

**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE CUENCA**

**CARRERA DE ELECTRÓNICA Y
AUTOMATIZACIÓN**

*Trabajo de titulación previo
a la obtención del título de
Ingeniero en Electrónica*

**PROYECTO TÉCNICO CON ENFOQUE INVESTIGATIVO:
"DISEÑO, CONSTRUCCIÓN Y PROGRAMACIÓN DE UN
PROTOTIPO DE ROBOT MÓVIL PARA APLICACIONES DE
TELEPRESENCIA "**

AUTORES:

MATEO RAFAEL MENDIETA MOLINA

PABLO ANDRÉS QUICHIMBO PLAZA

TUTOR:

ING. FREDY MARCELO RIVERA CALLE, Ph.D.

CO-TUTOR:

ING. JULIO CÉSAR MONTESDEOCA CONTRERAS, Ph.D.

CUENCA – ECUADOR

2022

CESIÓN DE DERECHOS DE AUTOR

Nosotros, Mateo Rafael Mendieta Molina con documento de identificación N° 1400679765 y Pablo Andrés Quichimbo Plaza con documento de identificación N° 0106709116, manifestamos nuestra voluntad y cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de titulación: **“DISEÑO, CONSTRUCCIÓN Y PROGRAMACIÓN DE UN PROTOTIPO DE ROBOT MÓVIL PARA APLICACIONES DE TELEPRESENCIA”**, mismo que ha sido desarrollado para optar por el título de: *Ingeniero en Electrónica*, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en nuestra condición de autores nos reservamos los derechos morales de la obra antes citada. En concordancia, suscribimos este documento en el momento que hago entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Cuenca, febrero de 2022.

Mateo Mendieta



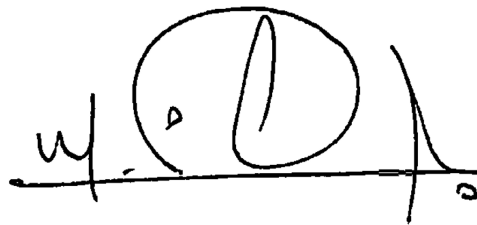
Mateo Rafael Mendieta Molina
C.I 1400679765

Pablo Andrés Quichimbo Plaza
C.I 0106709116

CERTIFICACIÓN

Yo, declaro que bajo mi tutoría, fue desarrollado el trabajo de titulación: “**DISEÑO, CONSTRUCCIÓN Y PROGRAMACIÓN DE UN PROTOTIPO DE ROBOT MÓVIL PARA APLICACIONES DE TELEPRESENCIA**”, realizado por Mateo Rafael Mendieta Molina y Pablo Andrés Quichimbo Plaza, obteniendo el *Proyecto Técnico con enfoque investigativo* que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana.

Cuenca, febrero de 2022.

A handwritten signature in black ink, consisting of a stylized 'F' followed by 'M. R. C.' and a large, circular flourish.

Ing. Fredy Marcelo Rivera Calle, Ph.D.
C.I. 0102151750

DECLARATORIA DE RESPONSABILIDAD

Nosotros Mateo Rafael Mendieta Molina con documento de identificación N° 1400679765 y Pablo Andrés Quichimbo Plaza con documento de identificación N° 0106709116, autores del trabajo de titulación: “**DISEÑO, CONSTRUCCIÓN Y PROGRAMACIÓN DE UN PROTOTIPO DE ROBOT MÓVIL PARA APLICACIONES DE TELEPRESENCIA**”, certificamos que el total contenido del *Proyecto Técnico con enfoque investigativo*, es de nuestra exclusiva responsabilidad y autoría.

Cuenca, febrero de 2022.

Mateo Mendieta



Mateo Rafael Mendieta Molina
C.I 1400679765

Pablo Andrés Quichimbo Plaza
C.I 0106709116

DEDICATORIA

Dedico este trabajo de titulación a mi madre que con su completo apoyo me permitió emprender este camino que hoy culmino, gracias por todo el apoyo incondicional y por ser un pilar fundamental en mi vida. A mis hermanos por siempre acompañarme en cada etapa de mi vida, estando ahí siempre sin importar lo que pase.

Mateo Mendieta

Este trabajo de titulación está dedicado a mi familia, especialmente a mis queridos padres Cesar y Sandra, quienes siempre fueron el pilar fundamental de mi vida, brindándome su fuerza y apoyo en la toda la trayectoria de mi carrera profesional.

Pablo Quichimbo

AGRADECIMIENTOS

Agradezco a mi madre Rosario Molina quien siempre estuvo presente brindándome su apoyo incondicional en todas las etapas de mi vida. A mis profesores y directores de tesis Fredy Rivera Y Julio Montesdeoca, ya que sin su ayuda y guía este trabajo de titulación no hubiese sido posible.

Mateo Mendieta

En primer lugar, agradezco a Dios por haberme brindado la vida para poder culminar mis estudios, a mi familia por el apoyo incondicional. A mis profesores Freddy Rivera y Julio Montesdeoca sin ellos este trabajo no hubiese sido posible y a la Universidad Politécnica Salesiana por ser mi alma mater en este proceso de formación.

Pablo Quichimbo

Índice

CESIÓN DE DERECHOS DE AUTOR	1
CERTIFICACIÓN	1
DECLARATORIA DE RESPONSABILIDAD	1
DEDICATORIA	1
AGRADECIMIENTOS	1
Resumen	7
Abstract	8
1. CAPÍTULO 1: INTRODUCCIÓN A LA ROBÓTICA MÓVIL	5
1.1. Robótica Móvil	5
1.2. Tipos de Robots Móviles	5
1.2.1. Robots móviles con ruedas	5
1.2.2. Robots móviles andantes o con patas	7
1.2.3. Robots con seguimiento	8
1.2.4. Robots híbridos	8
1.3. Telepresencia	9
1.3.1. Ventajas de la telepresencia	9
1.4. Robótica móvil en la telepresencia	10
2. CAPÍTULO 2: DISEÑO Y CONSTRUCCIÓN DEL PROTOTIPO DEL ROBOT	11
2.1. Diseño de la estructura del Robot	11
2.1.1. Accionamiento Diferencial	12
2.1.2. Sensores	12
2.1.3. Interfaz Humano - Robot	13
2.1.4. Diseño de la base interior del robot	14
2.1.5. Diseño de la base superior del robot	16
2.1.6. Construcción de la estructura del robot	20

2.1.7. Diseño de la interfaz del robot	24
3. CAPÍTULO 3: IMPLEMENTACIÓN Y PROGRAMACIÓN DEL ROBOT	29
3.1. Robot Operating System(ROS)	29
3.2. Nodos del robot	32
3.3. Nitrack	33
3.4. Modelo Matemático del motor	34
3.5. Diseño de PID	37
3.6. Control Digital PID	41
3.7. Modelo Cinemático	46
3.7.1. Controlador basado en Cinemática Inversa	47
3.8. Control de bajo nivel	48
3.8.1. Conversión de Velocidades	48
3.9. Modelado de la tarea	48
3.9.1. Objetivos de Control	49
3.9.2. Ley de control	50
3.10. Análisis de estabilidad	50
3.11. Simulaciones	50
4. CAPÍTULO 4: PRUEBAS Y ANÁLISIS DE RESULTADOS	51
4.1. Seguimiento de personas	51
4.2. Mapeo	55
4.3. Videollamada	56
4.4. Iluminación	58
5. CAPÍTULO 5: CONCLUSIONES Y RECOMENDACIONES	59
5.1. Conclusiones	59
6. Recomendaciones	60
7. Trabajo futuro	61
Referencias	62
APÉNDICES	63

Lista de Tablas

1.	Parámetros del Motor BLDC	36
2.	Parámetros del método Ziegler -Nichols (1)	39
3.	Parámetros obtenidos del controlado mediante Ziegler -Nichols	40
4.	Tabla de materiales utilizados	65

Lista de Figuras

3.	Robot Móvil con Ruedas	7
4.	Robot Móvil con Patas	8
5.	Robot Móvil de Seguimiento con Orugas	8
6.	Robot Móvil Híbrido	9
7.	Vista lateral del robot móvil para telepresencia	11
8.	Disposición de las ruedas del robot de tracción diferencial	12
9.	Sensores que equipa el robot. Lidar y Cámara de profundidad.	12
10.	Flujograma del funcionamiento del robot	14
11.	Diseño de la base y eje del robot	14
12.	Diseño de la base incorporado los demás elementos del robot	15
13.	Diseño de la tapa base inferior del robot	15
14.	Diseño de la tapa base inferior del robot integrada en la estructura	16
15.	Diseño del soporte para la pantalla del robot	16
16.	Diseño de los soportes acoplados a la pantalla y perfil del robot	17
17.	Tapa de la base superior del robot	17
18.	Caja del parlante del robot	18
19.	Diseño de la base superior del robot	18
20.	Diseño interior del robot	19
21.	Diseño final del prototipo de robot móvil para aplicaciones de telepresencia .	20
22.	Base inferior del robot	21
23.	Base inferior y tronco del robot	21
24.	Tapa inferior construida del robot	22
25.	Soportes acoplados a la pantalla y perfil del robot	22
26.	Estructura interior del robot	23
27.	Estructura Final del robot	24
28.	Diseño de la interfaz gráfica en el software de Qt Designer	25
29.	Activación de los sistemas de navegación	25
30.	Activación de software de telepresencia	26
31.	Activación de los sistemas de seguimiento	27
32.	Activación de la animación, iluminación y apagado del sistema	27
33.	GUI del robot para aplicaciones de telepresencia	28
34.	Desarrollo de la animación. Adobe After Effects	29

35.	Expresiones de la animación	29
36.	Iniciamos "Ros_Master"	30
37.	Conexión Raspberry con Arduino	31
38.	Publicación para encendido de la iluminación	32
39.	Gráfico de los nodos de la teleoperacion	32
40.	Velocidad Angular y Lineal del Robot	33
41.	Nodos presentes en el mapeo del entorno.	33
42.	Prueba de funcionamiento del SDK nitrack	34
43.	Gráfico de los nodos presentes en el seguimiento de personas	34
44.	Diagrama del circuito de devanado del estator.	35
45.	Diagrama de bloques de la planta.	37
46.	Respuesta al Escalón.	37
47.	Diagrama del control de velocidad para el motor BLDC.	38
48.	Sintonización de parámetros según Ziegler –Nichols.	38
49.	Parámetros de la respuesta al escalón de la planta.	39
50.	Diagrama de la planta + control PID.	40
51.	Respuesta del sistema en lazo cerrado con control PID (rojo) , respuesta del sistema en lazo cerrado sin control PID (azul).	41
52.	Conexión Arduino con drivers.	42
53.	Asignación de variables.	43
54.	Variables para la matriz de transformación.	43
55.	Configuración del Timer 4.	44
56.	Lectura de las velocidades angulares de cada una de las ruedas.	44
57.	Librería <i>PID_V2</i> y constante del controlador.	45
58.	Parámetros del controlador.	45
59.	Configuración de la dirección de las ruedas.	46
60.	Estructura usada para obtener el modelo cinemático del robot móvil.	47
61.	Modelado de la tarea de seguimiento de personas.	49
62.	Seguimiento de personas con diferentes condiciones iniciales	51
63.	Representación de la morfología corporal humana en Rviz	52
64.	Ángulo de detección del seguimiento de personas (Centro).	52
65.	Ángulo de detección del seguimiento de personas (Derecha).	53
66.	Ángulo de detección del seguimiento de personas (Izquierda).	53
67.	Datos de las articulaciones en Rviz.	54
68.	Datos de las articulaciones de 2 usuarios representados en Rviz.	54

69.	Mapeo del entorno.	55
70.	Visualización del mapeo del entorno en la pantalla del robot.	56
71.	Prueba de la transmisión de videollamada desde el robot.	57
72.	Configuración de audio y video.	57
73.	Iluminación encendida del robot.	58
74.	Iluminación apagada del robot.	58
75.	Dimensiones de la estructura base del prototipo	63
76.	Dimensiones de la estructura tapa del prototipo	64
77.	Partes del robot	66
78.	Primeras pruebas de funcionamiento Raspberry	66
79.	Construcción base inferior del robot	67
80.	Conexión de los componentes del robot	67
81.	Acople del cuerpo del robot	68
82.	Prueba de los drivers del robot	68
83.	Implementación de la pantalla del robot	69

Resumen

En este trabajo se presenta el diseño de un robot móvil de tracción diferencial para aplicaciones de telepresencia. El robot móvil tiene 2 ruedas independientes, las cuales son accionadas por motores sin escobillas, los cuales tienen sensores de efecto hall destinados para el control de bajo nivel. Cada uno de los motores tiene incorporado un controlador PID para el control de velocidad. Así mismo, mediante un sensor Lidar y una cámara RGB-D se implementan algoritmos para la percepción del ambiente.

En donde, el diseño mecánico del robot consta de varios segmentos para la integración con el hardware, los cuales están contruidos en acero al carbono, perfil de aluminio e impresión 3D. Por otra parte, a implementación de los diversos algoritmos para el funcionamiento del robot móvil se realiza en el framework ROS (Robot Operating System), el cual está instalado en un Raspberry Pi 4. Adicionalmente, se implementa la interfaz de interacción con el usuario y el control para los diferentes módulos que constituyen al robot. A estas funciones se accede mediante el uso de una interfaz presente en la pantalla, que busca ser intuitiva de cara al usuario. Así mismo, se propone un algoritmo para el seguimiento de personas y un software de video conferencia, que en conjunto facilitan la telepresencia. Así también, el robot puede ser teleoperado mediante un dispositivo de control externo.

Palabras clave: Tracción diferencial, Telepresencia, Teleoperado, Percepción del ambiente.

Abstract

This work presents the design of a mobile differential traction robot for telepresence applications. The mobile robot has 2 independent wheels, which are driven by brushless motors, which have hall effect sensors intended for low-level control. Each of the motors has a built-in PID controller for speed control. Likewise, by means of a Lidar sensor and an RGB-D camera, algorithms for the perception of the environment are implemented.

Where, the mechanical design of the robot consists of several segments for integration with the hardware, which are built in carbon steel, aluminum profile and 3D printing.

On the other hand, the implementation of the various algorithms for the operation of the mobile robot is carried out in the ROS (Robot Operating System) framework, which is installed on a Raspberry Pi 4. Additionally, the user interaction interface is implemented and the control for the different modules that make up the robot. These functions are accessed through the use of an interface on the screen, which seeks to be intuitive for the user. Likewise, an algorithm for tracking people and video conferencing software are proposed, which together facilitate telepresence. Likewise, the robot can be teleoperated by means of an external control device.

keywords: Natural Fibre, Carludovica Palmata, Cold Plasma at low Pressure, Contact Angle, Free Surface Energy.

INTRODUCCIÓN

La telepresencia, es considerada la transmisión de la imagen de una persona, acompañada de sonido, a lugares distantes, permitiendo al usuario superar las limitaciones creadas por la distancia y, por tanto, estar presente en más de dos lugares al mismo tiempo, aunque no sea de manera física. Uno de los objetivos de la telepresencia es eliminar la distancia geográfica entre dos o más personas, mediante videoconferencia o proyección.

La tele-educación digital es ahora un importante complemento de la educación presencial. Se están transformando, de manera acelerada las modalidades de aprender y enseñar mediante el uso de los robots de telepresencia.

Con la actual situación de pandemia a la que nos hemos tenido que adaptar, se hizo presente la necesidad de implementar nuevas tecnologías que nos permitan realizar de alguna manera la presencialidad e interacción. Es así, que mediante un sistema robotizado que nos permita realizar la telepresencia se puede de alguna manera dar solución a esta problemática, permitiéndonos ser parte del entorno.

El prototipo del robot se inicia con el diseño y la construcción de la estructura que dará soporte al módulo de navegación y a los diferentes sensores, para posteriormente realizar la implementación de las diferentes librerías necesarias para el funcionamiento de cada uno de estos elementos. Una vez lista la parte de hardware se procede al diseño de la interfaz del usuario, para finalmente realizar la programación requerida para la navegación del robot.

ANTECEDENTES DEL PROBLEMA DE ESTUDIO

Antecedentes

Los robots autónomos y sus aplicaciones en diversos campos están creciendo y ganando popularidad, especialmente en los sectores industriales y de servicio, estos robots han aumentado drásticamente durante la última década (2). Con el desarrollo tecnológico y la mejora continua de los robots móviles, se puede notar un aumento en su aplicación de los mismos en los medios mencionados anteriormente. Por lo tanto, para cada aplicación que se busca proporcionar a un robot móvil, se debe considerar sus parámetros estructurales, de forma y estéticos. Un ejemplo de esto sería que si se aplica a la telepresencia, debe contar con el equipamiento necesario para facilitar la comunicación con las personas, así como un modelo que sea atractivo para el usuario y se mueva con naturalidad (2).

Los robots de telepresencia tienen la capacidad de moverse en su entorno y no están fijos en una ubicación física. Un ejemplo de un robot de telepresencia que se usa comúnmente hoy en día es un vehículo guiado automático (AGV). Un AGV es generalmente un robot de telepresencia que sigue marcadores o cables en el suelo, o utiliza un sistema de visión o láseres para la navegación. Los robots de telepresencia se pueden encontrar en entornos industriales, militares y de seguridad. También aparecen como productos de consumo, para entretenimiento o para realizar determinadas tareas como asistencia domiciliaria (3).

Partiendo de la implementación de un robot móvil para telepresencia se puede mejorar las técnicas de comunicación entre individuos en diferentes campos de la vida humana, entre ellos sociales, laborales, académicos e industriales. En la actualidad, la comunicación entre diversas partes del mundo se ha convertido en una prioridad para el desarrollo de las personas presentes en el campo de la industria, medicina, educación y más. Se busca disminuir la complejidad de gestión y los costos que involucran la comunicación a distancia.(3)

Los robots móviles de telepresencia intervienen en diferentes actividades laborales, del hogar, medicina y de la vida cotidiana. Se tiene la necesidad de un control remoto que permita realizar diversas actividades a distancia con la finalidad de evitar desplazamientos innecesarios. (4)

En los últimos años, el internet ha tenido un papel fundamental en la comunicación que se ha vuelto indispensable, pero no es suficiente para las demandas actuales. Por tanto, la implementación de un robot móvil para telepresencia permitirá una comunicación dinámica y eficiente que representará una alternativa de la solución a los problemas actuales en la comunicación en diferentes campos de aplicación.

