

La investigación reportada en esta tesis es parte de los programas de investigación del CICESE (Centro de Investigación Científica y de Educación Superior de Ensenada, Baja California).

La investigación fue financiada por el CONAHCYT (Consejo Nacional de Humanidades, Ciencias y Tecnologías).

Todo el material contenido en esta tesis está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México). El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo o titular de los Derechos de Autor.

CICESE © 2023, Todos los Derechos Reservados, CICESE

Centro de Investigación Científica y de Educación Superior de Ensenada, Baja California



Maestría en Ciencias en Electrónica y Telecomunicaciones con orientación en Instrumentación y Control

Diseño e implementación de un guía robótico para mejorar la movilidad de personas invidentes

Tesis

para cubrir parcialmente los requisitos necesarios para obtener el grado de
Maestro en Ciencias

Presenta:

Abraham Arellano Argumedo

Ensenada, Baja California, México

2023

Tesis defendida por

Abraham Arellano Argumedo

y aprobada por el siguiente Comité

Dr. Jonatán Peña Ramírez

Codirector de tesis

Dr. Alejandro Rodríguez Ángeles

Codirector de tesis

M. C. Ricardo Francisco Núñez Pérez

Dr. Jesús Favela Vara

Dr. Omar Arturo Domínguez Ramírez



Dra. María del Carmen Maya Sánchez

Coordinadora del Posgrado en Electrónica y Telecomunicaciones

Dra. Ana Denise Re Araujo

Directora de Estudios de Posgrado

Resumen de la tesis que presenta Abraham Arellano Argumedo como requisito parcial para la obtención del grado de Maestro en Ciencias en Electrónica y Telecomunicaciones con orientación en Instrumentación y Control.

Diseño e implementación de un guía robótico para mejorar la movilidad de personas invidentes

Resumen aprobado por:

Dr. Jonatán Peña Ramírez

Codirector de tesis

Dr. Alejandro Rodríguez Ángeles

Codirector de tesis

El sentido de la vista juega un papel primordial en la capacidad de orientación, desplazamiento y autonomía de una persona. Es por ello que para las personas ciegas o invidentes, desplazarse en un ambiente desconocido puede ser una experiencia poco placentera, insegura e incluso traumática. Existen distintas herramientas y sistemas de apoyo para ayudar a las personas invidentes a desplazarse con cierto grado de autonomía. Una de ellas y que es la más comúnmente usada debido a su bajo costo, es el llamado bastón blanco, el cual consiste de una varilla ligera plegable que les sirve de guía para desplazarse de manera autónoma. A pesar de que en ciertos ambientes el bastón puede ser de mucha utilidad, presenta varias desventajas. Por ejemplo, la persona tiene que explorar constantemente áreas pequeñas enfrente de él(ella), solo puede detectar obstáculos a nivel del piso y se requiere de muchas horas de entrenamiento para poder beneficiarse de su uso. Por otra parte, existen los perros guía, que son animales altamente entrenados, pero en general es muy costoso adquirirlos, además de que requieren de mucho cuidado, algo que las personas invidentes quizás no estén en condición de realizar. Por lo tanto, sería útil poder crear desarrollos tecnológicos accesibles que permitan a las personas ciegas orientarse y desplazarse a través de ambientes conocidos y desconocidos de forma rápida y segura, y con ello contribuir a mejorar su calidad de vida. Es en este contexto que se enmarca esta tesis. En particular, en este trabajo se presenta la integración de un guía robótico para personas invidentes, el cual se construye utilizando un robot móvil comercial, conectado a un bastón blanco convencional por medio de una interfaz mecánica. Para garantizar la navegación autónoma del guía robótico, se implementa una estrategia de control de seguimiento de trayectoria y se diseña una estrategia de evasión de obstáculos tanto estáticos como dinámicos. Dicha estrategia de evasión se basa en modelos neuronales del tipo Hindmarsh-Rose interconectados. La efectividad de los controladores de seguimiento y evasión se ilustra por medio de simulaciones numéricas. Finalmente, se presentan los resultados de un estudio experimental que se realizó en conjunto con un grupo de seis personas ciegas con el fin de determinar la funcionalidad del guía robótico. Este estudio se realizó en la ciudad de Pachuca Hidalgo, con la colaboración de la Universidad Autónoma del Estado de Hidalgo y la Biblioteca Central Ricardo Garibay. Los resultados obtenidos sugieren que, a futuro y con mayor desarrollo y trabajo de investigación, el prototipo de guía robótico aquí presentado puede llegar a ser de utilidad para mejorar la movilidad de personas invidentes.

Palabras clave: robot móvil, guía robótico, navegación autónoma, invidente

Abstract of the thesis presented by Abraham Arellano Argumedo as a partial requirement to obtain the Master of Science degree in Electronics and Telecommunications with orientation in Instrumentation and Control.

Integration of a robotic assistant to improve the mobility of visually impaired people

Abstract approved by:

Dr. Jonatán Peña Ramírez

Thesis Co-Director

Dr. Alejandro Rodríguez Ángeles

Thesis Co-Director

The sense of sight plays a key role in the ability of a person to orientate and to autonomously displace. For this reason, blind or visually impaired people tend to feel uncomfortable, worried and unsafe while walking in unknown environments. In order to improve the mobility of blind people, there exist several tools and support systems like the white cane, which consists of an extensible aluminum rod with a handle on one end and a tip on the other and allows blind people to orientate and to move autonomously in certain environments, like those having tactile paving. Although the white cane is a good option for enhancing mobility, it has certain disadvantages, like for example: the person should constantly scan small areas in order to determine whether the path is free of obstacles or not, the type of obstacles that can be identified is limited, and to master the use of the white case the person has to follow several training programs. On the other hand, there exist guide dogs that are trained to help blind people to navigate and to avoid obstacles. However, buying a guide dog may be costly. Therefore, it may be interesting to create technical developments for allowing blind people to orientated and to displace through known and unknown environments in a safety and confident manner, which ultimately may result in an improvement of their quality of life. For this reason, this thesis aims to integrate a robotic guide by using a commercial mobile robot, which is connected to a white cane through a mechanical interface. In order to guarantee the autonomous navigation of the robotic guide, a tracking controller is implemented and a suitable collision avoidance strategy for static and dynamic obstacles is designed using pairs of Hindmarsh-Rose neuronal models interacting via static and dynamic couplings. The effectiveness of the tracking and collision avoidance controllers is illustrated by means of numerical simulations. Furthermore, an experimental study with a group of blind people is conducted in order to validate the functionality of the robotic guide. The study was conducted at Pachuca, Hidalgo in collaboration with Universidad Autonoma del Estado de Hidalgo and the Central Library Ricardo Garibay. The obtained results suggest that, in the future, the robotic guide may be a good alternative for improving the mobility of blind people.

Keywords: mobile robot, robotic guide, autonomous navigation, visually impaired person

Dedicatoria

Para el doc Jonatán Peña y Elins Ricardo Nuñez

Agradecimientos

Al Centro de Investigación Científica y de Educación Superior de Ensenada, Baja California (CICESE) por permitirme cursar los estudios para obtener el grado de Maestro en Ciencias.

Al Consejo Nacional de Humanidades, Ciencias y Tecnologías (CONAHCyT) por el apoyo económico durante mis estudios.

A la Universidad Autónoma del Estado de Hidalgo (UAEH) por brindarme la posibilidad de realizar una estancia.

A la Biblioteca Central del Estado de Hidalgo Ricardo Garibay por prestar sus instalaciones para los experimentos.

A mis codirectores de tesis, Dr. Jonatán Peña Ramírez y Dr. Alejandro Rodríguez Ángeles por darme la oportunidad de colaborar en este proyecto.

Al M.C. Ricardo Francisco Nuñez Pérez el famoso Elins, por haber compartido sus conocimientos y mostrarse amigo.

Al Dr. Omar Arturo Domínguez Ramírez por ser tan excelente anfitrión.

Al Dr. Jesús Favela Vara por su tiempo y esfuerzo a lo largo de este proyecto.

A Alexis, Arturo, Zacarías, Ivonne y todas las personas que participaron en los experimentos.

Al todo el taller de maquinado que se la rifaron con la interfaz, y en especial al sr. Francisco que la fabricó.

A mi prometida Daniella que me apoya en todo. Te amo.

A mi familia y amigos que han influido a lo largo de mi vida.

Nuevamente al Dr. Jonatán particularmente por su amistad, apoyo y cobertura.

Y a Jesucristo que es la imagen del Dios invisible, porque todas las cosas provienen de Él, y existen por su poder. A Él sea la gloria por siempre, Amén.

Tabla de contenido

	Página
Resumen en español	ii
Resumen en inglés	iii
Dedicatoria	iv
Agradecimientos	v
Lista de figuras	viii
Lista de tablas	xi
Capítulo 1. Introducción	
1.1. Antecedentes	2
1.2. Panorama general	4
1.2.1. Motivación	4
1.2.2. Planteamiento del problema	5
1.2.3. Hipótesis	5
1.2.4. Objetivo general	6
1.2.5. Objetivos específicos	6
1.2.6. Contribuciones	6
1.2.7. Estructura de la tesis	7
Capítulo 2. Preliminares	
2.1. Modelo cinemático de un robot móvil tipo unicycle	8
2.2. Modelo neuronal Hindmarsh-Rose (HR)	9
2.3. Consideraciones para la selección del robot móvil usado en la integración del guía robótico	10
2.4. Características y especificaciones del robot móvil utilizado	11
2.5. Entorno ROS	13
2.6. Red ROS	13
Capítulo 3. Metodología	
3.1. Control para seguimiento de trayectorias	15
3.2. Diseño de la trayectoria deseada	17
3.3. Concatenación de trayectorias	20
3.4. Estrategia de evasión de obstáculos	21
3.4.1. Evasión lateral	22
3.4.2. Evasión frontal	24
3.5. Controlador híbrido	26
3.6. Diseño de la interfaz para conectar el robot móvil con un bastón blanco	27
Capítulo 4. Resultados numéricos	
4.1. Resultados numéricos: control de seguimiento de trayectoria sin evasión	30
4.2. Resultados numéricos: generación de trayectoria	33
4.3. Resultados numéricos: Estrategia de evasión de obstáculos	34

4.4.	Resultados numéricos: controlador híbrido	37
4.5.	Resultados experimentales	39
Capítulo 5. Validación experimental del prototipo		
5.1.	Ajuste de los parámetros del robot	42
5.1.1.	Experimentos con un grupo de voluntarios ciegos	44
5.1.1.1.	Experimento 1: trayectoria en L	46
5.1.2.	Experimento 2: trayectoria zigzagante	47
5.2.	Resultados de los cuestionarios	49
Capítulo 6. Discusión y conclusiones		
6.1.	Conclusiones	52
6.2.	Trabajo a futuro	53
Literatura citada		54
Anexos		56

Lista de figuras

Figura	Página
1. Módulo sensor desarrollado por la Universidad de Stuttgart (Hub et al. (2003)).	2
2. Sistema guía para navegación autónoma de personas invidentes (Ulrich & Borenstein (2001)).	3
3. Sistema guía para navegación autónoma de personas invidentes (Lee et al. (2013)).	4
4. Diagrama esquemático del robot diferencial tipo 2.0 (Juarez-Lora & Rodriguez-Angeles (2023)).	8
5. Comportamiento Bursting, $l=3.0$	9
6. Comportamiento Bursting Caótico, $l=3.3$	10
7. Comportamiento Spiking, $l=3.7$	10
8. Sujeto de estudio desplazándose en un espacio abierto.	11
9. Robot ROSbot 2.0 Pro de la marca Husarion.	12
10. Diagrama a bloques de los diferentes módulos que integran el robot móvil ROSbot 2.0 PRO.	12
11. Representación de seguimiento de trayectoria implementando un robot virtual de referencia.	16
12. El robot móvil se representa por el círculo azul, el punto meta por el círculo verde y la separación entre ambos por d_{meta}	19
13. Coordenadas: Inicio(2,2), coordenada deseada 1: (7,1.5), meta final: (2,4.5). $d_{prox} = 0.3m$, $b_r = 0.2m$	21
14. Distribución de los sensores de proximidad en el robot móvil. En total se consideran 11 sensores: 5 para detección de obstáculos del lado izquierdo del robot (indicados en color azul), 5 para detección de obstáculos por el flanco derecho (mostrados en verde) y uno para detección frontal (en color rojo) (Sanchez et al. (2021)).	22
15. Elementos del controlador híbrido.	26
16. Grados de libertad considerados para la interfaz usuario-ROSbot.	27
17. Diseño de mecanismo con grado de libertad alrededor del eje Z.	28
18. Vista frontal del producto implementado en ROSbot: grado de libertad sobre el eje Z (1), segundo grado de libertad sobre el eje Y (2), y bastón y adaptador (3).	29
19. Seguimiento de trayectoria. Rojo: trayectoria deseada. Azul: trayectoria del robot móvil. Puede verse que después de un tiempo el robot móvil converge a la trayectoria deseada.	31
20. Seguimiento de trayectoria. Rojo: trayectoria deseada. Azul: trayectoria del robot móvil. Puede verse que se alcanza el objetivo de control (seguimiento).	31
21. Validación numérica del controlador de seguimiento para diferentes trayectorias deseadas (indicadas en color rojo). En todos los casos, el robot móvil logra alcanzar la trayectoria deseada, lo que muestra la efectividad del controlador.	32
22. Generación de una trayectoria deseada utilizando cuatro puntos meta.	33

23. Ejemplo en el que la trayectoria del robot real diverge de la trayectoria deseada, al no lograr entrar en la zona de llegada del punto meta 2. Para evitar este tipo de escenarios, se debe realiar una buena sintonización de las ganancias del controlador, así como garantizar que el robot virtual de referencia es lo 'suficientemente lento' para dar tiempo a que el robot real llegue al punto meta. 34
24. Número de espigas de los potenciales de membrana. Rojo: n_1 . Azul: n_2 . En el intervalo $0 \leq t < 8$ no hay ningún obstáculo en el camino del robot y se puede observar que $n_1 = n_2$. Sin embargo, en el intervalo $8 \leq t < 14$ se agrega un obstáculo en el flanco derecho del robot y como consecuencia $n_1 \neq n_2$. Finalmente, en el intervalo $14 \leq t \leq 30$ el robot ya evadió el obstáculo y el número de espigas vuelve a ser el mismo en los potenciales de membrana, es decir se tiene que $n_1 = n_2$ 35
25. Panel superior: desplazamiento del robot en el plano XY. Panel inferior: Orientación del robot. Nótese que cuando el obstáculo es detectado al tiempo $t = 8$, el robot gira hacia la izquierda. Una vez que evade el objeto (al tiempo $t = 14$) el robot empieza a presentar un comportamiento constante. 36
26. Potenciales de membrana de las neuronas N_1 y N_2 descritas en (24) y (25). En el intervalo $0 \leq t < 8$ no hay obstáculos alrededor del robot y las neuronas están sincronizadas. Cuando se agrega un obstáculo en el flanco derecho del robot en el intervalo $8 \leq t < 14$, las neuronas pierden la sincronía y la recuperan posteriormente en el intervalo $14 \leq t < 30$ cuando ya el obstáculo ha sido esquivado. 36
27. En la gráfica superior se observa el desplazamiento del robot en el plano XY. En la gráfica inferior se observa la variación de la orientación del robot. Se puede apreciar que el robot realiza un giro sobre su propio eje para evadir el obstáculo frontal y entonces continuar avanzando en línea recta. Nótese que después de evadir el obstáculo, la orientación del robot cambia aproximadamente 140 grados con respecto a la orientación inicial. 37
28. Resultados numéricos con controlador híbrido. La trayectoria del robot móvil (línea azul) converge a la trayectoria circular deseada (línea negra), gracias al controlador de seguimiento de trayectoria (10). Sin embargo, una vez que el obstáculo (cuadro negro en la figura) es detectado, se activa el controlador de evasión de obstáculos (34), y el robot gira hacia la izquierda y logra evadir el obstáculo. Una vez que el obstáculo es evadido, se vuelve a activar el controlador de seguimiento de trayectoria y el robot regresa a la trayectoria deseada. 38
29. Panel superior: velocidades de rueda. Azul: velocidad de rueda derecha. Roja: velocidad de rueda izquierda. Cuando el controlador de seguimiento está activo, las velocidades de rueda son suaves y tienden a estabilizarse alrededor de un valor constante. Sin embargo, cuando el obstáculo es detectado, se activa el controlador de evasión de obstáculos y, mientras el objeto está a una distancia detectable, las velocidades de rueda exhiben oscilaciones de alta frecuencia. Panel inferior: banderas para identificar qué controlador está activo. Un valor de 1 indica que el controlador de evasión está activo, un valor de cero corresponde al caso en que solamente el controlador de seguimiento de trayectoria está activo. Negro: bandera para el controlador de evasión frontal. Azul: bandera para el controlador de evasión lateral. 39
30. Distribución de las lecturas de interés en el sensor LiDAR. 40

Figura	Página
31. Resultados experimentales del control de seguimiento por retroalimentación por odometría.	40
32. Resultados experimentales. Velocidades del robot virtual y velocidades enviadas al robot.	41
33. Sesión de prueba con un voluntario con ceguera y su familia.	42
34. Resultados experimentales con el robot guiando a una persona invidente a lo largo de una trayectoria en L.	43
35. Alexis guiado por el robot instantes antes de llegar a un punto meta.	44
36. Reunión con los participantes del experimento en la que se explicó el protocolo experimental a seguir.	45
37. Resultados del experimento de adiestramiento. Las trayectorias descritas por los voluntarios se muestran con curvas de colores, mientras que la trayectoria del robot de referencia se representa con las rectas de color negro.	46
38. Comparación de las trayectorias ejecutadas por los participantes.	47
39. Trayectoria en zig-zag usada en el segundo experimento.	47
40. Gráficas de las trayectorias de cada voluntario en el segundo experimento.	48
41. Robot operando de manera indebida.	48
42. Ejemplo de configurar una ventana de comandos de la laptop de control, para navegar en el SO del ROSbot.	57
43. Algunos tópicos y nodos disponibles en el sistema operativo del ROSbot que presenta el fabricante [7].	59
44. Configuración para una red ROS entre una laptop de control y ROSbot.	60
45. Configuración del ROSbot como dispositivo maestro.	61
46. Ventana de configuración para el ambiente ROS en MATLAB.	62
47. Suscripción y Publicación del nodo Simulink para compartir mensajes.	63
48. Nodos habilitados para el control de velocidades con retroalimentación por odometría. .	63
49. Resultados visualizados en Matlab, valores de posición dados por el robot.	64
50. Lectura de LiDAR A3 en Simulink.	65
51. Cuestionario NASA TLX.	66
52. Cuestionario 1.	67
53. Cuestionario 2.	68

Lista de tablas

Tabla		Página
1.	Relación de voluntarios.	46
2.	Tiempo que tomó a cada participante completar el experimento 1.	47
3.	Tiempo que tomo a cada participante concluir el experimento 2.	49
4.	Tabla de interpretación de resultados NASA TLX	50
5.	Tabla de ponderación por dimensión y por participante.	50
6.	Tabla normalizada de puntaje de esfuerzo por dimensión y por usuario.	50
7.	Resultados NASA TLX para la tarea de guía lazarillo con ROSbot y el controlador de seguimiento de trayectoria.	51

Capítulo 1. Introducción

La ceguera así como la discapacidad visual aguda son padecimientos que impactan de manera drástica en la vida de las personas que las padecen. De hecho, debido a que la mayoría de la información que se requiere para desarrollar las actividades propias de la vida cotidiana se obtiene a través de la visión, las personas ciegas o invidentes experimentan grandes dificultades para realizar diversas tareas, por simples que éstas sean, de manera independiente. Más aún, la discapacidad visual afecta el aspecto social ya que una persona invidente tiende a aislarse, deprimirse o sufrir periodos de ansiedad.

Una limitación importante de las personas invidentes es el reconocimiento del entorno en que se encuentran, así como la movilidad, esto debido a que el sentido de la vista juega un papel primordial durante el desplazamiento de una persona de un punto a otro, ya sea en un ambiente conocido o desconocido. En particular, para las personas ciegas, desplazarse en un ambiente desconocido puede ser una experiencia poco placentera, insegura e incluso traumática. Aún en ambientes conocidos, la movilidad de las personas con discapacidad visual puede verse comprometida (Saitis & Kalimeri, 2016).

Con el fin de mejorar la calidad de vida de las personas invidentes, en particular su movilidad, se han desarrollado diferentes herramientas y sistemas de apoyo, como lo son el llamado bastón blanco y el uso de perros guía. Por un lado, el bastón blanco es un elemento que permite a las personas invidentes desplazarse con cierto nivel de autonomía a la vez que les ayuda a detectar posibles obstáculos localizados al nivel del piso o superficie de desplazamiento (Attia & Asamoah, 2020). Este tipo de herramienta es ideal para desplazarse en zonas que tengan superficies podotáctiles. Por otra parte, los perros guía son un excelente apoyo para las personas ciegas ya que les permiten moverse con autonomía y seguridad por diferentes ambientes (Glenk et al., 2019). Además, los perros guía también están entrenados para realizar diferentes tareas como presionar botones de ascensor, abrir puertas, y asistir en actividades cotidianas como lo puede ser vestirse o ir al baño, entre otras.

