6	2.333	0.079	2.28	0.04	0.056	0.038	0.068
7	2.377	-0.005	2.39	0.03	-0.010	-0.037	0.038
8	2.533	-0.083	2.61	-0.13	-0.073	0.042	0.085
9	2.341	-0.098	2.28	-0.09	0.057	-0.006	0.057
10	2.360	0.022	2.32	0.05	0.039	-0.025	0.047

Tabla 4.1: Error de estimación de la posición final del vehículo.

En la tabla (4.1) se puede observar que el error de estimación de la coordenada en el eje x es mayor que el de la coordenada en el eje y , esto coincide con lo esperado ya que el robot se desplazó en línea recta sobre el eje y , se calculó la magnitud de error de estimación en cada caso y se determinó que el promedio de la misma es de 0.064 lo cual representa el 2.6% de error de estimación de la pose considerando un promedio del desplazamiento real de 2.47 metros.

Generación del mapa de ocupación

Para afrontar el problema de localización el primer paso es crear el mapa de ocupación del lugar en el que se encuentra el robot. Para crear el mapa de ocupación se utilizó la librería de ROS llamada *gmapping*, esta librería permite crear el mapa de ocupación a partir de la medición del sensor láser en combinación con la estimación de la pose proporcionada por la cámara de seguimiento. Dado que la pose se encuentra en un marco de coordenadas diferente al sensor LIDAR, es necesario hacer uso de transformaciones de

coordenadas, para esto se utilizo el paquete *tf* de ROS, este paquete permite realizar los cambios de coordenadas del los sensores de una forma muy intuitiva.

Para poder visualizar el esquema de comunicación que existe entre los nodos se hizo uso del paquete *rqt graph*, este paquete permite generar un grafo que muestra la interacción entre los nodos y los tópicos activos en el robot, en la Figura (4.6) se muestra el grafo resultante.

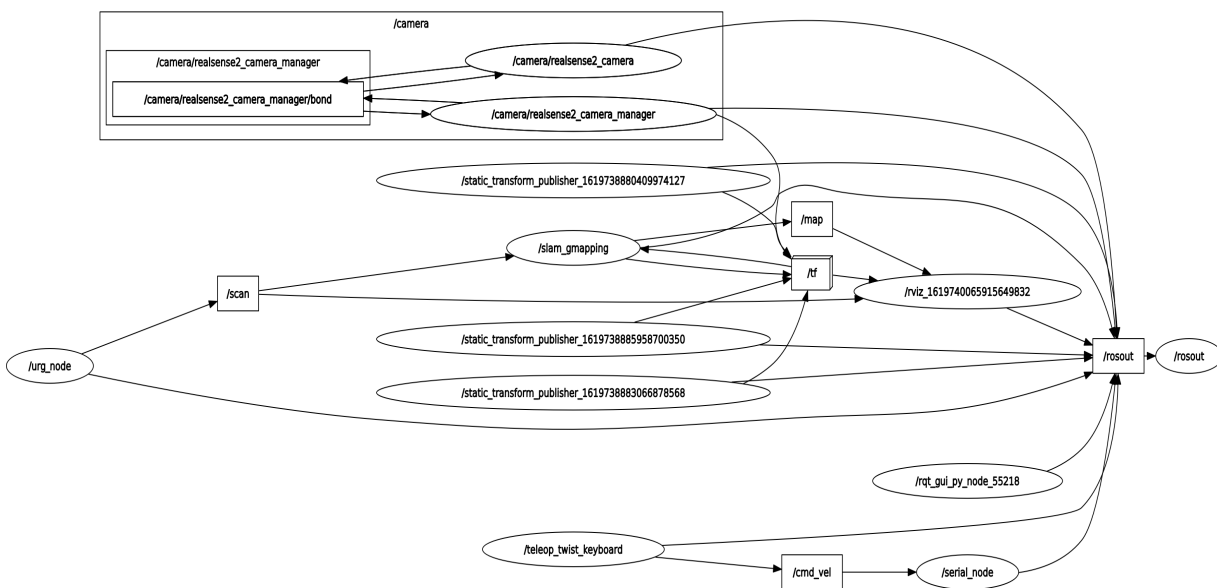
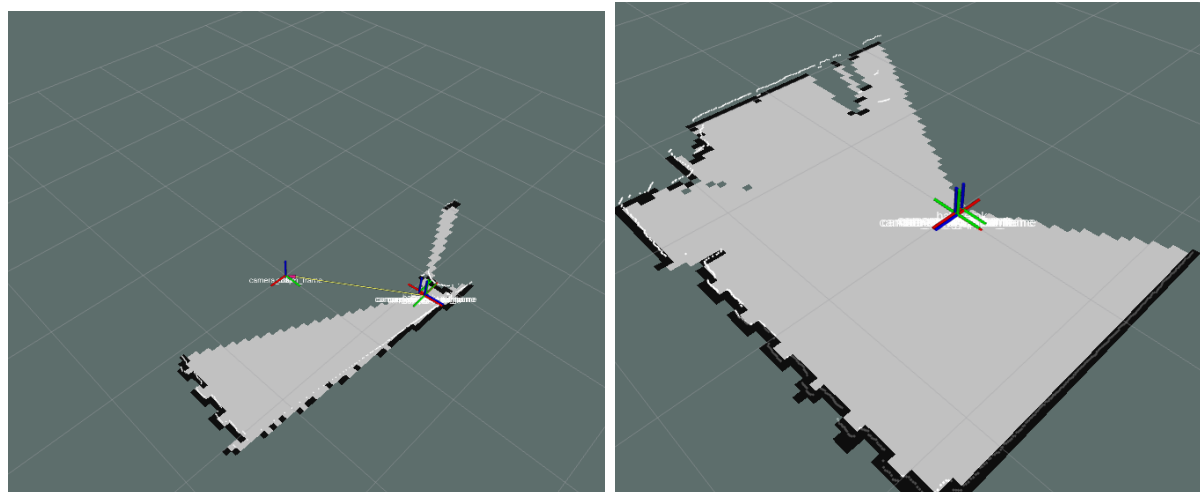


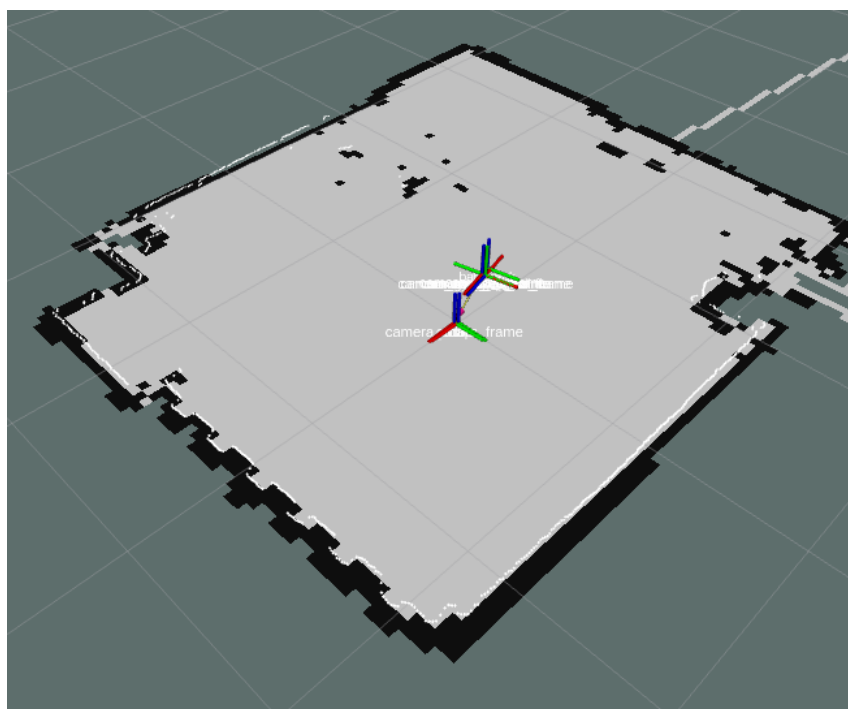
Figura 4.6: Grafo de los nodos y tópicos activos para la generación del mapa de ocupación.

Tras obtener los parámetros del prototipo se procedió a generar el mapa de ocupación del espacio de pruebas descrito anteriormente, en la Figura (4.7) se muestran las diferentes etapas de construcción del mapa de ocupación, inicialmente el programa crea una parte del mismo basado en la observación del sensor LIDAR en la posición inicial, este mapa se muestra en (4.7a), posteriormente se condujo el prototipo a lo largo del espacio para ir complementando el mapa como se muestra en (4.7b), hasta generar el mapa de ocupación final como el mostrado en (4.7c).



(a) Mapa inicial.

(b) Construcción del mapa al 50%.



(c) Mapa final.

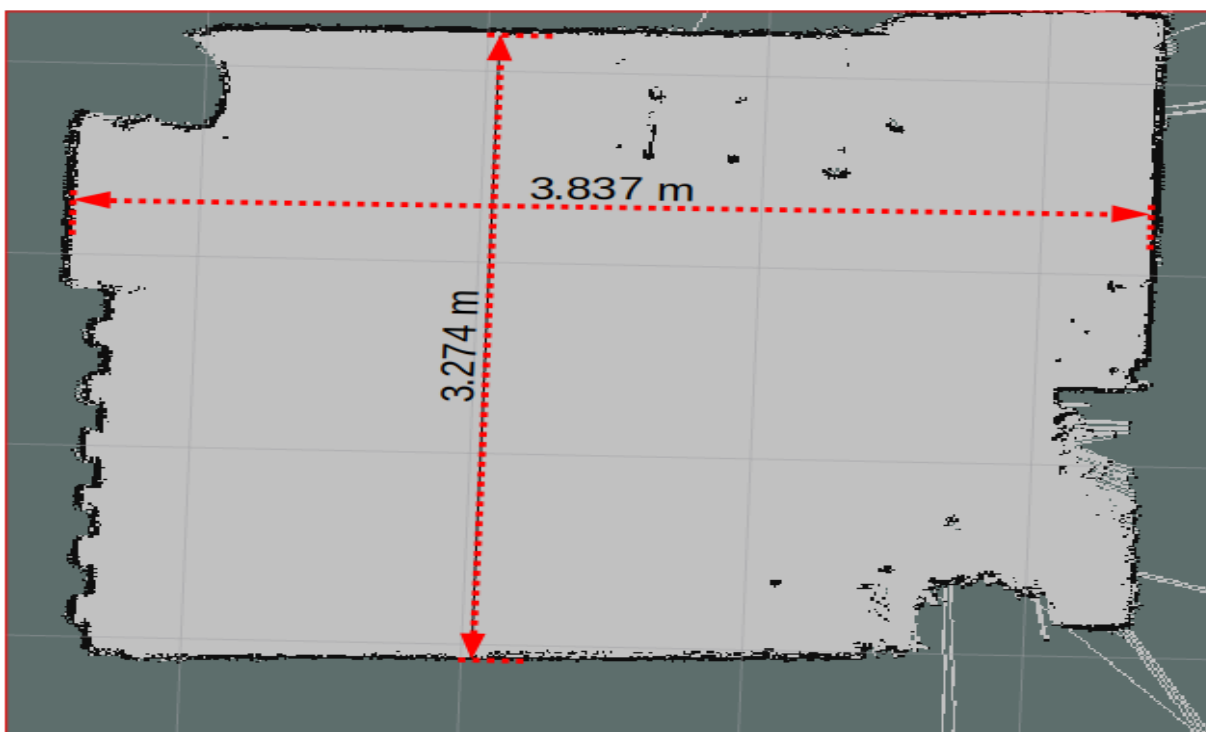
Figura 4.7: Proceso de construcción del mapa de ocupación.

Para evaluar la precisión del mapa obtenido se obtuvieron las medidas de la habitación obtenidas con las dimensiones reales del espacio. En la Figura (4.8) se muestran las medidas obtenidas mediante el algoritmo de *gmapping*, se evaluaron las distancias entre las paredes de la habitación a lo largo de los ejes x,y . Las medidas reales y las obtenidas se muestran

Eje	Distancia real [m]	Distancia obtenida [m]	Error [m]
Y	3.250	3.274	0.024
X	3.800	3.837	0.037

Tabla 4.2: Dimensiones principales de la habitación.

en la tabla (4.2). El error en la obtención de las dimensiones de la habitación es de 3.7cm y 2.4cm lo cual representa un 0.97 % y 0.73 % de error en los ejes x,y respectivamente. Estas cifras reflejan una precisión bastante alta lo cual a su vez es un indicador de la adecuada obtención de la pose del vehículo así como la ausencia de ruido en las mediciones del sensor LIDAR.

Figura 4.8: Medidas de la habitación obtenidas mediante el algoritmo de *gmapping*.

4.1.2. Control

Como primera aproximación se implementó un algoritmo de control clásico PID, para esto se propuso la estructura mostrada en la Figura (4.9) donde se utiliza el modelo cinemático directo e inverso del robot para obtener la señal de control para los motores y para obtener las velocidades lineal y angular del robot mediante la odometría de la cámara

de seguimiento. Para obtener las referencias de velocidad necesarias para realizar las tareas de navegación se seleccionó la pila de navegación de ROS que proporciona como salida la referencia de velocidad en coordenadas de la base del robot.

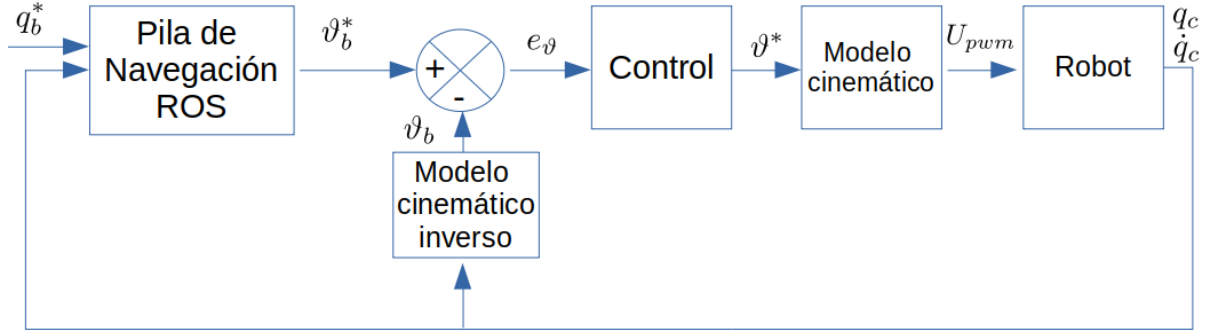


Figura 4.9: Descripción del sistema de control y navegación.