Justificación (Importancia y Alcances)

- Los robots de telepresencia pueden interactuar con humanos para proporcionar una serie de servicios, como un trabajador de la salud que brinde consultas médicas remotas, asistencia domiciliaria, asistencia comercial, entre otras. En el ejemplo de la asistencia domiciliaria, un robot de telepresencia puede ayudar a las personas mayores con las tareas diarias, que incluyen, entre otras, el mantenimiento de un régimen de medicación, asistencia para la movilidad, asistencia para la comunicación, monitoreo del hogar o del sitio (dentro y/o afuera), monitoreo de personas y/o provisión de un sistema de respuesta de emergencia personal (PERS). Para asistencia comercial, el robot de telepresencia puede proporcionar videoconferencias (por ejemplo, en un hospital), una terminal de punto de venta, una terminal interactiva de información / marketing, etc. (3)
- Además, con el desarrollo de software y hardware que se ha venido realizando en los últimos años se puede mejorar la forma en la que un robot interactúa con el usuario, utilizando las diferentes técnicas ya existentes para dar lugar a un robot móvil de telepresencia mejorado y eficiente de cara a las necesidades del usuario.
- Finalmente, con este proyecto se busca fijar un precedente de la robótica móvil en la Universidad Politécnica Salesiana para futuros proyectos e investigaciones que se quieran realizar en este campo

Objetivos

Objetivo General

Diseñar, construir y programar un prototipo de robot móvil para aplicaciones de telepresencia.

Objetivos Específicos

- Diseñar un robot móvil de 2 ruedas mediante la utilización de software de diseño para su posterior construcción.
- Instalar e implementar las librerías requeridas para el hardware del robot (Cámara RGBD, Lidar, Raspberry).
- Implementar los controladores de alto y bajo nivel para realizar el respectivo funcionamiento del robot
- Realizar los experimentos de telepresencia, incluyendo el seguimiento de personas y verificar el funcionamiento del robot.

1. CAPÍTULO 1: INTRODUCCIÓN A LA ROBÓTICA MÓVIL

1.1. Robótica Móvil

Los robots móviles son robots que pueden moverse de un lugar a otro de forma autónoma, es decir, sin la ayuda de un operador externo. A diferencia de la mayoría de los robots industriales que solo pueden moverse en un espacio de trabajo específico, los robots móviles tienen la característica especial de moverse libremente dentro de un espacio de trabajo predefinido para lograr el objetivo deseado. Esta capacidad de movilidad los hace adecuados para un amplio repertorio de aplicaciones en entornos estructurados y no estructurados. Los robots móviles terrestres se distinguen en robots móviles con ruedas (WMR) y robots móviles con patas (LMR). Los robots móviles también incluyen vehículos aéreos no tripulados (UAV) y vehículos submarinos autónomos (AUV). Los WMR son muy populares porque son apropiados para aplicaciones típicas con una complejidad mecánica y un consumo de energía relativamente bajos. Los robots con patas son adecuados para tareas en entornos no estándar, escaleras, montones de escombros, etc. Normalmente, los sistemas con dos, tres, cuatro o seis patas son de interés general, pero también existen muchas otras posibilidades. Los robots de una sola pierna encuentran aplicaciones raras porque solo pueden moverse saltando. Los robots móviles también incluyen manipuladores móviles (robots con ruedas o patas equipados con uno o más manipuladores de luz para realizar diversas tareas) (5).

1.2. Tipos de Robots Móviles

Los robots móviles están provistos de un sistema de locomoción, ruedas, orugas o patas, que les permiten desplazar de un sitio a otro (6).

Existen distintos tipos robots móviles:

1.2.1. Robots móviles con ruedas

Las ruedas son uno de los sistemas más importantes para la locomoción de robots, y los vehículos inteligentes autónomos (AIV's) se basa en principios como el reconocimiento de patrones y el procesamiento de señales e imágenes. Desempeñan un papel importante en el transporte, la logística y la distribución (6).

El uso de ruedas es más simple que usar orugas o patas, y es más fácil de diseñar, construir

y programar cuando el robot se mueve en un terreno plano y sin obstáculos. Tienden a ser mucho más baratos que sus contrapartes. El control de las ruedas es menos complejo y causan menos desgaste en la superficie donde se mueven en comparación con otras soluciones. Otra ventaja es que no presentan gran dificultad en cuanto a problemas de equilibrio, ya que el robot suele estar en contacto con una superficie (6).

La principal desventaja de las ruedas es que no son muy buenas para navegar sobre obstáculos, como terrenos rocosos, superficies afiladas o áreas con baja fricción (6).

Hay cuatro tipos básicos de ruedas:

- Rueda estándar fija: Son ruedas convencionales con un grado de libertad (DOF), rotación alrededor del punto de contacto.
- Rueda giratoria: tiene dos DOF y una articulación de dirección descentrada para girar.
- Rueda sueca: Tiene 3 DOF, que giran alrededor del eje de la rueda motriz, alrededor del punto de contacto y los rodillos.
- Rueda de bola o esférica: Su implementación es técnicamente compleja.

Es muy importante conocer el número y el tipo de ruedas del robot para modelar la cinemática y la dinámica del robot (6).

Los robots móviles con ruedas (WMR) también se pueden clasificar según el sistema de accionamiento:

- Robot móvil con ruedas de accionamiento diferencial.
- Robot móvil con ruedas tipo coche.
- Robot móvil con ruedas omnidireccionales.
- Robot móvil con ruedas de accionamiento sincronizado.



Figura 3: Robot Móvil con Ruedas

Fuente: Starship

1.2.2. Robots móviles andantes o con patas

Las piernas son otra forma común de locomoción, dando lugar a robots andantes. Aunque suelen ser más caras que las ruedas, las patas tienen varias ventajas sobre las ruedas. La mayor ventaja es su transversalidad y eficiencia y el hecho de que también pueden moverse en terrenos blandos y desnivelados, mejor movilidad, mayor eficiencia, estabilidad y menor impacto en el suelo. Los robots que caminan también tienen la ventaja de hacer frente a los obstáculos o grietas que se encuentran en el entorno (6).

Hay muchos tipos de robots andantes según el número de patas. Entre los más importantes se encuentran los bípedos (humanoides), los de cuatro patas (cuadrúpedos), los de seis patas, etc. (6).

En los robots que caminan, la estabilidad es el problema principal, ya que el equilibrio del cuerpo y la marcha son de extrema importancia, especialmente en el caso de un robot de dos piernas o un humanoide. Hay dos tipos de estabilidad: estática y dinámica. La estabilidad estática es la capacidad de mantener una configuración cuando se somete únicamente a fuerzas de reacción. La estabilidad dinámica está relacionada con la capacidad de permanecer erguido cuando se somete a fuerzas de reacción e inerciales (6).



Figura 4: Robot Móvil con Patas

Fuente: Spot Enterprise, Boston Dynamics

1.2.3. Robots con seguimiento

Se trata de un tipo de robots que utilizan bandas de rodadura o orugas en lugar de ruedas. Los robots con orugas tienen áreas de contacto con el suelo mucho más grande, este hecho juega un papel importante para mejorar su maniobrabilidad en superficies sueltas en comparación con los robots con ruedas convencionales. Sin embargo, debido a este gran parche de contacto con el suelo, el cambio de dirección del robot normalmente requiere de un giro deslizante y, por lo tanto, una gran parte de la pista debe deslizarse contra la superficie (6).



Figura 5: Robot Móvil de Seguimiento con Orugas

Fuente: Guardian S, Sarcos

1.2.4. Robots híbridos

Se trata de robots cuya estructura consiste en una combinación de los robots con ruedas, robots con patas o robots con seguimiento. Por ejemplo, un dispositivo segmentado articulado y con ruedas. Las soluciones híbridas combinan la idoneidad de las ruedas con la adaptabilidad de las patas (6).



Figura 6: Robot Móvil Híbrido

Fuente: Aunav.NEO HD, Aunav

1.3. Telepresencia

La telepresencia, es considerada la transmisión de la imagen de una persona, acompañada de sonido, a lugares lejanos, permitiendo al usuario superar las limitaciones que genera la distancia y, por consiguiente, estar presente en más de dos lugares al mismo tiempo, aunque no sea de manera física. Uno de los objetivos de la telepresencia es suprimir la distancia geográfica entre dos o más personas, por medio de una videoconferencia o una proyección (4).

1.3.1. Ventajas de la telepresencia

La telepresencia aplicada tanto a nivel industrial como en el hogar presenta varias ventajas como se detalla a continuación:

- Reducción en tiempos de transporte
- Agilidad en la toma de decisiones
- Reducción de gastos de movilización
- Comunicación ampliada a todos los lugares del mundo
- Mayor comodidad para el usuario (4).

1.4. Robótica móvil en la telepresencia

Un robot de telepresencia es un dispositivo con ruedas controlado a distancia que tiene conectividad inalámbrica a internet. Normalmente, el robot usa una tableta para proporcionar capacidades de video y audio (7).

Los robots de telepresencia se utilizan comúnmente para sustituir a guías turísticos, vigilante nocturno, inspectores de fábricas y consultores de atención médica. En una clase de educación a distancia, un robot de telepresencia puede moverse por la sala e interactuar cara a cara con estudiantes individuales, tal como lo haría un instructor en las instalaciones. En los negocios, un robot de telepresencia se puede utilizar para ayudar a un empleado cuya discapacidad o ubicación le impide viajar a tener una presencia física en la oficina (7).

2. CAPÍTULO 2: DISEÑO Y CONSTRUCCIÓN DEL PROTOTIPO DEL ROBOT

2.1. Diseño de la estructura del Robot

En la figura 7 se observa el robot móvil de tracción diferencial que se usa para las aplicaciones de telepresencia. Tiene consigo una pantalla la cual permite una fácil interacción con el usuario, además de un sensor Lidar y una cámara RGB-D que permite tener información del ambiente, como obstáculos u otros usuarios.

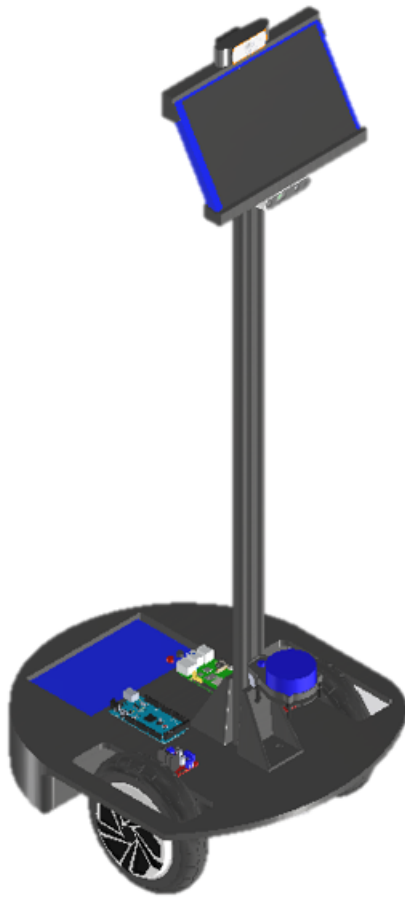


Figura 7: Vista lateral del robot móvil para telepresencia

A continuación, se detalla las partes y piezas que utiliza el robot para su posterior implementación.

2.1.1. Accionamiento Diferencial

Gracias a la fácil implementación y bajo coste se utiliza un accionamiento diferencial para el robot. Esto se realiza mediante 2 motores sin escobillas de 8 pulgadas situados sobre un mismo eje. Además, cada motor tiene un sensor de efecto hall de 90 pulsos por revolución, lo que permite realizar la retroalimentación para el control de velocidad. Así mismo, para brindar más estabilidad al robot, se incorpora 2 ruedas de apoyo.

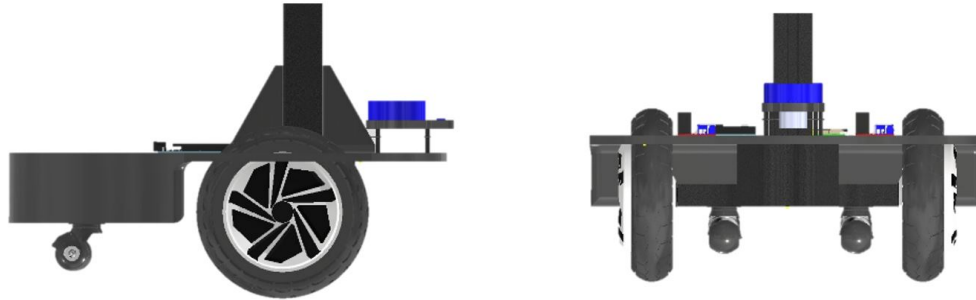


Figura 8: Disposición de las ruedas del robot de tracción diferencial

2.1.2. Sensores

Por otro lado, en el robot móvil se incorpora un sensor YD-Lidar y una cámara RGB-D Intel RealSense 435i, como se observa en la figura 9. Estos sensores junto con algoritmos específicos ayudan a rastrear el ambiente y reconocer personas.

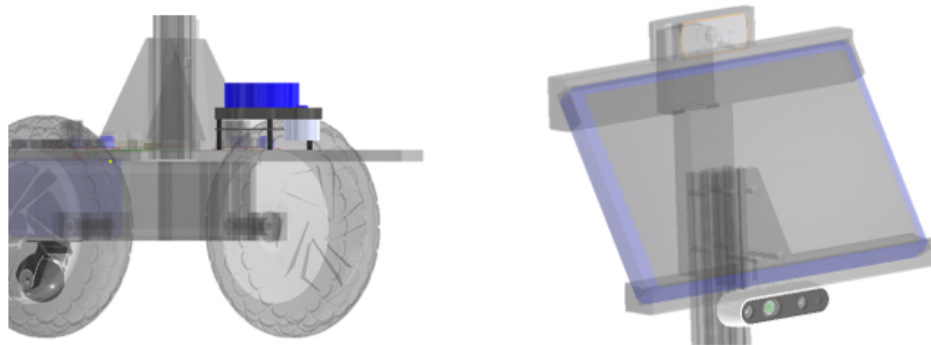


Figura 9: Sensores que equipa el robot. Lidar y Cámara de profundidad.

2.1.3. Interfaz Humano - Robot

El proceso de interacción entre el robot y el usuario se realiza mediante una pantalla de 10.8 pulgadas, que actúa como una interfaz humano maquina (HMI). En la pantalla se encuentra un menú con todas las acciones que el robot puede realizar. Esta se ubica a una altura de 1.50 metros con una inclinación hacia arriba de 30 grados brindando así una interacción cómoda e intuitiva. La estructura que da soporte a la misma está realizada en impresión 3D. El robot cuenta con una interfaz gráfica del usuario (GUI) para lograr una fácil y amigable interacción con el usuario. La programación del GUI se desarrolla usando Python por su versatilidad y fácil comunicación con los demás algoritmos implementados en el robot. Dentro de la interfaz gráfica se puede navegar por las diferentes funciones que el robot presenta, tales como iniciar una videollamada, activar el seguimiento de personas, la teleoperación del robot y las algunas animaciones implementadas.

Además, el robot cuenta con un “Modo Reposo”, en donde se presentan diversas animaciones que le dan al robot un carácter amigable y social para la interacción con el usuario. En la Figura 10 se describe de manera general el flujograma acerca del funcionamiento de la interfaz gráfica implementada.

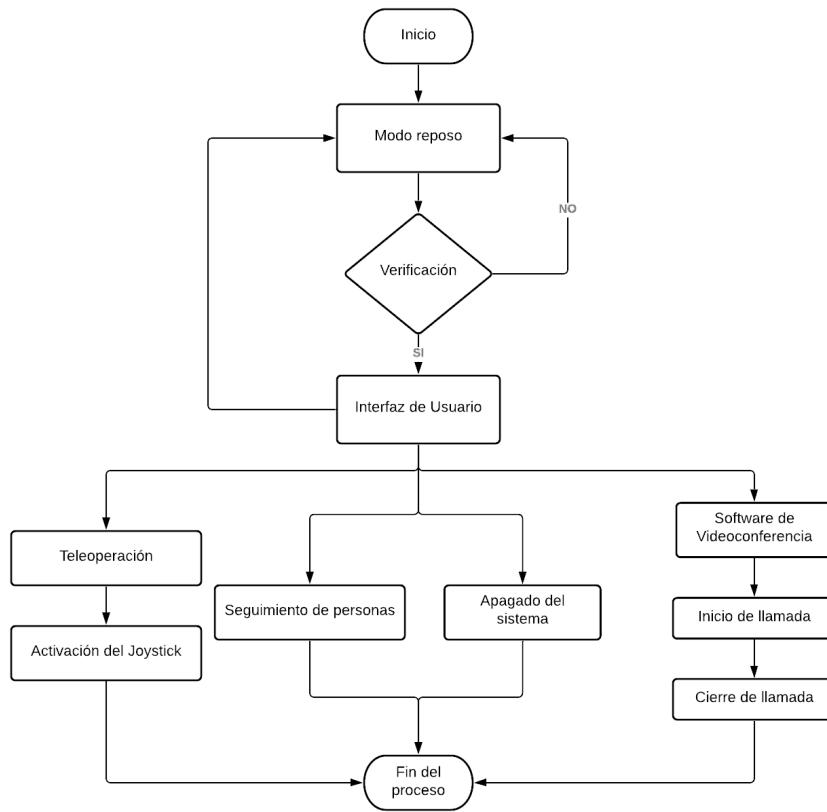


Figura 10: Flujograma del funcionamiento del robot

2.1.4. Diseño de la base interior del robot

Para la parte inferior del robot se realiza el diseño de la estructura base que se encarga de acoplar los componentes para el accionamiento diferencial del robot. Mediante el software inventor se realiza el diseño de parte y planos requeridos. Dicho diseño se muestra en la figura 11 y 12.