También, se han desarrollado aplicaciones móviles para apoyar a las personas invidentes a reconocer su entorno y desplazarse a través del mismo. Por ejemplo, en 2016 se creó la aplicación Lazzus, la cual proporciona información relevante del entorno en que la persona se encuentra, a través de un campo de visión auditivo y utilizando diversas fuentes de datos como Google Places y Open Street Maps (Real & Araujo, 2019).

De igual manera, se ha desarrollado tecnología asistencial para personas invidentes, las cuales proporcionan ayuda a nivel perceptivo o a nivel conceptual. A nivel perceptivo, se busca que la persona obtenga información del entorno a través de otros sentidos como, por ejemplo, el sentido del tacto o del oído.

Por otra parte, a nivel conceptual se busca usar modelos que ayuden a determinar las posibles rutas de navegación en un determinado entorno (Lahav et al., 2012).

Es en este contexto que esta tesis se enmarca. En particular, en este trabajo se presenta el desarrollo de un guía robótico para asistencia de personas ciegas o con discapacidad visual aguda. Para ello, se empleará la Teoría de Control y robótica móvil.

1.1. Antecedentes

En las últimas décadas se han llevado a cabo una variedad de desarrollos tecnológicos con la finalidad de mejorar la movilidad de personas invidentes. Estos sistemas pueden clasificarse en tres grupos: asistentes electrónicos de navegación (Electronic Travel Aids, ETAs), asistentes electrónicos de orientación (Electronic Orientation Aids, EOAs), y dispositivos de localización por posición (Position Locator Devices, PLDs) (Xu et al., 2023).

En particular, los ETAs basan su funcionamiento en la detección y evasión de obstáculos mientras el usuario se desplaza. Por ejemplo, en (Hub et al., 2003) se presenta una ETA que consiste en un módulo sensor para asistir a personas invidentes en ambientes cerrados, véase Figura 1. El módulo contiene dos cámaras, un teclado y una bocina, los cuales están conectados a una computadora portátil. El usuario puede interactuar con el módulo sensor y obtener información respecto a su orientación posición y navegación, así como algunas características de obstáculos cercanos. Algunas ventajas de este dispositivo son su diseño modular, que trabaja en tiempo real y que la interfaz es muy amigable con el usuario. Una de las desventajas es que el escaneo del área se hace de manera manual cada vez que el usuario presiona un botón.



Figura 1. Módulo sensor desarrollado por la Universidad de Stuttgart (Hub et al. (2003)).

Otro desarrollo relevante es el llamado 'guide cane' presentado en (Ulrich & Borenstein, 2001), el cual consiste de una varilla tipo bastón conectada a un servomecanismo, el cual a su vez está montado sobre un par de ruedas, tal y como se aprecia en la Figura 2. El dispositivo también está equipado con sensores ultrasónicos con los cuales se pueden detectar obstáculos pequeños en el piso y obstáculos laterales, como por ejemplo paredes. La información capturada por los sensores se procesa por un algoritmo de evasión de obstáculos de tal manera que el mecanismo cambia de dirección cuando encuentra objetos frente a él. Una de las ventajas de este dispositivo es que el escaneo del área de interés y la detección de obstáculos se hace de manera automática. Por otra parte, una desventaja es que el área de escaneo es relativamente pequeña y no se pueden detectar obstáculos que estén por arriba de la cintura de la persona, por ejemplo, objetos que estén colgando.

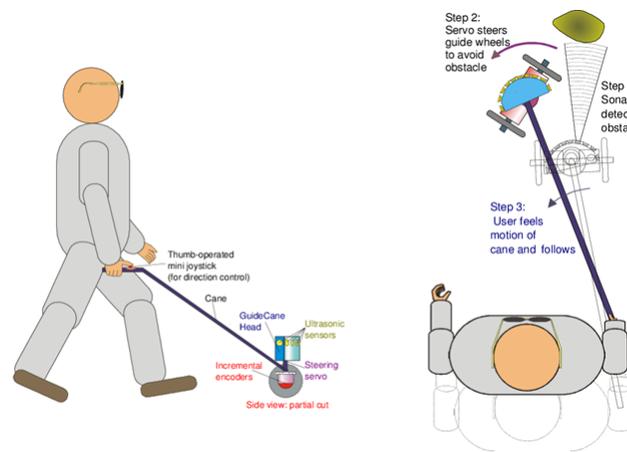


Figura 2. Sistema guía para navegación autónoma de personas invidentes (Ulrich & Borenstein (2001)).

Un tercer sistema de apoyo es el llamado robot social descrito en (Lee et al., 2013) y mostrado en la Figura 3. El sistema consiste de cuatro módulos: sensado, navegación, sistema de control remoto y una plataforma social. El módulo de sensado utiliza un dispositivo Kinect para reconstruir el ambiente en el que la persona se desplaza. El módulo de navegación recomienda la ruta óptima usando GPS y almacena en una base de datos los lugares frecuentes del usuario. Por otra parte, el módulo de control remoto permite operar el robot móvil en diferentes modos y en diferentes escenarios y terrenos. Finalmente, la plataforma social permite al usuario interactuar con otros usuarios y compartir, por ejemplo, rutas y lugares de interés, y permite la navegación en grupo.

Una de las ventajas de este sistema es que proporciona una ruta óptima y libre de obstáculos estáticos y está diseñado de tal manera que contribuye a reducir el miedo de las personas invidentes a desplazarse por ambientes desconocidos. Como punto en contra, este sistema es voluminoso y difícil de transportar.

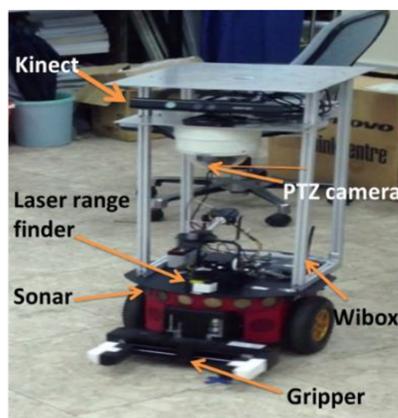


Figura 3. Sistema guía para navegación autónoma de personas invidentes (Lee et al. (2013)).

Además de los sistemas de apoyo arriba mencionados, existen otros desarrollos tecnológicos pensados para mejorar la movilidad de personas invidentes o débiles visuales. Por ejemplo, en (Kayukawa et al., 2020), se propone el diseño de un sistema de navegación basado en un robot móvil. El sistema utiliza sensores ultrasónicos para detectar obstáculos y una palanca háptica que simula un bastón y dirige a la persona en la dirección correcta. Por otra parte, en (Capi & Toda, 2011), se introduce y valida experimentalmente un sistema robótico equipado con sensor visual, detectores láser y bocinas, que proporciona información auditiva a la persona y el cual puede ser utilizado en ambientes interiores y exteriores desconocidos. Una característica interesante de este sistema es que es capaz de distinguir entre objetos y humanos. Finalmente, otra alternativa relevante es la que se presenta en (Toha et al., 2015), en el que se guía a una persona invidente a través de una ruta predefinida. El robot puede detectar cuando el usuario quiere aumentar o disminuir la velocidad.

1.2. Panorama general

En esta sección se presentan los detalles generales de este proyecto de tesis, lo cual incluye la motivación, planteamiento del problema e hipótesis, objetivos general y específicos, así como la estructura de tesis.

1.2.1. Motivación

De acuerdo con la organización mundial de la salud (OMS), en el año 2015 había 36 millones de personas ciegas, con una estimación de 114.6 millones para el año 2050.

La ceguera es una condición física que tiene un fuerte impacto en la calidad de vida de las personas que la padecen y es un factor importante en la generación de sentimientos de ansiedad, aislamiento social y depresión. Por otra parte, la falta o pérdida de visión limita la movilidad de las personas invidentes.

Por lo tanto, en este proyecto se busca desarrollar un prototipo de un guía robótico con la meta final de mejorar la movilidad de personas invidentes o con debilidad visual, a través de la investigación y del desarrollo tecnológico.

1.2.2. Planteamiento del problema

Una de las herramientas más comúnmente utilizadas por las personas invidentes, debido a su bajo costo, es el llamado bastón blanco, el cual consiste de una varilla ligera plegable que les sirve de guía para desplazarse de manera autónoma. A pesar de que en ciertos ambientes el bastón puede ser de mucha utilidad, presenta varias desventajas. Por ejemplo, el usuario(a) tiene que explorar constantemente áreas pequeñas enfrente de él(ella), solo puede detectar obstáculos a nivel del piso y el usuario(a) requiere de muchas horas de entrenamiento para poder beneficiarse de su uso. Por otra parte, existen los perros guía, que son animales altamente entrenados, pero son costosos. Además, requieren de mucho cuidado, algo que las personas invidentes quizás no estén en condición de realizar.

Por lo tanto, en este proyecto se pretende proporcionar una alternativa tecnológica al bastón blanco y al perro guía que permita a una persona invidente moverse de manera autónoma en ciertos ambientes interiores y exteriores como, por ejemplo, en aeropuertos y centros comerciales. Para ello, se pretende explotar resultados actuales en el área de robótica móvil. En particular, se adecuarán e implementarán algoritmos de seguimiento de trayectoria y evasión de obstáculos en un robot móvil, véase por ejemplo [6,7], y se diseñará una interfaz mecánica que permita al usuario seguir la trayectoria que le imponga.

1.2.3. Hipótesis

El uso de un guía robótico basado en un robot móvil puede contribuir a mejorar la movilidad de personas invidentes, al proporcionar un desplazamiento seguro y autónomo, en ambientes conocidos y desconocidos.

1.2.4. Objetivo general

Integrar un sistema de apoyo basado en un robot móvil con control de seguimiento de trayectoria y una estrategia de evasión de obstáculos, que permita a una persona invidente moverse de manera autónoma en ambientes desconocidos, con superficies planas y en presencia de objetos estáticos y dinámicos.

1.2.5. Objetivos específicos

- Identificar las características físicas y técnicas que debe tener el robot móvil que se usará para la integración del guía robótico
- Diseñar una ley de control de seguimiento de trayectoria, junto con una estrategia de evasión de obstáculos para el robot móvil.
- Diseñar y construir una interfaz que comunique al usuario con el robot.
- Realizar simulaciones numéricas del guía robótico utilizando software especializado, por ejemplo MatLab.
- Con la participación de al menos una persona invidente, evaluar de manera experimental el funcionamiento del prototipo construido

1.2.6. Contribuciones

Las contribuciones de este trabajo son: 1) se construyó un primer prototipo funcional de un guía robótico utilizando el robot móvil RosBot 2.0, 2) El algoritmo de navegación se basa en odometría y no en visión, lo que reduce el costo computacional 3) se logró realizar un estudio experimental relacionado con la utilidad del prototipo construido para lo cual se tuvo la participación de un grupo de personas invidentes. Dicho estudio se llevó a cabo en las instalaciones de la Universidad Autónoma del Estado de Hidalgo y en las instalaciones de la Biblioteca Central del Estado de Hidalgo 'Ricardo Garibay'.

1.2.7. Estructura de la tesis

Esta tesis está estructurada de la siguiente manera. El Capítulo 1 presenta una breve introducción al tema de movilidad de personas invidentes, así como una descripción de algunos desarrollos tecnológicos existentes para facilitar el desplazamiento autónomo de personas invidentes y, finalmente, se proporciona un panorama general de este trabajo de tesis.

Después, en el Capítulo 2 se discuten los modelos implementados para el control del robot móvil, y también se describe el procedimiento utilizado para seleccionar el robot móvil utilizado en la integración del guía robótico.

Por otra parte, la metodología utilizada en esta investigación se presenta en el Capítulo 3. Esto incluye la implementación del controlador de seguimiento de trayectoria, el diseño de la estrategia de evasión de obstáculos, así como el diseño y construcción de la interfaz mecánica para conectar el robot móvil con un bastón blanco.

Posteriormente, el Capítulo 4 presenta los resultados numéricos y experimentales obtenidos y en el Capítulo 5 se investiga experimentalmente la funcionalidad del guía robótico con la colaboración de un grupo de personas invidentes.

Finalmente, el Capítulo 6 presenta las conclusiones generales de esta tesis así como algunas recomendaciones para trabajo futuro.

Capítulo 2. Preliminares

En este capítulo se presenta una breve descripción de los modelos implementados para el control del robot móvil y se describe el procedimiento utilizado para seleccionar el tipo de robot que ayudará la integración del guía robótico.

2.1. Modelo cinemático de un robot móvil tipo unicycle

El modelo cinemático de un robot diferencial tipo 2.0, también llamado unicycle, está descrito por el siguiente conjunto de ecuaciones diferenciales (Juarez-Lora & Rodriguez-Angeles, 2023)

$$\begin{aligned} \dot{x} &= V \cos \theta, \\ \dot{y} &= V \sin \theta, \\ \dot{\theta} &= W, \end{aligned} \quad (1)$$

donde el punto (x, y) describe las coordenadas del centro de rotación del robot con respecto al marco de referencia fijo y θ representa la orientación, como se muestra en el diagrama esquemático mostrado en la Figura 4.

Por otra parte, V y W representan las velocidades traslacional y rotacional, respectivamente. En general, dichas velocidades son consideradas como las entradas de control al robot.

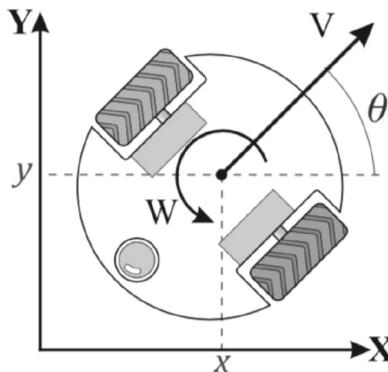


Figura 4. Diagrama esquemático del robot diferencial tipo 2.0 (Juarez-Lora & Rodriguez-Angeles (2023)).

2.2. Modelo neuronal Hindmarsh-Rose (HR)

Existen diversos modelos neuronales reportados en la literatura, como lo son el de Hodgkin y Huxley, el de FitzHugh-Nagumo, y el de Izhikevich. Sin embargo, en este trabajo, se considera el modelo neuronal de Hindmarsh-Rose, ya que es uno de los más completos en el sentido que puede reproducir diferentes formas de onda observadas en neuronas reales. El modelo, el cual fue originalmente propuesto por J.L. Hindmarsh y R.M. Rose en 1984, está descrito por el siguiente conjunto de ecuaciones diferenciales (Hindmarsh et al., 1984)

$$\begin{aligned}\dot{x} &= y + bx^2 - ax^3 - z + I, \\ \dot{y} &= c - dx^2 - y, \\ \dot{z} &= r[s(x - x_0) - z],\end{aligned}\tag{2}$$

donde x describe el potencial de membrana y las variables y, z son variables de recuperación y x_0 la condición inicial de x . Por otro lado, a, b, c, d, r, s son parámetros del sistema y la corriente sináptica se representa por I .

Este modelo exhibe al menos tres distintos comportamientos dinámicos, a saber: espigues (spiking), ráfagas periódicas (bursting) y ráfagas caóticas (bursting caótico). Para cambiar entre comportamientos, basta con variar la corriente sináptica I . Por ejemplo, si se consideran los parámetros $a = 1$, $b = 3$, $c = 1$, $d = 5$, $s = 4$, y $r = 0.005$, con condición inicial $x_0 = -1.6$, entonces pueden observarse los tres comportamientos antes mencionados, usando diferentes valores de I , tal y como se discute en (Velasco Equihua & Ramirez, 2018) y se muestra en las Figuras 5 a 7.

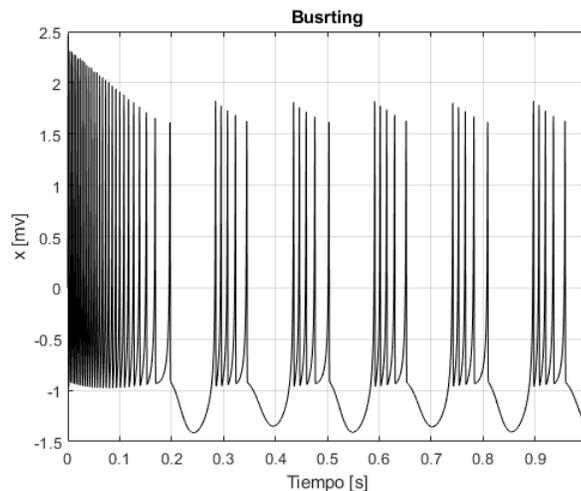


Figura 5. Comportamiento Bursting, $I=3.0$.

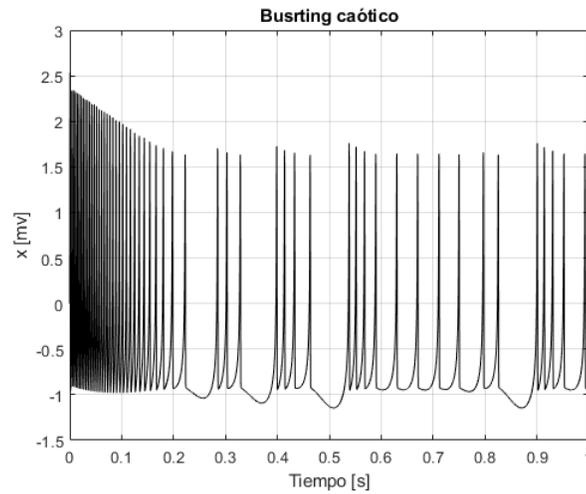


Figura 6. Comportamiento Bursting Caótico, $I=3.3$

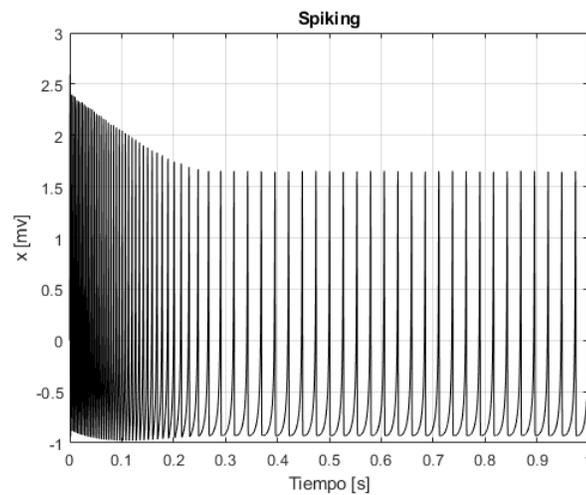


Figura 7. Comportamiento Spiking, $I=3.7$.

2.3. Consideraciones para la selección del robot móvil usado en la integración del guía robótico

Para poder determinar las características y especificaciones técnicas que debe de tener el robot móvil que se utilizará en el diseño del guía robótico, se realizaron una serie de entrevistas con una persona joven (19 años) invidente. En dichas entrevistas se le preguntó al sujeto de estudio sobre los métodos o ayudas que utiliza para desplazarse por lugares conocidos y desconocidos, así como las dificultades que llega a experimentar cuando se desplaza sin asistencia alguna.

Por otra parte, con el fin de poder determinar la velocidad de desplazamiento que debe tener el robot móvil, se le pidió al sujeto de estudio que caminara en un espacio abierto y conocido para él, utilizando un bastón blanco (véase la Figura 8).

Se consideró una superficie de desplazamiento completamente plana, sin desniveles y libre de obstáculos estáticos o dinámicos. Se le pidió al sujeto de estudio que dicha actividad la realizara a una velocidad a la que se sintiera cómodo. Un video de esta actividad se encuentra en el siguiente enlace: <https://www.youtube.com/watch?v=ld2ljT53roQ>

Después de varias pruebas, se determinó que para que el guía robótico sea funcional, el robot móvil a utilizar debe de tener una velocidad lineal mayor a 0.8 m/s.



Figura 8. Sujeto de estudio desplazándose en un espacio abierto.

2.4. Características y especificaciones del robot móvil utilizado

Utilizando la información recopilada de las entrevistas que sugieren cierta capacidad de instrumentación para retroalimentar al usuario, capacidades mecánicas, y pruebas realizadas con el sujeto de estudio mencionadas en la sección anterior, se decidió adquirir el robot ROSbot 2.0 PRO de la marca Husarion, para ser utilizado como base del guía robótico. Se trata de un robot autónomo, de arquitectura abierta y con las siguientes características y especificaciones:

- Velocidad lineal máxima: 1m/s
- Capacidad de carga máxima : 10kg
- Peso aproximado: 2.8kg
- Escáner láser de 360 grados (Lidar)

- Cámara RGBD Orbbec Astra
- Programable con ROS (Robot Operating System)



Figura 9. Robot ROSbot 2.0 Pro de la marca Husarion.

La unidad de procesamiento es una tarjeta Up-board que funciona con una versión de Ubuntu, y se combina con una segunda unidad de procesamiento de bajo nivel encargada del control de los motores, baterías y algunos otros periféricos. La Figura 10 muestra las conexiones entre los actuadores, sensores, puertos y unidades de procesamiento del robot.