El modelo cinemático del robot se definió en el capítulo (2) como en la ecuación (2.7), y está dado por

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & -d \sin \theta \\ \sin \theta & d \cos \theta \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (4.1)$$

donde $\dot{x}_1 = \dot{x}$ y $\dot{x}_2 = \dot{y}$ son las velocidades del robot en los ejes x, y respectivamente, $\dot{\theta}$ es la velocidad angular del robot en el eje z y v, ω son la velocidad lineal y angular del robot. El modelo cinemático inverso en función de \dot{x} y \dot{y} está dado por

$$\begin{bmatrix} v \\ w \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\frac{1}{d} \sin \theta & \frac{1}{d} \cos \theta \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} \quad (4.2)$$

de (4.2) tenemos que la velocidad lineal del robot es

$$v = \dot{x} \cos \theta + \dot{y} \sin \theta \quad (4.3)$$

La ley de control PID esta dada por

$$U = \begin{bmatrix} u_v \\ u_w \end{bmatrix} = K_p e_{\vartheta} + K_d \dot{e}_{\vartheta} + K_i \int e_{\vartheta} dt \quad (4.4)$$

donde K_p , K_d y K_i son matrices diagonales que corresponden a las ganancias proporcional, derivativa y e integral respectivamente. El error está dado por

$$e_{\vartheta} = \vartheta^* - \vartheta = \begin{bmatrix} v^* - v \\ w^* - w \end{bmatrix} \quad (4.5)$$

inicialmente en una primera implementación la derivada del error se calculaba mediante diferencias como

$$\dot{e}_{\vartheta}(t) = \frac{e_{\vartheta}(t) - e_{\vartheta}(t-1)}{\Delta t} \quad (4.6)$$

sin embargo como la señal es muy ruidosa la derivada provocaba oscilaciones en el desplazamiento del robot por lo que se implementó un diferenciador robusto exacto como en [31] donde para diferenciar una señal base se considera la ecuación auxiliar

$$\dot{x} = u \quad (4.7)$$

posteriormente se tiene que aplicando el algoritmo de segundo orden Super Twisting para mantener $x - f(t) = 0$, dada por

$$\begin{aligned} \dot{x} &= u = -\lambda|x - f(t)|^{1/2} \text{sgn}(x - f(t)) + u_1 \\ \dot{u}_1 &= -\alpha \text{sgn}(x - f(t)) \end{aligned} \quad (4.8)$$

donde $u(t)$ es la salida del diferenciador, es decir $u(t) = \dot{f}(t)$.

Por otra parte, tras implementar el controlador se determinó que la obtención de las velocidades del robot mediante la cámara no era viable debido a que esta presenta un retardo importante en la señal y además esta es muy susceptible a las vibraciones provocadas por la superficie, es por esto que se integró la odometría de las ruedas a la estructura de control. En la Figura (4.10) se muestra la gráfica de la velocidad lineal del robot obtenida mediante la cámara, los encoder y la referencia, en la figura se puede apreciar que la velocidad obtenida mediante la cámara presenta un retardo casi del doble del que presenta la señal obtenida mediante los encoder, además se puede apreciar que la

cámara muestra sobretiros y oscilaciones que no se ven reflejadas en el desplazamiento del robot. En la Figura (4.11) se muestra la gráfica de la velocidad angular del robot obtenida mediante la cámara, los encoder y la referencia, al igual que con la velocidad lineal, las gráfica muestra que el retardo de la cámara es más grande que los encoder y en este caso presenta sobretiros mas grandes que en la velocidad lineal. Debido a lo anterior se optó por cerrar el lazo de control de velocidad con los encoder y utilizar la odometría de la cámara para obtener la pose del vehículo.

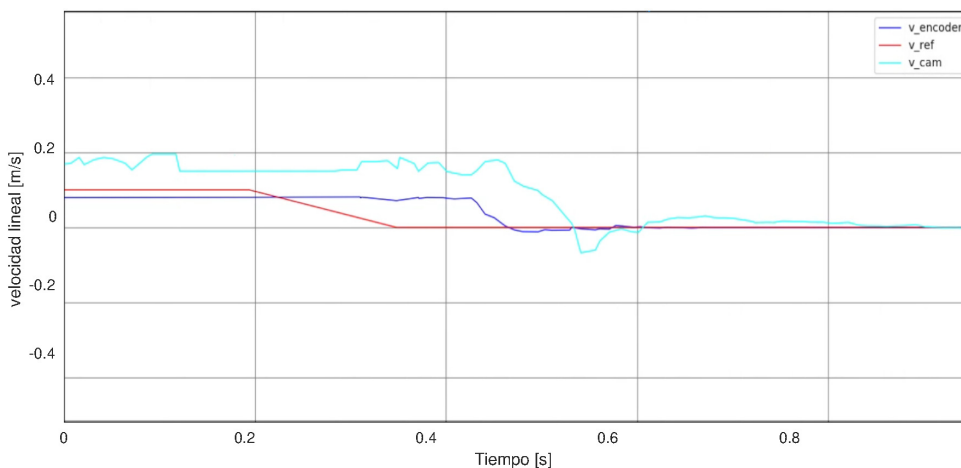


Figura 4.10: Velocidad lineal del robot, donde $v_{encoder}$, v_{cam} y v_{ref} son la velocidad lineal del robot medida mediante los encoder, la cámara y la referencia respectivamente.

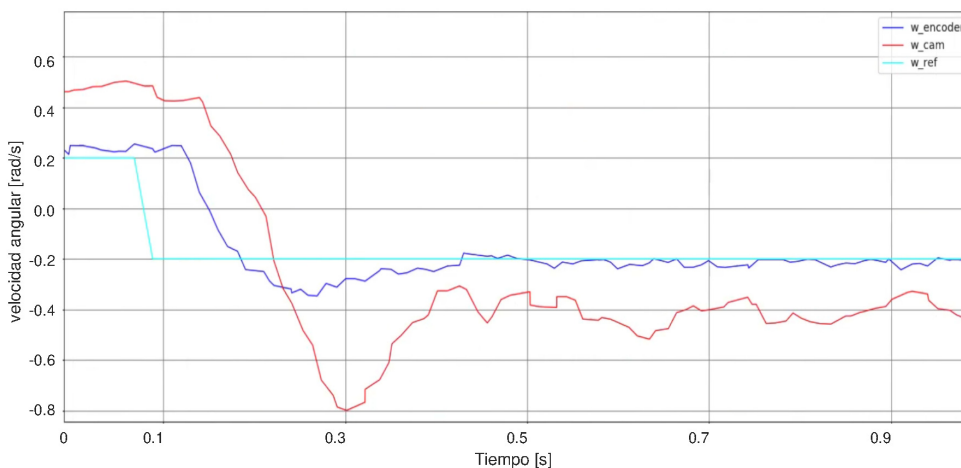


Figura 4.11: Velocidad angular del robot, donde $w_{encoder}$, w_{cam} y w_{ref} son la velocidad angular del robot medida mediante los encoder, la cámara y la referencia respectivamente.

Ahora con el lazo de control de velocidad implementado los encoder como fuente de las velocidades del robot se implementó el controlador PID y se ajustaron sus ganancias, en la Figura (4.12) se muestra la gráfica de la velocidad lineal del robot tras ajustar los parámetros del controlador, se puede observar que la respuesta del sistema es rápida y presenta poco ruido, las ganancias se ajustaron para reducir lo más posible los sobretiros ya que estos provocan que la cámara de seguimiento introduzca errores en la estimación de la pose. En la Figura (4.13) se muestra la gráfica de la velocidad angular del robot, en esta gráfica se puede apreciar que la respuesta es mas rápida que la velocidad lineal, además, presenta mayor error en estado estacionario, esto es debido a que la dinámica de la velocidad angular del robot es muy rápida y está fuertemente ligada a la velocidad lineal del robot, esto se puede apreciar en la Figura (4.14) donde se aprecia que cuando se aplica un escalón a la velocidad lineal se produce error en estado estacionario en la velocidad angular, en caso de que se aplique un escalón a la velocidad angular se aprecia que también afecta a la velocidad lineal pero la magnitud del error es pequeña por lo que no es significativa para el desempeño del robot.

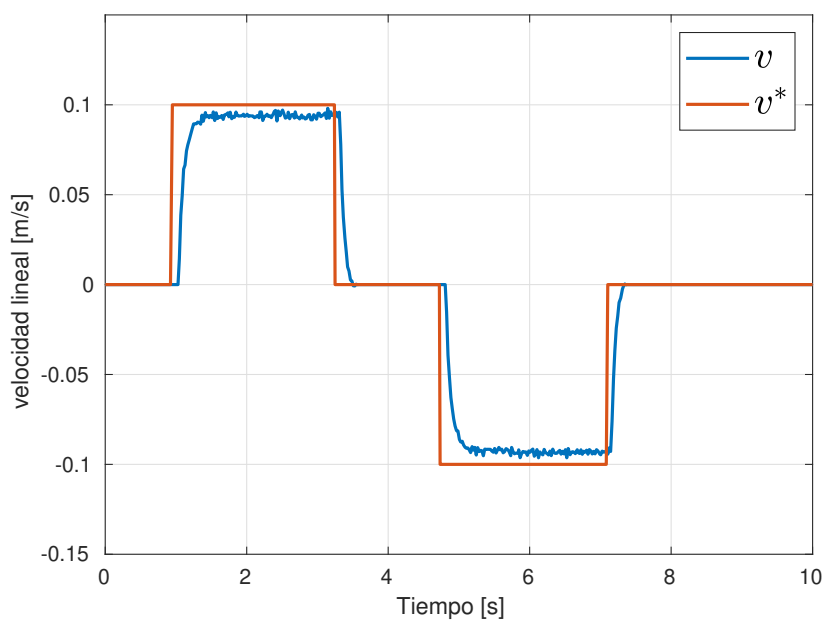


Figura 4.12: Gráfica de la velocidad lineal del robot con un control PID, donde v y v^* son la velocidad lineal del robot y su referencia respectivamente.

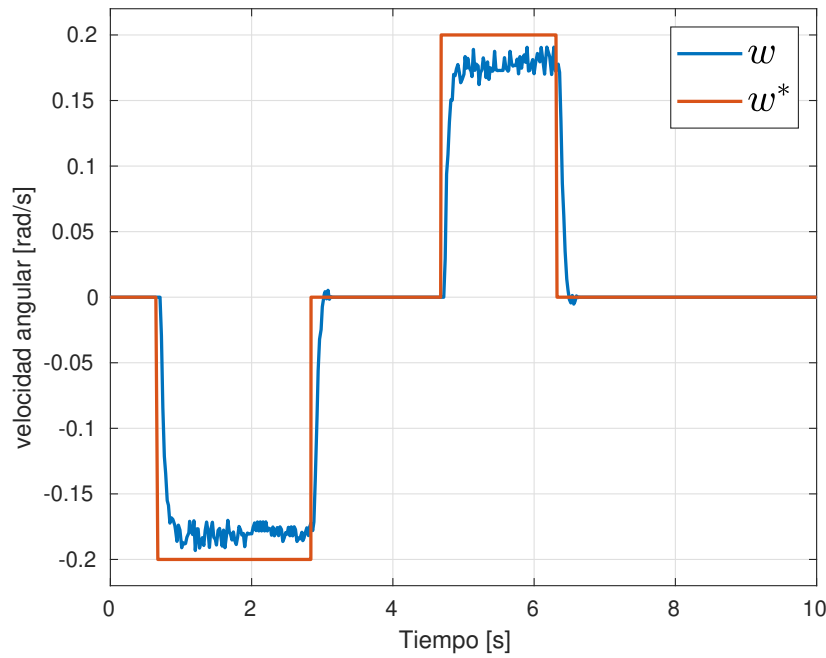


Figura 4.13: Gráfica de la velocidad angular del robot con un control PID, donde ω y ω^* son la velocidad angular del robot y su referencia respectivamente.

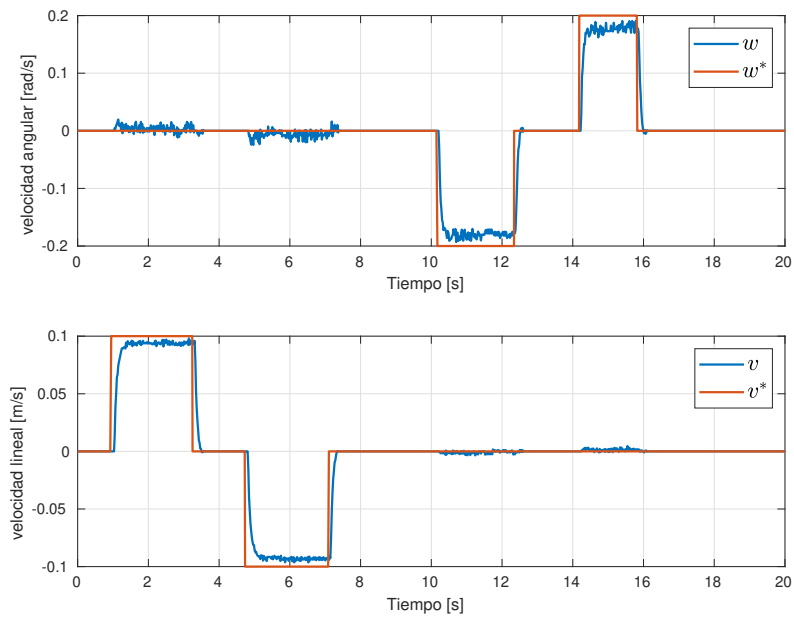


Figura 4.14: Gráfica de las velocidades lineal y angular del robot, donde ω y ω^* son la velocidad angular del robot y su referencia respectivamente, v y v^* son la velocidad lineal del robot y su referencia respectivamente.