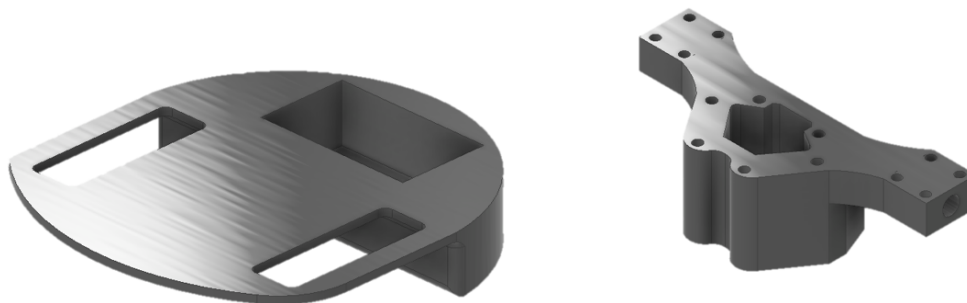


Figura 11: Diseño de la base y eje del robot

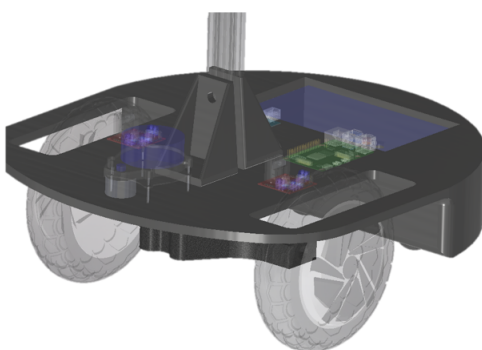


Figura 12: Diseño de la base incorporado los demás elementos del robot

En la figura 13 se muestra el diseño de tapa para la base inferior del robot, cuya finalidad es la de cubrir y resguardar el interior del mismo, donde se encuentra la computadora del robot, el microcontrolador y demás componentes. Además, permite la colocación del sensor LIDAR a una altura adecuada para su correcto funcionamiento.

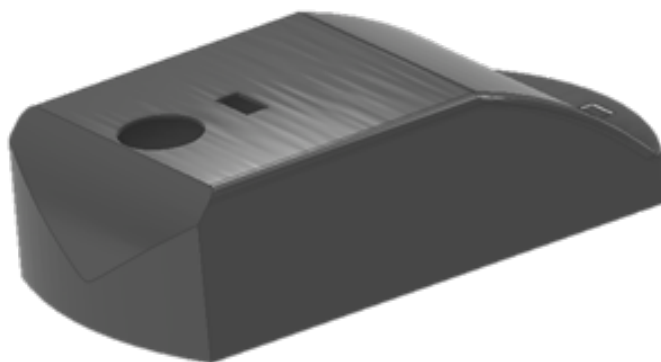


Figura 13: Diseño de la tapa base inferior del robot

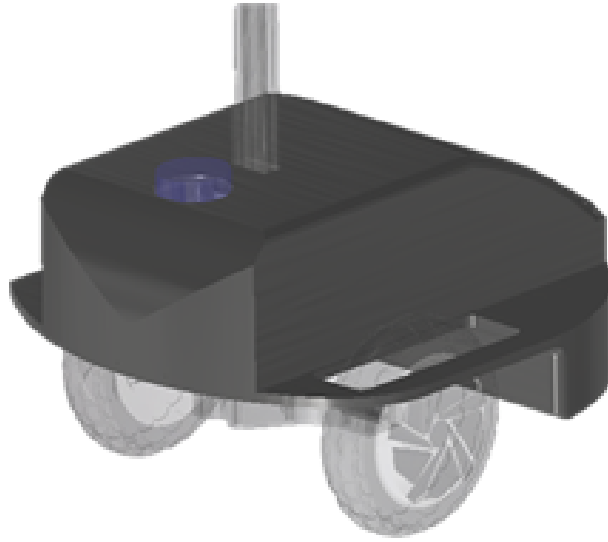


Figura 14: Diseño de la tapa base inferior del robot integrada en la estructura

2.1.5. Diseño de la base superior del robot

Para la parte superior del robot se realiza el diseño de dos piezas que sirvan de soporte para la pantalla del robot que a su vez puedan acoplarse al perfil de aluminio del robot. Dichas piezas se muestran en la figura 15.

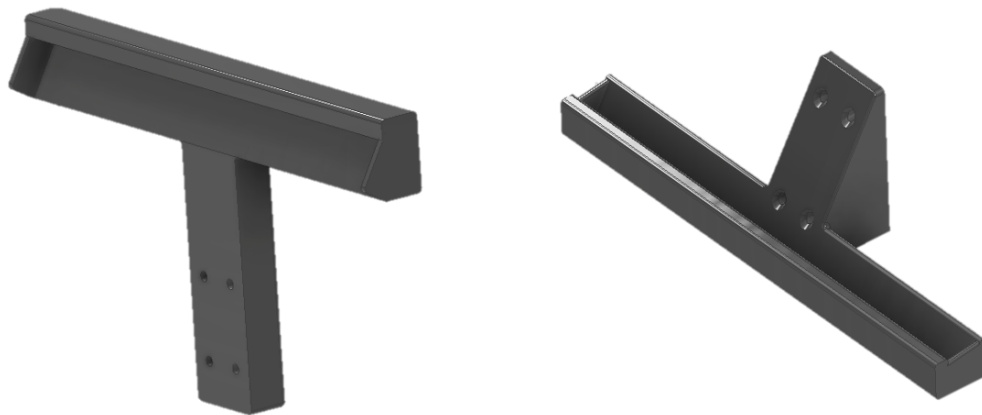


Figura 15: Diseño del soporte para la pantalla del robot

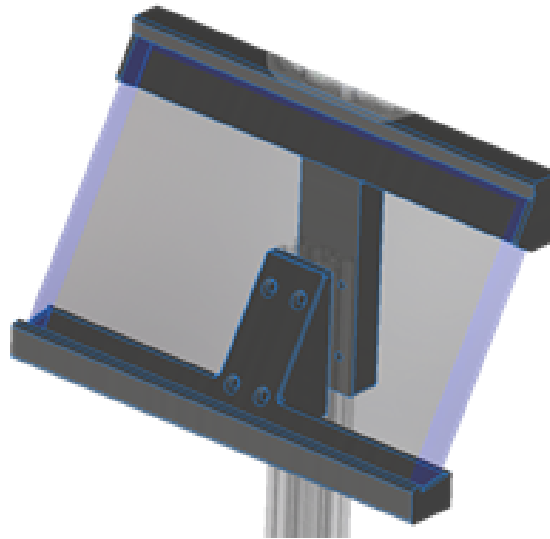


Figura 16: Diseño de los soportes acoplados a la pantalla y perfil del robot

En la figura 17 y 18 se muestra el diseño de tapa para la base superior del robot y la caja del parlante, cuya finalidad es la de tapar los cables de conexión de la pantalla y sostener el parlante del robot respectivamente.

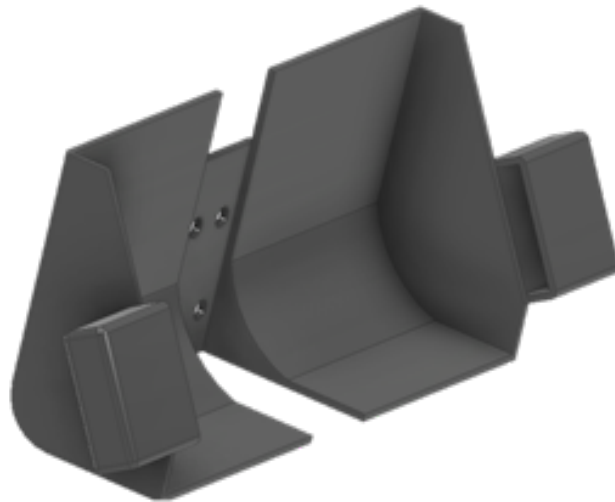


Figura 17: Tapa de la base superior del robot

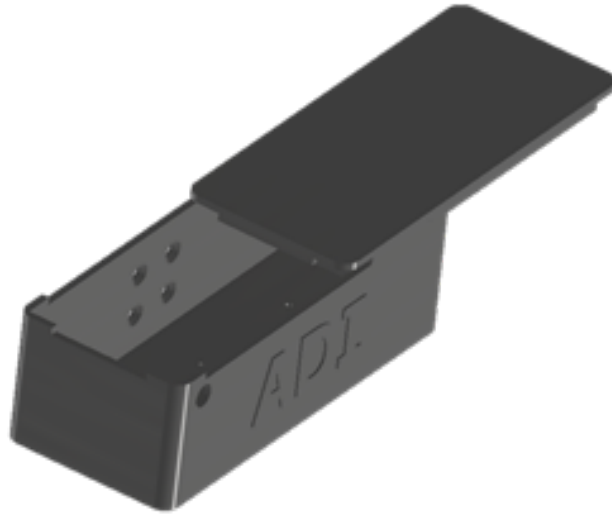


Figura 18: Caja del parlante del robot

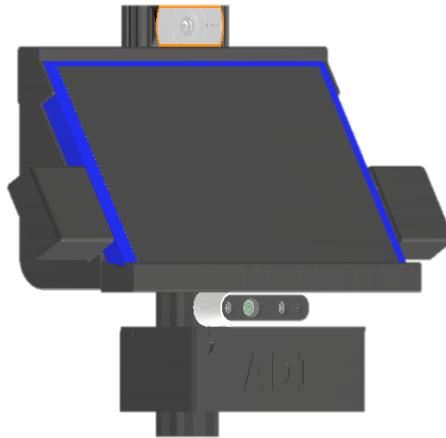


Figura 19: Diseño de la base superior del robot

En las siguientes figuras se muestra el diseño final del robot con todos sus componentes incorporados, desde diferentes perspectivas para su mejor visualización. En la figura 20 se muestra la estructura final del robot sin las tapas que recubren la parte superior e inferior del mismo.

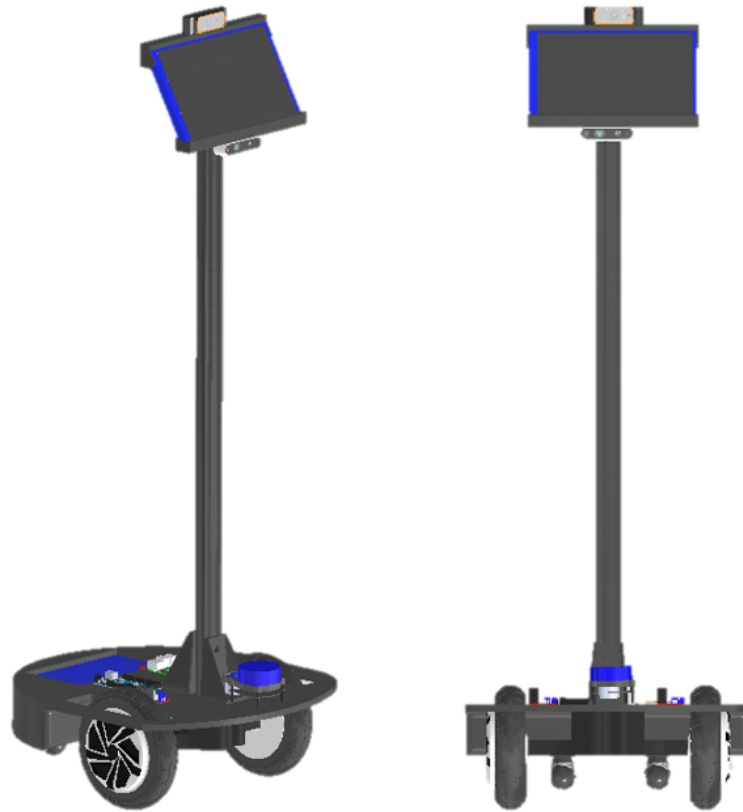


Figura 20: Diseño interior del robot

En la figura 21 se muestra la estructura final del robot con las respectivas piezas que recubren la estructura inferior y superior del mismo, además de la caja del parlante en la parte inferior de la pantalla.

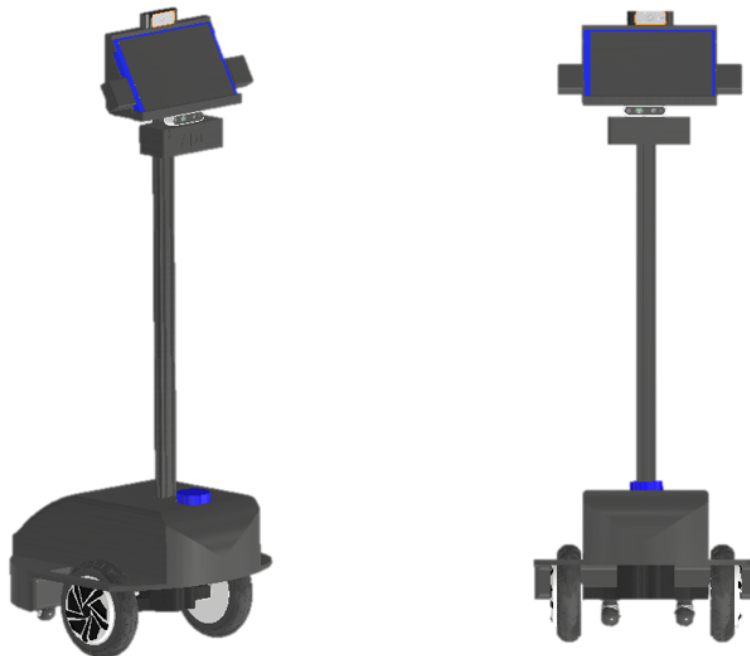


Figura 21: Diseño final del prototipo de robot móvil para aplicaciones de telepresencia

2.1.6. Construcción de la estructura del robot

Para la parte inferior del robot se utilizaron los dos motores sin escobillas de 8 pulgadas acoplados al eje de aleación de aluminio, además de dos ruedas de apoyo para brindar estabilidad, los cuales se incorporan a una estructura de acero al carbono previamente diseñada y fabricada con unas medidas aproximadas de Largo: 0.50m, Ancho: 0.40m y Alto: 0.17m. Dicha estructura se muestra en la figura 22

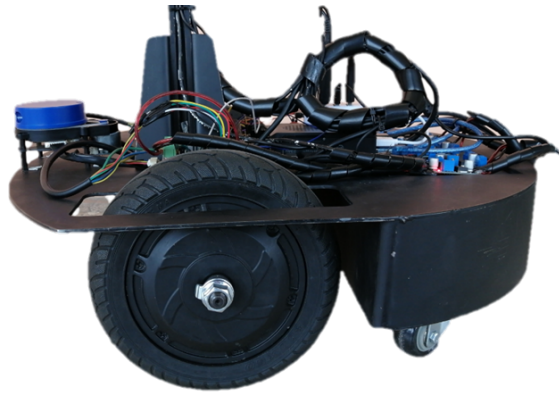


Figura 22: Base inferior del robot

Para el tronco del robot se utiliza un perfil de aluminio de 1m de largo, el cual se acopla a la parte inferior del robot mediante dos anclajes de acero de carbono, los cuales brindan firmeza a la unión de estas dos estructuras.



Figura 23: Base inferior y tronco del robot

La tapa inferior del robot se construye en acero de carbono con un grosor de 3mm, el cual le brinda protección a los componentes internos del robot.



Figura 24: Tapa inferior construida del robot

Mediante impresoras 3D se realizan las dos piezas en filamento petg de color negro que sirven de soporte para la pantalla del robot que a su vez puedan acoplarse al perfil de aluminio del robot.



Figura 25: Soportes acoplados a la pantalla y perfil del robot

En las figuras 26 y 27 se muestran la construcción final de la estructura interna y externa del robot con todos sus componentes incorporados.



Figura 26: Estructura interior del robot



Figura 27: Estructura Final del robot

2.1.7. Diseño de la interfaz del robot

La interfaz gráfica es diseñada en el software de Qt Designer. Qt Designer es una herramienta de Qt que proporciona una interfaz de usuario WYSIWYG (What You See Is What You Get). Permite crear GUIs arrastrando y soltando objetos. Puede utilizar diferentes administradores de diseño para organizarlos en una GUI coherente.

Qt Designer es independiente del lenguaje de programación. No produce código en ningún lenguaje de programación en particular, pero crea archivos .ui. Estos archivos son archivos XML con descripciones detalladas de como generar GUI basada en Qt. En la figura 28 se muestra el editor Qt Designer con la interfaz diseñada.