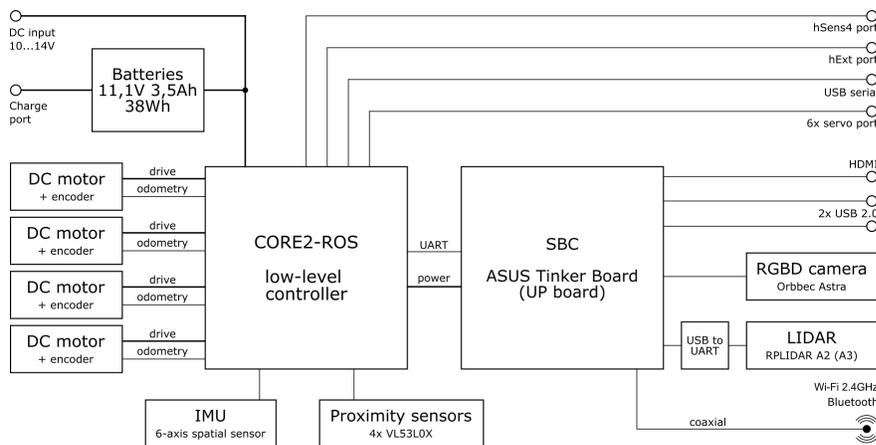


Figura 10. Diagrama a bloques de los diferentes módulos que integran el robot móvil ROSbot 2.0 PRO.

La programación del robot se realiza a través del software Robot Operating System (ROS). Incluir ROS en el sistema operativo se considera una ventaja para esta aplicación, ya que el entorno ROS brinda una capa de comunicación, que facilita el flujo de información entre los elementos activos en la red (red ROS). Reduciendo los tiempos de programación destinados a funciones que se encarguen de establecer

protocolos de comunicación con actuadores, sensores y unidades de procesamiento.

2.5. Entorno ROS

ROS es un paquete de programas destinados al desarrollo de sistemas robóticos autónomos. El cual cuenta con algoritmos comúnmente utilizados en robótica. Los elementos principales dentro de una red ROS son los nodos y los tópicos. ROS se distribuye en paquetes, un paquete es usualmente desarrollado para llevar a cabo un tipo de tarea y este puede contener uno o múltiples nodos. Los nodos son la unidad base de ROS, estos se encargan del manejo de dispositivos o el cómputo de algoritmos, se considera un nodo para una tarea específica.

Los nodos pueden comunicarse entre ellos utilizando tópicos. Un tópico es un canal utilizado para intercambiar información entre nodos, se utilizan para enviar mensajes frecuentes de un tipo de mensaje específico. Como puede ser la lectura de un sensor o la velocidad para un motor. Cada tópico se registra con un nombre único y con un tipo de mensaje definido. Los nodos pueden conectarse a un tópico como publicadores (enviar mensajes) o suscriptores (recibir mensajes), pero no ambos al mismo tiempo. Por otra parte, no existen restricciones al número de nodos diferentes que se publican y se suscriben.

2.6. Red ROS

Para establecer una red ROS, basta con ejecutar diferentes paquetes en el robot. Por otro lado y para fines de procesamiento computacional, es de interés crear una red ROS que contenga elementos externos al robot. De manera específica, crear una red ROS que permita la comunicación con el ordenador que ejecuta matlab con el controlador deseado para que este sea quien gobierne los actuadores del robot e interprete las lecturas realizadas por sus sensores.

Para establecer una red ROS hay que formar una red que conecte al robot con el ordenador, puede ser a través de una red LAN o internet. En este caso se utiliza una red WLAN que puede ser establecida por el modem y conexión a internet. Una vez conectando ambos dispositivos a la red WLAN, se obtiene las direcciones IP de cada dispositivo y se realizan los ajustes del entorno ROS de cada dispositivo. Suponiendo que las direcciones IP sean:

laptop (Host MATLAB) IP Address: 192.168.1.100

ROSbot (ROS Master) IP Address: 192.168.1.200

Para configurar el robot, se debe abrir la terminal de comandos y editar el archivo de ambiente `.bashrc`, y encontrar las líneas:

```
export ROS_MASTER_URI=http://master:11311
```

```
export ROS_IPV6=on
```

y cambiarlas por:

```
export ROS_MASTER_URI=http://192.168.1.200:11311
```

```
export ROS_IP=192.168.1.200
```

```
#export ROS_MASTER_URI=http://master:11311
```

```
#export ROS_IPV6=on
```

Para configurar el ordenador, se debe iniciar MATLAB y configurar la información de ambiente ROS con la herramienta `-Configure ROS Network Adresses-`

Posteriormente, en la sección de ROS Master (ROS), se agrega la dirección IP del robot y en la sección de Node Host (ROS), se agrega la dirección IP del ordenador.

Una vez realizada la configuración del entorno ROS en ambos dispositivos. Se realiza un ping en ambas direcciones. Se abre una ventana de comandos y se introduce la línea:

```
ping < IP_Address >
```

En `< IP_Address >` se introduce la dirección IP del dispositivo al que se le desea enviar, la señal. Hay que hacer en una ventana de comando del ordenador con la dirección IP del robot y también en sentido opuesto, del robot al ordenador.

Una vez que se obtiene un ping exitoso en ambos lados, se puede compartir información entre nodos a través de los tópicos disponibles.

Capítulo 3. Metodología

En este capítulo se presenta la integración de un primer prototipo de guía robótico para personas invidentes. El núcleo del prototipo es el robot móvil tipo unicycle presentado en el Capítulo 2. El otro componente del prototipo es un bastón blanco convencional, comúnmente usado por personas invidentes o débiles visuales.

Para que el robot móvil sea funcional como guía robótico, es esencial que sea capaz de navegar de manera autónoma. Por tanto, en este capítulo se presenta el diseño de una estrategia de control y de evasión de obstáculos para el robot móvil. Como primer punto se seleccionó un controlador disponible en la literatura para implementarlo en el robot móvil de tal manera que este siga una trayectoria deseada. Posteriormente, se diseñó una estrategia de evasión de obstáculos, la cual se basa en el uso del model neuronal de Hindmarsh-Rose introducido en el capítulo anterior. A continuación se obtiene un controlador híbrido, el cual permite conmutar entre el seguimiento de trayectoria y el algoritmo de evasión de obstáculos. El primero se usa cuando el camino del robot está libre de obstáculos mientras que el segundo se utiliza cuando hay objetos estáticos o dinámicos dentro de un radio de operación del robot móvil.

Por otro lado, en este capítulo también se discute el diseño de una interfaz mecánica para conectar el robot móvil con el bastón blanco.

3.1. Control para seguimiento de trayectorias

La estrategia de seguimiento de trayectoria utiliza dos robots móviles del tipo (2,0) en el plano (X_b, Y_b) y consiste de lo siguiente: Un robot virtual de referencia describe la trayectoria deseada mediante el apropiado cálculo de su velocidad lineal v_r , y rotacional ω_r , así como su orientación θ_r . El segundo robot, que es el que se desea controlar, realiza el seguimiento de trayectoria deseada mediante la ‘persecución’ del primer robot. Esta estrategia se muestra gráficamente en la Figura 11.

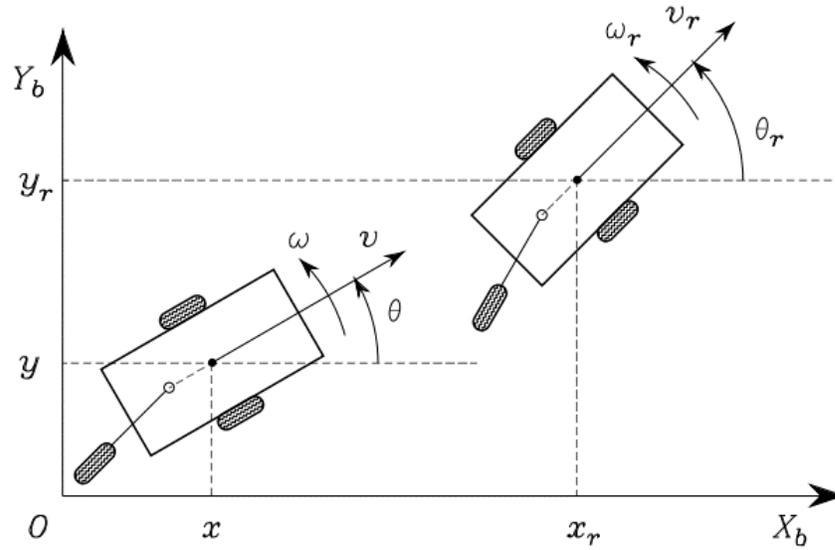


Figura 11. Representación de seguimiento de trayectoria implementando un robot virtual de referencia.

Las ecuaciones paramétricas de la posición deseada se determinan por $x_d = f(t)$ y $y_d = g(t)$, así como sus primeras derivadas \dot{x}_d , \dot{y}_d y segundas derivadas \ddot{x}_d , \ddot{y}_d . Con esta información, a continuación se obtienen las velocidades v_r , ω_r y la orientación θ_r que describen el comportamiento deseado del robot virtual. Dichas variables están dadas por:

$$v_r = \sqrt{\dot{x}_d^2 + \dot{y}_d^2}, \quad (3)$$

$$\omega_r = \frac{\dot{x}_d \ddot{y}_d - \dot{y}_d \ddot{x}_d}{v_d^2}, \quad (4)$$

$$\theta_r = \tan^{-1} \left(\frac{\dot{y}_d}{\dot{x}_d} \right), \quad (5)$$

donde v_r es la velocidad lineal, ω_r la velocidad angular del robot virtual de referencia y θ_r su orientación con respecto al eje X_b , tal y como se muestra en la Figura 11.

Las ecuaciones de movimiento del robot que se desea controlar están dadas en (1), mientras que el modelo del robot virtual está dado por:

$$\begin{bmatrix} \dot{x}_r \\ \dot{y}_r \\ \dot{\theta}_r \end{bmatrix} = \begin{bmatrix} v_r \cos \theta_r \\ v_r \sin \theta_r \\ \omega_r \end{bmatrix}. \quad (6)$$

El objetivo de control es lograr que el comportamiento del robot físico con modelo (1) realice un

seguimiento asintótico del robot virtual de referencia, es decir

$$\lim_{t \rightarrow \infty} |z_r(t) - z(t)| = 0, \quad (7)$$

donde $z(t) = [x(t), y(t), \theta(t)]$ y $z_r(t) = [x_r(t), y_r(t), \theta_r(t)]$.

Para alcanzar este objetivo de control, en (de Wit et al., 2012) se propone el siguiente controlador:

$$\begin{aligned} V &= k_1 e_1 + v_r \cos e_3, \\ W &= \omega_r + k_2 \tanh(v_r) e_2 + k_3 e_3, \end{aligned} \quad (8)$$

donde los errores de seguimiento están dados por

$$\begin{pmatrix} e_1 \\ e_2 \\ e_3 \end{pmatrix} = \begin{pmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_r - x \\ y_r - y \\ \theta_r - \theta \end{pmatrix}, \quad (9)$$

y las constantes k_1 , k_2 y k_3 son constantes positivas que tienen que sintonizarse de acuerdo a lo indicado en (de Wit et al., 2012) para garantizar que el sistema de lazo cerrado (1)-(8) sea asintóticamente estable.

Finalmente, nótese que el controlador (8) puede reescribirse de la siguiente manera para poder operar a nivel de velocidad de rueda

$$\omega_{ST} = \begin{cases} \omega_{der} \\ \omega_{izq} \end{cases} = \begin{bmatrix} \frac{1}{r_r} \left(V + \frac{b_r W}{2} \right) \\ \frac{1}{r_r} \left(V - \frac{b_r W}{2} \right) \end{bmatrix}, \quad (10)$$

donde ω_{der} es la velocidad de rueda derecha, ω_{izq} la velocidad de rueda izquierda, r_r el radio de rueda, b_r es la distancia entre ruedas, y V y W están dadas en (8).

3.2. Diseño de la trayectoria deseada

Existen diferentes estrategias para planificar la trayectoria de robots móviles, como por ejemplo usando las llamadas splines (Van Parys & Pipeleers, 2017; Zhao et al., 2022), redes neuronales combinadas con aprendizaje reforzado jerárquico (Yu et al., 2020), método de potenciales de campo (Khatib, 1985) y algoritmos genéticos (Lamini et al., 2018), entre otros (Campbell et al., 2020). Sin embargo, en este trabajo se propone que la trayectoria deseada se construya con ecuaciones paramétricas de la forma

$x_d = at$ y $y_d = bt$ donde a y b son constantes, y t denota el tiempo.

Utilizando las expresiones (3), (4) y (5), se obtiene

$$\begin{aligned} v_r &= \sqrt{a^2 + b^2}, \\ \omega_r &= 0, \\ \theta_r &= \tan^{-1} \left(\frac{b}{a} \right). \end{aligned} \quad (11)$$

Considerando valores para ' a ' y ' b ' que contengan la meta (punto deseado en el plano) dentro de las ecuaciones paramétricas de la trayectoria deseada, y considerando como origen la posición actual del robot, se tiene que

$$x_{robot} = at_0 \quad (12)$$

$$y_{robot} = bt_0 \quad (13)$$

donde t_0 es el tiempo inicial y

$$x_{meta} = at_f, \quad (14)$$

$$y_{meta} = bt_f, \quad (15)$$

con t_f siendo el tiempo al cual se quiere alcanzar el punto meta. Por lo tanto, si se considera una trayectoria dentro del intervalo de tiempo $\Delta t = t_f - t_0$, es posible obtener las siguientes expresiones

$$\Delta t = \frac{x_{meta} - x_{robot}}{a}, \quad (16)$$

y

$$\Delta t = \frac{y_{meta} - y_{robot}}{b}. \quad (17)$$

Al igualar las expresiones (16) y (17) y realizar un reacomodo de términos, se obtiene la siguiente expresión:

$$\frac{a}{b} = \frac{x_{meta} - x_{robot}}{y_{meta} - y_{robot}} = m, \quad (18)$$

la cual es la relación que se debe mantener entre a y b para garantizar que el robot llegue a la meta en un tiempo Δt .

Posicionamiento

Como siguiente paso, se define un parámetro de separación llamado d_{Meta} y un criterio de proximidad d_{prox} en el plano XY , los cuales están directamente asociados con la posición del robot respecto a un punto del plano de navegación. El parámetro de separación d_{Meta} , es la distancia de separación entre el robot móvil y la meta o par coordenado al que se desea llegar. Por lo que haciendo uso de la ecuación de la distancia entre dos puntos, y considerando la posición del robot como $[x_1, y_1]$ y la meta como $[x_{Meta}, y_{Meta}]$ se obtiene:

$$d_{Meta} = \sqrt{(x_{Meta} - x)^2 + (y_{Meta} - y)^2}. \quad (19)$$

Por otra parte, el criterio de proximidad d_{prox} es arbitrario, expresado en metros, y puede ajustarse en función de la aplicación, resultando en la condición de llegada

$$d_{Meta} \leq d_{prox}, \quad (20)$$

es decir, para efectos prácticos se considerará que el robot ha llegado a la meta deseada si se encuentra dentro de un círculo de radio d_{prox} con centro en el punto meta. En la Figura 12 se muestra el esquema del robot móvil en el plano de navegación, el punto meta y la separación d_{meta} entre ambos.

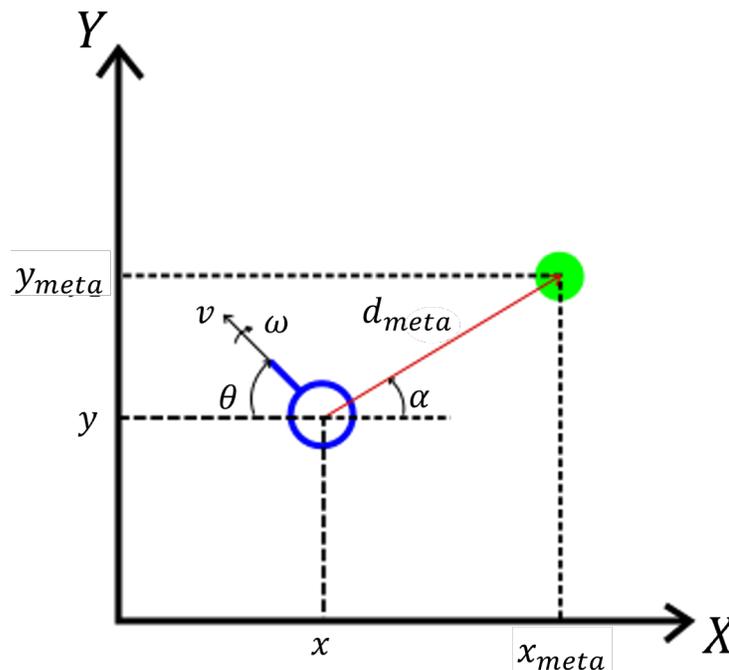


Figura 12. El robot móvil se representa por el círculo azul, el punto meta por el círculo verde y la separación entre ambos por d_{meta} .

3.3. Concatenación de trayectorias

Para generar trayectorias de mayor complejidad, se propone construir una trayectoria deseada T_d compuesta por subtrayectorias, tal que:

$$T_d = \begin{cases} T_{d1} \\ T_{d2} \\ \cdot \\ \cdot \\ \cdot \\ T_{dn}, \end{cases} \quad (21)$$

donde el subíndice n denota el número de subtrayectorias y el número de metas que se desea alcanzar para describir la trayectoria deseada. Una vez que una subtrayectoria inicia, esta tiene que finalizar para que se calcule la siguiente subtrayectoria. El criterio que se utiliza para iniciar una nueva subtrayectoria es el siguiente

$$T_d = \begin{cases} T_{d1} & \text{subtrayectoria inicial} \\ T_{d2} & \text{si } d_{Meta_1} \leq d_{prox} \\ T_{d3} & \text{si } d_{Meta_2} \leq d_{prox} \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ T_{dn} & \text{si } d_{Meta_{n-1}} \leq d_{prox}. \end{cases} \quad (22)$$

De la expresión anterior es claro ver que calculada la primera subtrayectoria deseada, se espera a que el robot cumpla con el criterio de proximidad para la primera meta. Una vez alcanzado el valor de umbral d_{prox} respecto a la primera meta, entonces la subtrayectoria T_{d1} es descartada y se habilita la condicionante para considerar la segunda subtrayectoria deseada y así sucesivamente. Debido a que se trabaja con un criterio de proximidad para considerar cierto grado de incertidumbre, la trayectoria deseada siguiente es calculada con la posición del robot al momento de cumplir el criterio. La Figura 13 ejemplifica la operación del robot al navegar con esta estrategia con un valor de umbral $d_{prox} = 0.3m$. En este ejemplo, el robot inicia en la coordenada (2,2) y avanza hacia la primera coordenada deseada que es el punto (7,1.5). Cuando el robot está a una distancia $d_{Meta} = d_{prox} = 0.3m$ del punto deseado, el robot continúa hacia el punto final que es el punto (2,4.5).

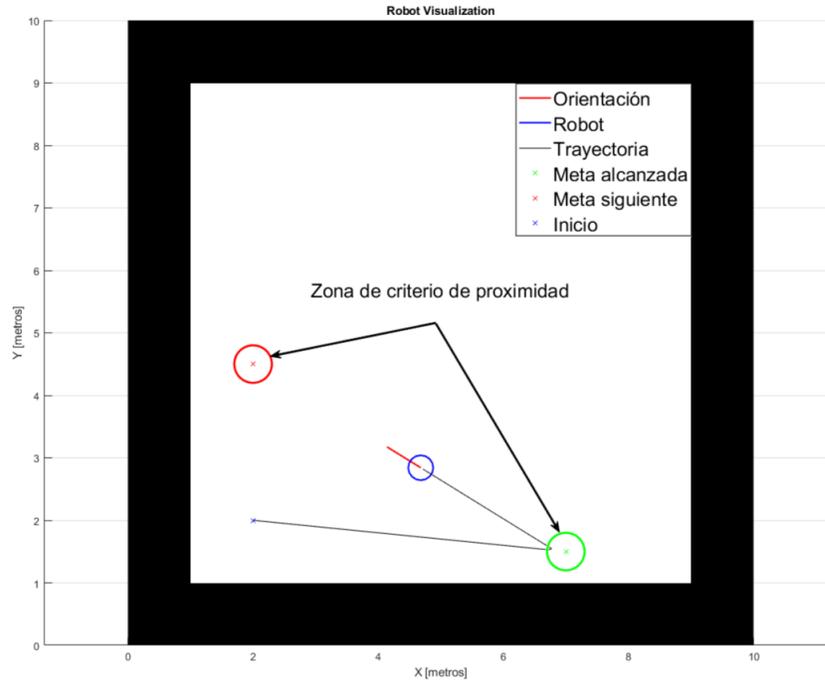


Figura 13. Coordenadas: Inicio(2,2), coordenada deseada 1: (7,1.5), meta final: (2,4.5). $d_{prox} = 0.3m$, $b_r = 0.2m$.

3.4. Estrategia de evasión de obstáculos

En la sección anterior se presentó el controlador de seguimiento de trayectoria. Ahora, en esta sección se presenta la estrategia de evasión de obstáculos para el robot móvil. Como primer punto, se asume que el robot móvil está instrumentado de tal manera que se tienen once sensores de proximidad, colocados en el semicírculo frontal del robot, con una separación de dieciocho grados entre ellos, tal y como se muestra en la Figura 14.