4.1.3. Navegación

La pila de navegación de ROS toma información de odometría, flujos de sensores y una pose objetivo y genera comandos de velocidad segura que se envían a una base móvil. Los requerimientos básicos para la utilización de la pila de navegación y los utilizados en este trabajo son los siguientes:

- Fuente de sensores: LIDAR (HOKUYO URG).
- Fuente de odometría: Cámara de seguimiento Intel Realsense T265.
- Transformación de sensores: Descripción mediante archivo URDF.
- Controlador base: control PID
- Mapa: Generado mediante el algoritmo *gmapping*

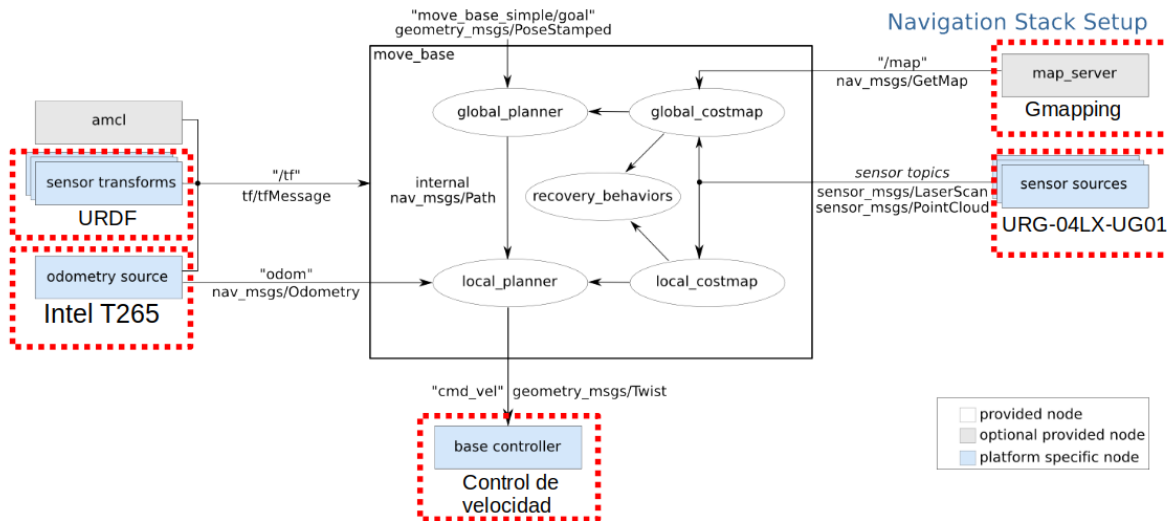


Figura 4.15: Configuración de la pila de navegación.

En la Figura (4.15) se muestra el diagrama a bloques de la configuración de la pila de navegación, en la figura se señalan en un recuadro rojo los elementos suministrados por la plataforma diseñada en este trabajo. Por otra parte, en la Figura (4.16) se muestra el diagrama de flujo del comportamiento del modo de recuperación de la pila de navegación,

es decir el comportamiento que tendrá el robot en caso de no poder continuar con la trayectoria planeada globalmente.

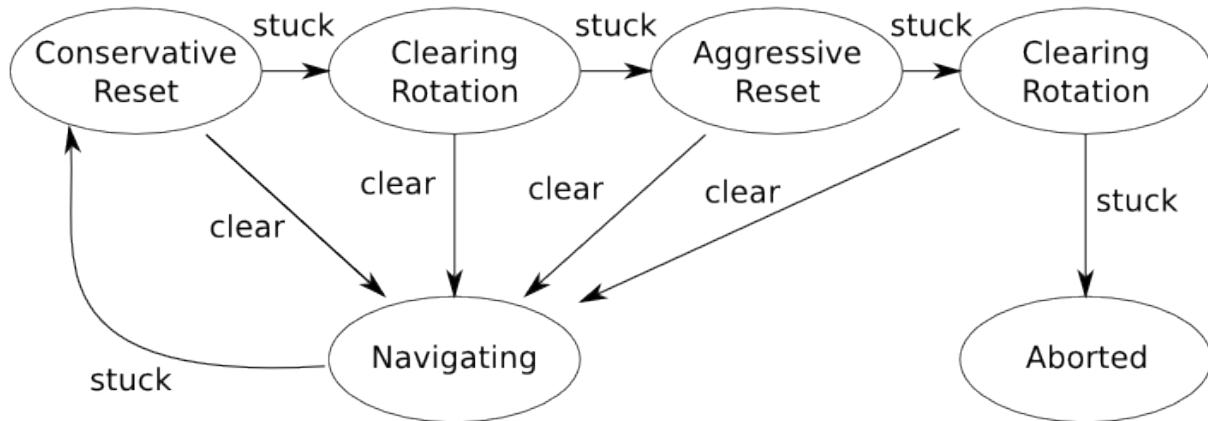


Figura 4.16: Comportamiento del modo de recuperación de la pila de navegación.

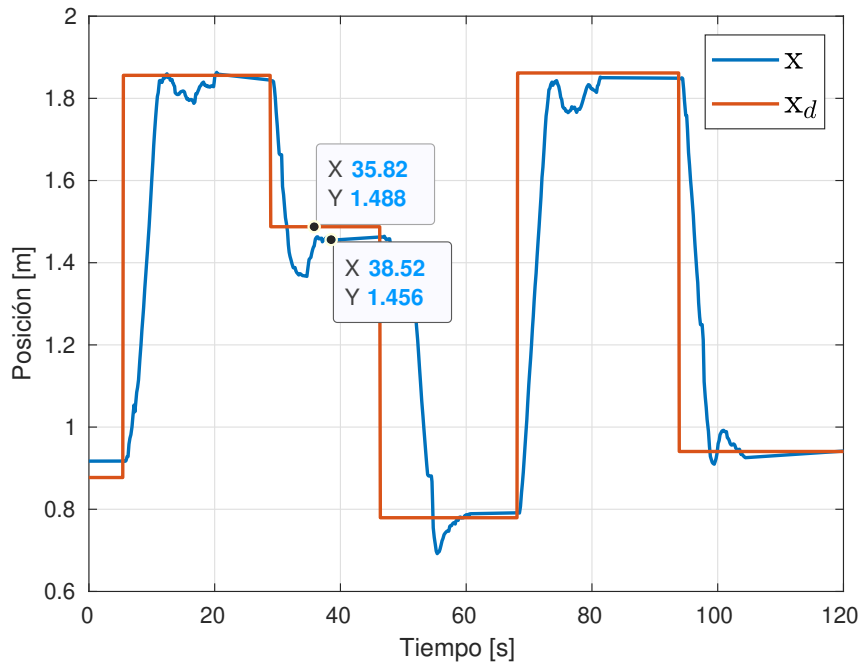


Figura 4.17: Gráfica de la posición del robot en la coordenada X, donde X y X_d son la posición actual y la posición deseada del robot con respecto al eje X del plano.

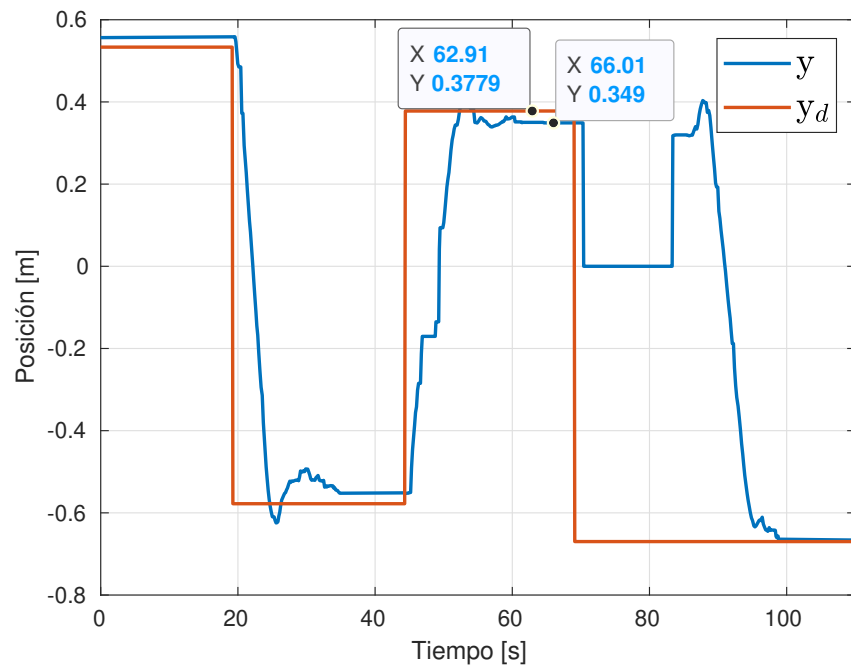


Figura 4.18: Gráfica de la posición del robot en la coordenada Y, donde Y y Y_d son la posición actual y la posición deseada del robot con respecto al eje Y del plano.

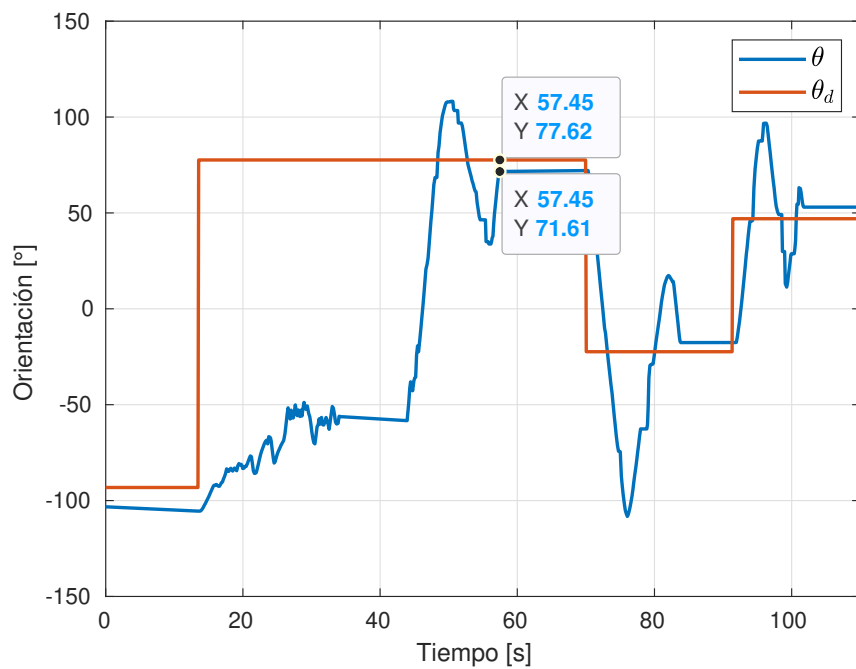


Figura 4.19: Gráfica de la orientación del robot, donde θ y θ_d son el ángulo actual y el ángulo deseado del robot en el eje de coordenadas Z.

En las Figuras de (4.17) a (4.19) se muestran las gráficas de las trayectorias seguidas por el robot para desplazarse a una posición objetivo. En la Figura (4.17) se muestra la gráfica correspondiente a la coordenada x , en esta gráfica se observa que el robot alcanza la referencia con un error en estado estacionario de 3.2cm aproximadamente, se observa además que existen oscilaciones en la trayectoria del robot, de acuerdo a los experimentos realizados se determinó que esto es debido a que la pose proporcionada por la cámara de seguimiento presenta un retardo. En la Figura (4.18) se muestra la gráfica correspondiente a la coordenada y , esta coordenada presenta un comportamiento similar, el error en estado estacionario es de aproximadamente 3cm, este valor también es aceptable, además en esta gráfica se puede observar que en el tiempo aproximado de 80 segundos el error aumenta, esto es debido a que la referencia es una coordenada global y aunque la posición final del robot tenderá a este punto no necesariamente coincide con el objetivo local que puede variar debido a un desplazamiento de reversa o para evitar un obstáculo en el camino. Finalmente en la Figura (4.19) se muestra la gráfica de la orientación del robot, en esta figura se observa que el error en estado estacionario es de aproximadamente 6.01 grados, este valor es aceptable sin embargo en la orientación del robot es donde más se observan los efectos del retardo en la obtención de la pose lo que genera oscilaciones de mayor amplitud que en las otras coordenadas, esto es debido a la velocidad de la dinámica de la orientación del vehículo como se mencionó anteriormente. Es por lo anterior que aunque el robot logró desplazarse a las referencias es necesario mejorar la odometría del vehículo mediante la fusión de la odometría de la cámara con la de las ruedas.

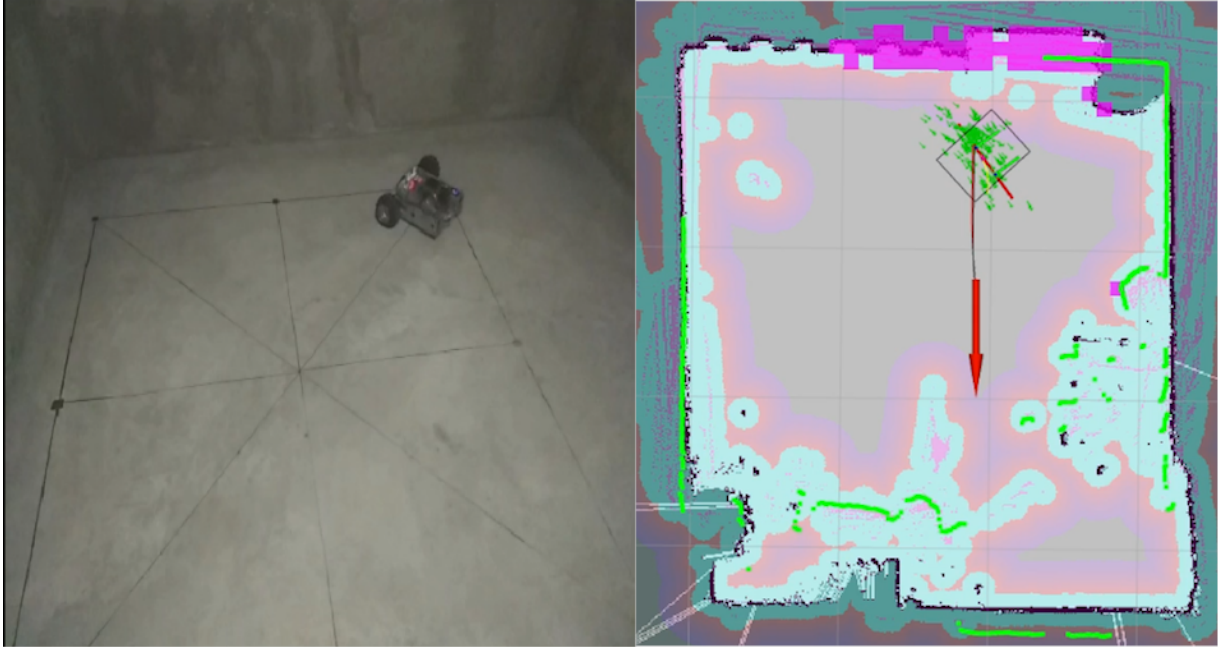


Figura 4.20: Interfaz RViz de ROS junto con el prototipo en tiempo real durante su desplazamiento a una pose objetivo.

En la Figura (4.20) se muestra del lado izquierdo una imagen del robot mientras se desplaza para alcanzar una pose objetivo y del lado derecho se observa la interfaz RViz de ROS en el mismo momento, en esta interfaz se pueden observar la pose objetivo, la pose actual del robot, las posibles trayectorias del robot que serán evaluadas por el algoritmo de navegación, las mediciones del sensor LIDAR y el mapa de ocupación del lugar que se encuentra siendo actualizado mientras el robot se desplaza.