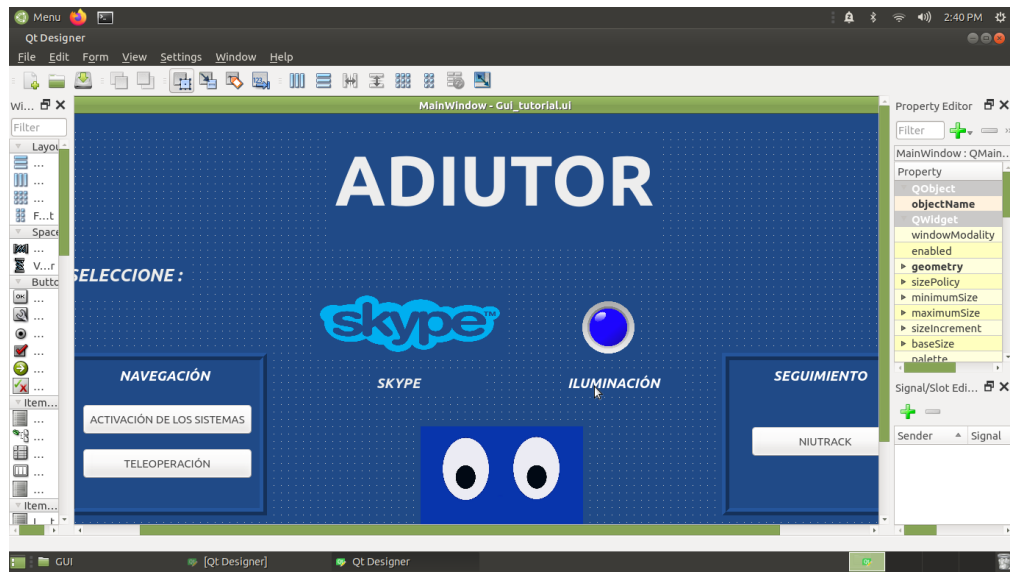


Figura 28: Diseño de la interfaz gráfica en el software de Qt Designer

En la figura 28 se presenta el GUI del robot, en el cual dispone de los siguientes botones:

Para la activación de los sistemas:

En la parte izquierda la interfaz nos presenta un marco de navegación con dos botones de activación y teleoperación como se muestra en la figura 29

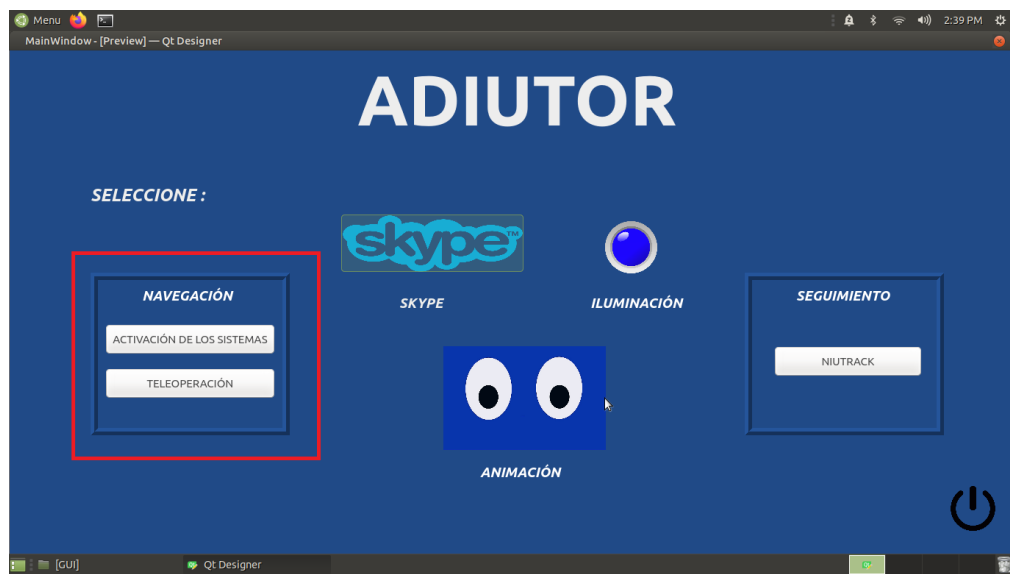


Figura 29: Activación de los sistemas de navegación

- Botón “Activación de los sistemas”: Este botón permite realizar la conexión con el microcontrolador Arduino mega para mediante el mismo comandar los drivers de los

motores.

- Botón “Tele operación”: Este nos permite lanzar el nodo del joystick analógico con el cual controlar el movimiento del robot en su modo de tele operación.

Para la telepresencia:

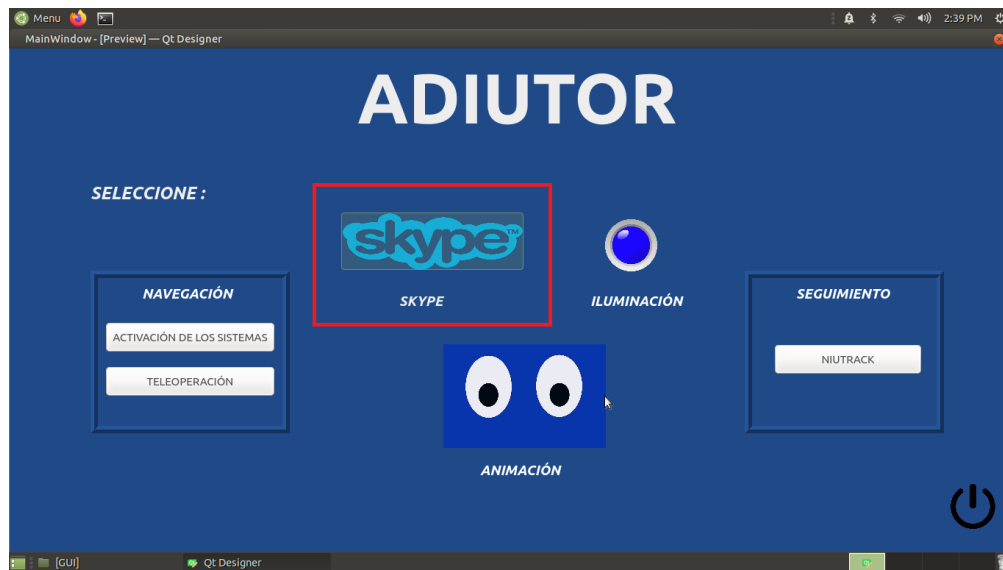


Figura 30: Activación de software de telepresencia

- Botón “Skype”: El botón Skype presente en el GUI con el logotipo de SKYPE, permite la apertura de la página web de “Skype web”, lo que se podrá iniciar sesión y realizar la respectiva videollamada.

Para la activación del seguimiento:

En la parte derecha la interfaz nos presenta un marco de seguimiento con un boton de NuiTrack como se muestra en la figura 31

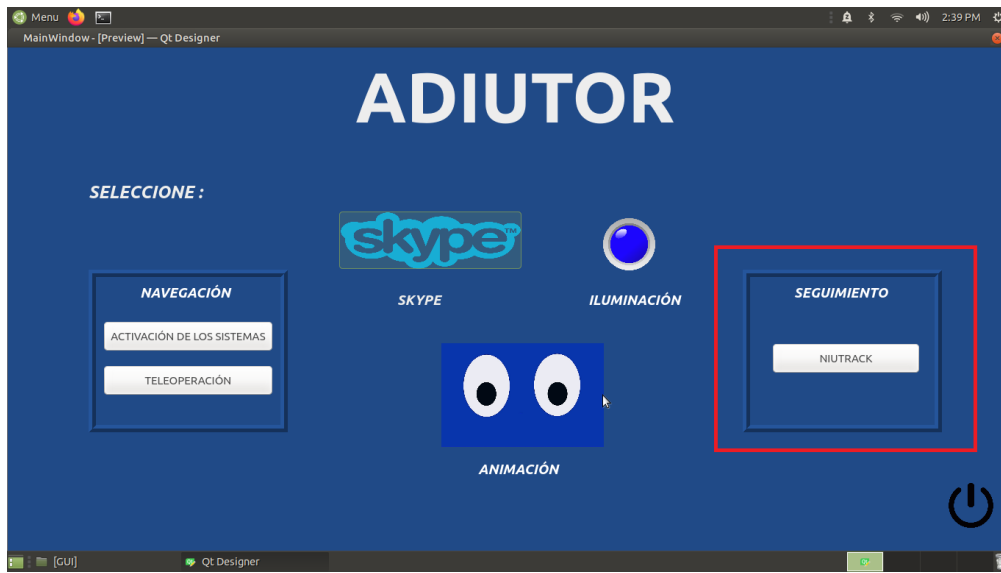


Figura 31: Activación de los sistemas de seguimiento

- Botón “Activación de Nuitrack”: Este botón permite iniciar el nodo de nuitrack que mediante la cámara de profundidad realsense nos permitirá realizar el seguimiento de personas.

Otros:

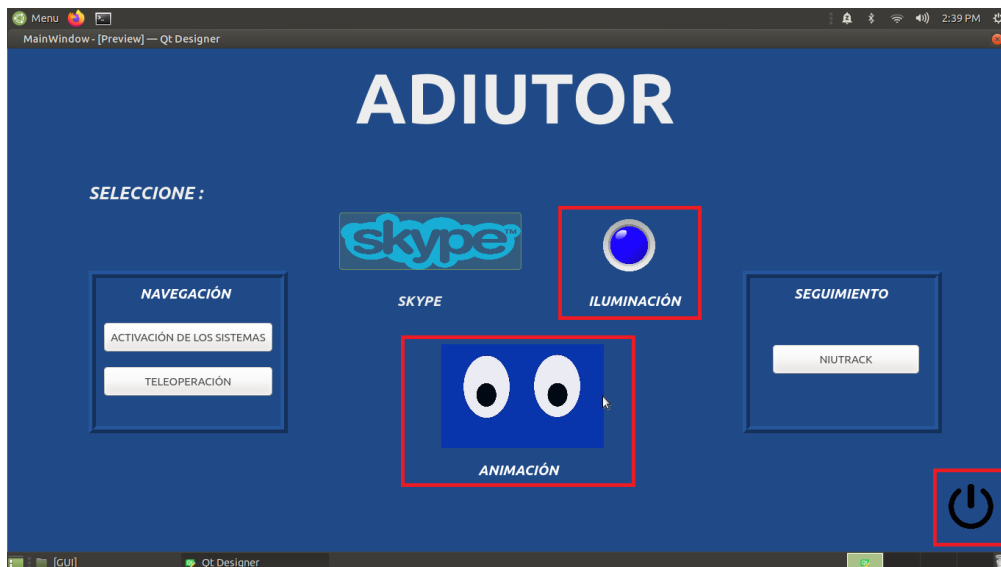


Figura 32: Activación de la animación, iluminación y apagado del sistema

- Botón “Animación”: Nos permite acceder a la animación del robot en cualquier momento

- Botón “Iluminación”: Enciende la iluminación led de color azul en la parte inferior del robot
- Botón “Apagado”: Nos presenta una animación de apagado y cierra todos los procesos.



Figura 33: GUI del robot para aplicaciones de telepresencia

Para el desarrollo de la animación se hace uso del software Adobe After Effects como se muestra en la figura 34. El mismo busca dar un carácter amigable al robot con distintas expresiones como se aprecia en la figura 35.

Las animaciones se presentan al inicio del sistema del robot y en su apagado, además de que puede ser activado en cualquier momento mientras el robot esta en funcionamiento.

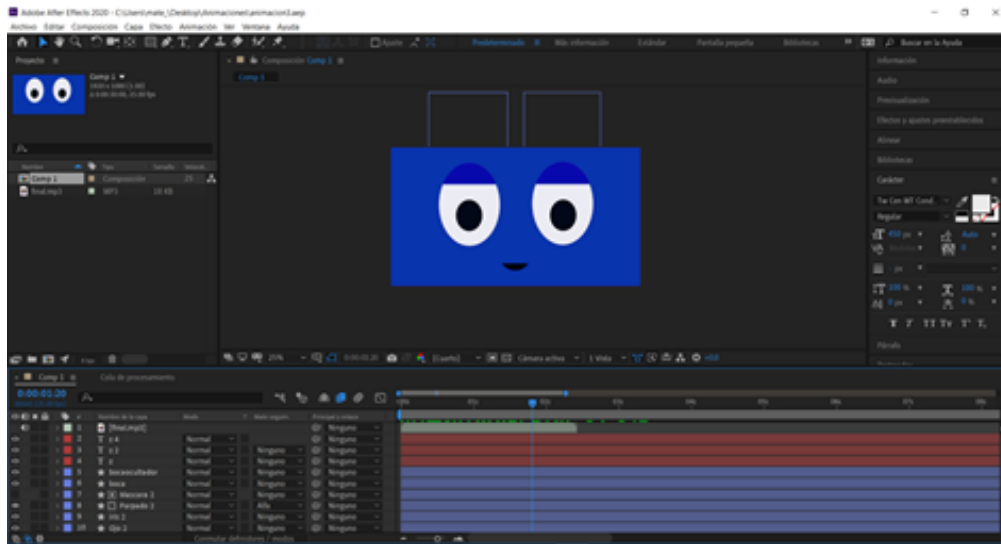


Figura 34: Desarrollo de la animación. Adobe After Effects



Figura 35: Expresiones de la animación

3. CAPÍTULO 3: IMPLEMENTACIÓN Y PROGRAMACIÓN DEL ROBOT

3.1. Robot Operating System(ROS)

Para que el sistema funcione se hace uso del framework de ROS, y para que entre en operación es necesario iniciar el Ros_Master (8), que en este caso será el mismo raspberry ya que no contaremos con un equipo externo conectado al mismo. Esto lo hacemos mediante el comando "roscore" asegurándonos que las direcciones IP esten correctamente configuradas (9).

```
roscore http://172.16.221.148:11311/
File Edit View Search Terminal Tabs Help
roscore http://172.16.221.148:11311/ x pablo@pablo-desktop: ~
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://172.16.221.148:38643/
ros_comm version 1.15.13

SUMMARY
=====

PARAMETERS
* /rostdistro: noetic
* /rosversion: 1.15.13

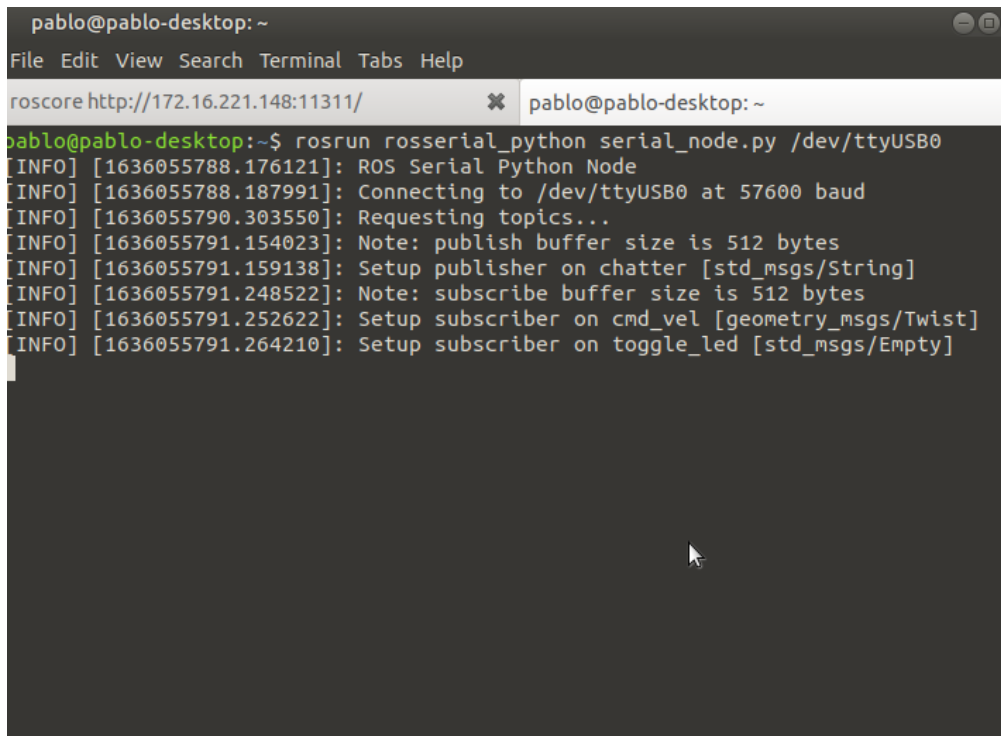
NODES

auto-starting new master
process[rosmaster]: started with pid [3231]
ROS_MASTER_URI=http://172.16.221.148:11311/

setting /run_id to 41c7a068-3da9-11ec-a133-8561e38ddaac
process[rosout-1]: started with pid [3241]
started core service [/rosout]
```

Figura 36: Iniciamos "Ros_Master"

Se realiza la comunicación del raspberry con el Arduino con ayuda del puerto serial. Mediante el CMD del raspberry ejecutamos el comando "roslaunch rosserial_python serial_node.py /dev/ttyUSB0". Esto nos permitirá acceder al programa en el Arduino para comandar los drivers del motor y la iluminación incorporada en la parte inferior del robot.



```
pablo@pablo-desktop: ~
File Edit View Search Terminal Tabs Help
roscore http://172.16.221.148:11311/ x pablo@pablo-desktop: ~
pablo@pablo-desktop:~$ rosrunc serial_python serial_node.py /dev/ttyUSB0
[INFO] [1636055788.176121]: ROS Serial Python Node
[INFO] [1636055788.187991]: Connecting to /dev/ttyUSB0 at 57600 baud
[INFO] [1636055790.303550]: Requesting topics...
[INFO] [1636055791.154023]: Note: publish buffer size is 512 bytes
[INFO] [1636055791.159138]: Setup publisher on chatter [std_msgs/String]
[INFO] [1636055791.248522]: Note: subscribe buffer size is 512 bytes
[INFO] [1636055791.252622]: Setup subscriber on cmd_vel [geometry_msgs/Twist]
[INFO] [1636055791.264210]: Setup subscriber on toggle_led [std_msgs/Empty]
```

Figura 37: Conexión Raspberry con Arduino

Para controlar el encendido y apagado de las luces en el robot es necesario realizar una publicación en el suscriptor del mismo, esto mediante el comando `rostopic pub /toggle_led std_msgs/Empty --once` (10) de esta manera se habilitará o deshabilitará la salida digital del Arduino correspondiente a la iluminación.

```
pablo@pablo-desktop: ~  
File Edit View Search Terminal Tabs Help  
roscore http://172.... x pablo@pablo-desk... x /home/pablo/catki... x pablo@pablo-desk...  
pablo@pablo-desktop:~$ rostopic pub /toggle_led std_msgs/Empty --once  
publishing and latching message for 3.0 seconds  
pablo@pablo-desktop:~$
```

Figura 38: Publicación para encendido de la iluminación

3.2. Nodos del robot

A través del uso de *rqt_graph* disponible en las herramientas de Rqt, se muestra las conexiones de forma gráfica de cada uno de los nodos presentes en el funcionamiento de la navegación mediante teleoperación como se muestra en la figura 39.

El nodo JoyRobot y el nodo Joy se están comunicando entre sí a través de un topico de ROS (11). Joy está publicando las pulsaciones de los botones en el joystick sobre un tema mientras que JoyRobot se suscribe a este tema para recibir las pulsaciones de los mismo. Este transforma la velocidad angular y lineal en velocidad en las llantas y JoyRobot posteriormente pública los valores de velocidad a los cuales serial_node se suscribe para poder finalmente realizar el movimiento del robot.