Nótese que, debido a que el robot móvil cuenta con escaneo láser de 360 grados (LiDAR), véase la Sección 2.4, entonces es muy sencillo poder cumplir con la distribución mostrada en la Figura 14. A continuación, se obtienen los valores mínimos de proximidad de cada grupo, es decir,

$$\begin{aligned}
 d_{der} &= \min(d_1, d_2, d_3, d_4, d_5), \\
 d_{izq} &= \min(d_7, d_8, d_9, d_{10}, d_{11}), \\
 d_{front} &= d_6,
 \end{aligned} \tag{23}$$

donde d_{der} representa la distancia mínima con respecto a un obstáculo localizado en el flanco derecho del robot, d_{izq} es la distancia mínima a un obstáculo ubicado en el flanco izquierdo y d_{front} es la distancia que existe hacia un obstáculo que se encuentre en la parte frontal del robot.

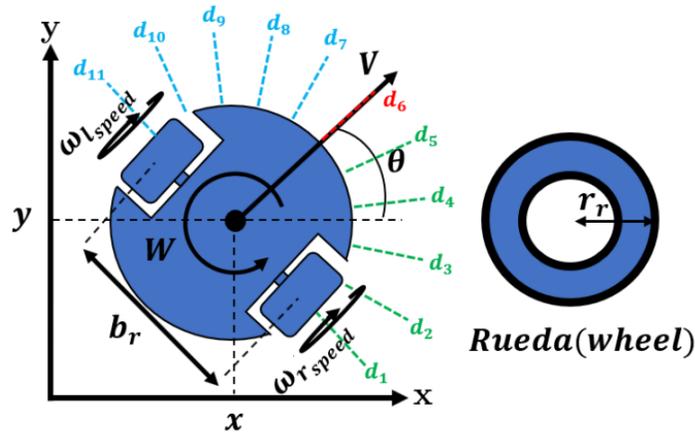


Figura 14. Distribución de los sensores de proximidad en el robot móvil. En total se consideran 11 sensores: 5 para detección de obstáculos del lado izquierdo del robot (indicados en color azul), 5 para detección de obstáculos por el flanco derecho (mostrados en verde) y uno para detección frontal (en color rojo) (Sanchez et al. (2021)).

Estas distancias son un punto medular de la estrategia de evasión, la cual está inspirada en los trabajos previos de (Vromen et al., 2016; Velasco Equihua & Ramirez, 2018; Sanchez et al., 2021) y está dividida en dos partes: una estrategia para evasión de colisiones laterales y otra estrategia para evasión de colisiones frontales. Para la primera se utilizan dos neuronas Hindmarsh-Rose comunicadas mediante un acoplamiento difuso y los valores de d_{der} , d_{izq} arriba mencionados, mientras que para la segunda estrategia se consideran dos neuronas comunicadas mediante un acoplamiento dinámico junto con el valor d_{front} dado en (23). Esto se discute a continuación.

3.4.1. Evasión lateral

La estrategia de evasión lateral hace uso de dos modelos neuronales HR, cada uno descrito por la ecuación (2), conectados mediante acoplamiento difuso, es decir

$$\begin{aligned}
 \dot{x}_1 &= y_1 + bx_1^2 - ax_1^3 - z_1 + I_1 + u_1, \\
 \mathbf{N}_1 : \quad \dot{y}_1 &= c - dx_1^2 - y_1, \\
 \dot{z}_1 &= r[s(x_1 - x_{10}) - z_1],
 \end{aligned} \tag{24}$$

$$\begin{aligned}
 \dot{x}_2 &= y_2 + bx_2^2 - ax_2^3 - z_2 + I_2 + u_2, \\
 \mathbf{N}_2 : \quad \dot{y}_2 &= c - dx_2^2 - y_2, \\
 \dot{z}_2 &= r[s(x_2 - x_{20}) - z_2],
 \end{aligned} \tag{25}$$

donde las entradas u_1 y u_2 están dadas por

$$\begin{aligned} u_1 &= \epsilon(x_2 - x_1), \\ u_2 &= \epsilon(x_1 - x_2), \end{aligned} \quad (26)$$

y las corrientes sinápticas I_1 e I_2 están en función de una distancia de proximidad entre el robot y un obstáculo. Específicamente, I_1 está en función del parámetro de proximidad d_{der} correspondiente a un valor de umbral derecho del robot, e I_2 está en función del parámetro de proximidad d_{izq} que se determina con respecto a obstáculos en el lado izquierdo del robot.

Estas corrientes sinápticas se determinan de la siguiente manera:

$$I_1 = \begin{cases} a_1 I_{min} & d_{der} \leq x_{min}, \\ b_1 I_{min} \left(\frac{I_{max}}{I_{min}} \right)^{\frac{d_{der} - x_{min}}{x_{max} - x_{min}}} & x_{min} < d_{der} < x_{max}, \\ c_1 I_{max} & x_{max} \leq d_{der}, \end{cases} \quad (27)$$

$$I_2 = \begin{cases} a_1 I_{min} & d_{izq} \leq x_{min}, \\ b_1 I_{min} \frac{I_{max}}{I_{min}} \frac{d_{izq} - x_{min}}{x_{max} - x_{min}} & x_{min} < d_{izq} < x_{max}, \\ c_1 I_{max} & x_{max} \leq d_{izq}, \end{cases} \quad (28)$$

donde I_{max} e I_{min} son las cotas máximas y mínimas de la corriente sináptica I . Por otro lado, x_{max} y x_{min} son las distancias máxima y mínima que pueden detectar los sensores y los parámetros a_1 , b_1 y c_1 son parámetros de ajuste.

Como siguiente paso, se realiza lo siguiente:

1. Las corrientes I_1 e I_2 dadas en (27)-(28) se aplican al par de neuronas acopladas (24)-(25) con control (26).
2. Del sistema (24)-(25) se miden las variables x_1 y x_2 , las cuales corresponden a los potenciales de membrana de las neuronas N_1 y N_2 , respectivamente.
3. Se define una ventana de tiempo y se cuenta el número de espigas que tienen las variables x_1 y x_2 dentro de esa ventana/intervalo de tiempo. A lo largo de este trabajo se considera una ventana de tiempo de 100 ms.
4. El número de espigas para x_1 se denota por n_1 y el número de espigas de la variable x_2 se denomina n_2 .

Finalmente, la velocidad de rueda para la estrategia de evasión lateral queda definida por

$$\omega_{ST} = \begin{cases} \omega_{der*} = n_1 p, \\ \omega_{izq*} = n_2 p. \end{cases} \quad (29)$$

donde ω_{der*} es la velocidad de rueda derecha, ω_{izq*} es la velocidad de rueda izquierda, n_1 y n_2 son el número de espigas que presentan las variables x_1 y x_2 tal y como se explicó arriba y p es un factor de escalamiento.

Observación 3.1 *El funcionamiento del controlador (29) puede explicarse de una manera casi intuitiva: si no hay obstáculos en el camino del robot entonces las corrientes sinápticas (27)-(28) son iguales y como consecuencia las neuronas (24)-(25) se sincronizan. Como consecuencia, el número de espigas de las variables x_1 y x_2 es igual, es decir $n_1 = n_2$ y por tanto de la ecuación (29) se tiene que $\omega_{der*} = \omega_{izq*}$, es decir, la velocidad de rueda derecha es igual a la velocidad de rueda izquierda y el robot se desplazará en línea recta. Por otra parte, si se detecta un obstáculo en uno de los lados del robot, entonces las corrientes sinápticas (27)-(28) serán diferentes, lo que resulta en que las neuronas pierdan la sincronización de tal forma que el número de espigas sera diferente, es decir $n_1 \neq n_2$ y por tanto las velocidades de rueda ω_{der*} y ω_{izq*} serán diferentes, lo que al final resultará en que el robot gire a la izquierda o a la derecha, dependiendo de si $n_1 > n_2$ o $n_2 > n_1$.*

3.4.2. Evasión frontal

Para la estrategia de evasión frontal se usan dos neuronas que interactúan a través de un acoplamiento dinámico. Las neuronas están descritas por

$$\begin{aligned} \dot{x}_3 &= y_3 + bx_3^2 - ax_3^3 - z_3 + I + h_2, \\ \mathbf{N}_3 : \dot{y}_3 &= c - dx_3^2 - y_3, \\ \dot{z}_3 &= r[s(x_3 - x_{30}) - z_3], \end{aligned} \quad (30)$$

$$\begin{aligned} \dot{x}_4 &= y_4 + bx_4^2 - ax_4^3 - z_4 + I - Gh_2, \\ \mathbf{N}_4 : \dot{y}_4 &= c - dx_4^2 - y_4, \\ \dot{z}_4 &= r[s(x_4 - x_{40}) - z_4], \end{aligned} \quad (31)$$

donde las variables de estado x_i, y_i y z_i , $i = 3, 4$ son las variables de estado que describen el comportamiento de la neurona, y los parámetros a, b, c, d, r, s y x_0 se describieron con anterioridad en la ecuación (2) y ambas neuronas reciben la misma corriente sináptica I . Por otra parte, h_2 se genera por el siguiente sistema lineal de segundo orden

$$\begin{aligned} \dot{h}_1 &= \alpha h_1 + h_2, \\ \dot{h}_2 &= \gamma_1 h_1 - \gamma_2 h_2 - k(x_3 - x_4), \end{aligned} \quad (32)$$

donde los parámetros $\alpha, \gamma_1, \gamma_2$ y k son positivos.

Por otro lado, la función G en (31) depende de la lectura de proximidad frontal d_{front} y del valor de umbral x_{fmin} y está dada por

$$G = -11e^{-3(d_{front} - x_{fmin})}. \quad (33)$$

La función G expresada en la ecuación (33), tiene una influencia directa en el comportamiento de la variable de estado x_4 . En particular cuando $d_{front} < x_{fmin}$, el estado x_4 exhibe comportamiento de espiguelo mientras que, cuando dicha condición no se satisface entonces el estado x_4 se encuentra en estado de relajación. Para poder identificar en que estado se encuentra el sistema de neuronas acopladas, se introduce la variable binaria x_{4A} , de tal manera que $x_{4A} = 1$ cuando el estado x_4 se encuentra en modo de espiguelo, y $x_{4A} = -1$ cuando el estado x_4 se encuentra en estado de relajación.

Finalmente, esta variable binaria se introduce en el control, por lo que las velocidades de rueda utilizadas para realizar la evasión frontal están dadas por

$$\omega_{EO} = \begin{bmatrix} \omega_{der} \\ \omega_{izq} \end{bmatrix} = \begin{bmatrix} n_1 p x_{4A} \\ n_2 p \end{bmatrix}, \quad (34)$$

donde n_1, n_2 y p se definieron previamente en (29).

Observación 3.2 *El funcionamiento de este controlador puede explicarse de la siguiente manera. Cuando la distancia d_{front} es menor que el valor de umbral x_{fmin} , es decir, que hay un objeto frontal ‘demasiado’ cerca, el estado x_4 entra en relajación y entonces $x_{4A} = -1$. Además, dado que las neuronas están sincronizadas, entonces se tiene que $n_1 = n_2$. Reemplazando esto en (34) se puede ver que la velocidad de rueda derecha ω_{der} tendrá signo contrario a la velocidad de rueda izquierda ω_{izq} , lo que provocará que el robot gire sobre su propio eje y así evitar la colisión frontal.*

3.5. Controlador híbrido

Combinando el controlador de seguimiento de trayectoria dado en (29) con el controlador de evasión de obstáculos dado en (34), se puede crear un control híbrido, que conmuta entre el modo de seguimiento y el modo de evasión dependiendo de si existe algún obstáculo, frontal o lateral, en el área de trabajo del robot móvil. Esto se ilustra en el diagrama de la Figura 15.

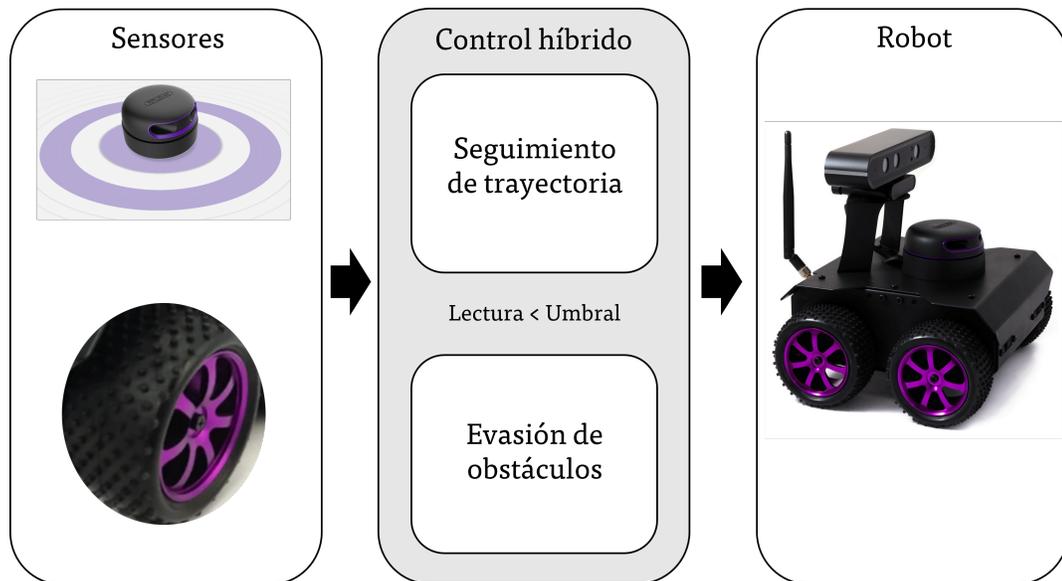


Figura 15. Elementos del controlador híbrido.

Para determinar qué modo utilizar (seguimiento o evasión), es conveniente definir la mínima distancia hacia un obstáculo en cualquier dirección (izquierda, derecha, frente)

$$d = \min(d_{der}, d_{izq}, d_{front}), \quad (35)$$

donde d_{der} , d_{izq} , d_{front} se definieron en (23).

Finalmente, utilizando esta distancia mínima d , se define el siguiente controlador híbrido

$$\text{Velocidad de rueda} = \begin{cases} \omega_{ST} & \text{si } d > d_c, \\ \omega_{EO} & \text{si } d \leq d_c, \end{cases} \quad (36)$$

donde d_c es una distancia crítica predeterminada, ω_{ST} está dado en (29) y ω_{EO} se describe en (34).

Observación 3.3 *La operación del controlador híbrido es relativamente sencilla: si hay un obstáculo localizado a una distancia d que sea menor o igual a una distancia crítica d_c , entonces el robot navegará utilizando el controlador de evasión (34). De otra manera, el comportamiento del robot estará gobernado por el controlador de seguimiento de trayectoria (29).*

3.6. Diseño de la interfaz para conectar el robot móvil con un bastón blanco

El proceso de diseño e implementación de la interfaz se llevó a cabo en tres etapas. Primero se realizó un bosquejo con base a las consideraciones necesarias para la adaptabilidad del sistema. A partir del bosquejo se procedió a la presentación del diseño en formato CAD. Por último, se maquinaron las piezas necesarias en el taller de mecánica fina del CICESE.

Se decidió utilizar el bastón blanco como principal enlace entre el robot y el usuario dada la familiaridad y practicidad que ofrece a las personas invidentes. El bastón blanco se ajusta a las características físicas del usuario como altura y distancia de monitoreo a su alrededor, y en este caso, dicha distancia sería la separación entre el sujeto y el robot. También, el bastón blanco utilizado permite que el usuario mantenga sus hábitos de postura correcta, además de ser un símbolo de la discapacidad visual proveyendo información para el resto de los transeúntes.

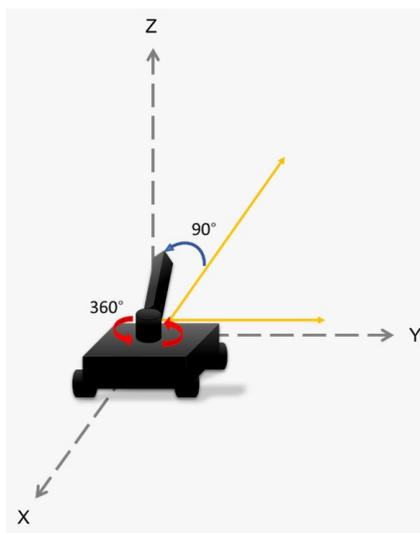


Figura 16. Grados de libertad considerados para la interfaz usuario-ROSBot.

El principal objetivo del bosquejo fue plantear las bases de un vínculo para el bastón blanco, que no obstaculizara las funciones del robot. En la Figura 16 se observan los grados de libertad seleccionados, estos se propusieron teniendo presente que el robot debe tener movilidad de 360° alrededor del eje Z . Además, fue necesario adaptarse a distintas alturas por lo que también se consideró un segundo grado de libertad sobre el plano vertical Y' .

A partir de las consideraciones mencionadas, se propuso el diseño en CAD mostrado en la Figura 17. Una de las actividades a realizar en esta etapa, fue conocer las herramientas y materiales para llevar a cabo la interfaz, así como los mecanismos existentes y su efectividad.

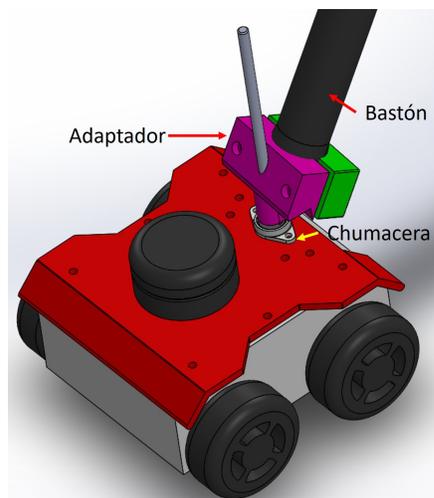


Figura 17. Diseño de mecanismo con grado de libertad alrededor del eje Z .

Posteriormente se decidió que las piezas fueran maquinadas por el equipo técnico de CICESE. Se eligió adecuar el adaptador al soporte de la cámara que viene de fábrica en el ROSbot. De esta manera nos aseguramos de que el desempeño de la cámara frontal, así como el sensor de obstáculos no se vean comprometidos.

En la Figura 18 se muestra el producto final. Las piezas fueron maquinadas en aluminio de galvanizado. El eslabón 1 se realizó con un rodamiento de aluminio que está fijo sobre el soporte y unido al eje 2 por la articulación tipo revoluta. Posteriormente, el eslabón 2 se une por medio de un tornillo pasado con el adaptador para el bastón, para obtener el segundo grado de libertad. Finalmente, se introduce el bastón blanco en el adaptador y se sujeta a presión con un tornillo. Este mecanismo es seguro y no corre el riesgo que las piezas se separen.

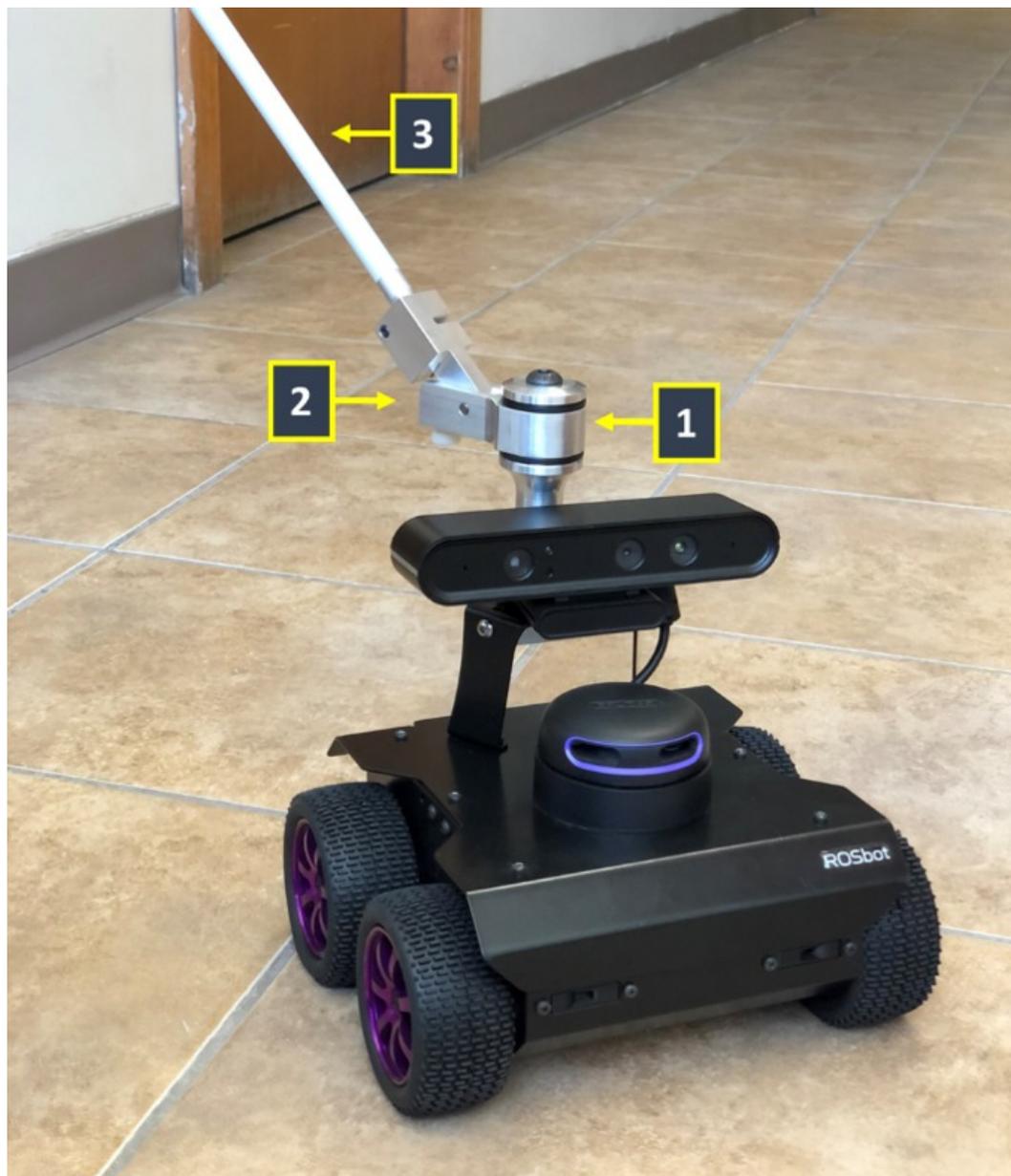


Figura 18. Vista frontal del producto implementado en ROSbot: grado de libertad sobre el eje Z (1), segundo grado de libertad sobre el eje Y (2), y bastón y adaptador (3).