Capítulo 5

Conclusiones y trabajos futuros

5.1. Conclusiones

Como resultado del desarrollo de este trabajo, se pueden mencionar las siguientes conclusiones:

- Los resultados obtenidos de la construcción del mapa de ocupación mostraron que la implementación de la librería de ROS GMapping (filtro de partículas) en conjunto con los sensores LIDAR y la cámara de seguimiento seleccionados generan un mapa con dimensiones precisas, cercanas a las dimensiones del entorno real.
- Con la implementación del algoritmo de localización adaptativo Monte Carlo de ROS no es necesario tener conocimiento a priori del espacio donde va a desplazarse el robot, sin embargo, al generar un mapa previamente, se puede realizar la puesta en marcha del robot más rápidamente y éste es capaz de actualizar el mapa conforme el ambiente cambia o se presentan elementos dinámicos en el entorno.
- Mediante la ley de control PID para las velocidades angular y lineal del robot en conjunto con la librería de navegación, fue posible realizar las tareas de regulación de postura satisfactoriamente, ya que se logra posicionar el robot en las poses objetivo adecuadamente.
- Se propuso la instrumentación mínima necesaria para realizar las tareas de navega-

ción y regulación de postura de un robot móvil, además se diseñó y construyó un prototipo para implementar los algoritmos de navegación y control.

- Existe mucha información en la literatura con respecto al control de robots móviles, en este caso se decidió estudiar la implementación de los algoritmos en una plataforma experimental y con ello sentar las bases y los requerimientos para realizar la implementación de robot móviles con aplicaciones en el interior de instalaciones médicas.
- El envío de las posiciones objetivo del robot se realizan desde una estación de trabajo mediante la interfaz grafica RViz de ROS, en esta interfaz se carga el mapa previamente construido donde es fácil identificar la posición deseada y seleccionarla como objetivo.

5.2. Trabajos futuros

En este trabajo se obtuvieron resultados satisfactorios ya que en efecto se logra cumplir con la tarea de posicionamiento del robot móvil diseñado. No obstante, aún quedan bastantes problemas por abordar al finalizar esta tesis, de tal forma que en trabajos futuros se pretende:

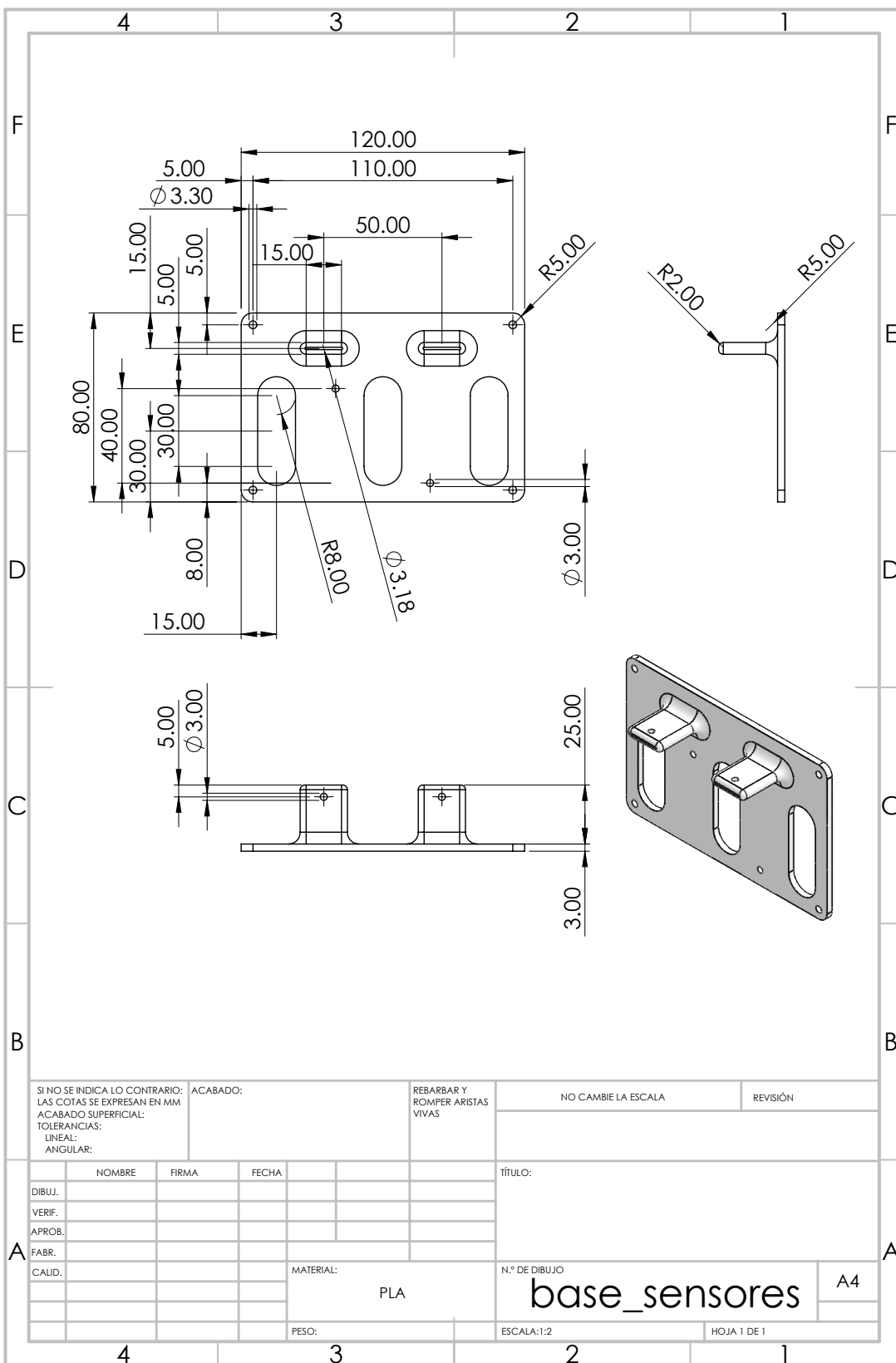
- Diseñar un esquema de control que permita controlar el robot en tiempo real. Además, se propone implementar este esquema en una versión de ROS2 ya que en la versión de ROS utilizada en este trabajo no se considera el tiempo real, y en la versión futura ya se ha definido y diseñado esta característica.
- Al trabajar con una cámara, se observó que la estimación de la posición es muy sensible al ruido y a pesar de proporcionar una buena estimación de la pose, esto no ocurre con la medición de las velocidades. Por lo anterior, se propone realizar la fusión de los datos de la cámara con los de la odometría de las ruedas, para ello es necesario seleccionar otro microcontrolador para realizar la lectura de los encoder de forma independiente a la activación de los motores.

- Evaluar el desempeño de los algoritmos en diferentes mapas o entornos con más variables aleatorias.

- Evaluar otras estrategias de control para regular la velocidad de los motores para así obtener un sistema más robusto. En estas estrategias de control se debe priorizar el obtener una respuesta suave, esto por la misma razón de eliminar el problema de estimación de la pose mediante la cámara de seguimiento.

Apéndice A

Diagramas y especificaciones de la estructura



SI NO SE INDICA LO CONTRARIO:
 LAS COTAS SE EXPRESAN EN MM
 ACABADO SUPERFICIAL:
 TOLERANCIAS:
 LINEAL:
 ANGULAR:

ACABADO:

REBARBAR Y
 ROMPER ARISTAS
 VIVAS

NO CAMBIE LA ESCALA

REVISIÓN

	NOMBRE	FIRMA	FECHA	
DIBUJ.				
VERIF.				
APROB.				
FABR.				
CALID.				

TÍTULO:

MATERIAL:

PLA

N.º DE DIBUJO

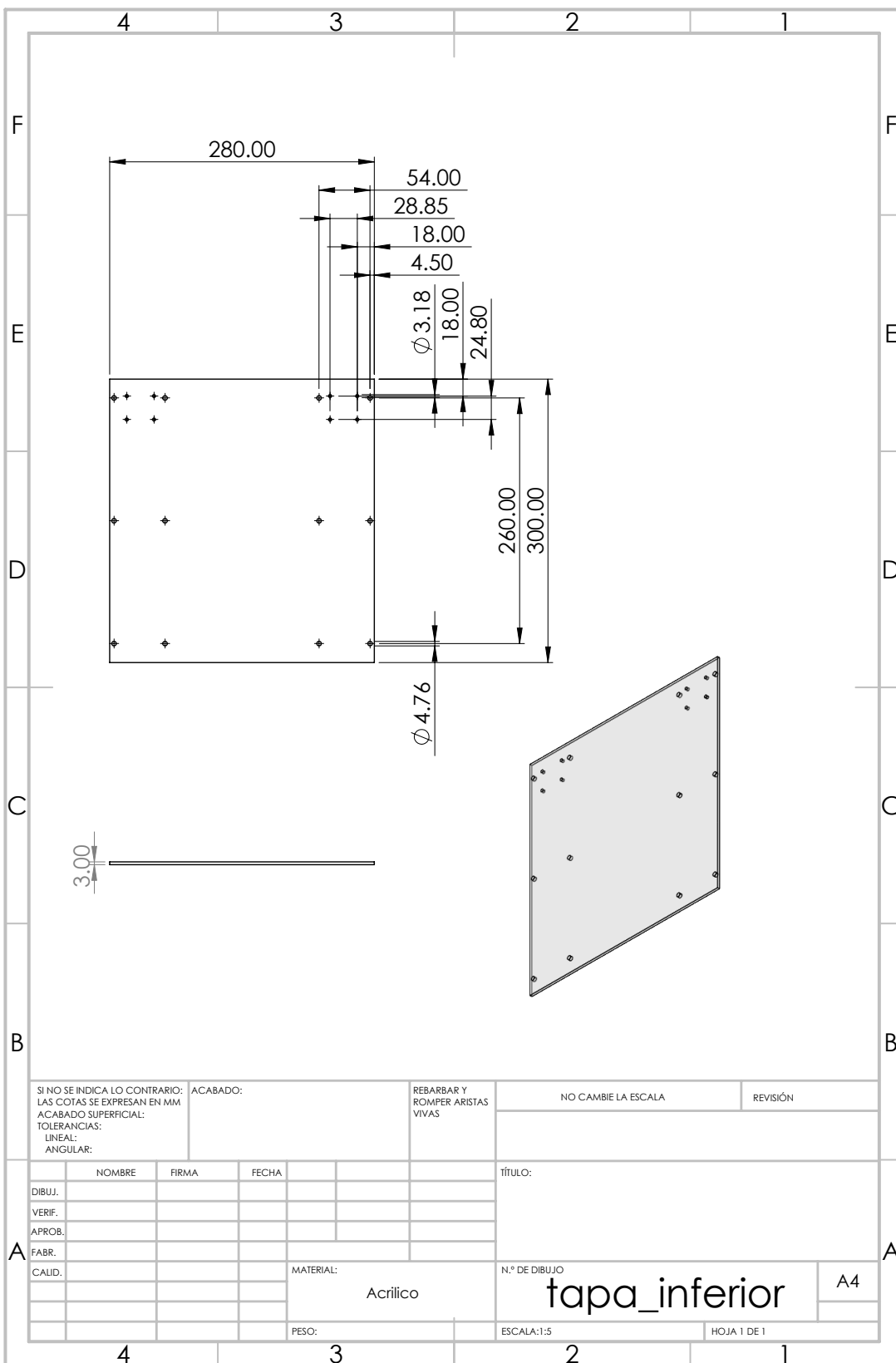
base_sensores

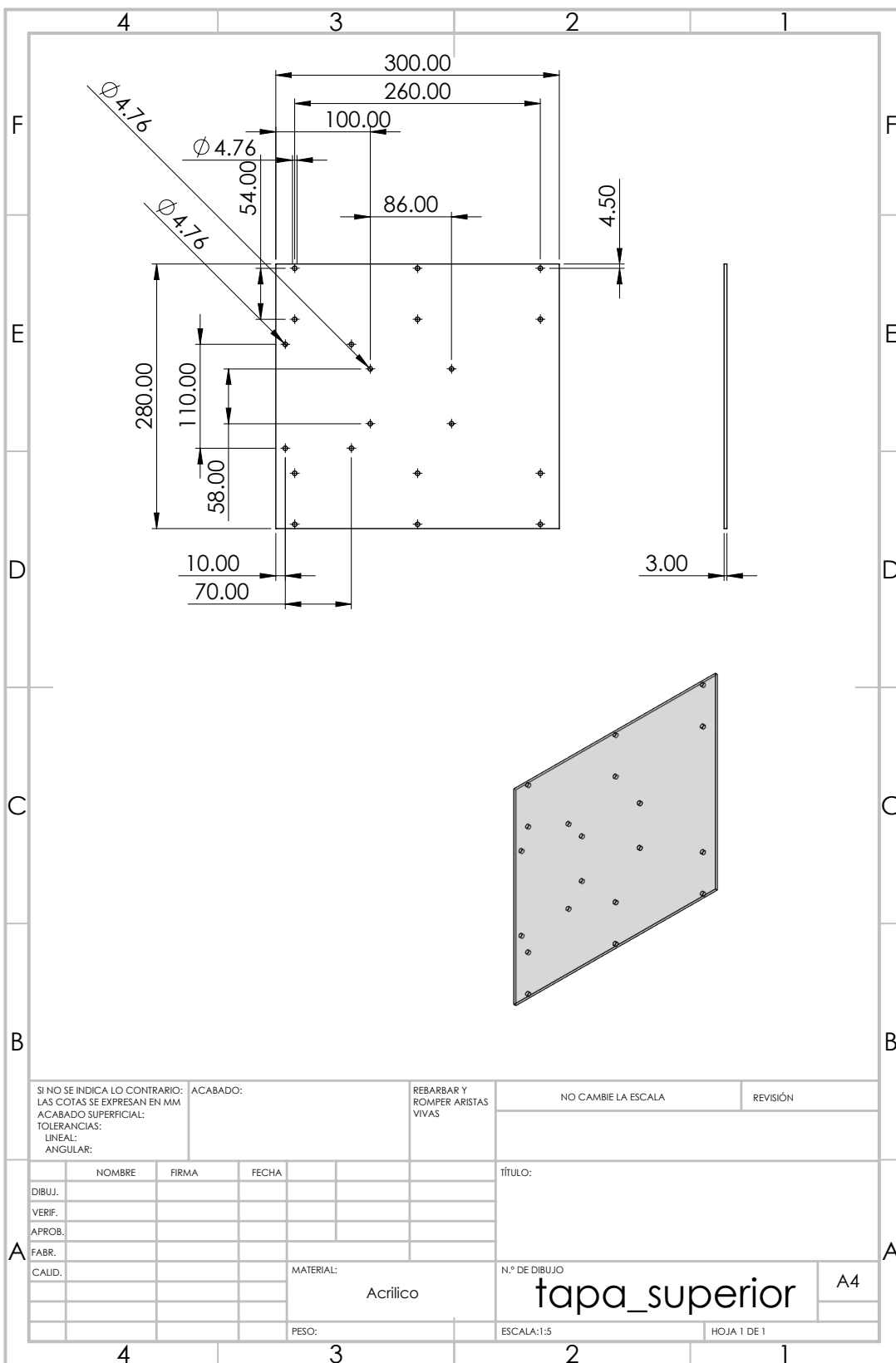
A4

PESO:

ESCALA:1:2

HOJA 1 DE 1





SI NO SE INDICA LO CONTRARIO:
 LAS COTAS SE EXPRESAN EN MM
 ACABADO SUPERFICIAL:
 TOLERANCIAS:
 LINEAL:
 ANGULAR:

ACABADO:
 REBARBAR Y
 ROMPER ARISTAS
 VIVAS

NO CAMBIE LA ESCALA

REVISIÓN

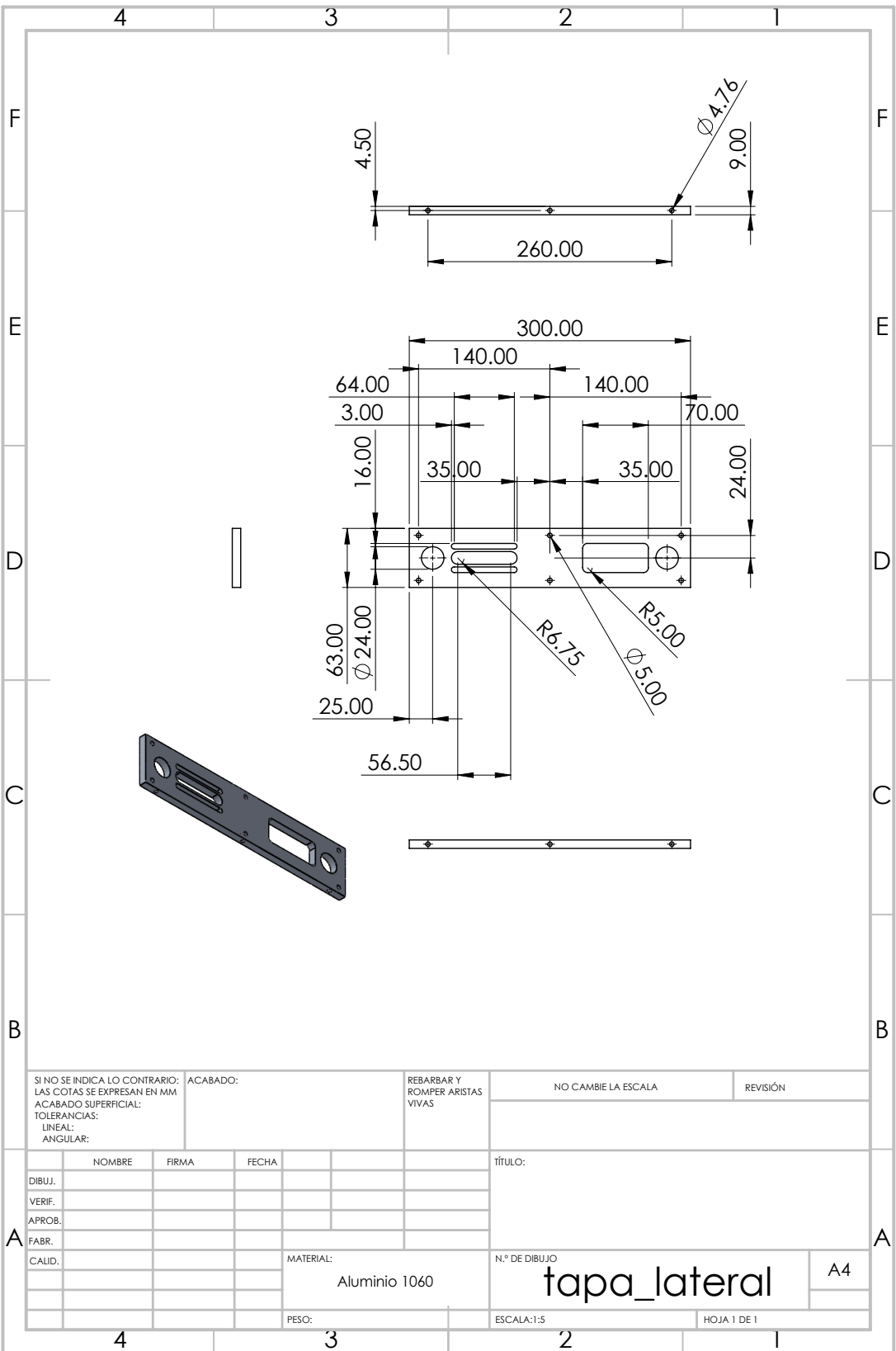
	NOMBRE	FIRMA	FECHA		TÍTULO:
DIBUJ.					
VERIF.					
APROB.					
FABR.					
CALID.					

MATERIAL:
 Acrílico

N.º DE DIBUJO
 tapa_superior A4

PESO:

ESCALA:1:5 HOJA 1 DE 1



SI NO SE INDICA LO CONTRARIO:
 LAS COTAS SE EXPRESAN EN MM
 ACABADO SUPERFICIAL:
 TOLERANCIAS:
 LINEAL:
 ANGULAR:

ACABADO:
 REBARBAR Y
 ROMPER ARISTAS
 VIVAS

NO CAMBIE LA ESCALA

REVISIÓN

	NOMBRE	FIRMA	FECHA
DIBUJ.			
VERIF.			
APROB.			
FABR.			

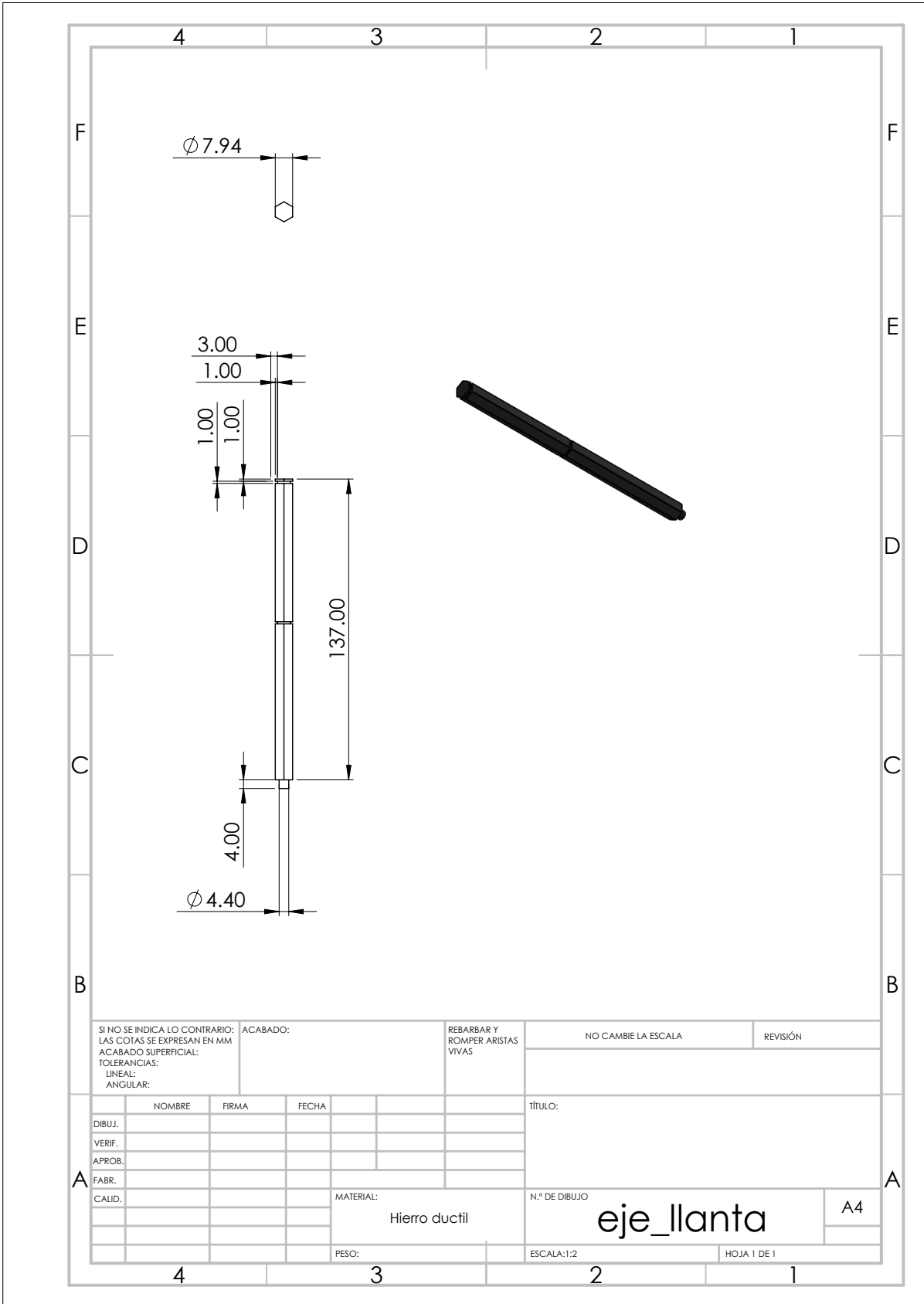
TÍTULO:

MATERIAL:
 Aluminio 1060

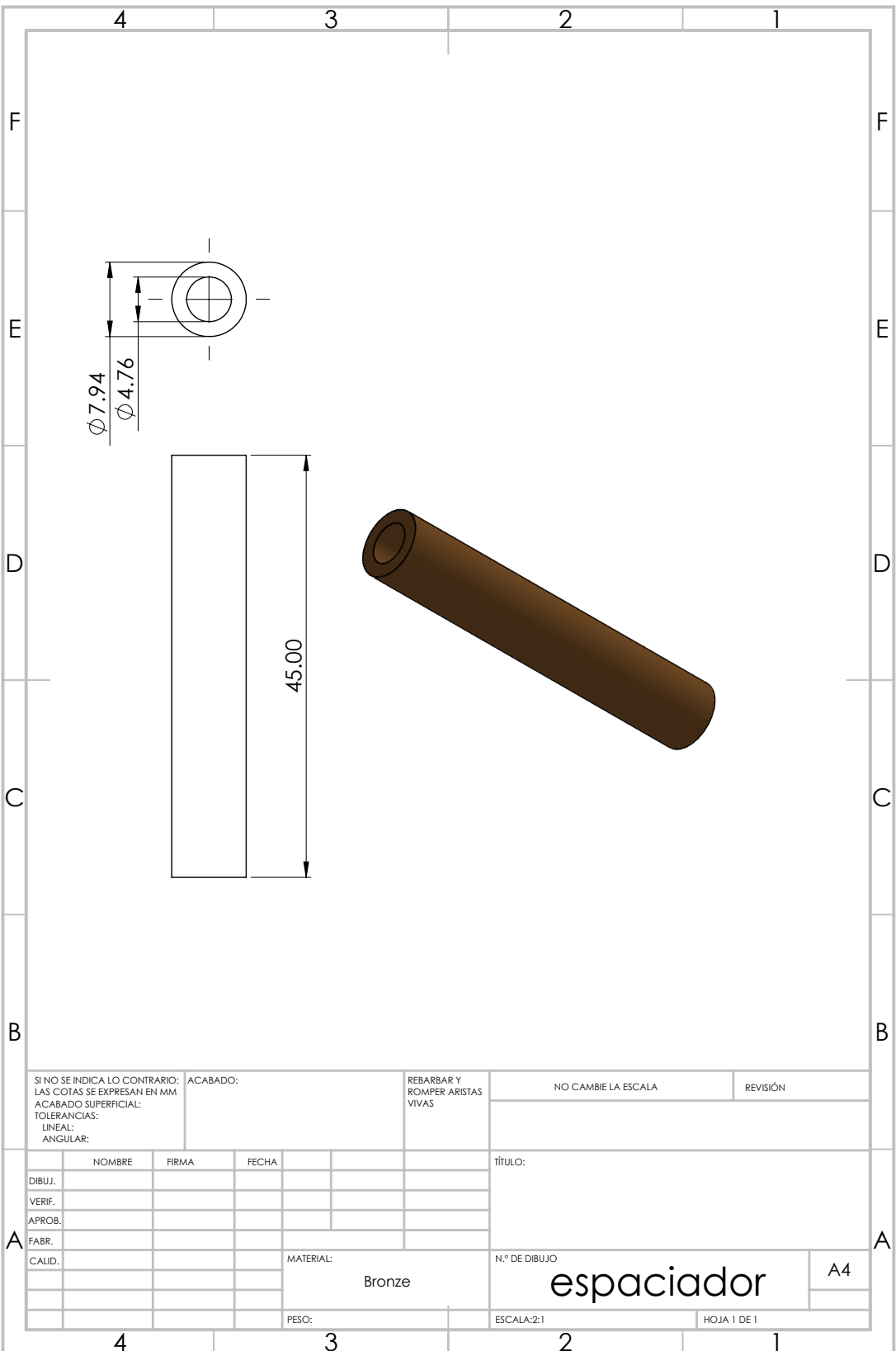
N.º DE DIBUJO
tapa_lateral A4

PESO:

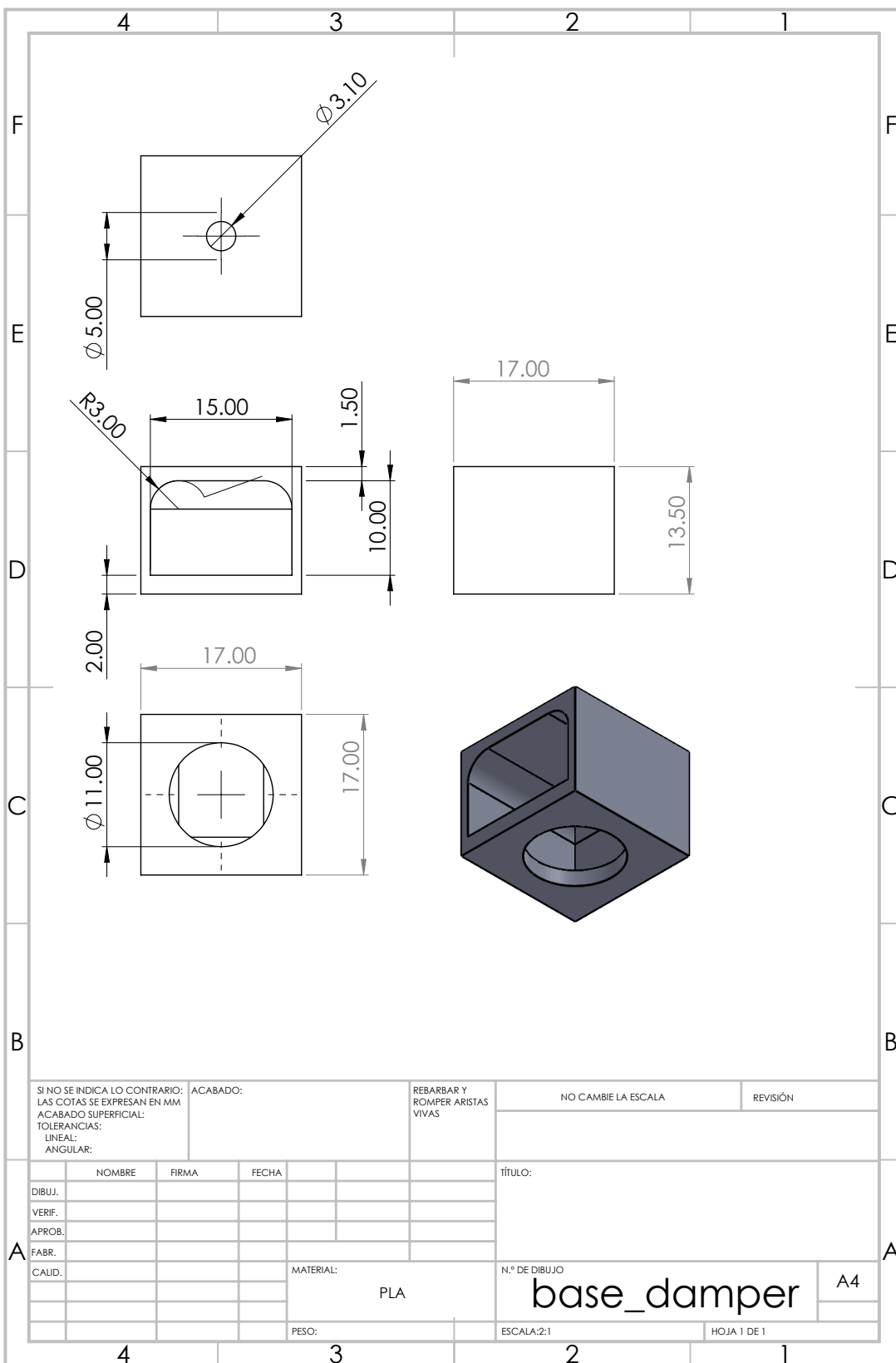
ESCALA:1:5 HOJA 1 DE 1



SI NO SE INDICA LO CONTRARIO: LAS COTAS SE EXPRESAN EN MM ACABADO SUPERFICIAL: TOLERANCIAS: LINEAL: ANGULAR:		ACABADO:	REBARBAR Y ROMPER ARISTAS VIVAS	NO CAMBIE LA ESCALA	REVISIÓN
NOMBRE	FIRMA	FECHA		TÍTULO:	
DIBUJ.					
VERIF.					
APROB.					
FABR.					
CALID.			MATERIAL: Hierro ductil	N.º DE DIBUJO eje_llanta	A4
			PESO:	ESCALA:1:2	HOJA 1 DE 1



SI NO SE INDICA LO CONTRARIO: LAS COTAS SE EXPRESAN EN MM ACABADO SUPERFICIAL: TOLERANCIAS: LINEAL: ANGULAR:		ACABADO:	REBARBAR Y ROMPER ARISTAS VIVAS	NO CAMBIE LA ESCALA	REVISIÓN
DIBUJ.	NOMBRE	FIRMA	FECHA	TÍTULO:	
VERIF.					
APROB.					
FABR.					
CALID.			MATERIAL: Bronze	N.º DE DIBUJO espaciador	A4
			PESO:	ESCALA:2:1	HOJA 1 DE 1



SI NO SE INDICA LO CONTRARIO:
 LAS COTAS SE EXPRESAN EN MM
 ACABADO SUPERFICIAL:
 TOLERANCIAS:
 LINEAL:
 ANGULAR:

ACABADO:

REBARBAR Y
 ROMPER ARISTAS
 VIVAS

NO CAMBIE LA ESCALA

REVISIÓN

	NOMBRE	FIRMA	FECHA	
DIBUJ.				
VERIF.				
APROB.				
FABR.				
CALID.				

TÍTULO:

MATERIAL:
 PLA

N.º DE DIBUJO

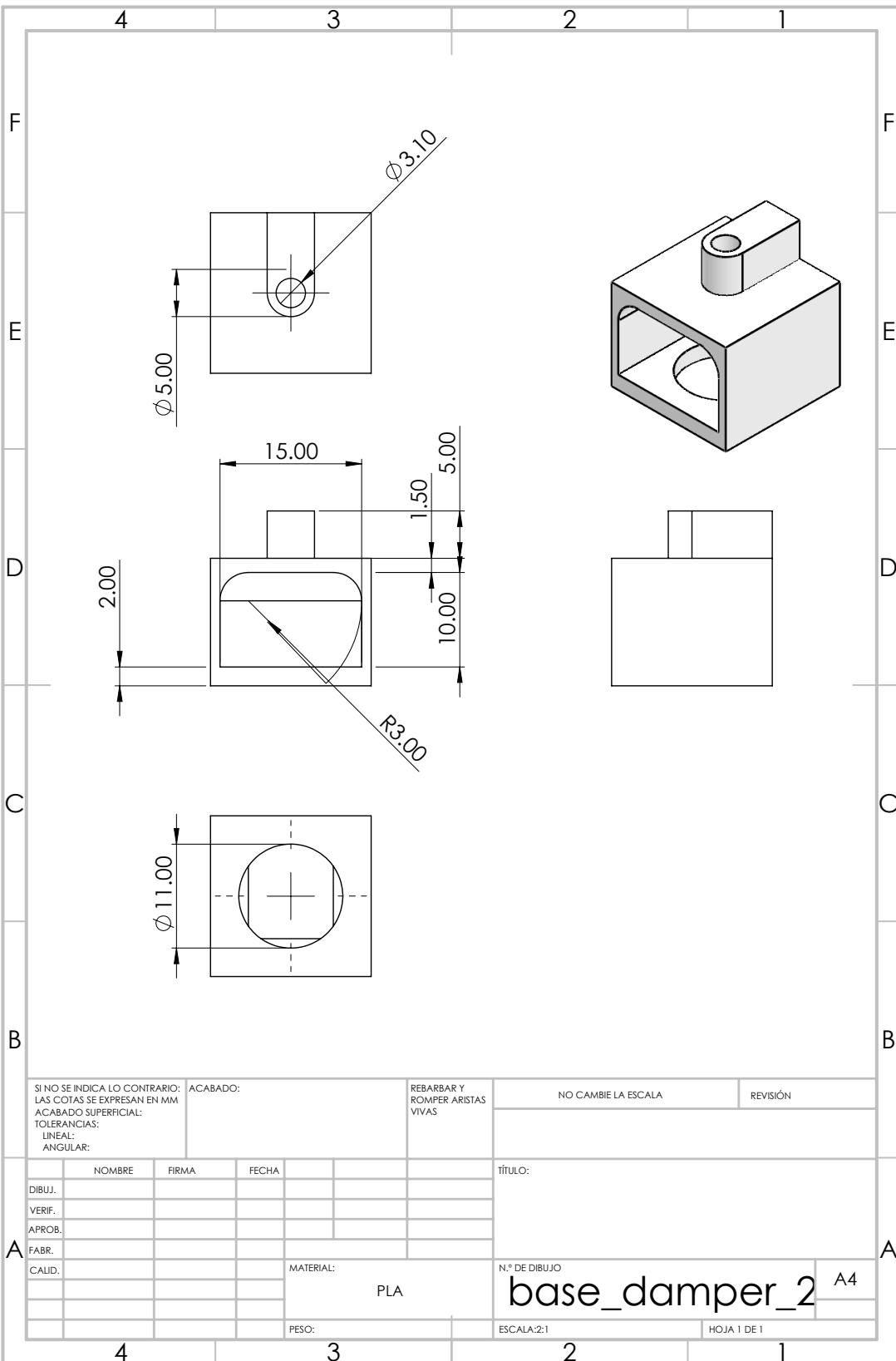
base_damper

A4

PESO:

ESCALA:2:1

HOJA 1 DE 1



SI NO SE INDICA LO CONTRARIO:
 LAS COTAS SE EXPRESAN EN MM
 ACABADO SUPERFICIAL:
 TOLERANCIAS:
 LINEAL:
 ANGULAR:

ACABADO:

REBARBAR Y
 ROMPER ARISTAS
 VIVAS

NO CAMBIE LA ESCALA

REVISIÓN

	NOMBRE	FIRMA	FECHA	
DIBUJ.				
VERIF.				
APROB.				
FABR.				
CALID.				

TÍTULO:	
N° DE DIBUJO	
base_damper_2 A4	
PESO:	ESCALA:2:1
HOJA 1 DE 1	

MATERIAL:
 PLA

Apéndice B

Especificaciones técnicas de la instrumentación seleccionada

Electrical Characteristics (unless otherwise specified, $T_a = 25^\circ\text{C}$, $V_{CC} = 3\text{ V}$, $V_M = 5\text{ V}$)

Characteristics	Symbol	Test Condition	Min	Typ.	Max	Unit	
Supply current	$I_{CC}(3V)$	STBY = $V_{CC} = 3\text{ V}$, $V_M = 5\text{ V}$	—	1.1	1.8	mA	
	$I_{CC}(5.5V)$	STBY = $V_{CC} = 5.5\text{ V}$, $V_M = 5\text{ V}$	—	1.5	2.2		
	$I_{CC}(\text{STB})$	STBY = 0 V	—	—	1	μA	
	$I_M(\text{STB})$		—	—	1		
Control input voltage	V_{IH}		$V_{CC} \times 0.7$	—	$V_{CC} + 0.2$	V	
	V_{IL}		-0.2	—	$V_{CC} \times 0.3$		
Control input current	I_{IH}	$V_{IN} = 3\text{ V}$	5	15	25	μA	
	I_{IL}	$V_{IN} = 0\text{ V}$	—	—	1		
Standby input voltage	$V_{IH}(\text{STB})$		$V_{CC} \times 0.7$	—	$V_{CC} + 0.2$	V	
	$V_{IL}(\text{STB})$		-0.2	—	$V_{CC} \times 0.3$		
Standby input current	$I_{IH}(\text{STB})$	$V_{IN} = 3\text{ V}$	5	15	25	μA	
	$I_{IL}(\text{STB})$	$V_{IN} = 0\text{ V}$	—	—	1		
Output saturating voltage	$V_{\text{sat}}(\text{U+L})1$	$I_O = 1\text{ A}$, $V_{CC} = V_M = 5\text{ V}$	—	0.5	0.7	V	
	$V_{\text{sat}}(\text{U+L})2$	$I_O = 0.3\text{ A}$, $V_{CC} = V_M = 5\text{ V}$	—	0.15	0.21		
Output leakage current	$I_L(\text{U})$	$V_M = V_{\text{OUT}} = 15\text{ V}$	—	—	1	μA	
	$I_L(\text{L})$	$V_M = 15\text{ V}$, $V_{\text{OUT}} = 0\text{ V}$	-1	—	—		
Regenerative diode VF	$V_F(\text{U})$	$I_F = 1\text{ A}$	—	1	1.1	V	
	$V_F(\text{L})$		—	1	1.1		
Low voltage detecting voltage	UVLD	(Designed value)	—	1.9	—	V	
Recovering voltage	UVLC		—	2.2	—		
Response speed	t_r	(Designed value)	—	24	—	ns	
	t_f		—	41	—		
	Dead time	H to L	Penetration protect time (Designed value)	—	50		—
		L to H		—	230		—
Thermal shutdown circuit operating temperature	TSD	(Designed value)	—	175	—	$^\circ\text{C}$	
Thermal shutdown hysteresis	ΔTSD		—	20	—		

Figura B.1: Características eléctricas del controlador TB6612FNG [32].

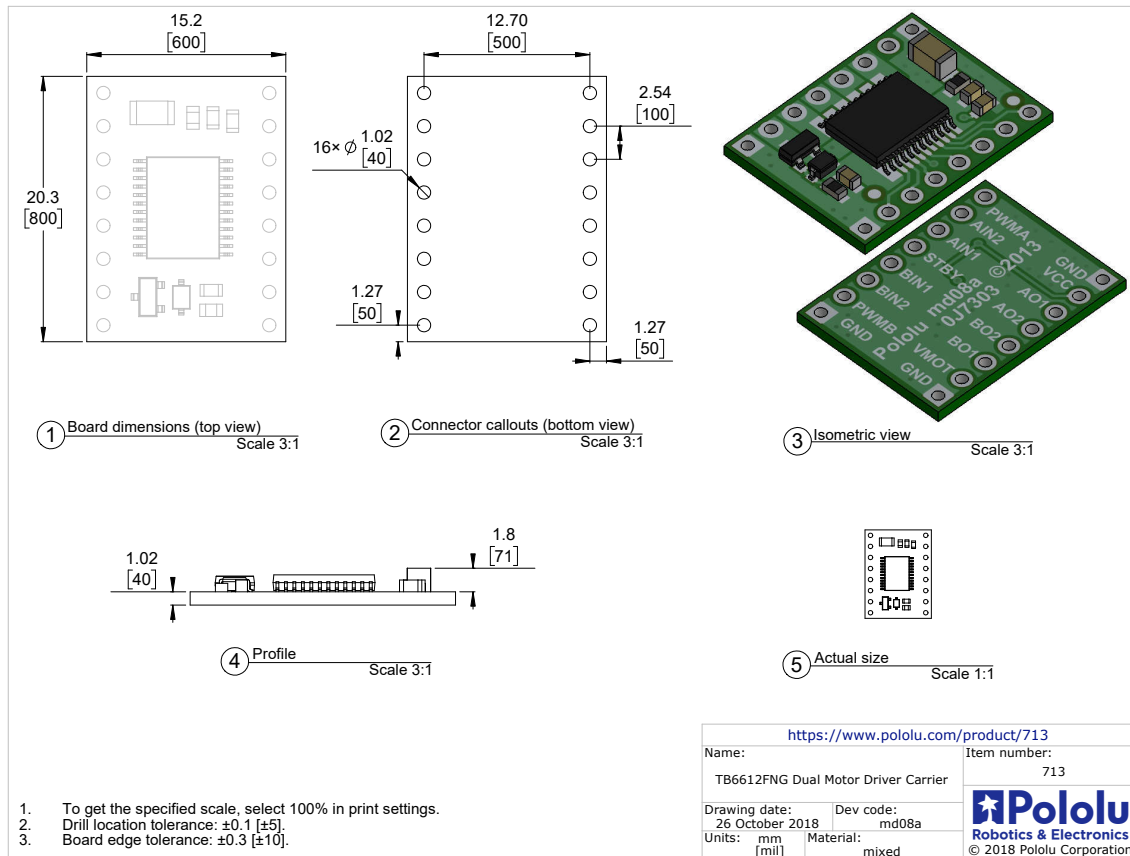


Figura B.2: Dimensiones del controlador TB6612FNG [32].