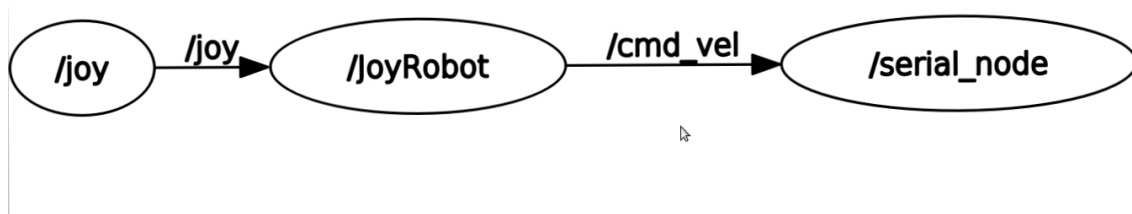
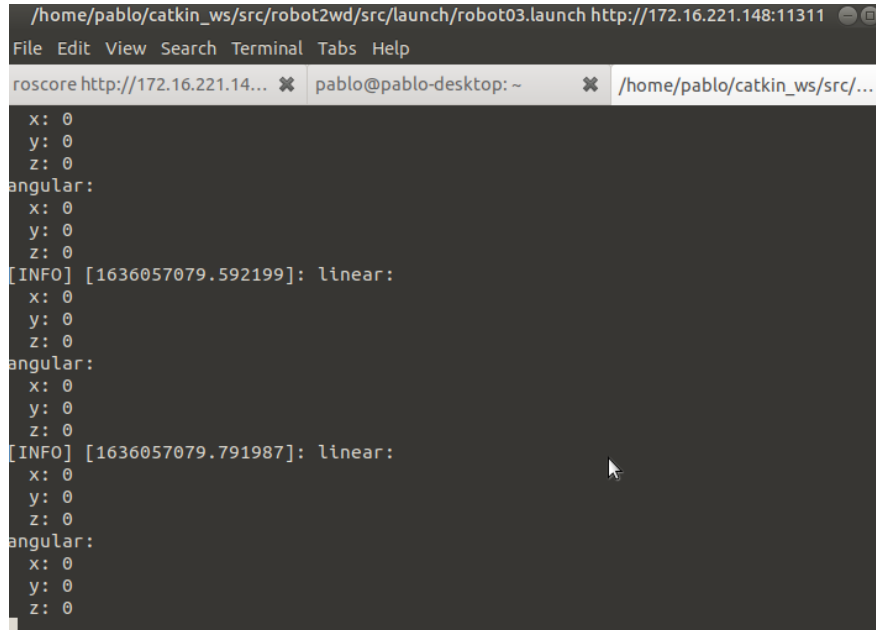


Figura 39: Gráfico de los nodos de la teleoperacion

Mediante el CMD se visualiza los valores de velocidad angular y lineal en x,y,z que nos arroja el nodo JoyRobot, los cuales son enviados al microcontrolador para el movimiento del robot.



```
/home/pablo/catkin_ws/src/robot2wd/src/launch/robot03.launch http://172.16.221.148:11311
File Edit View Search Terminal Tabs Help
roscore http://172.16.221.14... x pablo@pablo-desktop: ~ x /home/pablo/catkin_ws/src/... :
x: 0
y: 0
z: 0
angular:
x: 0
y: 0
z: 0
[INFO] [1636057079.592199]: linear:
x: 0
y: 0
z: 0
angular:
x: 0
y: 0
z: 0
[INFO] [1636057079.791987]: linear:
x: 0
y: 0
z: 0
angular:
x: 0
y: 0
z: 0
```

Figura 40: Velocidad Angular y Lineal del Robot

Los nodos presentes en el mapeo del entorno se muestra en la figura 41, el cual permite la visualización de los diferentes datos de mapeo en Rviz.

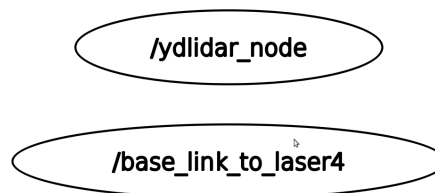


Figura 41: Nodos presentes en el mapeo del entorno.

3.3. NuiTrack

En la figura 42, se muestra el funcionamiento del SDK NuiTrack, el cual dispone de 19 articulaciones para el seguimiento de cuerpo completo.

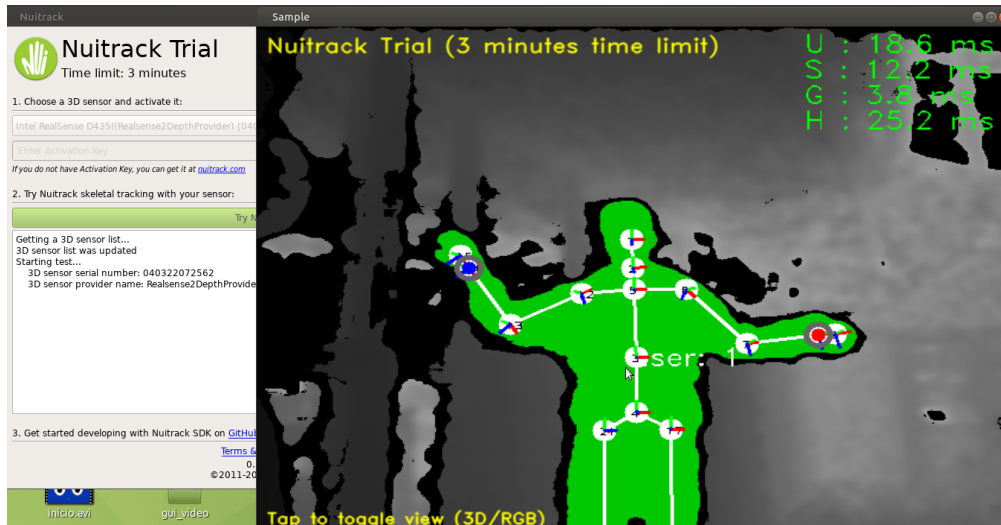


Figura 42: Prueba de funcionamiento del SDK nuitrack

Mediante el uso de *rqt_graph* disponible en las herramientas de Rqt, se muestra las conexiones de forma gráfica de cada uno de los nodos presentes en el funcionamiento del SDK nuitrack como se muestra en la figura 43. El nodo *nuitrack_user_viewer* y el nodo *nuitrack_core* se están comunicando entre sí a través de un tema de ROS.

NuiTrack_core está publicando los datos de las articulaciones para el seguimiento de personas mientras que *nuiTrack_user_viewer* se suscribe a este tema para recibir estos valores.

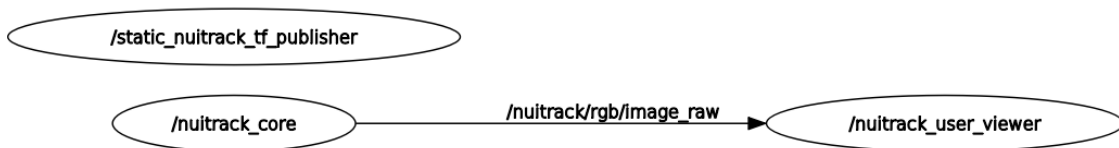


Figura 43: Gráfico de los nodos presentes en el seguimiento de personas

3.4. Modelo Matemático del motor

El modelado que se plantea de los motores BLDC se visualiza en la figura 44.

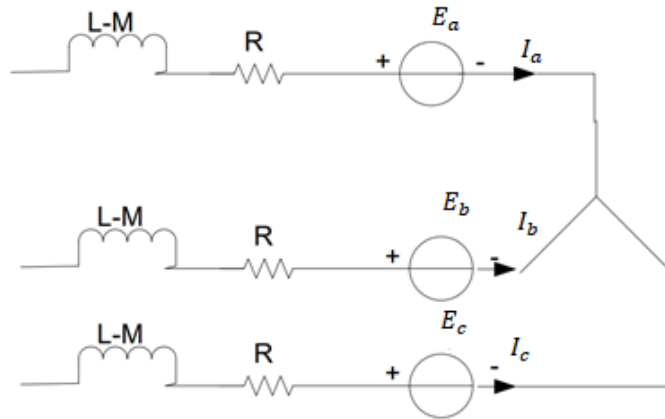


Figura 44: Diagrama del circuito de devanado del estator.

Las ecuaciones del circuito del motor pueden describirse mediante las siguientes ecuaciones:

$$V_a = I_a R + L \frac{di_a}{dt} + E_a \quad (1)$$

$$V_b = I_b R + L \frac{di_b}{dt} + E_b \quad (2)$$

$$V_c = I_c R + L \frac{di_c}{dt} + E_c \quad (3)$$

Por lo tanto, la función de transferencia se obtiene utilizando la relación del radio y velocidad angular con la fuente de voltaje:

$$G(s) = \frac{\omega_m}{V_s} = \frac{\frac{1}{K_e}}{\tau_m \tau_e s^2 + \tau_m \cdot s + 1} \quad (4)$$

$$\tau_m = \frac{RJ}{K_e K_t} \quad (5)$$

$$\tau_e = \frac{L}{3xR} \quad (6)$$

$$K_e = \frac{3R\phi J}{K_t \tau_m} \quad (7)$$

Las variables relacionadas con la ecuación son:

τ_m : variable mecánica (constante en el tiempo)

τ_e : par de endtrada del devanado

$R = R_\phi$ es la resitencia de fase a fase

J es la inercia del motor

L es la inductacia de armadura

K_t es la constante par

P es le numero de polos

B es el coeficiente calculado a partir del momento de inercia

Los parámetros del motor que se utilizaron son los mostrados en la Tabla 1.

No	Parámetro	Valor
1	Resistencia de armadura (Ra)	0,94 Ω
2	Inductancia (La)	1,19x10 ⁻³ H
3	Momento de inercia (J)	5,1x10 ⁻³ Kgm ²
4	Constante Par (Kt)	4,17Nm/A
5	Constante de tiempo (mecánica) (Tm)	0.00491s

Tabla 1: Parámetros del Motor BLDC

Estos valores se reemplazan en las ecuaciones 4, 5, 6 y 7; obteniéndose:

$$\tau_e = \frac{L}{3xR} = \frac{1,19x10^{-3}}{3x0,94} = 422x10^{-6}$$

$$K_e = \frac{3R\phi J}{K_t\tau_m} = \frac{3x0,94x5,1x10^{-6}}{0,426x0,00491} = 6,87x10^{-3}V \cdot s/rad$$

$$G(s) = \frac{\omega_m}{V_s} = \frac{\frac{1}{6,87x10^{-3}}}{0,00491x422x10^{-6}s^2+0,00491 \cdot s+1}$$

Dando como resultado:

$$G(s) = \frac{145,5}{2,072x10^{-6}s^2 + 0,00491 \cdot s + 1} \quad (8)$$

Después de obtención de la función de transferencia se modela la misma mediante simulink tal como se visualiza en la figura 45.

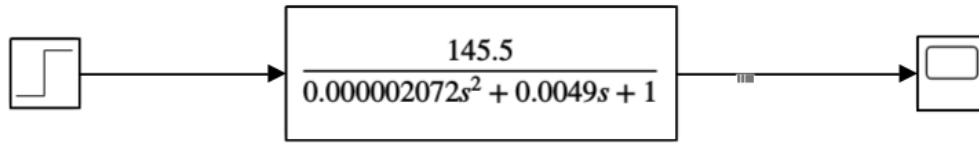


Figura 45: Diagrama de bloques de la planta.

Se simula con una respuesta al escalón para conocer como responde la planta mostrado en en la figura 46.

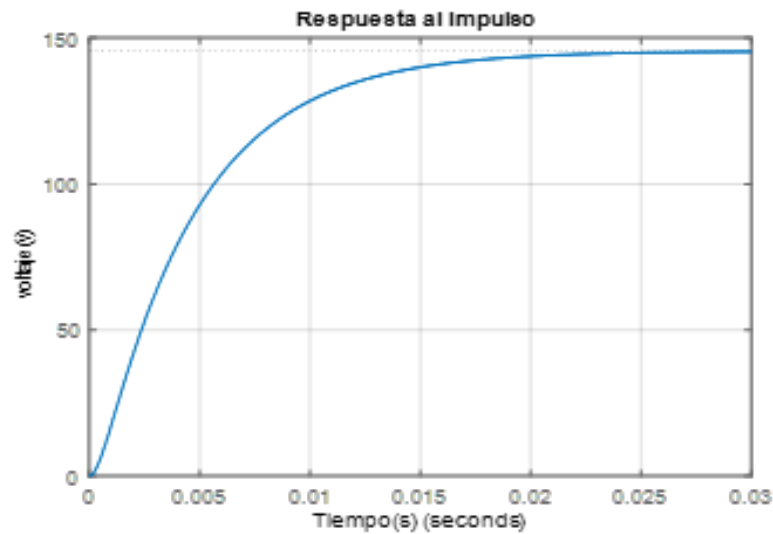


Figura 46: Respuesta al Escalón.

3.5. Diseño de PID

Para el diseño del algoritmo que se va utilizar para el control, se requiere conocer el comportamiento de la planta que se va a controlar, a partir de la figura 47 se obtiene el modelo del motor BLDC (6).

Para la planta antes mostrada se aplica un controlador digital PID de velocidad, como se puede visualizar en la figura 47.

(1)

Tipo de PID	K_p	$TI = \frac{K_p}{K_i}$	$TD = \frac{KD}{K_p}$
P	$\frac{T}{L}$	∞	0
PI	$0,9x\frac{T}{L}$	$\frac{L}{L0,3}$	0
PID	$1,2x\frac{T}{L}$	$2xL$	$0,5xL$

Tabla 2: Parámetros del método Ziegler -Nichols (1)

Tomando en cuenta estos parámetros se procede a simular en el software Simulink para poder evaluar los mismos, como se puede ver en la figura 49.

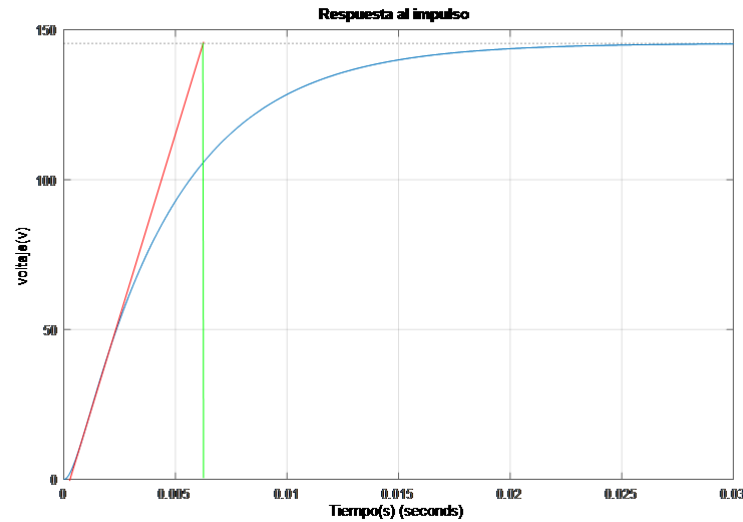


Figura 49: Parámetros de la respuesta al escalón de la planta.

Según la gráfica se pudo obtener los siguientes valores:

$$L = 1,86x10^{-3}T = 6,16x10^{-3}K = 145 \quad (9)$$

Dados los valores se procede a reemplazarlos en la siguiente tabla:

Tipo de PID	K_p	$TI = \frac{K_p}{K_i}$	$TD = \frac{K_D}{K_p}$
P	3.31	∞	0
PI	2.98	0.0062	0
PID	3.97	0.0037	0.0093

Tabla 3: Parámetros obtenidos del controlado mediante Ziegler –Nichols

Hallados los parámetros del controlador se lo asocia al sistema propuesto en lazo cerrado como se observa en la figura 50.

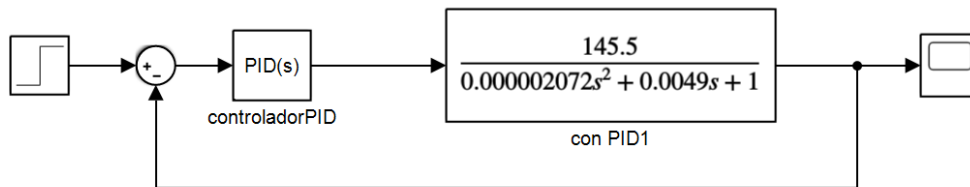


Figura 50: Diagrama de la planta + control PID.

Se realiza la simulación del sistema haciendo una comparativa entre el control PID y el sistema sin control PID como se ve a continuación en la figura 51.

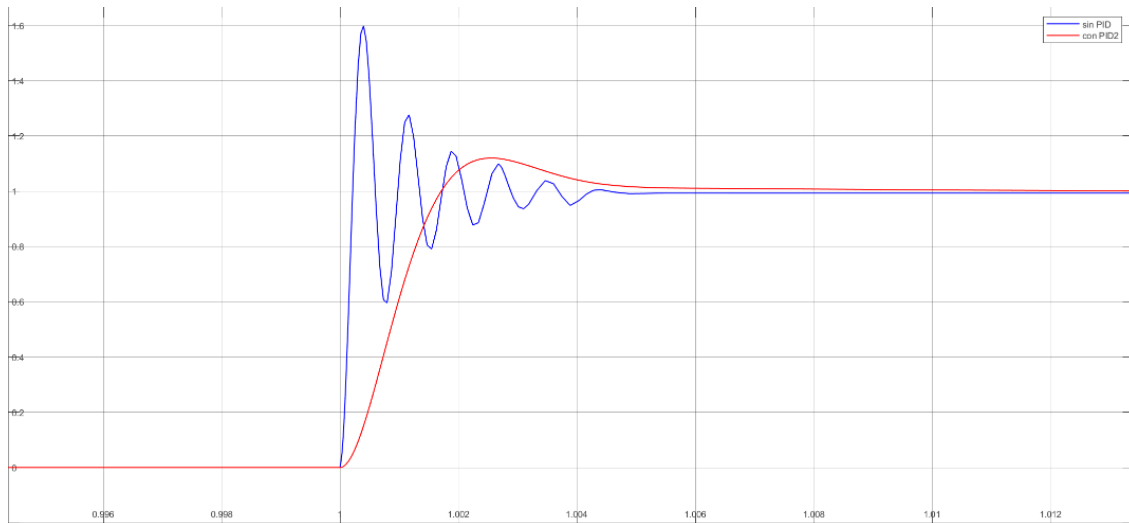


Figura 51: Respuesta del sistema en lazo cerrado con control PID (rojo) , respuesta del sistema en lazo cerrado sin control PID (azul).

Se observa que la respuesta de la planta mejora considerablemente en cuanto a los siguientes criterios : sobreimpulso, tiempo de establecimiento y de subida. Con esto se puede controlar de manera precisa la señal del PWM que varía el voltaje promedio a la entrada del motor .