Capítulo 4. Resultados numéricos

En este capítulo se presentan resultados numéricos que ilustran la efectividad de los controladores de seguimiento de trayectoria y de evasión de obstáculos presentados en el capítulo anterior. Dichos resultados se obtienen por medio de simulaciones en computadora utilizando el software especializado matlab y simulink. Por otra parte, también se presentan resultados experimentales que demuestran la efectividad del controlador de seguimiento de trayectoria. Para dichos experimentos se utilizó el robot RosBot 2.0 Pro mencionado en el Capítulo 2. Cabe aclarar que los experimentos se hicieron solamente con el robot navegando de manera autónoma, sin la intervención de un usuario (persona invidente).

Tanto en los resultados numéricos como experimentales, primero se muestran los resultados obtenidos para cada controlador por separado (primero seguimiento de trayectoria y después evasión de obstáculos) y posteriormente se presentan los resultados que corresponden al controlador híbrido que combina ambos controles.

4.1. Resultados numéricos: control de seguimiento de trayectoria sin evasión

En un primer resultado numérico, se ilustra la efectividad del control de seguimiento de trayectoria introducido en la Sección 3.1. Para ello se considera el modelo del robot móvil dado en la ecuación (1) con V y W dadas en (8) con $k_1 = k_2 = k_3 = 1.2$. Las condiciones iniciales del robot móvil y del robot virtual de referencia son $[x(0), y(0), \theta(0)]^T = [2, 2, 0]^T$ y $[x_r, y_r, \theta_r]^T = [0, 0, 0]^T$, respectivamente.

Como trayectoria deseada se considera una lemniscata, por lo que se emplean las velocidades v_r y ω_r del robot virtual dadas en (3)-(4), con $x_r = \cos t \sin t$ y $y_r = \sin t$.

La Figura 19 muestra los resultados obtenidos. Puede verse que después de un cierto tiempo transitorio, la trayectoria del robot (línea azul) converge a la trayectoria deseada indicada por la línea roja.

En una segunda simulación la trayectoria deseada se cambia a un círculo de radio unitario, por lo que ahora, las velocidades lineal y translacional del robot virtual de referencia se construyen con $x_r = \cos t$ y $y_r = \sin t$ y los valores de k_1 , k_2 y k_3 son los mismos que los usados en la simulación anterior.

El resultado de la simulación se muestra en la Figura 20, de donde se puede ver que el robot móvil converge a la trayectoria deseada.

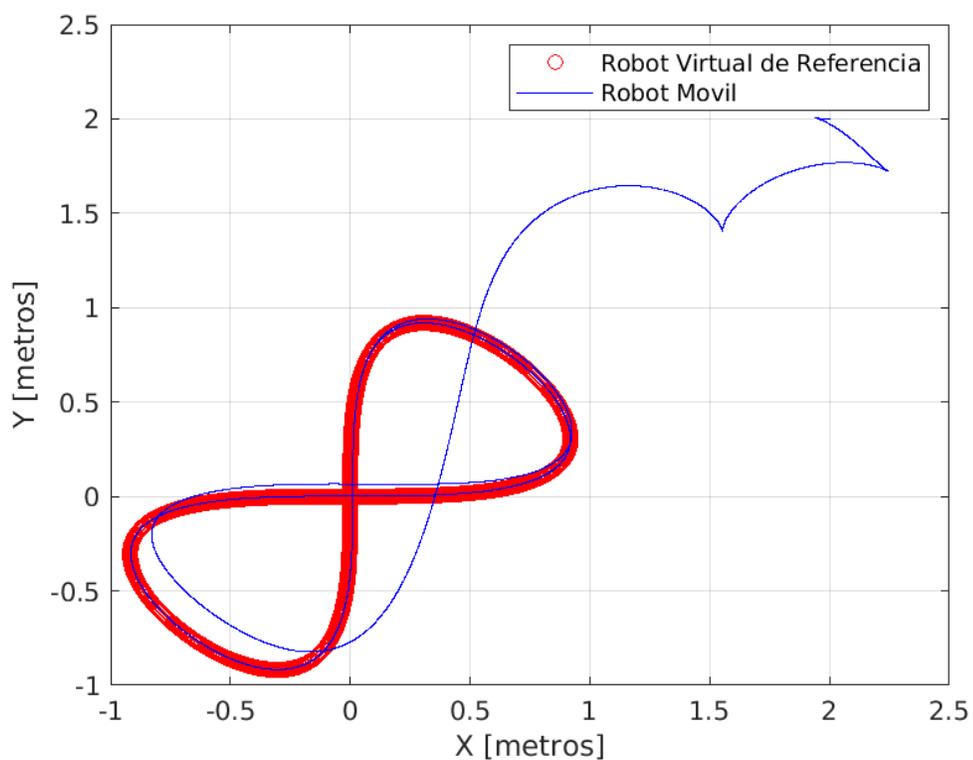


Figura 19. Seguimiento de trayectoria. Rojo: trayectoria deseada. Azul: trayectoria del robot móvil. Puede verse que después de un tiempo el robot móvil converge a la trayectoria deseada.

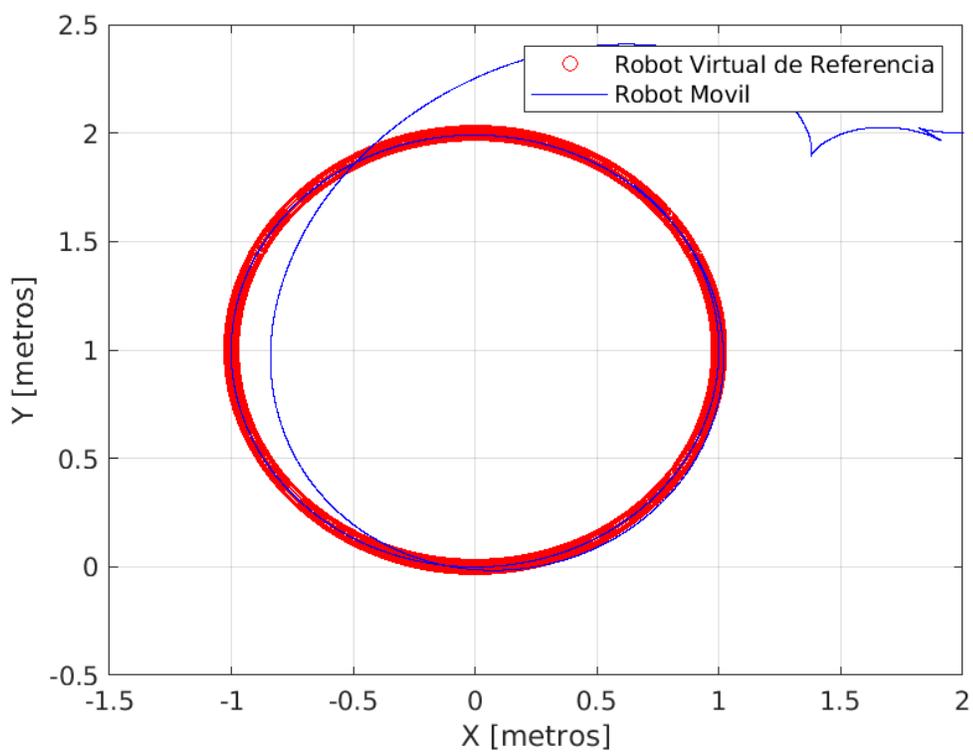


Figura 20. Seguimiento de trayectoria. Rojo: trayectoria deseada. Azul: trayectoria del robot móvil. Puede verse que se alcanza el objetivo de control (seguimiento).

Finalmente, se realiza una serie de simulaciones con diferentes trayectorias deseadas y los resultados obtenidos se presentan en la Figura 21. Las expresiones para x_d y y_d usadas en (3)-(4) para generar la trayectoria deseada se muestran en la parte superior de cada panel dentro de la figura y los valores de las ganancias en el controlador (8) son $k_1 = k_2 = k_3$. La duración de la simulación es de $t = 10\text{seg}$ y las condiciones iniciales del robot móvil son $x(0) = 1.3$, $y(0) = 1.3$, y $\theta(0) = 0$ y son las mismas para las cuatro simulaciones. Además, en la simulación solo se consideran valores positivos de la velocidad lineal del robot virtual de referencia, es decir, $v_r > 0$. Como se aprecia en la Figura 21, en todos los casos el robot converge a la trayectoria deseada.

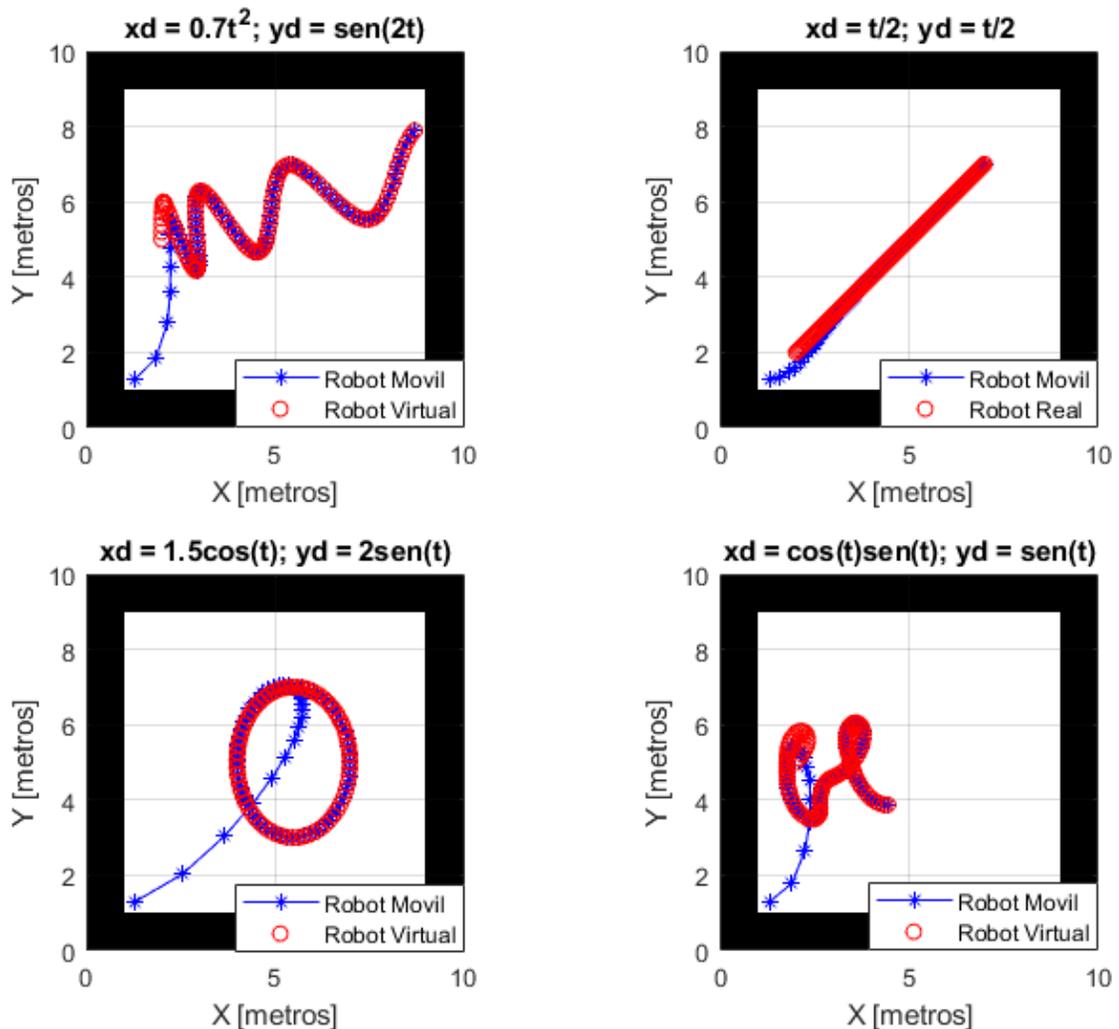


Figura 21. Validación numérica del controlador de seguimiento para diferentes trayectorias deseadas (indicadas en color rojo). En todos los casos, el robot móvil logra alcanzar la trayectoria deseada, lo que muestra la efectividad del controlador.

4.2. Resultados numéricos: generación de trayectoria

En la Sección 3.2 se presentó una estrategia para generar la trayectoria deseada, utilizando rectas concatenadas. Para ilustrar la efectividad de este método, aquí se presenta una simulación numérica. Para ello, se considera una trayectoria compuesta de cuatro puntos meta: punto $Meta1 = (6.5, 1.5)$, $Meta2 = (2, 4.5)$, $Meta3 = (2, 7.5)$, $Meta4 = (8, 7.5)$. Para cada uno de estos puntos meta se considera que los valores de a y b en (11) son $a = 0.3$ y $b = 0.3$. EL valor crítico d_{prox} en (20) es $d_{prox} = 0.3$ [m].

La Figura 22 corresponde a una simulación donde la trayectoria deseada es la generada usando los puntos meta arriba mencionados y el robot móvil está gobernado por el controlador a nivel de rueda dado en (10), con V y W calculadas de la Ec. (8) y con constantes $k_1 = k_2 = 1.2$, y $k_3 = 3$. Además, se asume que las ruedas del robot tienen radio $r_r = 156$ mm, y la distancia de separación entre ruedas es: $b_r = 85$ mm. Además, se considera que las velocidades del robot están limitadas, tal que $V \leq 1$ m/s y $w \leq 1$ m/s, esto para emular las condiciones del robot móvil real.

Como se aprecia en la Figura 22 el robot móvil sigue la trayectoria deseada, la cual ha sido diseñada concatenando cuatro puntos meta.

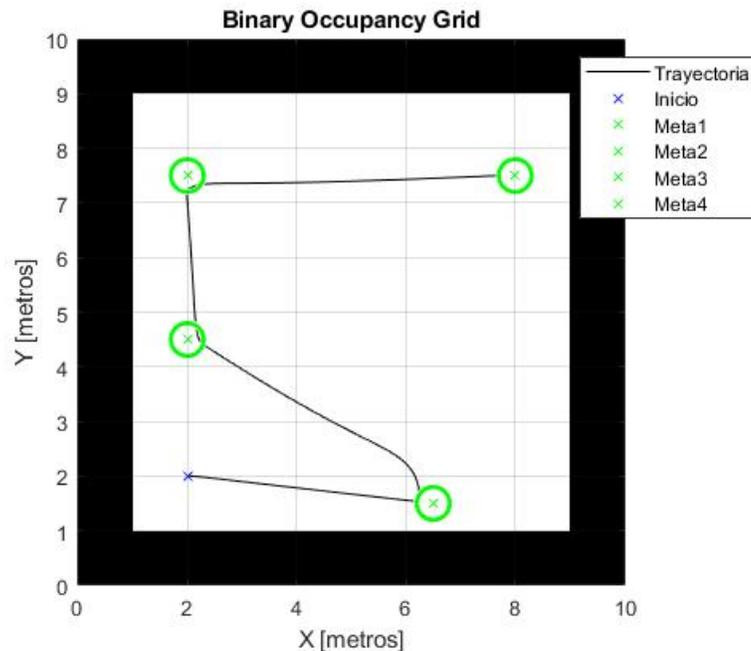


Figura 22. Generación de una trayectoria deseada utilizando cuatro puntos meta.

Observación 4.1 En este punto es importante mencionar que si las ganancias del controlador (8) no se sintonizan de manera adecuada o si la velocidad lineal del robot virtual de referencia es mucho mayor que la velocidad lineal del robot móvil real entonces puede darse el caso de que el robot real no llegue al punto meta deseado. Por ejemplo, si en la simulación presentada en la Figure 22 se cambian los valores de a y b tal que $a = b = 3$, lo que equivale a acelerar el robot virtual, entonces el robot móvil real no converge al punto meta deseado y más bien diverge, tal y como se muestra en la Figura 23.

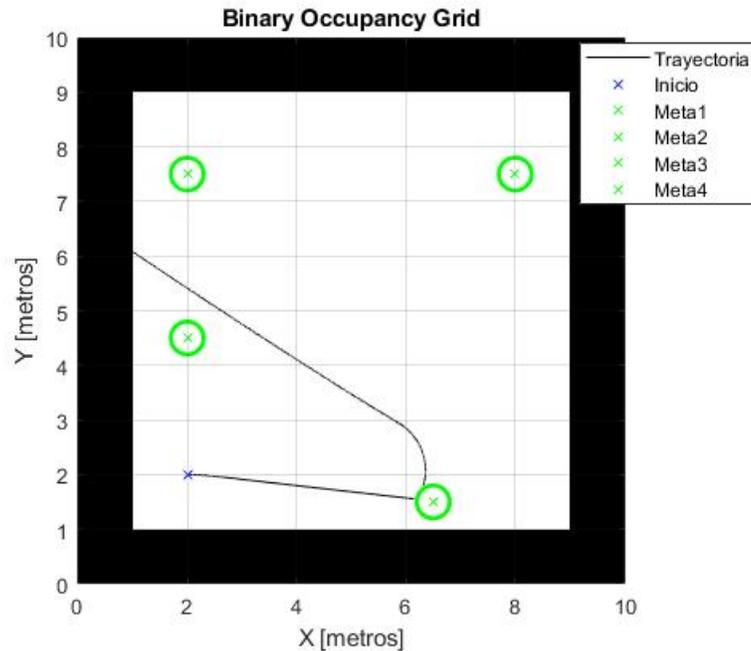


Figura 23. Ejemplo en el que la trayectoria del robot real diverge de la trayectoria deseada, al no lograr entrar en la zona de llegada del punto meta 2. Para evitar este tipo de escenarios, se debe realizar una buena sintonización de las ganancias del controlador, así como garantizar que el robot virtual de referencia es lo 'suficientemente lento' para dar tiempo a que el robot real llegue al punto meta.

4.3. Resultados numéricos: Estrategia de evasión de obstáculos

Los resultados numéricos presentados han mostrado que el controlador de seguimiento de trayectoria funciona correctamente.

Como siguiente paso, en esta sección se evalúa la efectividad de la estrategia de evasión de obstáculos. Primero, se realiza una simulación de 30 segundos, usando el controlador de evasión lateral (29) con $p = 0.007$ y con los siguientes parámetros para los modelos neuronales acoplados (24)-(26): $a = 1$, $b = 3$, $c = 1$, $d = 5$, $r = 0.005$, $s = 4$, $x_0 = -1.6$, y $\epsilon = 0.5$. Para las corrientes sinápticas (27)-(28) se usan los parámetros: $a_1 = 0.95$, $b_1 = 1.35$, $c_1 = 0.85$, $I_{min} = 3.25$, $I_{max} = 10$, $x_{min} = 0.4$ [m]

y $x_{max} = 5$. Durante los primeros 8 segundos de simulación no hay ningún obstáculo en el área de desplazamiento del robot, por lo que las corrientes sinápticas son iguales lo que implica que las neuronas están sincronizadas, tal y como se puede ver en la Figura 26, y se tiene el mismo número de espigas en los potenciales de membrana, es decir $n_1 = n_2$, tal como se aprecia en la Figura 24. Como consecuencia, el robot móvil se desplaza en línea recta, tal y como se observa en la Figura 25.