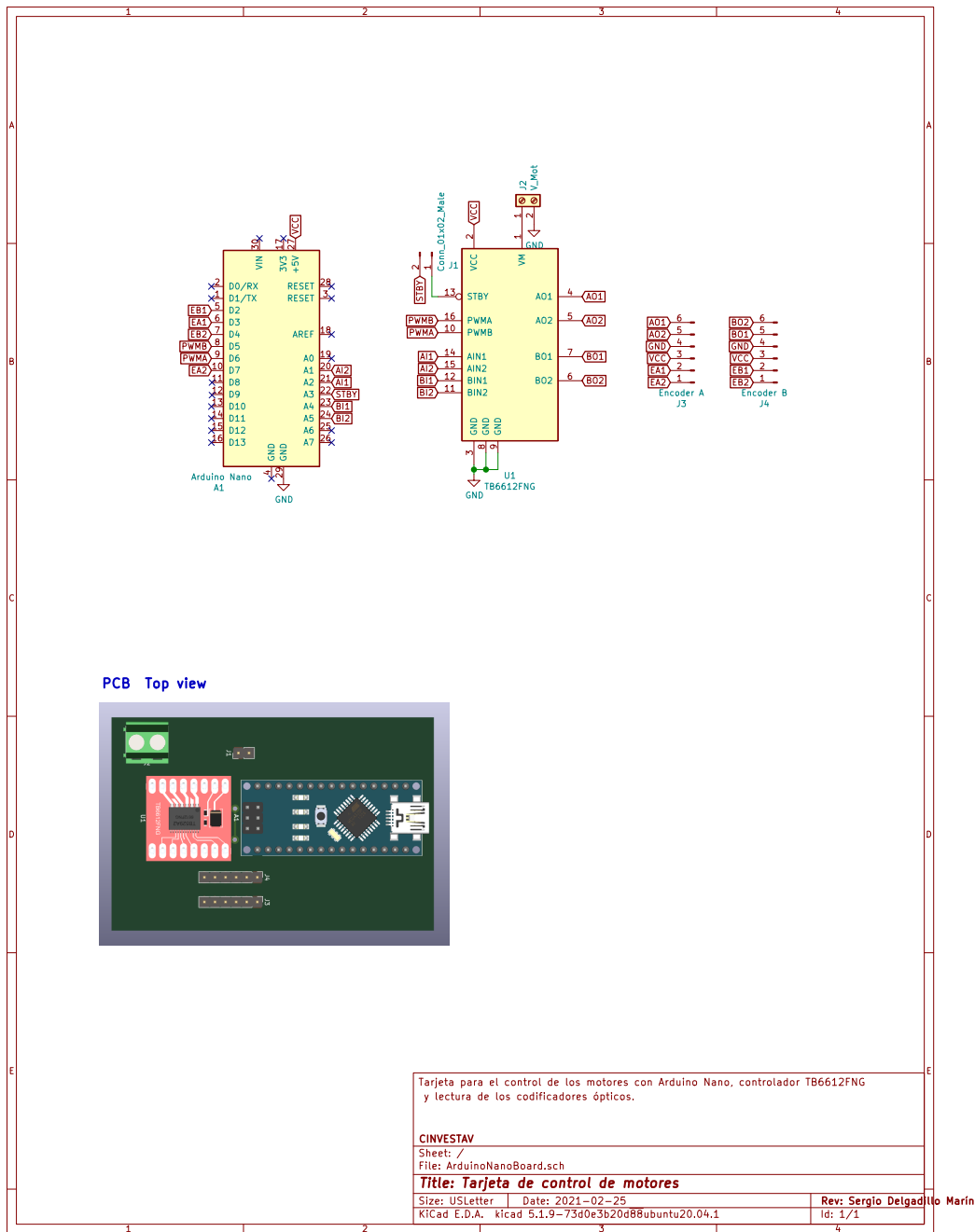


Figura B.3: Diagrama esquemático de la tarjeta de control de motores.

Parameter	Camera Sensor Properties
Active Pixels	848 X 800
Sensor Aspect Ratio	1.06
Format	8bit, 10-bit RAW
Filter Type	IR Cut Filter
Focus	Fixed
Shutter Type	Global Shutter
Signal Interface	MIPI CSI-2, 2 X Lanes

Figura B.4: Especificaciones de los sensores de imagen de la cámara seguimiento Intel RealSense T265 [29].

Parameter	Properties
Degrees of Freedom	6
Acceleration Range	±4g
Accelerometer Sample Rate	62.5Hz
Gyroscope Range	±2000 Deg/s
Gyroscope Sample Rate	200Hz

Figura B.5: Especificaciones de la unidad de medición inercial de la cámara seguimiento Intel RealSense T265 [29].

Parameter		Min	Nom	Max	Unit
VCC	Supply Voltage	4.5	5	5.25	V
ICC	Supply Current		300	300	mA

Figura B.6: Requerimientos energéticos de la cámara seguimiento Intel RealSense T265 [29].

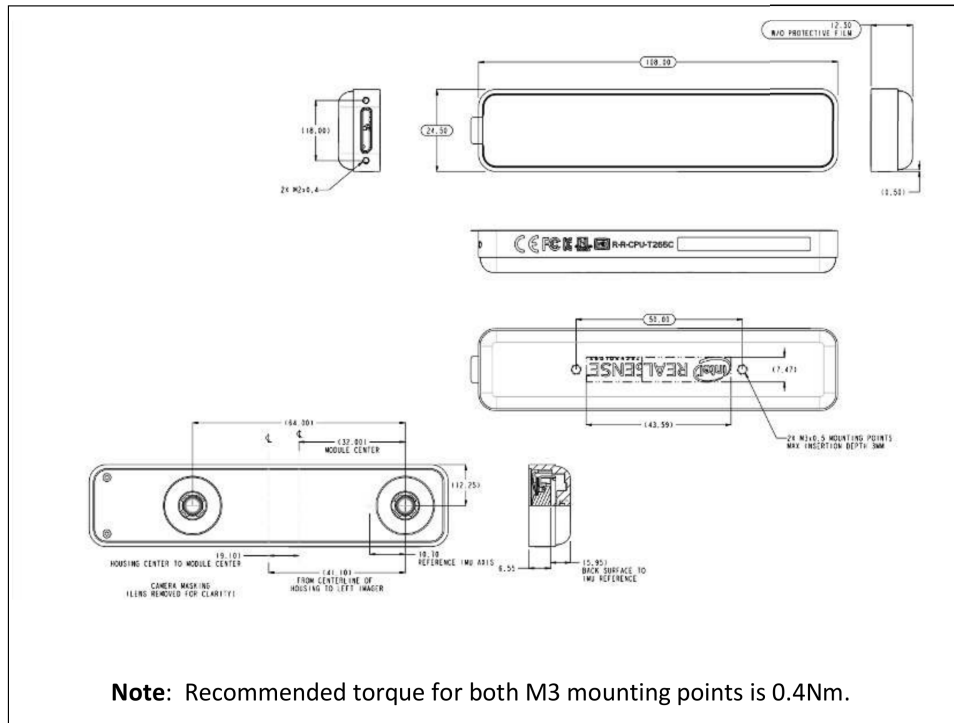


Figura B.7: Dimensiones de la cámara de seguimiento Intel RealSense T265 [29].

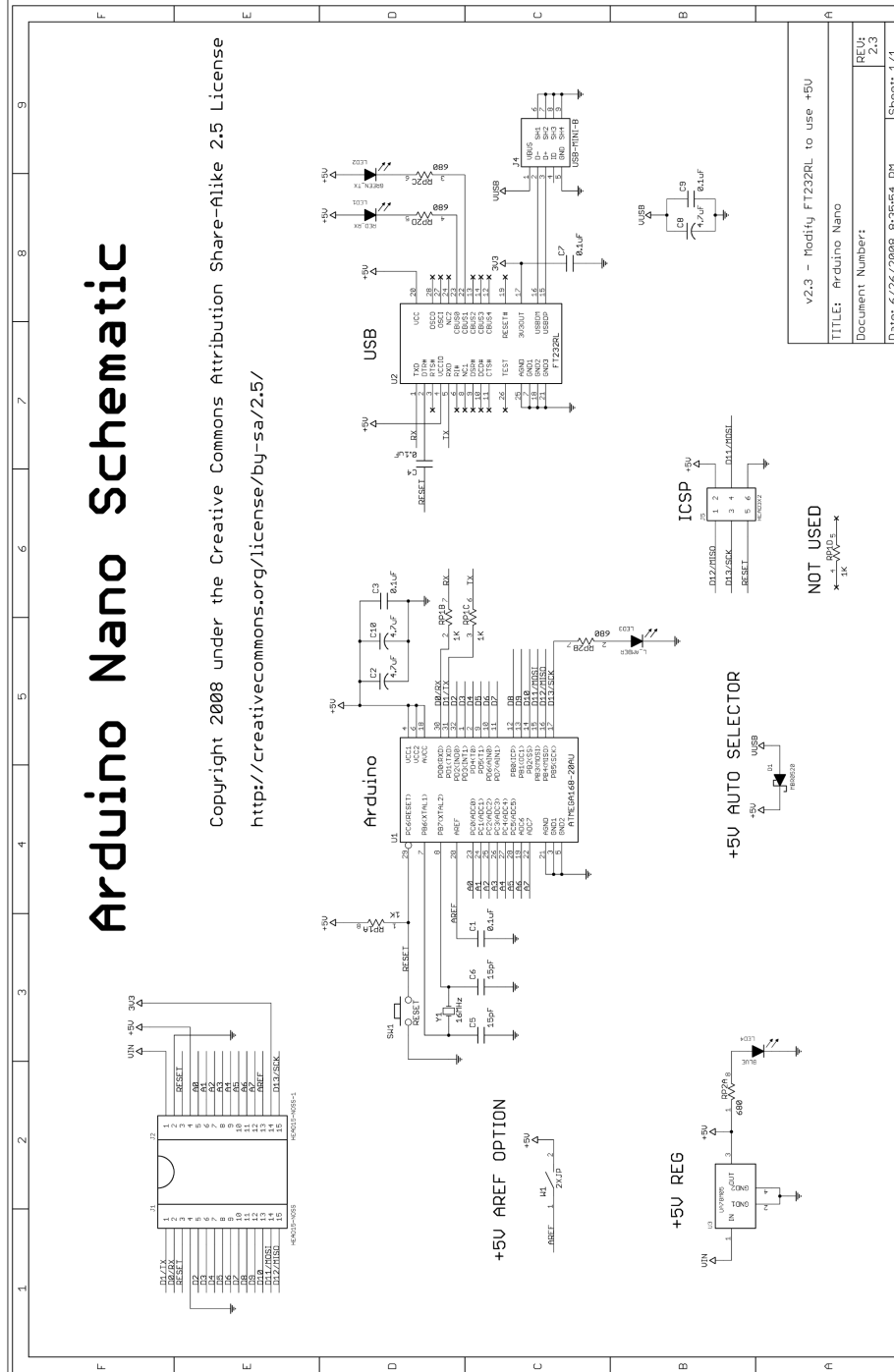


Figura B.8: Diagrama esquemático de la tarjeta Arduino Nano [33].

Microcontroller	ATmega328
Architecture	AVR
Operating Voltage	5 V
Flash Memory	32 KB of which 2 KB used by bootloader
SRAM	2 KB
Clock Speed	16 MHz
Analog IN Pins	8
EEPROM	1 KB
DC Current per I/O Pins	40 mA (I/O Pins)
Input Voltage	7-12 V
Digital I/O Pins	22 (6 of which are PWM)
PWM Output	6
Power Consumption	19 mA
PCB Size	18 x 45 mm
Weight	7 g
Product Code	A000005

Figura B.9: Especificaciones técnicas de la tarjeta Arduino Nano [33].

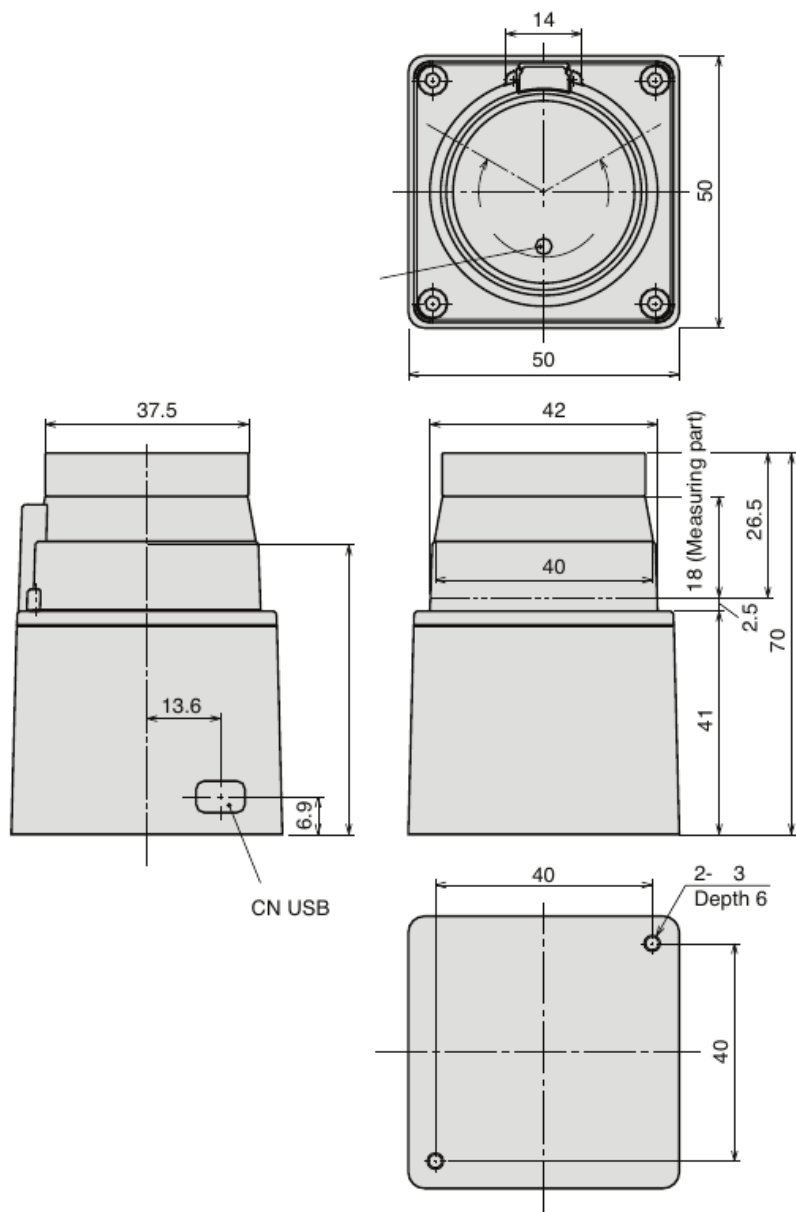


Figura B.10: Dimensiones del sensor Lidar URG-04LX-UG01 [34].