3.6. Control Digital PID

En la figura ?? se presenta la forma de conexión de los drivers hacia la tarjeta Arduino Mega del robot, el cual cuenta con dos drivers para cada una de sus llantas respectivamente.

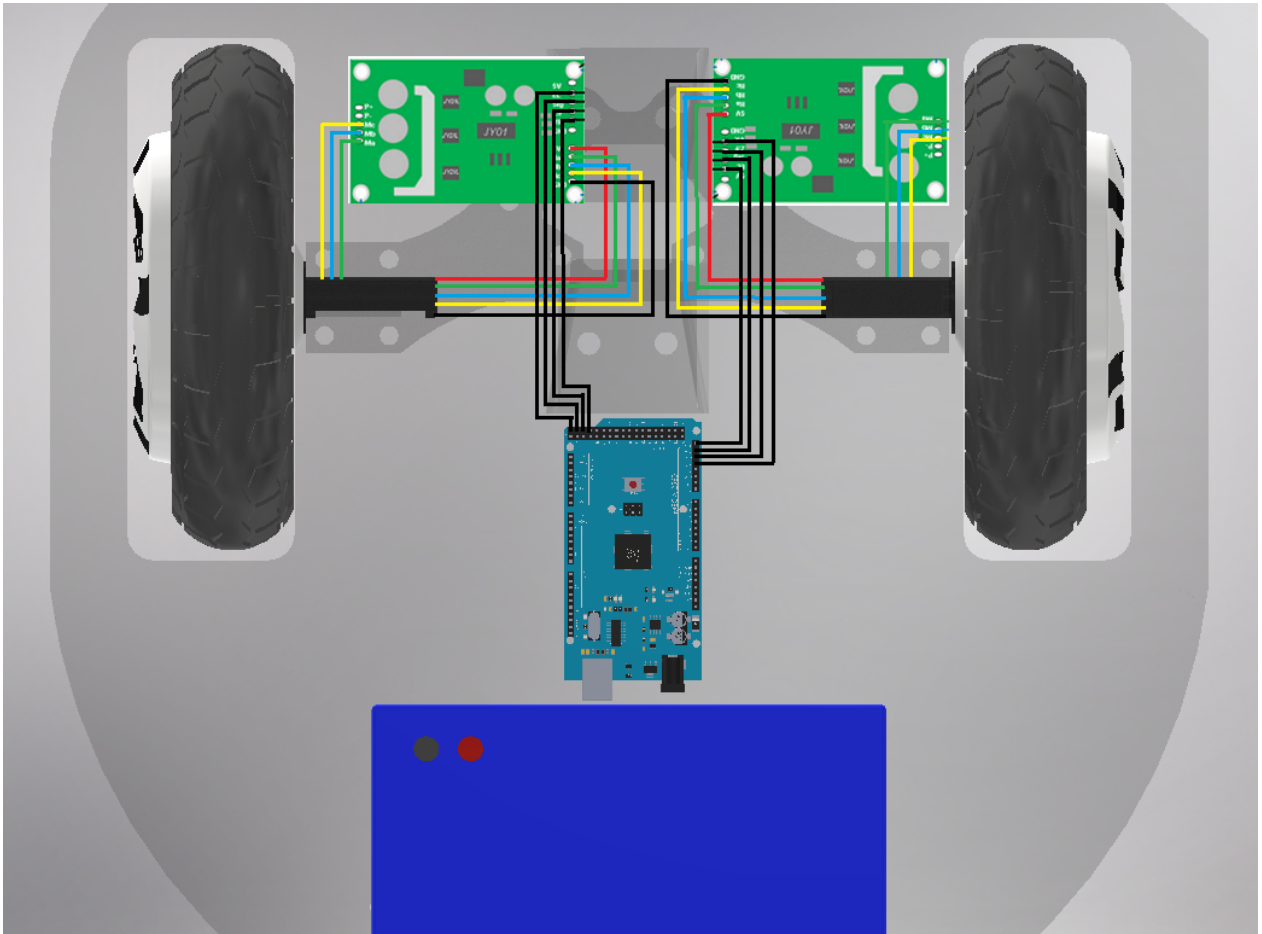


Figura 52: Conexión Arduino con drivers.

Tomando en cuenta el diseño del controlador PID, el procesamiento digital del mismo se realiza mediante las librerías de Arduino.

En la figura 53 se muestra la asignación de pines para la lectura de los sensores y control PWM para los motores.

```

const byte interruptC = 20; // Rueda delantera derecha
const byte interruptD = 21; // Rueda delantera izquierda

const byte pinbreakC = 27; // Rueda delantera derecha
const byte pinbreakD = 29; // Rueda delantera izquierda

const int pinPWC = 4; // Rueda delantera derecha
const int pinPWD = 5; // Rueda delantera izquierda

const int pindirectC = 32; // Rueda delantera derecha
const int pindirectD = 34; // Rueda delantera izquierda

```

Figura 53: Asignación de variables.

Se asigna las variables para la matriz de transformación.

```

volatile float u=0;
volatile float w=0;
float r = 0.08;
float d = 0.28;

volatile float w_r ;
volatile float w_l;

```

Figura 54: Variables para la matriz de transformación.

Se usa el timer 4 debido a que presenta una mejor aproximación para el tiempo de muestreo y no interfiere con los pines de interrupción de los sensores. Se crea interrupciones cada 100ms para que actúe el PID.

```

// Configuración de Lectura

attachInterrupt(digitalPinToInterrupt(interruptA), encoderA, CHANGE);
attachInterrupt(digitalPinToInterrupt(interruptB), encoderB, CHANGE);
attachInterrupt(digitalPinToInterrupt(interruptC), encoderC, CHANGE);
attachInterrupt(digitalPinToInterrupt(interruptD), encoderD, CHANGE);

noInterrupts(); // disable all interrupts
//set timer4 interrupt at 10Hz
TCCR4A = 0;// set entire TCCR1A register to 0
TCCR4B = 0;// same for TCCR1B
TCNT4 = 0;//initialize counter value to 0
// set compare match register for 1hz increments
//OCR4A = 15624/1;// = (16*10^6) / (1*1024) - 1 (must be <65536)
// set compare match register for 10hz increments
OCR4A = 1561;// = (16*10^6) / (10*1024) - 1 (must be <65536)
// turn on CTC mode
TCCR4B |= (1 << WGM12);
// Set CS12 and CS10 bits for 1024 prescaler
TCCR4B |= (1 << CS12) | (1 << CS10);
// enable timer compare interrupt
TIMSK4 |= (1 << OCIE4A);
interrupts(); // enable all interrupts

}

```

Figura 55: Configuración del Timer 4.

Se usa la función callback que está asociada a la interrupción para obtener la lectura de la velocidad angular de las ruedas independientemente en rad/s.

```

ISR(TIMER4_COMPA_vect) {

velA = (countA/(0.1*resolucion))* (2*pi); //rad/s
velB = (countB/(0.1*resolucion))* (2*pi); //rad/s
velC = (countC/(0.1*resolucion))* (2*pi); //rad/s
velD = (countD/(0.1*resolucion))* (2*pi); //rad/s

//Serial.println(velA);
//Serial.println(velB);
//Serial.println(velC);
//Serial.println(velD);

countA = 0;
countB = 0;
countC = 0;
countD = 0;

}

```

Figura 56: Lectura de las velocidades angulares de cada una de las ruedas.

Se usa un control PID en tiempo discreto definiendo la ecuación tal que:

$$u(k) = u(k - 1) + q_0e(k) + q_1e(k - 1) + q_2e(k - 2) \quad (10)$$

Para la parte del controlador digital, se usa la librería de arduino *PID_V2* y se asigna las constantes del controlador obtenidas teóricamente y ajustadas en la parte práctica.

```
//Constantes PID

double Kp =0.95, Ki = 0.9, Kd = 0.065;

PID_v2 myPID_C(Kp, Ki, Kd, PID::Direct);
PID_v2 myPID_D(Kp, Ki, Kd, PID::Direct);
```

Figura 57: Librería *PID_V2* y constante del controlador.

Se define el ingreso, salida y setpoint para el controlador.

```
//PID

myPID_C.Start(analogRead(velC), pinPWMC, 0);
myPID_D.Start(analogRead(velD), pinPWMD, 0);
```

Figura 58: Parámetros del controlador.

Se configura la dirección de acuerdo a la lectura que toman las variables de velocidad lineal y velocidad angular.

```

void direccion(){

    if(w_r<0 && w_l >0){

        digitalWrite(pindirectC, HIGH); // Rueda delantera derecha 32
        digitalWrite(pindirectD, HIGH); //Rueda delantera izquierda 34
    }

    else if (w_r>0 && w_l<0){

        digitalWrite(pindirectC, LOW); // Rueda delantera derecha 32
        digitalWrite(pindirectD, LOW); //Rueda delantera izquierda 34
    }

    else if (w_r<0 && w_l<0){

        digitalWrite(pindirectC, HIGH); // Rueda delantera derecha 32
        digitalWrite(pindirectD, LOW); //Rueda delantera izquierda 34
    }
    |
    else {

        digitalWrite(pindirectC, LOW); // Rueda delantera derecha 32
        digitalWrite(pindirectD, HIGH); //Rueda delantera izquierda 34
    }

    w_r=abs(w_r);
    w_l=abs(w_l);

}

```

Figura 59: Configuración de la dirección de las ruedas.

3.7. Modelo Cinemático

La figura 60 muestra la estructura del robot móvil de tracción con diferencial que se usa para obtener el modelo cinemático extendido propuesto por (12). En donde, (x^ω, y^ω) especifica las coordenadas cartesianas globales 2D; (x, y) representa el punto de interés del robot y está localizado entre las ruedas del robot, ϕ denota la orientación del robot que se mide positiva en sentido contrario a las manecillas del reloj respecto a x^ω . De este modo, las coordenadas generalizadas el robot están dadas por $q = [x, y, \phi]^T$. La velocidad lineal es u , la velocidad lateral es v , respectivamente, y ω representa la velocidad de rotación del robot.

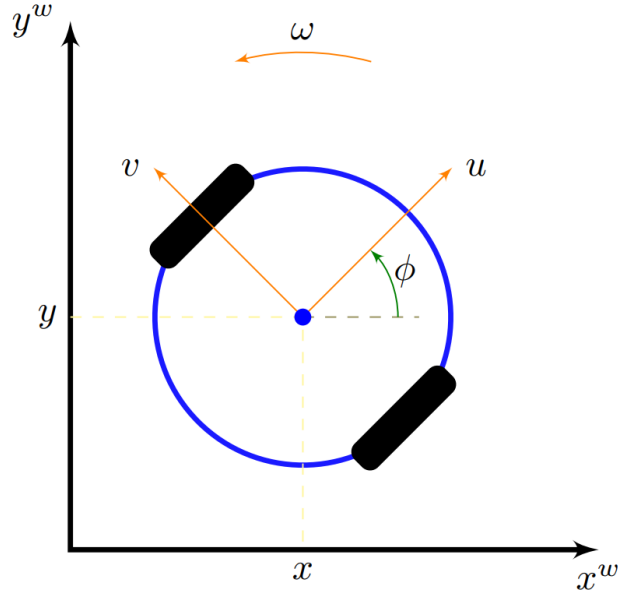


Figura 60: Estructura usada para obtener el modelo cinemático del robot móvil.

El modelo cinemático extendido para el robot de tracción diferencial se muestra en (1), en donde se puede representar por $\dot{q} = A(q)u$. Note que el sistema propuesto es siempre invertible, lo cual lleva a $u = A^{-1}(q)\dot{q}$; en donde \dot{q} es el vector de estados generalizados, u es el vector de entradas de control $A(q)$ es la matriz cinemática.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} u \\ v \\ \omega \end{bmatrix} \quad (11)$$

3.7.1. Controlador basado en Cinemática Inversa

Teniendo en cuenta lo antes expuesto, se propone un controlador basado en el modelo cinemático extendido que se muestra en (2), cuya notación compacta está dado por $u_c = A^{-1}(q)n$. En donde, u_c es el vector de entradas de control, n es el vector de la ley de control, la cual será diseñada para satisfacer los objetivos de control propuestos.

$$\begin{bmatrix} u_c \\ v \\ \omega_c \end{bmatrix} = \begin{bmatrix} \cos\phi & \sin\phi & 0 \\ -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} n_x \\ n_y \\ n_\phi \end{bmatrix} \quad (12)$$

3.8. Control de bajo nivel

Para el correcto funcionamiento del robot es imprescindible la implementación de un controlador de bajo nivel. En este caso, un controlador PID para cada motor está programado en un Arduino Mega, el cual se comunica con ROS para la recepción de las acciones de control. Para la lectura de velocidad de las ruedas se usan el sensor de efecto hall que tiene incorporado cada motor y se conecta a una entrada de interrupción del Arduino. Se propone un tiempo de muestro de $t_s = 100ms$, debido a la baja resolución del sensor de efecto hall.

$$\omega = 2\pi \frac{\#pulsos}{t_s Resol} [rad/s] \quad (13)$$

3.8.1. Conversión de Velocidades

Debido a que las acciones de control tienen las siguientes magnitudes, velocidad lineal m/s y velocidad angular rad/s ; es necesario realizar la conversión para que el controlador de bajo nivel tenga las referencias acorde a las magnitudes de velocidad angular.

$$\begin{bmatrix} u_c \\ \omega_c \end{bmatrix} = \begin{bmatrix} \frac{r}{2} & \frac{r}{2} \\ \frac{r}{d} & -\frac{r}{d} \end{bmatrix} \begin{bmatrix} \omega_r \\ \omega_l \end{bmatrix} \quad (14)$$

Obteniendo la inversa, se obtiene las velocidades angulares para cada una de las ruedas cuya magnitud es rad/s .

$$\begin{bmatrix} \omega_r \\ \omega_l \end{bmatrix} = \begin{bmatrix} \frac{1}{r} & \frac{d}{2r} \\ \frac{1}{r} & -\frac{d}{2r} \end{bmatrix} \begin{bmatrix} u_c \\ \omega_c \end{bmatrix} \quad (15)$$

3.9. Modelado de la tarea

En la figura 79 se muestra una vista superior del modelado de la tarea de seguimiento de personas. En donde, se considera la nomenclatura usada en la figura 60, además ϕ_d representa el ángulo deseado al cual debe situarse el robot. Así mismo, (x_h, y_h) representa la posición global del usuario.

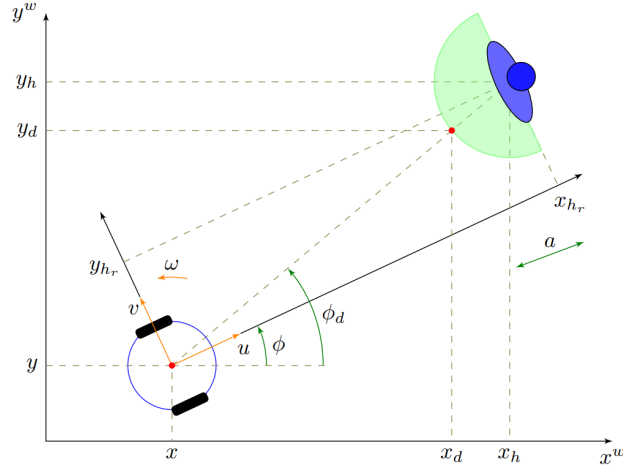


Figura 61: Modelado de la tarea de seguimiento de personas.

En donde, la posición del usuario con respecto al marco de referencia global se calcula como se indica en (6)

$$\begin{bmatrix} x_h \\ y_h \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} x_{hr} \\ y_{hr} \end{bmatrix} \quad (16)$$

De esta manera se definen los valores deseados de la siguiente manera:

$$x_d = x_h - a\cos\phi_d, \quad (17a)$$

$$y_d = y_h - a\sin\phi_d \quad (17b)$$

$$\phi_d = \arctan\left(\frac{y_h - y}{x_h - x}\right) \quad (18)$$

3.9.1. Objetivos de Control

Como objetivo de control se propone que el robot realice el seguimiento del usuario manteniéndolo dentro del campo de visión de la cámara mientras mantiene una video llamada. Por lo tanto, se definen los siguientes errores de control como $\tilde{x} = x_d - x$, $\tilde{y} = y_d - y$ y $\tilde{\phi} = \phi_d - \phi$. Note que ϕ_d se obtiene directamente de (8). Usando la misma notación, las coordenadas generalizadas están dadas por $q_d = [x_d \ y_d \ \phi_d]^T$. Así, el vector de errores se expresa como $\tilde{q} = (q_d - q)$.

Note que cuando $\tilde{\phi}(t) \rightarrow 0$ significa que el usuario está localizado en el centro de la cámara, con lo cual se cumple el objetivo de control propuesto.

3.9.2. Ley de control

A fin de cumplir con el objetivo de control se propone la siguiente ley de control.

$$n_x = \dot{x}_d + l_x \tanh(k_x l_x^{-1} \tilde{x}), \quad (19a)$$

$$n_y = \dot{y}_d + l_y \tanh(k_y l_y^{-1} \tilde{y}), \quad (19b)$$

$$n_\phi = \dot{\phi}_d + l_\phi \tanh(k_\phi l_\phi^{-1} \tilde{\phi}) \quad (19c)$$

De forma compacta se puede expresar como:

$$n = \dot{q}_d + L \tanh(KL^{-1} \tilde{q}) \quad (20)$$

en donde, $L = \text{diag}(l_x, l_y, l_\phi)$ representa un parámetro de saturación, mientras que, $k = \text{diag}(k_x, k_y, k_\phi)$ representa la ganancia del controlador.

3.10. Análisis de estabilidad

Para este análisis, se asume un seguimiento perfecto de velocidad, es decir, para este caso en específico $u = u_c$. Entonces, el sistema en lazo cerrado resulta:

$$\dot{q} = \dot{q}_d + L \tanh(KL^{-1} \tilde{q}) \quad (21)$$

Expresando en términos de error $\tilde{q} = q_d - q$ se tiene

$$\dot{\tilde{q}} = -L \tanh(KL^{-1} \tilde{q}) \quad (22)$$

Para el punto de equilibrio $\tilde{q} = 0$ se escoge la siguiente candidata de Lyapunov.

$$\dot{V} = -\tilde{q}^T L \tanh(KL^{-1} \tilde{q}) < 0. \quad (23)$$

Debido a que la función $\tanh(\cdot)$ es impar, es posible concluir que \dot{V} es una función definida negativa. En otras palabras, $\tilde{q}(t) \rightarrow 0$ cuando $t \rightarrow \infty$.