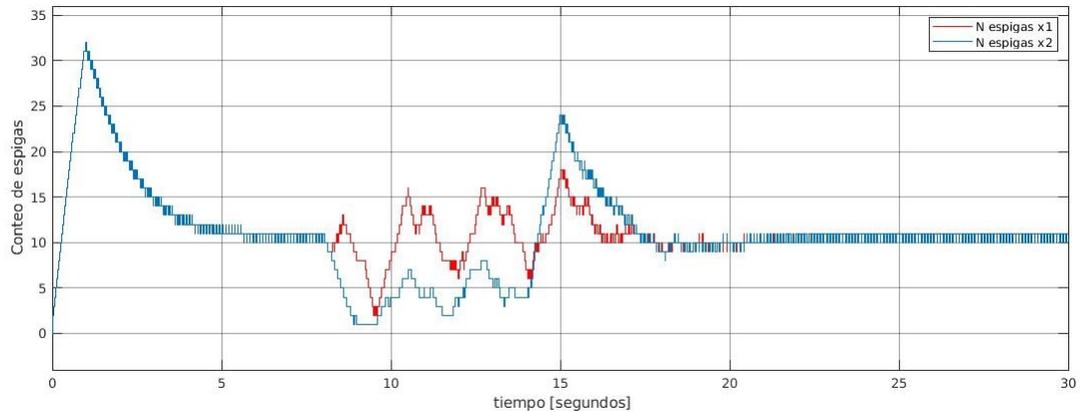


Figura 24. Número de espigas de los potenciales de membrana. Rojo: n_1 . Azul: n_2 . En el intervalo $0 \leq t < 8$ no hay ningún obstáculo en el camino del robot y se puede observar que $n_1 = n_2$. Sin embargo, en el intervalo $8 \leq t < 14$ se agrega un obstáculo en el flanco derecho del robot y como consecuencia $n_1 \neq n_2$. Finalmente, en el intervalo $14 \leq t \leq 30$ el robot ya evadió el obstáculo y el número de espigas vuelve a ser el mismo en los potenciales de membrana, es decir se tiene que $n_1 = n_2$.

Posteriormente, a los 8 segundos de simulación se introduce un obstáculo en el flanco derecho del robot. Al detectarse el obstáculo el número de espigas en cada potencial es diferente como se ve en la Figura 24, lo que resulta en que las corrientes sinápticas cambien y las neuronas pierden la sincronía tal y como se aprecia en la parte central de la Figura 26), y como consecuencia final, el robot gira hacia la izquierda tal y como se muestra en la Figura 25.

Finalmente, a los 14 segundos de simulación el robot ha esquivado por completo el obstáculo y como consecuentemente el número de espigas en los potenciales de membrana vuelve a ser el mismo, las neuronas regresan al estado de sincronía y el robot se mantiene avanzando en línea recta, tal y como se puede apreciar en la sección final de las Figuras (24) a (25). Estos resultados muestran que la estrategia de evasión lateral funciona adecuadamente.

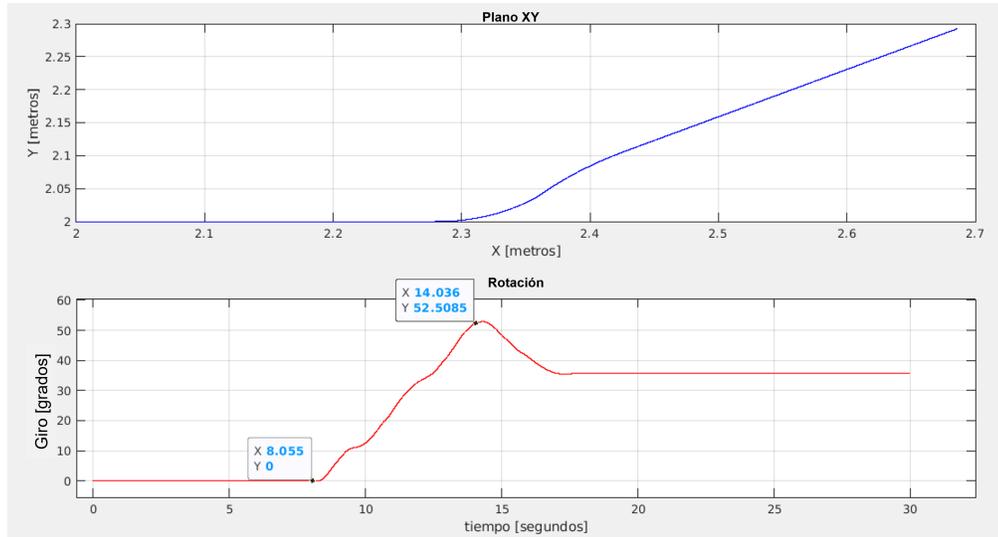


Figura 25. Panel superior: desplazamiento del robot en el plano XY. Panel inferior: Orientación del robot. Nótese que cuando el obstáculo es detectado al tiempo $t = 8$, el robot gira hacia la izquierda. Una vez que evade el objeto (al tiempo $t = 14$) el robot empieza a presentar un comportamiento constante.

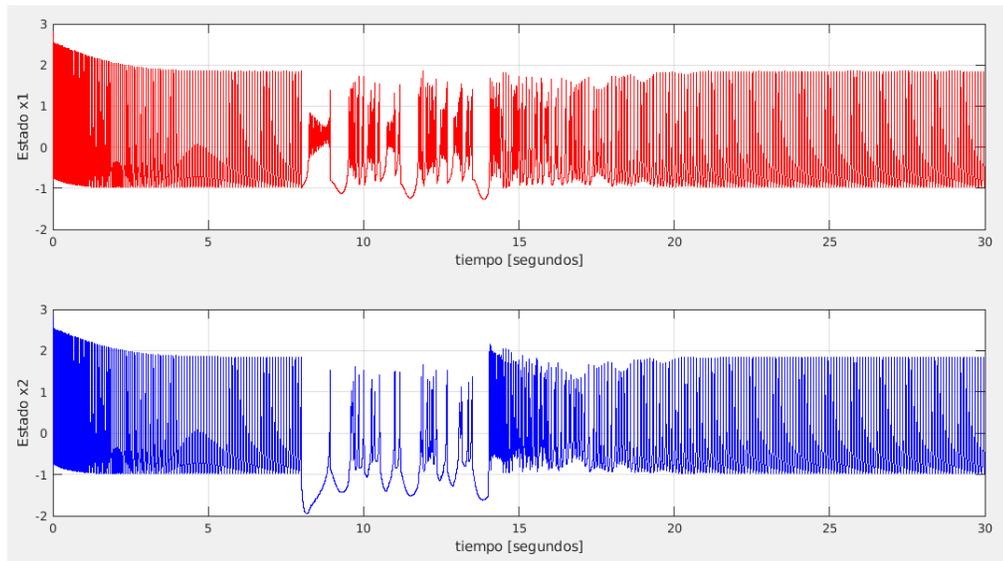


Figura 26. Potenciales de membrana de las neuronas N_1 y N_2 descritas en (24) y (25). En el intervalo $0 \leq t < 8$ no hay obstáculos alrededor del robot y las neuronas están sincronizadas. Cuando se agrega un obstáculo en el flanco derecho del robot en el intervalo $8 \leq t < 14$, las neuronas pierden la sincronía y la recuperan posteriormente en el intervalo $14 \leq t < 30$ cuando ya el obstáculo ha sido esquivado.

En una segunda simulación se investiga la estrategia de evasión frontal. Para ello, se utiliza el controlador dado en la Ec. (34) y el par de neuronas (30)-(31) con acoplamiento dinámico (32). Los parámetros de las neuronas y el parámetro p tienen los mismos valores utilizados en la simulación anterior, los parámetros para el acoplamiento dinámico (32) son $\gamma_1 = \alpha = 1$, $\gamma_2 = k = 10$, el valor de la corriente sináptica en (30)-(31) es $I = 3.281$ y el parámetro de la función G en (33) es $x_{fmin} = 0.5$.

La simulación tiene una duración de 600 segundos, y se introduce un obstáculo en el espacio de trabajo

del robot móvil con una detección de proximidad frontal dentro del intervalo [260-265] segundos. Los resultados obtenidos se muestran en la Figura 27, en la que se puede apreciar claramente como el robot giró sobre su propio eje al encontrar el obstáculo frontal y una vez que el objeto quedó fuera de la zona de detección, el robot siguió avanzando en línea recta.

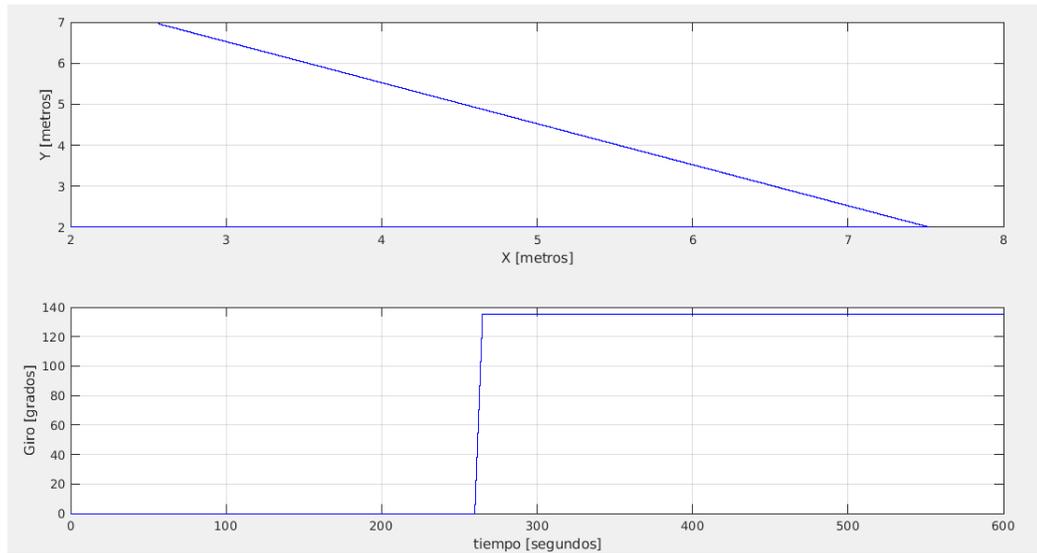


Figura 27. En la gráfica superior se observa el desplazamiento del robot en el plano XY. En la gráfica inferior se observa la variación de la orientación del robot. Se puede apreciar que el robot realiza un giro sobre su propio eje para evadir el obstáculo frontal y entonces continuar avanzando en línea recta. Nótese que después de evadir el obstáculo, la orientación del robot cambia aproximadamente 140 grados con respecto a la orientación inicial.

4.4. Resultados numéricos: controlador híbrido

En la sección anterior se ilustró la efectividad de los controladores de seguimiento y de evasión de obstáculos por separado. Sin embargo, para que el robot móvil sea completamente autónomo se requiere que éstos controladores trabajen de manera complementaria, es decir, se requiere del controlador híbrido presentado en la Sección 3.5.

Por tanto, en esta sección se realiza un estudio numérico para validar el controlador híbrido dado en la Ecuación (36), el cual está expresado en términos de velocidad de rueda. En la simulación se considera el modelo (1) con control (36) y condiciones iniciales $[x(0), y(0), \theta_0]^T = [6, 6, \pi]^T$. Para generar la velocidad lineal v_r del robot virtual se considera (3) con $x_d = 2\cos(t/2)$, y $y = 2\sin(t/2)$ y las velocidades lineal y rotacional del robot móvil se calculan de acuerdo a (8) con $k_1 = 1.2$, $k_2 = 1.2$ y $k_3 = 3$.

Por otro lado, los parámetros del controlador de seguimiento de trayectoria (10) son $r_r = 0.085$ [m] y $b_r = 0.186$ [m] y el parámetro p del controlador de evasión de obstáculos (34) es $p = 2.1$. Finalmente, los parámetros de los modelos neuronales y su respectivo acoplamiento son idénticos a los usados en las simulaciones de la Sección 4.3.

La simulación dura 30 segundos y los resultados obtenidos se presentan en la Figura 28 en la cual se observa que la trayectoria del robot móvil (indicada por la línea azul) converge a la trayectoria deseada (indicada por la línea negra). La figura también muestra que, al encontrar el obstáculo (indicado por el cuadro negro en la figura) el robot se desvía hacia la izquierda y, una vez que ha evadido completamente el obstáculo, regresa a la trayectoria deseada.

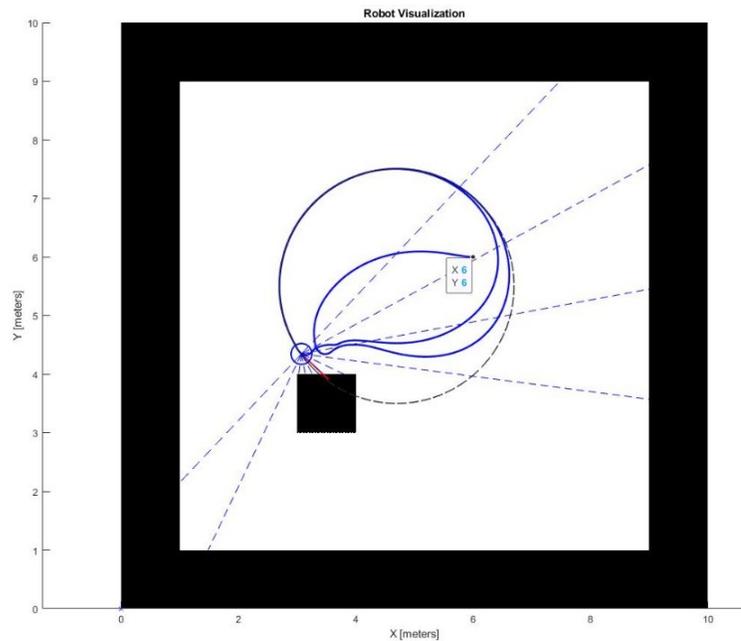


Figura 28. Resultados numéricos con controlador híbrido. La trayectoria del robot móvil (línea azul) converge a la trayectoria circular deseada (línea negra), gracias al controlador de seguimiento de trayectoria (10). Sin embargo, una vez que el obstáculo (cuadro negro en la figura) es detectado, se activa el controlador de evasión de obstáculos (34), y el robot gira hacia la izquierda y logra evadir el obstáculo. Una vez que el obstáculo es evadido, se vuelve a activar el controlador de seguimiento de trayectoria y el robot regresa a la trayectoria deseada.

El resultado obtenido es posible gracias al controlador híbrido, cuya serie de tiempo se muestra en la Figura 29. Nótese que cuando el controlador de seguimiento está activo, las velocidades de rueda son suaves y tienden a estabilizarse alrededor de un valor constante. Sin embargo, cuando el obstáculo es detectado, las velocidades de rueda exhiben oscilaciones de alta frecuencia propias de la naturaleza híbrida del controlador. Los resultados obtenidos muestran claramente que el control híbrido aquí propuesto es

efectivo, ya que permite al robot móvil navegar de manera autónoma mientras evade obstáculos.

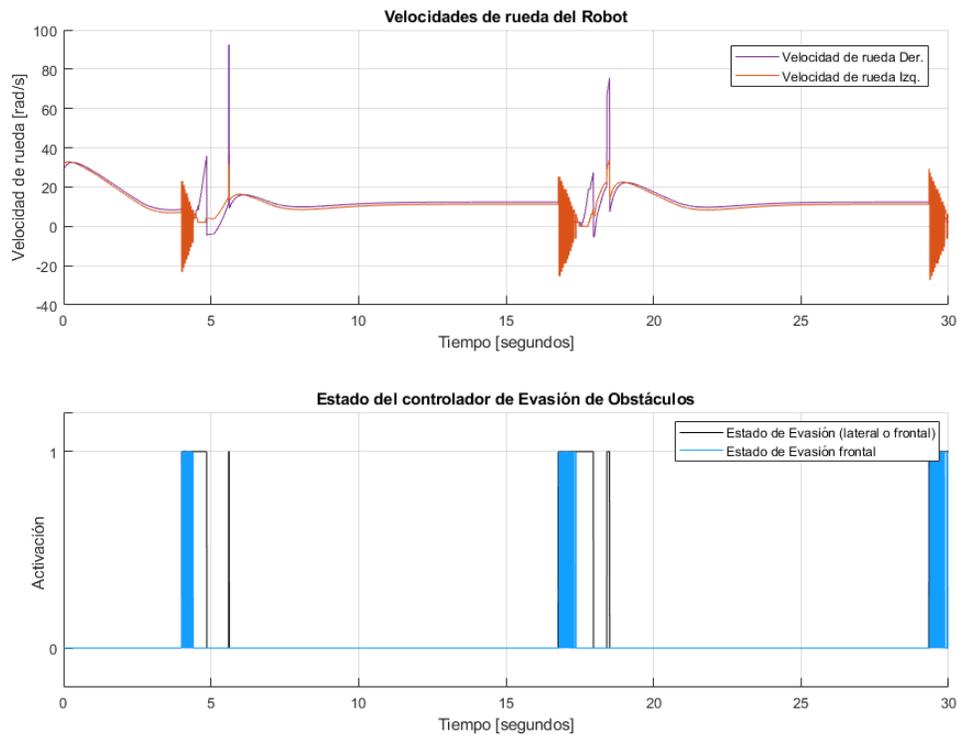


Figura 29. Panel superior: velocidades de rueda. Azul: velocidad de rueda derecha. Roja: velocidad de rueda izquierda. Cuando el controlador de seguimiento está activo, las velocidades de rueda son suaves y tienden a estabilizarse alrededor de un valor constante. Sin embargo, cuando el obstáculo es detectado, se activa el controlador de evasión de obstáculos y, mientras el objeto está a una distancia detectable, las velocidades de rueda exhiben oscilaciones de alta frecuencia. Panel inferior: banderas para identificar qué controlador está activo. Un valor de 1 indica que el controlador de evasión está activo, un valor de cero corresponde al caso en que solamente el controlador de seguimiento de trayectoria está activo. Negro: bandera para el controlador de evasión frontal. Azul: bandera para el controlador de evasión lateral.

4.5. Resultados experimentales

Ahora se presentarán los resultados experimentales obtenidos utilizando únicamente el robot móvil, sin tener conectado el bastón blanco y sin intervención de un usuario. Estos resultados solo tienen el fin de validar el correcto funcionamiento de los controladores. Los experimentos que involucran usuarios se presentan en el siguiente capítulo.

Para los experimentos se utiliza la versión 'melodic' de ROS ya que dicha versión permite trabajar directamente a nivel de velocidades lineales y angulares.

Además, se hacen los ajustes pertinentes para que el sensor LiDAR solo capture las 11 mediciones de interés mostradas la Figura 30, las cuales corresponden a 5 mediciones en el flanco derecho, cinco en el lado izquierdo y una frontal.

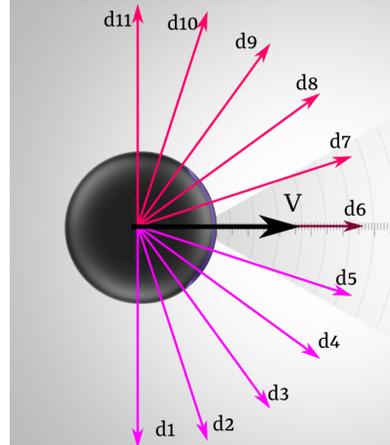


Figura 30. Distribución de las lecturas de interés en el sensor LiDAR.

En un primer experimento se considera una trayectoria deseada compuesta de 4 puntos meta: $Inicio = (0, 0)$, $Meta1 = (0.91, 0)$, $Meta2 = (1.52, -2.44)$, $Meta3 = (3.05, 2.44)$, $Meta4 = (4.57, 0)$ y se utiliza el controlador (8) con $k1 = k2 = 1.2$, $k3 = 3$. y para construir la velocidad v_r del robot virtual de referencia se usa $a = 0.3$ y $b = 0.3$. Los resultados obtenidos se muestran en la Figura 31. Puede verse que el robot móvil visita todos los punto meta y llega al punto meta final. El tiempo que le tomó al robot completar la tarea fue de 23.90 segundos.

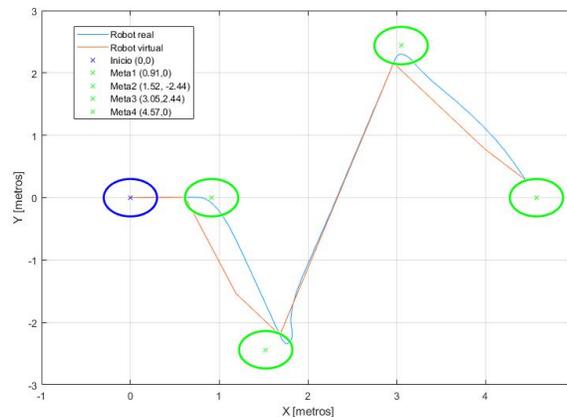


Figura 31. Resultados experimentales del control de seguimiento por retroalimentación por odometría.

Como comentario adicional, nótese que, entre puntos meta, existe una diferencia entre la trayectoria del robot virtual y la del robot móvil real. La razón es que la estrategia de generación de trayectoria considera que el robot ha llegado al punto meta si se encuentra en una vecindad de radio d_{prox} del punto meta, tal y como se discutió en la Sección 3.2. En el caso de este experimento se consideró $d_{prox} = 0.3$ [m]. Por otra parte, las velocidades que se enviaron al robot y las velocidades del robot de referencia se muestran en la Figura 32.