Product Name	Scanning Laser Range Finder
Model	URG-04LX-UG01
Light source	Semiconductor laser diode ($\lambda=785\text{nm}$), Laser safety Class 1 (21 CFR part 1040.10 and 1040.11) Laser power: 0.8mW or less (Class 1 compliant by scanning)
Power source	5V DC $\pm 5\%$ (Supplied by USB bus power)
Current consumption	500mA or less (Rush current 800mA)
Detection distance and standard object	Accuracy:60-4,095mm (white paper 70mm x 70mm or bigger)* Detectable range: 20-5,600mm
Accuracy	Described in the data sheet attached to each unit Guaranteed accuracy: 0.06-1m: $\pm 30\text{mm}^*$, 1-4m: 3% of the detected distance* (Standard object: white paper 70mm x 70mm)
Resolution	1 mm
Scan Angle	240°
Angular Resolution	Approx.0.36° (360°/1024)
Scan Time	100msec/scan
Interface	USB Version 2.0 FS mode (12Mbps) SCIP2.0
Ambient (Temperature/Humidity)	-10 ~ 50°C / 85% or less (without dew and frost)
Preservation temperature	-25 ~ 75°C
Ambient Light Resistance	10000Lx or less (Sunlight)
Vibration Resistance	Double amplitude 1.5mm 10 ~ 55Hz, 2 hours each in X, Y and Z direction, and 98m/s ² 55Hz ~ 150Hz in 2 minutes sweep, 1 hours each in X, Y and Z direction
Impact Resistance	196 m/s ² , 10 times each in X, Y and Z direction
Protective Structure	Optics : IP64 Case : IP40
Insulation Resistance	10M Ω for DC 500Vmegger
Weight	Approx. 160 g
Case	Polycarbonate
External dimension (W×D×H)	50×50×70mm (Reference design sheet No. C-40-3362)

*Under standard test conditions with white Kent sheet 70mm×70mm

Figura B.11: Especificaciones técnicas del sensor Lidar URG-04LX-UG01 [34].

Maxwell GPU

128-core GPU | End-to-end lossless compression | Tile Caching | OpenGL[®] 4.6 | OpenGL ES 3.2 | Vulkan[™] 1.1 | CUDA[®] | OpenGL ES Shader Performance (up to): 512 GFLOPS (FP16) | Maximum Operating Frequency: 921MHz

CPU

ARM[®] Cortex[®]-A57 MPCore (Quad-Core) Processor with NEON Technology | L1 Cache: 48KB L1 instruction cache (I-cache) per core; 32KB L1 data cache (D-cache) per core | L2 Unified Cache: 2MB | Maximum Operating Frequency: 1.43GHz

Audio

Industry standard High Definition Audio (HDA) controller provides a multichannel audio path to the HDMI interface.

Memory

Dual Channel | System MMU | Memory Type: 4ch x 16-bit LPDDR4 | Maximum Memory Bus Frequency: 1600MHz | Peak Bandwidth: 25.6 GB/s | Memory Capacity: 4GB

Storage

eMMC 5.1 Flash Storage | Bus Width: 8-bit | Maximum Bus Frequency: 200MHz (HS400) | Storage Capacity: 16GB

Boot Sources

eMMC and USB (recovery mode)

Networking

10/100/1000 BASE-T Ethernet | Media Access Controller (MAC)

Imaging

Dedicated RAW to YUV processing engines process up to 1400Mpix/s (up to 24MP sensor) | MIPI CSI 2.0 up to 1.5Gbps (per lane) | Support for x4 and x2 configurations (up to four active streams).

Operating Requirements

Temperature Range (T_j): -25 – 97C* | Module Power: 5 – 10W | Power Input: 5.0V

Display Controller

Two independent display controllers support DSI, HDMI, DP, eDP: MIPI-DSI (1.5Gbps/lane): Single x2 lane | Maximum Resolution: 1920x960 at 60Hz (up to 24bpp) | HDMI 2.0a/b (up to 6Gbps) | DP 1.2a (HBR2 5.4 Gbps) | eDP 1.4 (HBR2 5.4Gbps) | Maximum Resolution (DP/eDP/HDMI): 3840 x 2160 at 60Hz (up to 24bpp)

Clocks

System clock: 38.4MHz | Sleep clock: 32.768kHz | Dynamic clock scaling and clock source selection

Multi-Stream HD Video and JPEG**Video Decode**

H.265 (Main, Main 10): 2160p 60fps | 1080p 240fps
H.264 (BP/MP/HP/Stereo SEI half-res): 2160p 60fps | 1080p 240fps
H.264 (MVC Stereo per view): 2160p 30fps | 1080p 120fps
VP9 (Profile 0, 8-bit): 2160p 60fps | 1080p 240fps
VP8: 2160p 60fps | 1080p 240fps
VC-1 (Simple, Main, Advanced): 1080p 120fps | 1080i 240fps
MPEG-2 (Main): 2160p 60fps | 1080p 240fps | 1080i 240fps

Video Encode

H.265: 2160p 30fps | 1080p 120fps
H.264 (BP/MP/HP): 2160p 30fps | 1080p 120fps
H.264 (MVC Stereo per view): 1440p 30fps | 1080p 60fps
VP8: 2160p 30fps | 1080p 120fps

JPEG (Decode and Encode): 600 MP/s

Peripheral Interfaces

xHCI host controller with integrated PHY: 1 x USB 3.0, 3 x USB 2.0 | USB 3.0 device controller with integrated PHY | EHCI controller with embedded hub for USB 2.0 | 4-lane PCIe: one x1/2/4 controller | single SD/MMC controller (supporting SDIO 4.0, SD HOST 4.0) | 3 x UART | 2 x SPI | 4 x I2C | 2 x I2S: support I2S, RJM, LJM, PCM, TDM (multi-slot mode) | GPIOs

Mechanical

Module Size: 69.6 mm x 45 mm | PCB: 8L HDI | Connector: 260 pin SO-DIMM

Figura B.12: Especificaciones técnicas de la tarjeta Nvidia Jetson Nano [30].

Bibliografía

- [1] Zeashan Hameed Khan, Afifa Siddique, and Chang Won Lee. Robotics utilization for healthcare digitization in global covid-19 management. *International Journal of Environmental Research and Public Health*, 17(11):3819, 2020.
- [2] José Rafael García-Sánchez, Salvador Tavera-Mosqueda, Ramón Silva-Ortigoza, Victor Manuel Hernández-Guzmán, Jacobo Sandoval-Gutiérrez, Mariana Marcelino-Aranda, Hind Taud, and Magdalena Marciano-Melchor. Robust switched tracking control for wheeled mobile robots considering the actuators and drivers. *Sensors*, 18(12):4316, 2018.
- [3] M Luo, D Wang, M Pham, CB Low, JB Zhang, DH Zhang, and YZ Zhao. Model-based fault diagnosis/prognosis for wheeled mobile robots: a review. In *31st Annual Conference of IEEE Industrial Electronics Society, 2005. IECON 2005.*, pages 6–pp. IEEE, 2005.
- [4] Frida-Gail Rojas-Contreras, Alberto-Ismael Castillo-Lopez, Leonid Fridman, and Victor-Javier Gonzalez-Villela. Trajectory tracking using continuous sliding mode algorithms for differential drive robots. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 6027–6032. IEEE, 2017.
- [5] Parisa Yazdjerdi and Nader Meskin. Fault tolerant control of differential drive mobile robots using sliding mode controller. In *2017 5th International Conference on Control, Instrumentation, and Automation (ICCIA)*, pages 270–274. IEEE, 2017.
- [6] Yasmine Koubaa, Mohamed Boukattaya, and Tarak Damak. Adaptive sliding mode

- control for trajectory tracking of nonholonomic mobile robot with uncertain kinematics and dynamics. *Applied Artificial Intelligence*, 32(9-10):924–938, 2018.
- [7] Ming Yue, Junjie Gao, Teng Ma, and Jun Chen. Finite-time sliding mode control for trajectory tracking of wmrs with wheel-soil friction disturbance adaptability. *Applied Artificial Intelligence*, 31(1):80–101, 2017.
- [8] Alexandr Stefek, Thuan Van Pham, Vaclav Krivanek, and Khac Lam Pham. Energy comparison of controllers used for a differential drive wheeled mobile robot. *IEEE Access*, 8:170915–170927, 2020.
- [9] Yi-Xiang Wang and Ching-Lung Chang. Ros-base multi-sensor fusion for accuracy positioning and slam system. In *2020 International Symposium on Community-centric Systems (CcS)*, pages 1–6. IEEE, 2020.
- [10] Sumegh Pramod Thale, Mihir Mangesh Prabhu, Pranjali Vinod Thakur, and Pratik Kadam. Ros based slam implementation for autonomous navigation using turtlebot. In *ITM Web of Conferences*, volume 32, page 01011. EDP Sciences, 2020.
- [11] Riccardo Giubilato, Sebastiano Chiodini, Marco Pertile, and Stefano Debei. An evaluation of ros-compatible stereo visual slam methods on a nvidia jetson tx2. *Measurement*, 140:161–170, 2019.
- [12] Zhaojun Meng, Chuang Wang, Zhao Han, and Zirong Ma. Research on slam navigation of wheeled mobile robot based on ros. In *2020 5th International Conference on Automation, Control and Robotics Engineering (CACRE)*, pages 110–116. IEEE, 2020.
- [13] P Sankalprajan, Thri Lochan Sharma, Hamsa Datta Perur, and Prithvi Sekhar Pagala. Comparative analysis of ros based 2d and 3d slam algorithms for autonomous ground vehicles. In *2020 International Conference for Emerging Technology (INCET)*, pages 1–6. IEEE, 2020.
- [14] Maria Pia Fanti, Agostino Marcello Mangini, Michele Roccotelli, and Bartolomeo Silvestri. Hospital drugs distribution with autonomous robot vehicles. In *2020 IEEE*

- 16th International Conference on Automation Science and Engineering (CASE)*, pages 1025–1030. IEEE, 2020.
- [15] Abu Ubaidah bin Shamsudin, Chia Kim Seng, Hisyam bin Abdul Rahman, Dirman Hanafi, and Hazwaj bin Mhd Poad. Development of 7 dof robotic arm with electronic bluetooth stethoscope for covid patient medical assesment. In *2020 IEEE Student Conference on Research and Development (SCORED)*, pages 1–5. IEEE, 2020.
- [16] Francesco Capezio, Fulvio Mastrogiovanni, Antonello Scalmato, Antonio Sgorbissa, Paolo Vernazza, Tullio Vernazza, and Renato Zaccaria. Mobile robots in hospital environments: an installation case study. In *ECMR*, pages 61–68, 2011.
- [17] John M Evans. Helpmate: An autonomous mobile robot courier for hospitals. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'94)*, volume 3, pages 1695–1700. IEEE, 1994.
- [18] Masaki Takahashi, Takafumi Suzuki, Francesco Cinquegrani, Rosario Sorbello, and Enrico Pagello. A mobile robot for transport applications in hospital domain with safe human detection algorithm. In *2009 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 1543–1548. IEEE, 2009.
- [19] Carlos Antonio Acosta Calderon and Asadollah Norouzi. Towards design and development of autonomous mobile platforms for delivery and collection. In *2015 10th International Conference on Information, Communications and Signal Processing (ICICS)*, pages 1–6. IEEE, 2015.
- [20] Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, and Giuseppe Oriolo. *Robotics: modelling, planning and control*. Springer Science & Business Media, 2010.
- [21] Farbod Fahimi. Autonomous robots. *Modeling, Path Planning and Control*, 2009.
- [22] J Álvarez-Gallegos, R Castro-Linares, and R García-Saldívar. Modelado dinámico y control de un robot móvil con distribución de masa no uniforme. In *Memorias del XVI Congreso Latinoamericano de Control Automático, CLCA*, pages 457–462, 2014.

- [23] R Rosenberg. *Analytical dynamics*. Springer, 1977.
- [24] Sebastian Thrun. Probabilistic robotics. *Communications of the ACM*, 45(3):52–57, 2002.
- [25] Zhan Wang, Shoudong Huang, and Gamini Dissanayake. *Simultaneous Localization and Mapping: Exactly Sparse Information Filters*, volume 3. World Scientific, 2011.
- [26] Lentin Joseph and Jonathan Cacace. *Mastering ROS for Robotics Programming: Design, build, and simulate complex robots using the Robot Operating System*. Packt Publishing Ltd, 2018.
- [27] Aaron Martinez and Enrique Fernández. *Learning ROS for robotics programming*. Packt Publishing Ltd, 2013.
- [28] Spyros G Tzafestas. *Introduction to mobile robot control*. Elsevier, 2013.
- [29] Varios Autores. Intel realsense tracking module datasheet. https://www.intelrealsense.com/wp-content/uploads/2019/09/Intel_RealSense_Tracking_Camera_Datasheet_Rev004_release.pdf, 09/2019. Accedido 01-10-2020.
- [30] Varios Autores. Data sheet nvidia jetson nano system-on-module. <https://developer.nvidia.com/embedded/dlc/jetson-nano-system-module-datasheet>, 2014. Accedido 23-04-2021.
- [31] Arie Levant. Robust exact differentiation via sliding mode technique. *automatica*, 34(3):379–384, 1998.
- [32] Pololu Robotics and Electronics. Tb6612fng dual motor driver carrier. <https://www.pololu.com/product/713/specs>, 10/2018. Accedido 16-01-2021.
- [33] Varios Autores. Arduino nano (v2.3) user manual. <https://www.arduino.cc/en/uploads/Main/ArduinoNanoManual23.pdf>, 06/2008. Accedido 16-01-2021.
- [34] Varios Autores. Scanning laser range finder urg-04lx specifications. https://hokuyo-usa.com/application/files/5115/8947/8197/URG-04LX-UG01_Specifications_Catalog.pdf, 10/2005. Accedido 23-04-2021.