3.11. Simulaciones

En la figura 62 se observa el desempeño del robot durante la tarea de seguimiento de personas. Se observa que el robot converge a la trayectoria descrita por el usuario.

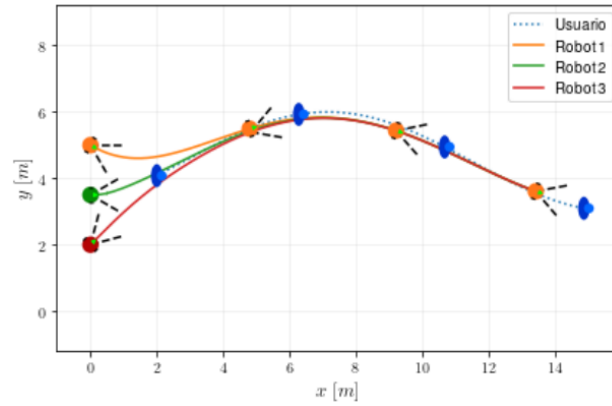


Figura 62: Seguimiento de personas con diferentes condiciones iniciales

4. CAPÍTULO 4: PRUEBAS Y ANÁLISIS DE RESULTADOS

4.1. Seguimiento de personas

Mediante el uso del SDK Nitrack que implementa el robot, el cual tiene la capacidad de captar 19 articulaciones de la morfología corporal humana, se desarrollan las siguientes pruebas correspondiente al seguimiento de personas.

A su vez, en la figura 63 se muestra mediante Rviz la recepción de los datos enviados por el nodo del SDK Nitrack, representando la morfología corporal humana del usuario en las 19 articulaciones.

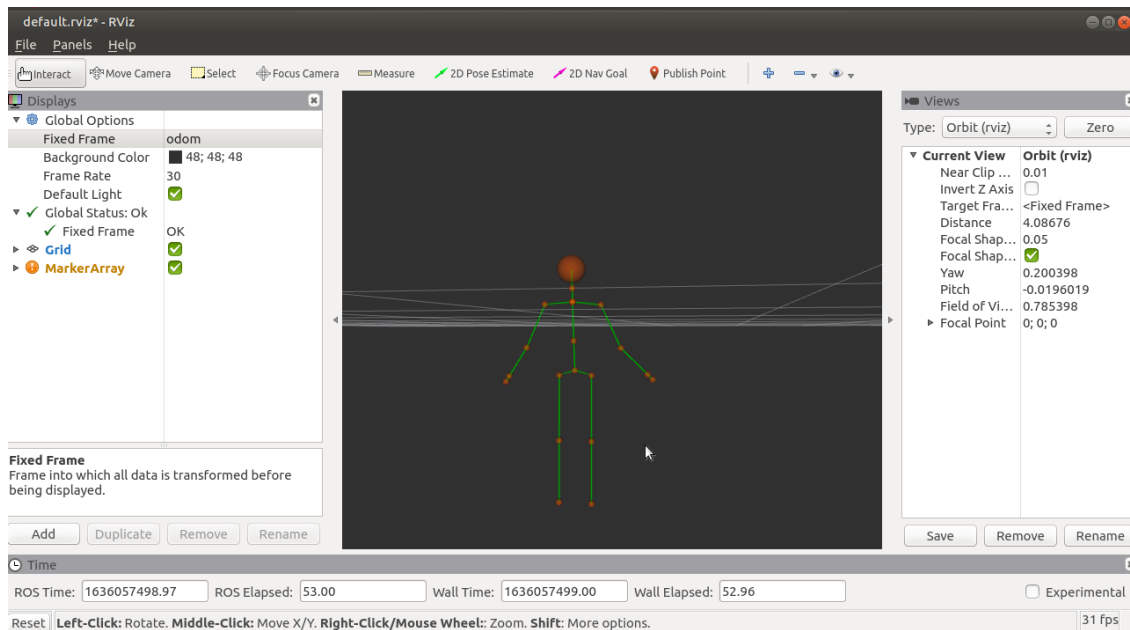


Figura 63: Representación de la morfología corporal humana en Rviz

El ángulo de detección del seguimiento del cuerpo se muestra en la variable z presente en la figura 64, en esta imagen el usuario 1 se encuentra a una distancia promedio de 1,80 metros con respecto al robot y en promedio de 0 grados con respecto al centro de la cámara de profundidad que implementa.



Figura 64: Ángulo de detección del seguimiento de personas (Centro).

En la figura 65, se muestra que el usuario 1, se desplazo hacia el lado derecho, por lo que el ángulo que presenta con respecto a la cámara de profundidad es aproximadamente de -20 grados.

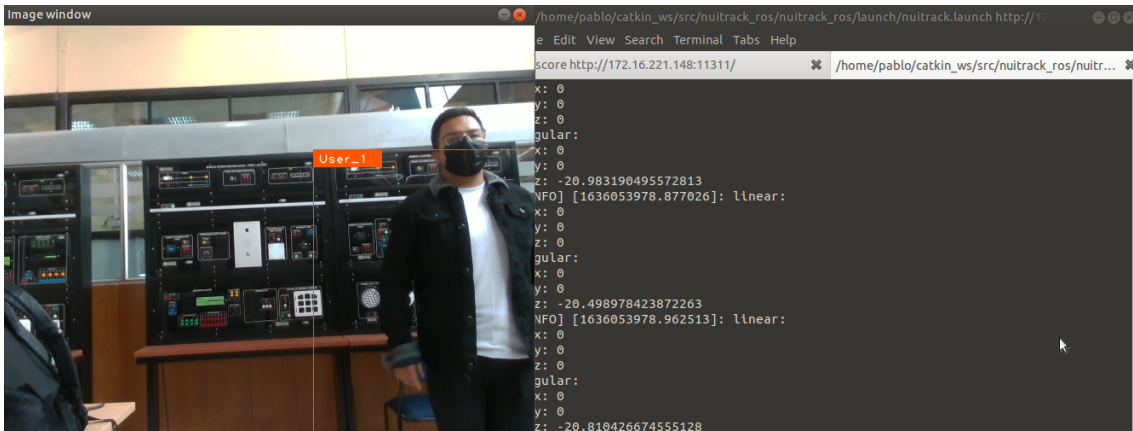


Figura 65: Ángulo de detección del seguimiento de personas (Derecha).

En la figura 66, el usuario 1 se desplaza hacia la izquierda con respecto a la cámara de profundidad, por lo que el ángulo representado por z, varía a un nuevo valor aproximado de 15 grados respecto al robot.



Figura 66: Ángulo de detección del seguimiento de personas (Izquierda).

Y a su vez en la figura 67 se visualiza los datos de las articulación en Rviz, el cual imita los movimientos del usuario distribuidos en 19 articulaciones, permitiendo verificar el correcto funcionamiento del nodo lanzado.

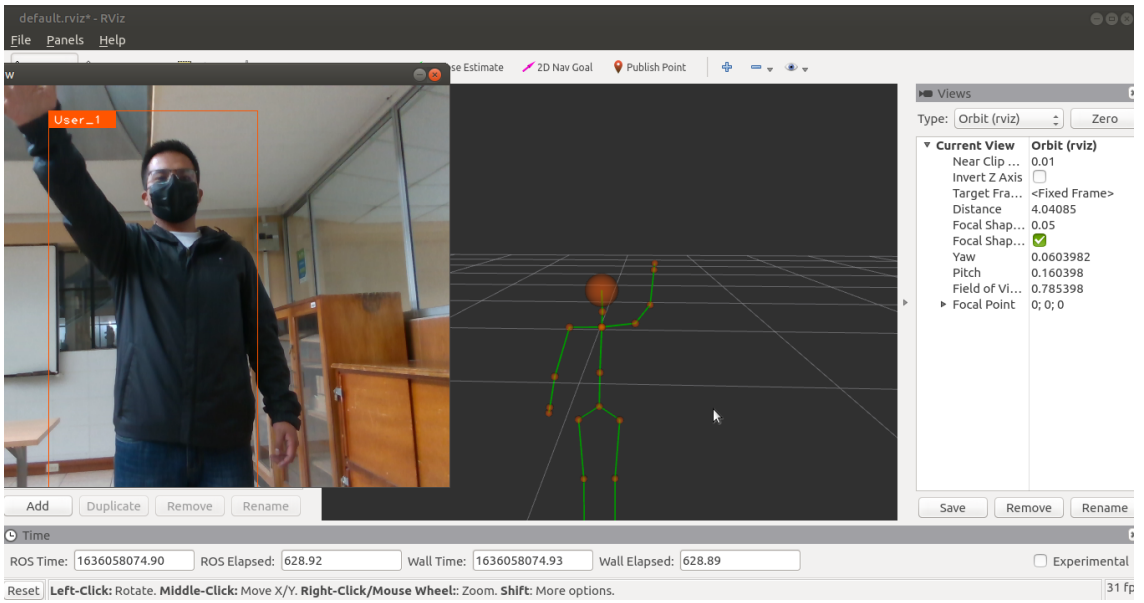


Figura 67: Datos de las articulaciones en Rviz.

Además, en la figura 68 se visualiza la respuesta del SDK nitrack al presentar un nuevo usuario *usuario_2*. En esta figura se aprecia al usuario 1 y 2 realizar movimientos, y a su vez se muestra las articulaciones de cada uno representados en Rviz.

Cada usuario dispone de 19 articulaciones, distribuidas en toda su morfología.

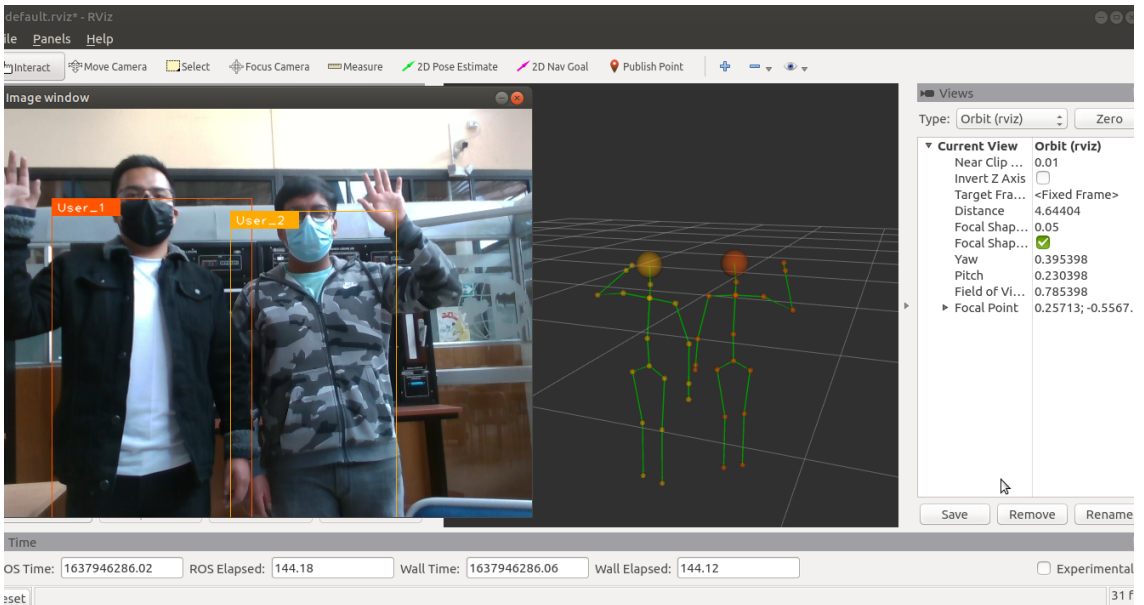


Figura 68: Datos de las articulaciones de 2 usuarios representados en Rviz.

4.2. Mapeo

El mapeo del entorno se realiza haciendo uso del sensor YDLidar que esta montado sobre el Robot, el cual genera una nube de puntos proyectando los objetos que se encuentran en el entorno y se visualiza en Rviz con el tópic “/scan” una serie de puntos rojos que se obtiene del láser como se muestra en la figura 69.

Esto permite conocer los objetos presentes en la trayectoria del robot.

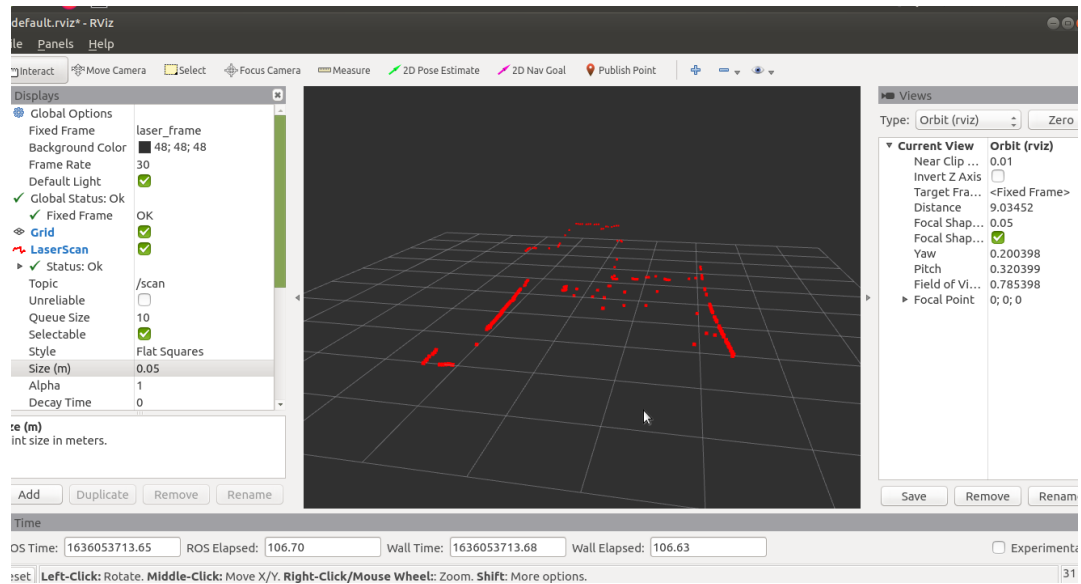


Figura 69: Mapeo del entorno.

En la figura 70, se presenta la visualización del mapeo del entorno en la pantalla del robot, dando a conocer los objetos cercanos al mismo.



Figura 70: Visualización del mapeo del entorno en la pantalla del robot.

4.3. Videollamada

En la figura 71 se presenta una prueba de videollamada en software "Skype", realizada haciendo uso de la cámara web, micrófono y parlante integrados en el robot móvil.

El usuario de la imagen central viene siendo el receptor de llamada por consiguiente el usuario de la derecha superior es quien se encuentra haciendo uso del Robot.

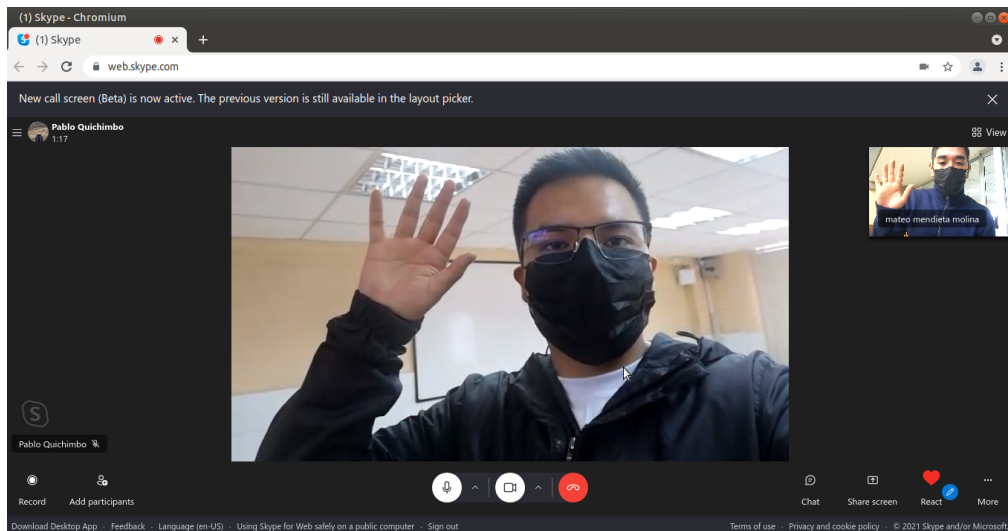


Figura 71: Prueba de la transmisión de videollamada desde el robot.

En la figura 72, se presenta la configuración del sistema de audio y video en el software de Skype, para la parte de entrada y salida de audio se hace uso de una tarjeta de sonido externa conectada al CPU del robot mediante un puerto USB. Mientras para la transmisión de video se hace uso de una cámara web.

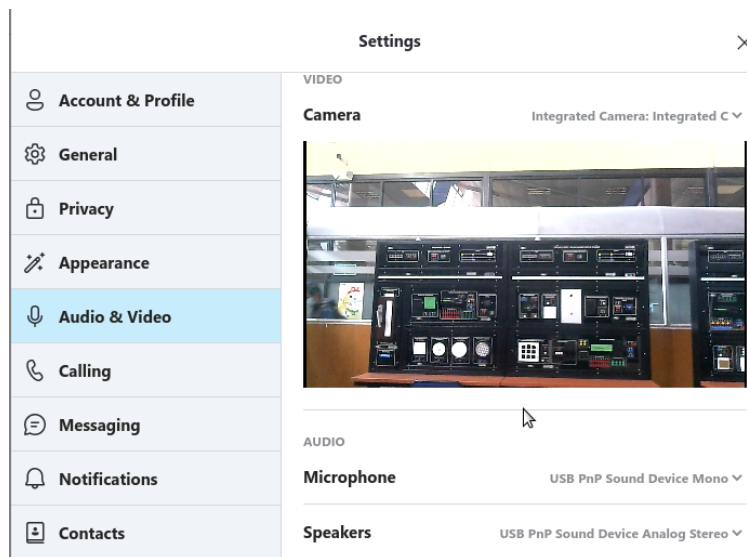


Figura 72: Configuración de audio y video.