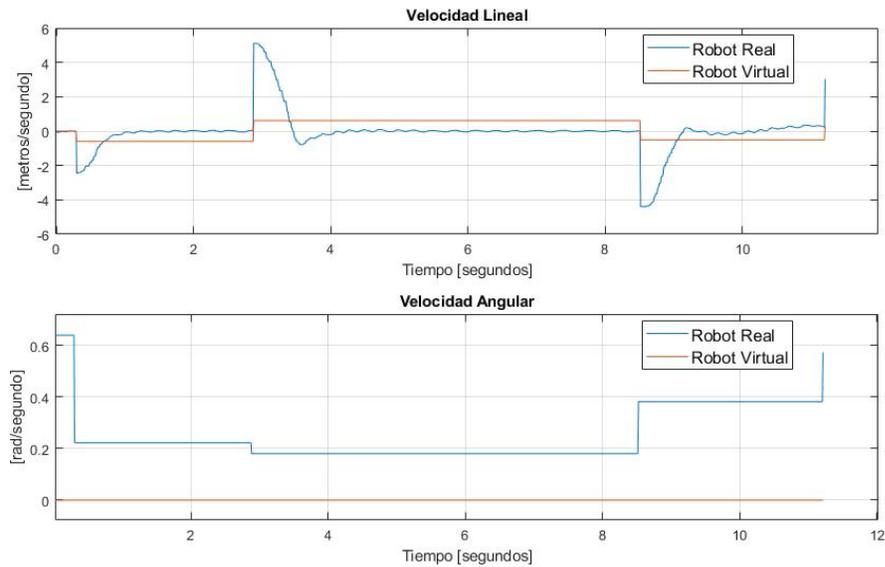


Figura 32. Resultados experimentales. Velocidades del robot virtual y velocidades enviadas al robot.

En lo que respecta al controlador de evasión de obstáculos, se tuvo un problema con la implementación porque el tiempo de procesamiento requerido para realizar la integración numérica de los modelos neuronales y el tiempo de procesamiento del robot eran muy distintos. Una posible alternativa para solucionar este problema es hacer una implementación electrónica de los modelos neuronales. Esto sin embargo, es algo que se propone como trabajo a futuro.

Capítulo 5. Validación experimental del prototipo

Para poder determinar la posible aplicabilidad del robot móvil como guía de personas ciegas o débiles visuales, se realizó un estudio experimental con la participación de seis voluntarios. Dichos experimentos se realizaron en las instalaciones de la Biblioteca Central del Estado de Hidalgo Ricardo Garibay (Biblioteca Central), en la ciudad de Pachuca de Soto, Hidalgo. Dentro de la Biblioteca Central existe la Sala Braille, donde se dedican a actividades que incluyen a personas ciegas en procesos de desarrollo. Estas actividades frecuentemente incluyen a los familiares de las personas invidentes y a personas no ciegas que deseen aprender sobre la ceguera y los retos y desafíos que una persona ciega enfrenta.

A continuación se describen los resultados obtenidos de los experimentos realizados con voluntarios para probar el funcionamiento del robot. Se inicia con pruebas para el ajuste de ganancias del controlador, seguido de pruebas de seguimiento que se considera sencilla, y finalizando con pruebas de seguimiento un poco más complejas.

5.1. Ajuste de los parámetros del robot

Previo a la sesión de experimentos, se realizó una sintonización de los parámetros y ganancias del robot móvil, para lo cual se contó con la participación de un voluntario con ceguera, véase la Figura 33. Esta actividad se realizó en el Laboratorio de Electrónica de la Universidad Autónoma del Estado de Hidalgo (UAEH).



Figura 33. Sesión de prueba con un voluntario con ceguera y su familia.

Además de sintonizar los parámetros del robot y verificar su correcto desempeño, esta sesión de prueba también tuvo como objetivos:

1. Identificar posibles condiciones o situaciones que pusieran en riesgo al usuario.
2. Conocer las sensaciones del usuario al ser guiado por el robot móvil.
3. Encontrar un conjunto de parámetros con los que el usuario se sintiera cómodo (por ejemplo, determinar la velocidad del robot guía a la cual el usuario se sienta seguro).

El experimento realizado consistió en realizar una trayectoria compuesta en forma de 'L', para lo cual se definieron los siguientes puntos meta para generar la trayectoria deseada: $Inicio = (0, 0)$, $Meta1 = (4.88, 0)$, $Meta2 = (4.88, 4.88)$. Estos puntos meta se marcaron con cinta adhesiva en el piso.

Después de una serie de experimentos, ajuste de ganancias $k1$, $k2$, $k3$ y velocidad de referencia V_{ref} , se observó que los parámetros que dieron las mejores sensaciones para el usuario al tiempo que se garantizaba el objetivo de seguimiento de la trayectoria deseada fueron los siguientes: $V_{ref} = 4.42m/s$, $V_{robot_{MAX}} = 1m/s$, $\omega_{robot_{MAX}} = 1rad/s$, $k1 = k2 = 0.3$, $k3 = 1$. El usuario completó la trayectoria deseada en un tiempo total de 18.20 segundos. Los resultados obtenidos se muestran en la Figura 34. El robot virtual y el robot real parten del punto de inicio y se puede observar que es un seguimiento casi perfecto. Una vez que el robot real entra al área de llegada, el robot de referencia calcula su siguiente trayectoria hacia la Meta2. El robot real cambia su orientación y trata de converger a la trayectoria deseada. Una vez que el robot real llega al punto meta, el experimento termina.

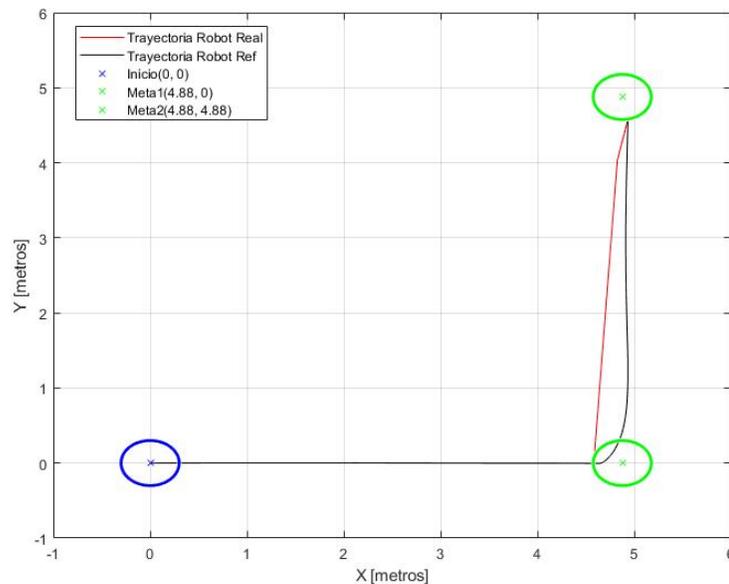


Figura 34. Resultados experimentales con el robot guiando a una persona invidente a lo largo de una trayectoria en L.

En la Figura 35 muestra al usuario siendo guiado por el robot en uno de los experimentos.



Figura 35. Alexis guiado por el robot instantes antes de llegar a un punto meta.

5.1.1. Experimentos con un grupo de voluntarios ciegos

Después de la sesión de ajustes, al día siguiente se realizó la sesión experimental con el grupo de personas invidentes. Para ello, se siguió el siguiente protocolo experimental, el cual se explicó en detalle a cada uno de los participantes en una reunión previa que se tuvo, ver Figura 36.

Protocolo experimental

- Se recopilaron algunos datos de cada una de las personas voluntarias (nombre completo, edad, tiempo con ceguera, causa de la ceguera), para tener un registro y asignar un número identificador (ID) a cada participante.
- Después, se aplicó el cuestionario: 'Características de un guía', el cual se muestra en la sección de

Anexos, con la finalidad de conocer aptitudes que pueden aportar a la tarea de guiar del individuo.

- Posteriormente, se iniciaron con las pruebas experimentales en el que el participante era guiado por el robot móvil a lo largo de una trayectoria. Se iniciaron las pruebas con la trayectoria en 'L'. Cada participante ejecutó esta trayectoria dos veces de manera consecutiva.
- Para la siguiente etapa del experimento, se cambió la trayectoria a una trayectoria del tipo 'zig-zag' y también cada participante la realizó dos veces de manera consecutiva.
- Una vez que la persona participante terminó sus pruebas con ambas trayectorias, se le aplicó el cuestionario: 'Desempeño del Robot', (ver Anexos).
- Finalmente, se aplicó la prueba NASA TLX a cada participante, esta prueba está diseñada para analizar respuestas subjetivas y medir el esfuerzo o carga mental que el usuario requiere (ver Anexos).



Figura 36. Reunión con los participantes del experimento en la que se explicó el protocolo experimental a seguir.

En resumen, los experimentos realizados consistieron de dos etapas. En la primera etapa se consideró una trayectoria en ' L ', y para la segunda etapa se utilizó una trayectoria zigzagante. Previo a los experimentos se aplicó una encuesta relativa a las características que debe tener un guía (un guía en términos generales), y posterior a los experimentos se aplicaron 2 encuestas relativas al desempeño del robot. Los datos de los participantes voluntarios que participaron en este experimento se presentan en la Tabla 1.

Tabla 1. Relación de voluntarios.

ID	Nombre	Edad	Tiempo con ceguera	Padecimiento
01	Arturo	51 años	10 años	Retinosis pigmentaria
02	Alexis	21 años	21 años	Retinopatía del prematuro
03	Zacarías	49 años	20 años	Glaucoma
04	Ivonne	29 años	29 años	Aniridia congénita bilateral
05	María	15 años	15 años	Atrofia Ocular
06	Laura	27 años	27 años	Herencia

5.1.1.1. Experimento 1: trayectoria en L

En un primer experimento la trayectoria deseada consistió de una trayectoria en L utilizando los siguientes puntos meta: $Inicio = (0, 0)$, $Meta1 = (4.88, 0)$, $Meta2 = (4.88, 4.88)$. Además, se utilizaron los siguientes parámetros: $V_{ref} = 4.42m/s$, $k1 = k2 = 0.3$, $k3 = 1$. La Figura 37 muestra las trayectorias descritas por el robot real guiando a los diferentes usuarios.

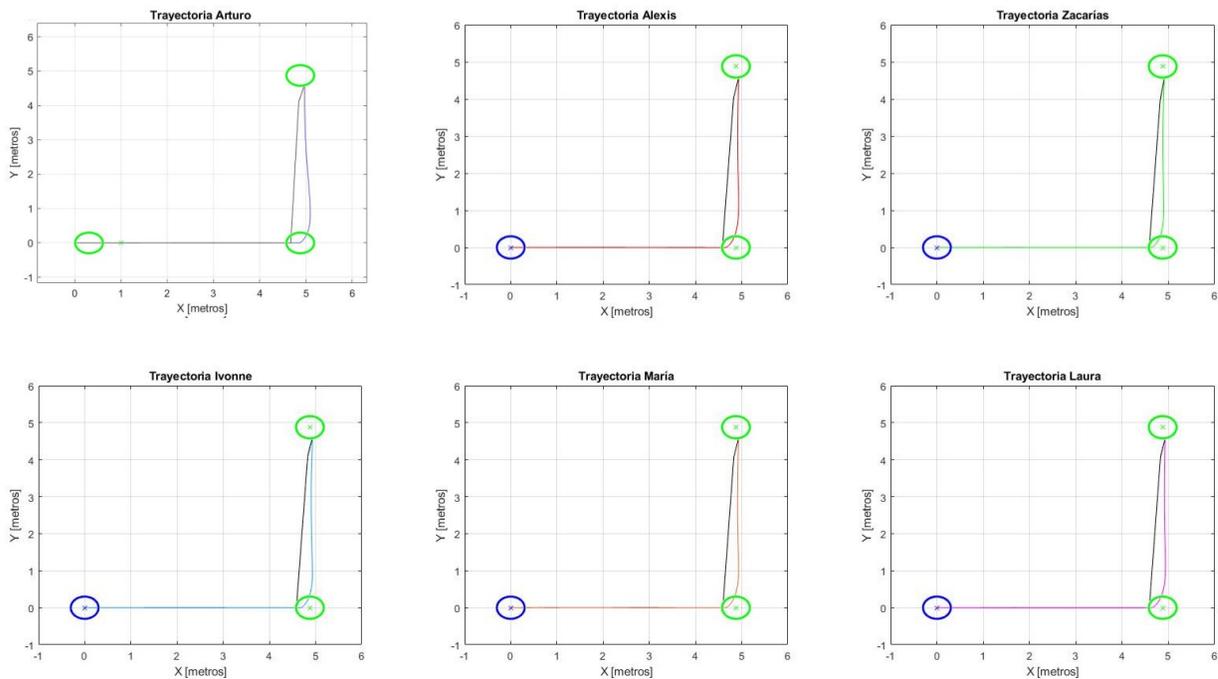
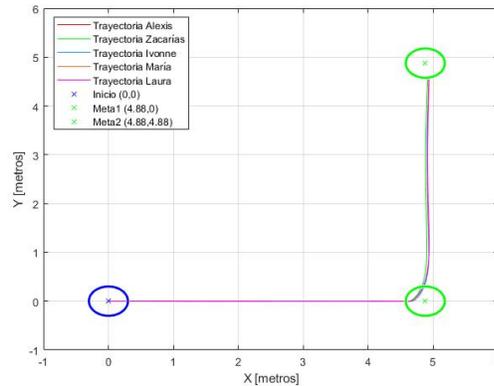


Figura 37. Resultados del experimento de adiestramiento. Las trayectorias descritas por los voluntarios se muestran con curvas de colores, mientras que la trayectoria del robot de referencia se representa con las rectas de color negro.

Por otra parte, en la figura 38 se muestra la comparación de las trayectorias entre los diferentes participantes. El tiempo que tomó a cada uno de los participantes para completar la trayectoria deseada se muestra en la Tabla 2.

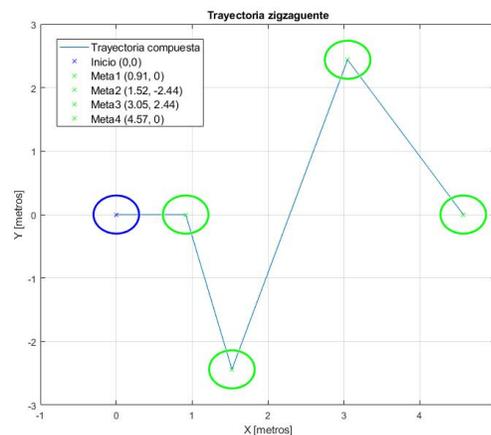
Tabla 2. Tiempo que tomó a cada participante completar el experimento 1.

Trayectoria	Tiempo
Alexis	18.20 segundos
Zacarías	17.58 segundos
Ivonne	17.72 segundos
María	18.55 segundos
Laura	18.50 segundos

**Figura 38.** Comparación de las trayectorias ejecutadas por los participantes.

5.1.2. Experimento 2: trayectoria zigzagueante

Para el segundo experimento, se tomó en consideración la trayectoria que se muestra en la Figura 39, descrita por los siguientes puntos: $Inicio(0, 0)$, $Meta1(0.91, 0)$, $Meta2(1.52, -2.44)$, $Meta3(3.05, 2.44)$, $Meta4(4.57, 0)$. En este caso, los parámetros del experimento son: $V_{ref} = 4.42m/s$, $\omega_{ref} = 0m/s$, $k1 = k2 = 0.3$, $k3 = 1$.

**Figura 39.** Trayectoria en zig-zag usada en el segundo experimento.

Los resultados del experimento se muestran en la Figura 40, en la que se aprecia que la mayoría de los participantes lograron concluir satisfactoriamente el experimento, excepto uno (ver panel central inferior de la figura). Un inconveniente ocasional en esta ronda de experimentos fue que cuando el robot cambiaba su velocidad, se generaban fuerzas que hacían que el robot se levantara de la parte de enfrente y anduviera solo sobre sus ruedas traseras, tal y como se muestra en la Figura 41, ocasionando que el robot no pudiera controlar sus movimientos. Este voluntario no pudo concluir el experimento, a pesar de que se intentó hasta en tres ocasiones y en las tres se observó el mismo comportamiento del robot móvil.

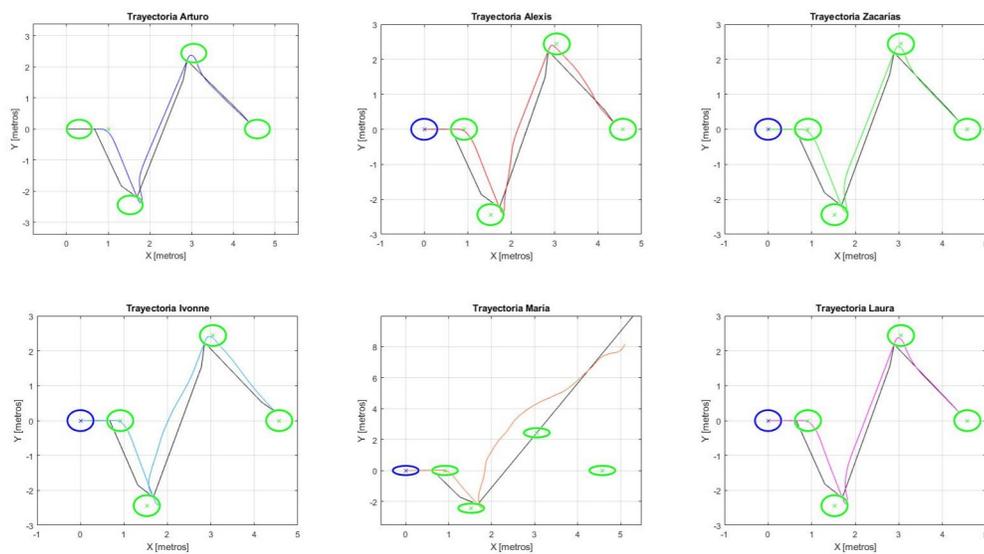


Figura 40. Gráficas de las trayectorias de cada voluntario en el segundo experimento.



Figura 41. Robot operando de manera indebida.

La Tabla 3 muestra los tiempos que les tomó a los participantes concluir satisfactoriamente el experimento.

Tabla 3. Tiempo que tomo a cada participante concluir el experimento 2.

Trayectoria	Tiempo
Alexis	13.03 segundos
Zacarías	12.85 segundos
Ivonne	13.08 segundos
María	Inconcluso
Laura	12.96 segundos

5.2. Resultados de los cuestionarios

Además de las pruebas experimentales realizadas, se aplicaron tres distintos cuestionarios a los sujetos de prueba con el fin de obtener retroalimentación sobre el desempeño de ROSbot y sobre sus necesidades al ser guiados. El primer cuestionario titulado: 'Características de un guía', véase el Anexo B, se enfocó en conocer las particularidades que un guía debe de tener de acuerdo a sus experiencias previas. En este cuestionario se observó que los seis usuarios mencionan la importancia de ser anticipados de los potenciales obstáculos. Buscan que se les comunique sobre las características del entorno y se les indiquen de manera clara los obstáculos. Cuatro de seis usuarios especificaron que las distracciones que genere el guía dificulta el trayecto, además de no ser de su agrado. Un guía que comete errores es la principal razón para ya no buscar la ayuda del guía. Conjuntamente, cinco de seis usuarios señalan la empatía como la principal característica que buscan en un guía.

En el segundo cuestionario titulado: 'Desempeño del robot', el cual se incluye en el Anexo C, se investigaron los detalles posteriores a las pruebas experimentales de seguimiento de trayectoria realizadas. En promedio, los usuarios calificaron la comodidad del uso del robot con una ponderación de 8/10 teniendo en consideración variables como: velocidad, practicidad y diseño. En cuanto a la seguridad que apreciaron utilizando el ROSbot lo calificaron en promedio con 7/10. Cuatro de seis usuarios mencionan que la velocidad utilizada es más lenta de lo que les gustaría.

Debido a que las pruebas experimentales realizadas toman en consideración variables basadas en la percepción, se decidió aplicar el método NASA TLX (Task Load Index), el cual se incluye en el Anexo D. Este método propone un medio para ponderar la carga mental desde una perspectiva multidimensional

sobre una actividad en específico. La tarea evaluada fue la guía de lazarillo con ROSbot. Para aplicar este método se consideran seis dimensiones de esfuerzo: Esfuerzo Mental (M), Esfuerzo Físico (F), Demanda Temporal (T), Rendimiento (R), Esfuerzo (E) y Nivel de frustración (NF).

Los resultados de la prueba son interpretados con base en la siguiente tabla (Prabaswari et al., 2019)

Tabla 4. Tabla de interpretación de resultados NASA TLX

Carga o esfuerzo de la tarea	Puntaje
Fácil	0-9
Medio	10-29
Poco elevado	30-49
Difícil	50-79
Muy difícil	80-100

En la Tabla 5 se muestran pesos o ponderaciones obtenidos de cinco de los seis participantes, el participante con ID número 05 no pudo realizar el cuestionario NASA TLX debido a era una joven menor que tenía que retirarse con sus padres.

Tabla 5. Tabla de ponderación por dimensión y por participante.

Peso o ponderación						
ID	M	F	T	R	E	NF
01	5	2	4	3	1	0
02	5	2	3	4	1	0
03	3	2	2	4	3	1
04	2	1	5	4	3	0
06	2	2	3	4	4	0

En la Tabla 6 se muestran el puntaje que los usuarios dieron al esfuerzo requerido durante el experimento; calificando en una escala de cero a veinte. Entre más alta sea la puntuación, mayor es el esfuerzo requerido. Posteriormente, se representan los datos en una escala de 0 a 100 dentro de la misma tabla.