4.4. Iluminación

En la figura 73, se muestra la iluminación que dispone el robot en la parte inferior. El cual es controlado directamente desde la interfaz del usuario, permitiendo la visualización de la presencia del robot en entornos de poca iluminación.



Figura 73: Iluminación encendida del robot.



Figura 74: Iluminación apagada del robot.

5. CAPÍTULO 5: CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

Para satisfacer las necesidades de las aplicaciones de telepresencia se consideró ciertos aspectos en diseño que son fundamentales para lograr una buena experiencia de cara al usuario. Se considero la altura optima de 1.32 metros para que un usuario promedio pueda interactuar de manera adecuada con el robot. Debido a que además el mismo cuenta con una pantalla de 10.8 pulgadas ubicada en la parte superior con una inclinación de 45 grados, lo que permite a los usuarios visualizar de manera clara la interfaz del robot. Además, se consideró la colocación del audio y video del robot ya que el usuario deberá escuchar y escucharse de manera clara cuando se encuentra en la video conferencia y a su vez el receptor de la llamada debe poder visualizar al usuario de manera clara. Para la estructura inferior se tomó en cuenta una base que soporte los motores con su respectivo eje y las ruedas de apoyo, para que de esta manera no pierda su estabilidad y mantenga una tracción diferencial óptima. Y a su vez la distribución del peso sea adecuada y el robot pueda moverse libremente y no pierda equilibrio en el frenado. Se busco que el diseño sea lo más compacto posible sin perder su eficiencia y los espacios necesarios para albergar el hardware.

Para la programación del robot se optó por la instalación del sistema operativo de Linux con distribución Ubuntu en su versión 20.04 LTS. Ya que nos permitió la instalación del framework de ROS con la cual se controló al robot, a su vez este sistema nos permitió la instalación optima de los paquetes y librerías necesarias para el funcionamiento de los compontes que alberga el mismo. Para la cámara de profundidad RGBD realsense D435i se instalaron los paquetes que el fabricante dispone y en adición a este se instalaron los paquetes de SDK Nuitrack que tiene compatibilidad con la misma, lo que nos permitió utilizar cada uno de sus sensores para la aplicación de seguimiento de personas planteada, ya que posterior al lanzamiento de los nodos de nuitrack se creó un suscriptor al que pudimos acceder para obtener los valores de ángulo y profundidad de los usuarios. El fabricante del sensor YDLidar también dispone de paquetes para el uso de sus productos por lo que realizamos la instalación de los mismos y lanzamos el nodo correspondiente al modelo adquirido, para posteriormente realizar las pruebas.

Para la implementación del controlador, se utiliza Arduino Mega, porque a diferencia de otros dispositivos, puede manejar seis pines de interrupción. En nuestro ejemplo, usamos

cuatro interrupciones para leer el codificador de velocidad y una interrupción adicional para sincronizar el tiempo de muestreo. Es necesario calibrar correctamente el controlador de bajo nivel, porque en el controlador de alto nivel, se asume que la velocidad de referencia enviada al motor está perfectamente rastreada. En este trabajo, el método de ajuste gráfico se utiliza primero para aproximar, y luego se ajustan los parámetros del controlador para obtener un mejor rendimiento del robot móvil.

Para la prueba de la telepresencia, se realizó una llamada haciendo uso del software de Skype. Previamente a la llamada se inició sesión y se configuró las entradas correctas correspondientes al sistema de audio y video, Una vez lograda la configuración se procedió con la llamada siendo esta eficiente tanto en audio y video. Para la navegación del robot se hace uso de un joystick el cual permitió controlar la velocidad angular y lineal, logrando mediante la implementación de un controlador PID en cada una de las llantas. Mientras que en el seguimiento de personas se desarrolló pruebas de posición del ángulo en el que se encuentra el usuario tanto en el centro como hacía de derecha e izquierda. La medición del ángulo es respecto a la posición de la cámara de profundidad RGBD que dispone el robot. Además, se realizó la verificación del funcionamiento de Nitrack con las 19 articulaciones que dispone, es probado con 1 usuario como con 2 usuarios, dando como resultado la proyección de las 19 articulaciones en cada morfología humana presente en la cámara la misma es proyecta en Rviz. La prueba de funcionamiento del sensor YDLidar se da en un entorno controlado dando como resultado un mapa del entorno actual en el que se encuentra, el mapeo generado se visualiza en la pantalla del robot. Finalmente, se desarrolla la prueba de iluminación que incorpora el robot, la misma es activada como desactivada desde la interfaz del robot, la cual tiene como finalidad la visualización de la presencia del robot en entornos de poca iluminación.

6. Recomendaciones

En la elaboración de este proyecto, surgieron varios inconvenientes que debieron ser resueltos para la culminación del mismo, por ende, vemos prudente dar ciertas recomendaciones que sirvan de ayuda si se busca emprender proyectos de índole parecida.

- En el diseño de la estructura del robot debe considerarse con especial atención el tamaño que tendrá el mismo y el peso que cargara, ya que al ser de tracción diferencial apoyado únicamente por dos llantas en la parte de atrás, este puede perder estabilidad si los pesos no se encuentran debidamente distribuido.
- El tronco del robot debe estar adherido completamente a la superficie inferior del robot,

ya que si no se encuentra de esta manera puede soltarse, o la parte superior puede presentar inestabilidad

- Las partes que se realicen en impresión 3D deben estar correctamente medidas en su diseño dejando un margen de error, debido a que las impresoras carecen de exactitud, por lo que se recomienda un sobre dimensionamiento de cada pieza.
- Para alimentar todo el sistema se debe considerar las corrientes que tendrá cada componente del robot, ya que estos pueden demandar demasiados recursos y el sistema puede empezar a fallar. Por eso además se debe tomar en cuenta una alimentación que satisfaga las demandas de cada componente. Se optó por el uso de una batería de 36V 20A para alimentar todo el robot, y ya que cada componente requiere voltaje diferente se realizó las reducciones necesarias mediante el uso de la placa step down que nos permite reducir el voltaje de hasta 50V en lo que necesitemos con la limitante de 3A máximos en la carga.
- Para la instalación de las librerías SDK se recomienda verificar previamente si el procesador del CPU y el sistema operativo del robot es compatible con cada una de ellas. Puesto que se tuvo ciertos inconvenientes en la instalación del SDK NuiTrack, con la versión de Ubuntu 18.04.5 LTS, lo que se solucionó con la versión de Ubuntu 20.04 LTS.
- Si se busca realizar un robot móvil que albergue varios componentes de hardware ya sean sensores o actuadores, se recomienda usar un CPU con la mayor robustez posible debido a que el CPU utilizado para este robot resultó no ser suficiente para lanzar los diferentes procesos.

7. Trabajo futuro

Para trabajo a futuro se propone el ampliamiento de las aplicaciones del robot, debido que por su diseño y componentes de hardware nos permite plantear otras funcionalidades, como la implementación de bases en el tronco del robot para ser ocupado como cargador, delivery o programación de la interfaz para que sirva de camarero en un espacio de venta de alimentos. Además, debido a que incorpora sensores de navegación, realizar levantamiento de mapas de ambientes para la navegación autónoma o generación de trayectorias.

Referencias

- [1] K. Ogata, *Ingeniería de Control Moderna*. Prentice Hall, 1970.
- [2] L. J. M. Cunha, B., “Introduction to the special issue “robotica 2016”,” *Disponible en: <https://bibliotecas.ups.edu.ec:2582/10.1007/s10846-018-0934-1>*, 2019.
- [3] C. G. C. Yulun Wang., *Interfacing with a mobile telepresence robot*. United States of America Patente US, 2012.
- [4] M. T. F. E. Cordova Lara, S. A., “Diseño y construcción del prototipo de un robot móvil para telepresencia controlado a través de internet,” *Disponible en: <https://hosteleriaecuador.com/producto/camara-de-fermentacion/>*, 2018.
- [5] S. G. Tzafestas, *Introduction to mobile Robot Control*. Elsevier, 2014.
- [6] L.-A. C. Rubio F, Valero F, *A review of mobile robots: Concepts, methods, theoretical framework, and applications*. International Journal of Advanced Robotic Systems, 2019.
- [7] M. Rouse, “Enterprise ai,” *Disponible en: <https://searchenterpriseai.techtarget.com/definition/telepresencerobot>*, 2016.
- [8] M. Luqman, “Ubuntu install of ros noetic,” *Disponible en: <http://wiki.ros.org/noetic/Installation/Ubuntu>*, 2021.
- [9] L. J. D. L. Yoonseok Pyo, Hancheol Cho, *ROS Robot Programming (English)*. ROBOTIS, 12 2017.
- [10] R. Zickler, “Arduino ide setup,” *Disponible en: http://wiki.ros.org/rosserial_arduino/Tutorials/Arduino*, 2020.
- [11] W. K. G. B. Quigley M., Gerkey B., “Joy,” *Disponible en: http://wiki.ros.org/joyPublished_Topics*, 2021.
- [12] J. T. y. R. C. J. C. Montesdeoca Contreras, D. Herrera, “Controllers design for differential drive mobile robots based on extended kinematic modeling,,” *European Conference on Mobile Robots (ECMR)*, 2017.

APÉNDICES

APÉNDICE A: MEDIDAS DE LA ESTRUCTURA BASE DEL PROTOTIPO

A continuación, se muestran las medidas reales de la base del prototipo.

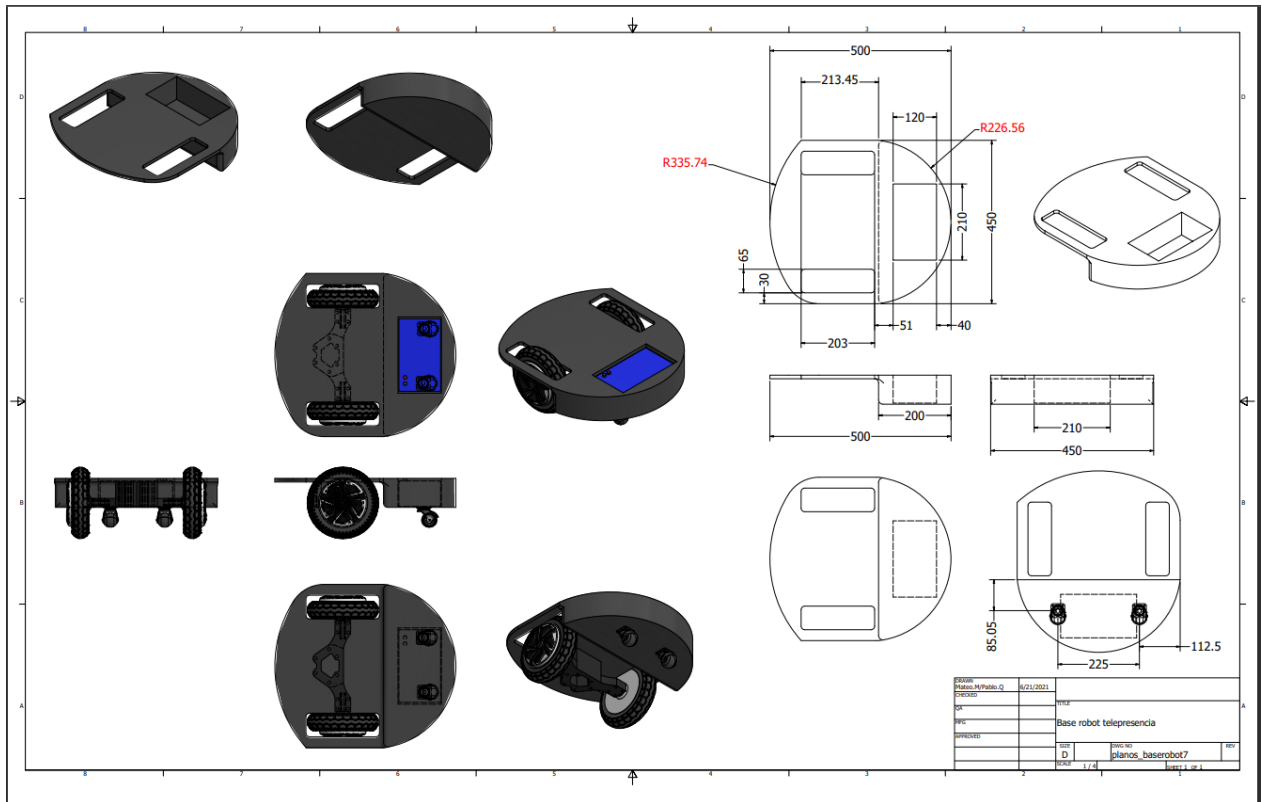


Figura 75: Dimensiones de la estructura base del prototipo

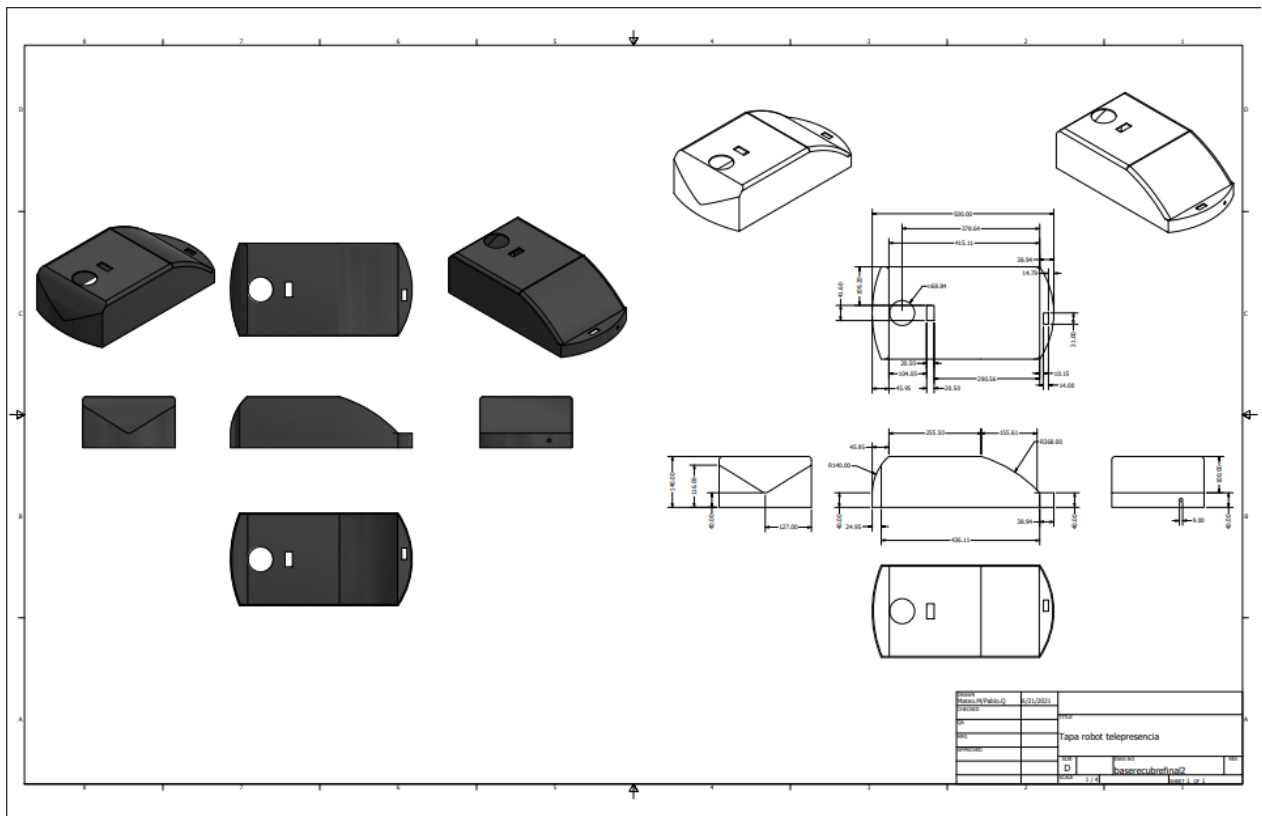


Figura 76: Dimensiones de la estructura tapa del prototipo

APÉNDICE B: Listado de materiales

Item	Cantidad	Valor	Envió	Subtotal
Motores	2	37\$	112\$	149
Soportes	2	31\$	45\$	76
Drivers	6	8\$	38\$	46
Bateria	1	137\$	40\$	177
YDLidar	1	77\$	304	381
Pantalla	1	130\$	30\$	160\$
Parlante	1	50\$	0	50\$
RealSense	1	200\$	30\$	230\$
Microfono	1	15\$	0	15\$
Raspberry	1	110\$	0	110\$
Arduino Mega	1	30\$	0	30\$
Camaraweb	1	17\$	0	17\$
Joystick	1	15\$	0	15\$
Componentes electrónicos	8	30\$	0	30\$
Base del robot	1	90\$	0	90\$
Acople	1	20\$	0	20\$
Ruedas de apoyo	2	5\$	0	5\$
Tapa inferior del robot	1	40\$	0	40\$
Piezas en impresión 3D	1	80\$	0	80\$
Otros	-	100\$	0	100\$
				1.821\$

Tabla 4: Tabla de materiales utilizados

APÉNDICE C: Proceso de Construcción



Figura 77: Partes del robot



Figura 78: Primeras pruebas de funcionamiento Raspberry



Figura 79: Construccion base inferior del robot



Figura 80: Conexion de los componentes del robot

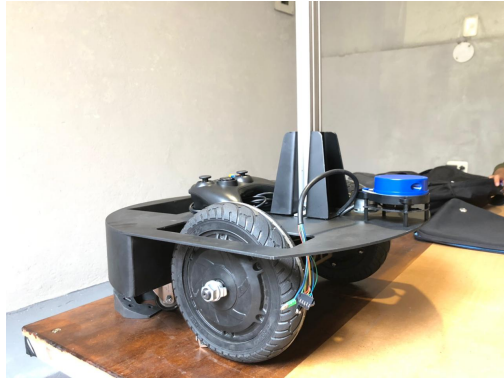


Figura 81: Acople del cuerpo del robot



Figura 82: Prueba de los drivers del robot



Figura 83: Implemetación de la pantalla del robot