Tabla 6. Tabla normalizada de puntaje de esfuerzo por dimensión y por usuario.

Valoraciones 0 - 20							Valoraciones escala 0 - 100					
ID	M	F	T	R	E	NF	M	F	T	R	E	NF
01	10	5	5	2	11	7	50	25	25	10	55	35
02	11	1	1	2	8	9	55	5	5	10	40	45
03	2	3	6	2	3	2	10	15	30	10	15	50
04	1	4	8	2	2	1	5	20	40	10	10	5
06	6	6	7	2	6	1	30	30	35	10	30	5

Dadas las ponderaciones de las dimensiones de los participantes, y la puntuación con la que calificaron la carga de trabajo del experimento, se pueden obtener los resultados de la carga de trabajo que le tomó a cada participante realizar el experimento de ser guiados por el guía robótico ROSbot con el controlador de seguimiento de trayectoria. Los resultados se muestran en la siguiente tabla.

Tabla 7. Resultados NASA TLX para la tarea de guía lazarillo con ROSbot y el controlador de seguimiento de trayectoria.

ID	Puntaje
01	32.33
02	25.33
03	14.33
04	20.00
06	25.67

De los resultados mostrados en la Tabla 7, se observa que el esfuerzo mental promedio que requirieron los participantes para los experimentos es de 23.532. Y de acuerdo con la Tabla 4, este valor corresponde con un esfuerzo medio, por lo que se concluye que el uso del prototipo de guía robótico desarrollado en este trabajo no representa una carga mental significativa para el usuario.

Capítulo 6. Discusión y conclusiones

A lo largo del proyecto se observaron diversos aspectos que no se tenían contemplados al inicio del mismo. Algunos de estos aspectos, afectaron de manera considerable el alcance de este proyecto, por ejemplo, el tiempo de procesamiento para el control de evasión de obstáculos. Al inicio del proyecto se consideró que el poder de procesamiento de una computadora “común”, podría ser suficiente para ejecutar el controlador híbrido. Se intentó cargar parte del programa en la computadora del robot y que otra parte lo ejecutara la computadora externa (laptop), pero aún así no se logró resolver este problema de tiempo de procesamiento

Una observación importante es que el controlador para seguimiento de trayectoria permitía velocidades negativas, generando condiciones que no se consideraban para la experimentación al momento de utilizar un robot como guía. Restringir la velocidad lineal del robot a valores positivos permitió observar un comportamiento diferente del controlador, pero que seguía cumpliendo con el objetivo.

Aunado a las velocidades, el ajuste de las ganancias fue importante para lograr adecuar las velocidades dentro de los parámetros de giro y avance del robot. Se apreciaron experimentos con usuarios y simulaciones en computadora en los que el robot podía cumplir el objetivo o fallar, solo ajustando las ganancias del controlador.

En la etapa de experimentación, se pudieron observar dos oportunidades de mejora fundamentales. Todos los voluntarios estuvieron de acuerdo en que el robot necesita algún tipo de retroalimentación por parte del usuario. Por otro lado, los voluntarios también estuvieron de acuerdo en que los movimientos del robot podrían ser más suaves, esto con el fin de brindarles mayor confort y equilibrio.

En la etapa de experimentación se apreció que el mecanismo que une al robot con el bastón blanco, funciona correctamente, sin embargo, hubo un momento en el que al ejercer cierta presión en el bastón blanco, el robot fácilmente levantaba sus ruedas delanteras. Esto ocasionaba que el robot no fuera operable. Posiblemente a causa de la baja estatura del participante y la ubicación de la interfaz. Esto puede dar origen a una parametrización y/o personalización de la interfaz para diferentes usuarios.

6.1. Conclusiones

La integración del robot con la interfaz mecánica y el control de seguimiento de trayectoria, es capaz de ser utilizado como guía de personas ciegas navegando a través de superficies planas.

Los resultados numéricos del control de navegación híbrido sugieren que este método de navegación es funcional para la aplicación experimental en la guía de personas invidentes, siempre y cuando se logren obtener en tiempo real los resultados de la parte de evasión de obstáculos.

La selección del robot es adecuada debido a que cuenta con las características mecánicas, instrumentales y computacionales para ser un prototipo experimental en aplicaciones de guía de personas ciegas. Lo que permitirá continuar y aumentar las investigaciones en este tema.

Con base en los resultados experimentales, las seis personas con ceguera fueron sumamente abiertas y proactivas a participar y retroalimentar este proyecto, lo que indica que esta población está dispuesta a participar en proyectos que mejoren o impulsen su desempeño.

6.2. Trabajo a futuro

Utilizar tarjetas electrónicas que simulen el comportamiento del modelo neuronal HR que se utiliza para la parte de evasión de obstáculos, para así, reducir los tiempos de respuesta en esta sección del controlador y poder realizar experimentación con el controlador híbrido.

Agregar un control de seguimiento de trayectoria que permita movimientos uniformemente acelerados. Esto permitirá un mejor comportamiento del robot, ayudará a la integridad de los motores y mecanismos del robot y brindará una sensación de navegación más placentera.

Agregar al robot la capacidad de transmitir información del entorno al usuario. Información sobre obstáculos y las características del espacio a través de sonidos como alarmas cortas y/o palabras. Esto ayudaría al usuario a estar mentalmente preparado y mejoraría el desempeño de los experimentos.

Integrar una interfaz háptica que permita intercambio de información en dos vías entre el robot y el usuario. Esto permite la posibilidad de ajustar parámetros como aceleración, tiempos, trayectorias con base al usuario.

Considerar una interfaz que pueda ajustarse al usuario en función de sus características físicas.

Evaluaciones o experimentaciones con una mayor cantidad de participantes, así como una mayor cantidad de pruebas o tareas a ejecutar.

Literatura citada

- Attia, I. & Asamoah, D. (2020). The white cane. its effectiveness, challenges and suggestions for effective use: The case of akropong school for the blind. *Journal of Education, Society and Behavioural Science*, 33(3), 47–55. <https://doi.org/10.9734/jesbs/2020/v33i330211>.
- Campbell, S., O'Mahony, N., Carvalho, A., Krpalkova, L., Riordan, D., & Walsh, J. (2020). Path planning techniques for mobile robots a review. In *2020 6th International Conference on Mechatronics and Robotics Engineering (ICMRE)*, 12–16. <https://doi.org/10.1109/ICMRE49073.2020.9065187>.
- Capi, G. & Toda, H. (2011). A new robotic system to assist visually impaired people. In *2011 RO-MAN*, 259–263. <https://doi.org/10.1109/ROMAN.2011.6005259>.
- de Wit, C. C., Siciliano, B., & Bastin, G. (Eds.). (2012). *Theory of Robot Control*. Springer Science & Business Media, London.
- Glenk, L. M., Příbylová, L., Stetina, B. U., Demirel, S., & Weissenbacher, K. (2019). Perceptions on health benefits of guide dog ownership in an austrian population of blind people with and without a guide dog. *Animals*, 9(7), 428. <https://doi.org/10.3390/ani9070428>.
- Hindmarsh, J. L., Rose, R. M., & Huxley, A. F. (1984). A model of neuronal bursting using three coupled first order differential equations. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 221(1222), 87–102. <https://doi.org/10.1098/rspb.1984.0024>.
- Hub, A., Diepstraten, J., & Ertl, T. (2003). Design and development of an indoor navigation and object identification system for the blind. In *Proceedings of the 6th International ACM SIGACCESS Conference on Computers and Accessibility, Assets '04*, 147–152, New York, NY, USA. Association for Computing Machinery. <https://doi.org/10.1145/1028630.1028657>.
- Juarez-Lora, A. & Rodriguez-Angeles, A. (2023). Bio-inspired autonomous navigation and formation controller for differential mobile robots. *Entropy*, 25(4). <https://doi.org/10.3390/e25040582>.
- Kayukawa, S., Ishihara, T., Takagi, H., Morishima, S., & Asakawa, C. (2020). Blindpilot: A robotic local navigation system that leads blind people to a landmark object. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems, CHI EA '20*, 1–9, New York, NY, USA. Association for Computing Machinery. <https://doi.org/10.1145/3334480.3382925>.
- Khatib, O. (1985). Real-time obstacle avoidance for manipulators and mobile robots. In *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, volume 2, 500–505. <https://doi.org/10.1109/ROBOT.1985.1087247>.
- Lahav, O., Schloerb, D., Kumar, S., & Srinivasan, M. (2012). Navigation systems for the blind and visually impaired: Past work, challenges, and open problems. *Journal of Assistive Technologies*, 19(15), 38–52. <https://doi.org/10.3390/s19153404>.
- Lamini, C., Benhlima, S., & Elbekri, A. (2018). Genetic algorithm based approach for autonomous mobile robot path planning. *Procedia Computer Science*, 127, 180–189. Proceedings of the Firts International Conference on Intelligent Computing in Data Sciences, ICDS2017, <https://doi.org/https://doi.org/10.1016/j.procs.2018.01.113>.
- Lee, M.-F. R., Chiu, F. H. S., & Zhuo, C. (2013). Novel design of a social mobile robot for the blind disabilities. In *Proceedings of the 2013 IEEE/SICE International Symposium on System Integration*, 161–166. <https://doi.org/10.1109/SII.2013.6776710>.
- Prabaswari, A. D., Basumerda, C., & Utomo, B. W. (2019). The mental workload analysis of staff in study program of private educational organization. *IOP Conference Series: Materials Science and Engineering*, 528(1), 012018. <https://doi.org/10.1088/1757-899X/528/1/012018>.

- Real & Araujo (2019). A virtual environment for people who are blind — a usability study. *Sensors*, 6(1), 3404. <https://doi.org/10.1108/17549451211214346>.
- Saitis, C. & Kalimeri, K. (2016). Identifying urban mobility challenges for the visually impaired with mobile monitoring of multimodal biosignals. In Antona, M. & Stephanidis, C. (Eds.), *Universal Access in Human-Computer Interaction. Users and Context Diversity*, 616–627, Cham. Springer International Publishing.
- Sanchez, E. M., Ramirez, J. P., & Angeles, A. R. (2021). Autonomous navigation of a mobile robot using a network of hindmarsh-rose (hr) neurons. In *2021 18th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*, 1–6. <https://doi.org/10.1109/CCE53527.2021.9633035>.
- Toha, S. F., Yusof, H. M., Razali, M. F., & Halim, A. H. A. (2015). Intelligent path guidance robot for blind person assistance. In *2015 International Conference on Informatics, Electronics & Vision (ICIEV)*, 1–5. <https://doi.org/10.1109/ICIEV.2015.7334040>.
- Ulrich, I. & Borenstein, J. (2001). The guidecane-applying mobile robot technologies to assist the visually impaired. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 31(2), 131–136. <https://doi.org/10.1109/3468.911370>.
- Van Parys, R. & Pipeleers, G. (2017). Spline-based motion planning in an obstructed 3d environment. *IFAC-PapersOnLine*, 50(1), 8668–8673. 20th IFAC World Congress, <https://doi.org/https://doi.org/10.1016/j.ifacol.2017.08.1525>.
- Velasco Equihua, G. G. & Ramirez, J. P. (2018). Synchronization of hindmarsh-rose neurons via Huygens-like coupling. *IFAC-PapersOnLine*, 51(33), 186–191. 5th IFAC Conference on Analysis and Control of Chaotic Systems CHAOS 2018, <https://doi.org/https://doi.org/10.1016/j.ifacol.2018.12.115>.
- Vromen, T. G. M., Steur, E., & Nijmeijer, H. (2016). Training a network of electronic neurons for control of a mobile robot. *International Journal of Bifurcation and Chaos*, 26(12), 1650196. <https://doi.org/10.1142/S0218127416501960>.
- Xu, P., Kennedy, G. A., Zhao, F.-Y., Zhang, W.-J., & Van Schyndel, R. (2023). Wearable obstacle avoidance electronic travel aids for blind and visually impaired individuals: A systematic review. *IEEE Access*, 11, 66587–66613. <https://doi.org/10.1109/ACCESS.2023.3285396>.
- Yu, J., Su, Y., & Liao, Y. (2020). The path planning of mobile robot by neural networks and hierarchical reinforcement learning. *Frontiers in Neurorobotics*, 14. <https://doi.org/10.3389/fnbot.2020.00063>.
- Zhao, J., Hu, Y., Liu, C., Tian, M., & Xia, X. (2022). Spline-based optimal trajectory generation for autonomous excavator. *Machines*, 10(7). <https://doi.org/10.3390/machines10070538>.

Anexos

A continuación se describe como establecer una comunicación bilateral con el ROSbot. Se establece el protocolo de comunicación vía VPN desarrollada por el proveedor, para poder acceder al CPU del robot desde una computadora personal externa con sistema operativo Windows, y permitir la operación del robot via Wifi. Se explica brevemente el entorno ROS y se configura una red ROS para el intercambio de información entre nodos externos (laptop de control remoto) y nodos internos del robot (CPU del robot). La información que se intercambia, son las velocidades de movimiento calculadas por la computadora externa y la información recolectada por los sensores del robot. De manera concreta, se realiza el control de velocidades en lazo cerrado y se inicia la configuración del LiDAR para el muestreo deseado.

Comunicación remota por VPN

El robot móvil ROSbot, cuenta con un CPU [1] integrado como unidad de procesamiento. El CPU cuenta con Ubuntu como sistema operativo (SO) configurado por el proveedor Husarion [2], en donde ya se encuentran cargados los paquetes o algoritmos que permiten la manipulación del ROSbot. Para ejecutar los algoritmos, es necesario acceder al CPU del robot y ejecutarlos; lo mismo para escribir los algoritmos propios. Para acceder al SO del robot, se pueden conectar periféricos como monitor y teclado; también se puede acceder de manera remota via Wi-fi. Existen diferentes formas de acceder al SO del robot via Wi-fi, sin embargo, la opción elegida en este trabajo, es hacerlo a través del servicio VPN llamado Husarnet.

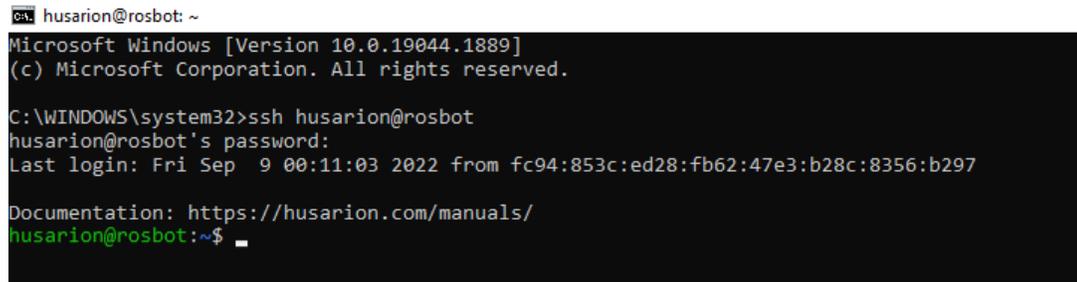
Para establecer este protocolo de comunicación hay que realizar las siguientes acciones:

- 1 Conectar ROSbot a Internet [3]
- 2 Crear una cuenta en <https://app.husarnet.com> el cual es la VPN Husarnet
- 3 Instalar la aplicación en la laptop de control y en el ROSbot [4].
- 4 Agregar los dispositivos al servicio VPN [5] (laptop y ROSbot).

Una vez que ambos dispositivos se encuentran con conexión a internet y dados de alta en la VPN, se abre una ventana de comandos en la laptop de control y se ejecuta la instrucción

```
ssh husarion@NombreRosbotEnVPN
```

Posteriormente ROSbot pedirá la clave de usuario. Ahora la ventana de comandos nos permite enviar instrucciones directamente al SO del robot. En la Figura 42 se muestra una ventana de comandos (laptop de control remoto), entrar al SO del robot.



```
ca husarion@rosbot: ~
Microsoft Windows [Version 10.0.19044.1889]
(c) Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>ssh husarion@rosbot
husarion@rosbot's password:
Last login: Fri Sep  9 00:11:03 2022 from fc94:853c:ed28:fb62:47e3:b28c:8356:b297

Documentation: https://husarion.com/manuals/
husarion@rosbot:~$
```

Figura 42. Ejemplo de configurar una ventana de comandos de la laptop de control, para navegar en el SO del ROSbot.

NOTA: Hay que realizar el mismo inicio de sesión para cada ventana de comandos que se ejecute. La clave de usuario del SO del ROSbot es *husarion*

De esta manera, podemos tener acceso al SO del robot con la ventaja de ejecutar algoritmos y que el robot pueda navegar libremente bajo la cobertura de una red Wi-fi.

Entorno ROS

Robot Operating System (ROS) es un paquete de programas destinados al desarrollo de sistemas robóticos autónomos. El cual cuenta con algoritmos comunmente utilizados en robótica. Los elementos principales dentro de una red ROS son los nodos (nodes), los tópicos (topics) y los servicios (services). ROS es distribuido en paquetes que contienen algoritmos destinados a un fin particular, por ejemplo, paquetes destinados a sensores lidar, o paquetes destinados al control de velocidades de llanta. Los algoritmos contenidos en un paquete son llamados nodos. Un paquete puede contener uno o múltiples nodos.

Nodo

Los nodos son la unidad base de ROS, estos se encargan del manejo de dispositivos o cómputo de datos. Un nodo es un algoritmo dedicado a una tarea específica. Los nodos se pueden comunicar entre ellos utilizando tópicos o servicios.

Tópico

Los tópicos son el canal utilizado para intercambiar información entre nodos en forma de *mensajes*. Los tópicos son utilizados para enviar mensajes. Un tópico es registrado con un nombre único y con un tipo de mensaje determinado. Un tópico no puede ser utilizado para enviar diferentes tipos de mensajes. Un ejemplo puede ser la lectura de un sensor, o la velocidad para un motor. Los nodos pueden conectarse a un tópico como publicadores (emiten mensajes) o subscriptores (reciben mensajes), pero no pueden usar el mismo tópico para publicar y suscribirse. Por otra parte, no existen restricciones al número de nodos diferentes que publican o se suscriben a través de un mismo tópico.

Servicio

Comunicación por servicios puede parecerse al modelo de cliente-servidor. En este modo, un nodo (servidor) registra servicios al sistema. Posteriormente otro nodo puede solicitar el servicio y obtener respuesta. A diferencia de los tópicos, los servicios permiten comunicación bilateral, ya que la solicitud de un servicio puede contener información tanto en la solicitud como en la respuesta.

La Figura 43 muestra una lista de tópicos y nodos disponibles en ROSbot. La lista no es exhaustiva y además pueden agregarse nuevos paquetes. A manera de ejemplo, si ejecutamos el nodo *robot_ekf*, este publicará la odometría a través del tópico */odom*. De la misma manera, el nodo estará recibiendo los comandos de velocidad a través del tópico */cmd_vel*.

Topic	Message type	Direction	Node	Description
/mpu9250	rosbot_ekf/Imu	publisher	/serial_node	Raw IMU data in custom message type
/range/fl	sensor_msgs/Range	publisher	/serial_node	Front left range sensor raw data
/range/fr	sensor_msgs/Range	publisher	/serial_node	Front right range sensor raw data
/range/rl	sensor_msgs/Range	publisher	/serial_node	Rear left range sensor raw data
/range/rf	sensor_msgs/Range	publisher	/serial_node	Rear right range sensor raw data
/joint_states	sensor_msgs/JointState	publisher	/serial_node	Wheels rotation angle
/battery	sensor_msgs/BatteryState	publisher	/serial_node	Battery voltage
/buttons	std_msgs/UInt8	publisher	/serial_node	User buttons state, details in User buttons section
/pose	geometry_msgs/PoseStamped	publisher	/serial_node	Position based on encoders
/odom/wheel	nav_msgs/Odometry	publisher	/msgs_conversion	Odometry based on wheel encoders
/velocity	geometry_msgs/Twist	publisher	/serial_node	Odometry based on encoders
/imu	sensor_msgs/Imu	publisher	/msgs_conversion	IMU data wrapped in standard ROS message type
/odom	nav_msgs/Odometry	publisher	/rosbot_ekf	Odometry based on sensor fusion
/tf	tf2_msgs/TFMessage	publisher	/rosbot_ekf	ROSbot position based on sensor fusion
/set_pose	geometry_msgs/ PoseWithCovarianceStamped	subscriber	/rosbot_ekf	Allow to set custom state of EKF
/cmd_vel	geometry_msgs/Twist	subscriber	/serial_node	Velocity commands
/config	rosbot_ekf/Configuration	service server	/serial_node	Allow to control behaviour of CORE2 board, detaild in CORE2 config section

Figura 43. Algunos tópicos y nodos disponibles en el sistema operativo del ROSbot que presenta el fabricante [7].

Configurar una red ROS

Una red Ros se establece al momento de ejecutar dos o más nodos que intercambien información. Estos nodos pueden encontrarse en el mismo dispositivo o ser de diferentes dispositivos. Para el primer caso,

solo basta ejecutar los nodos de interés, ya que el direccionamiento de los archivos queda establecido al encontrarse en el mismo dispositivo. Para el segundo caso, se necesita establecer una capa de direccionamiento por medio de las direcciones IP de los dispositivos involucrados.

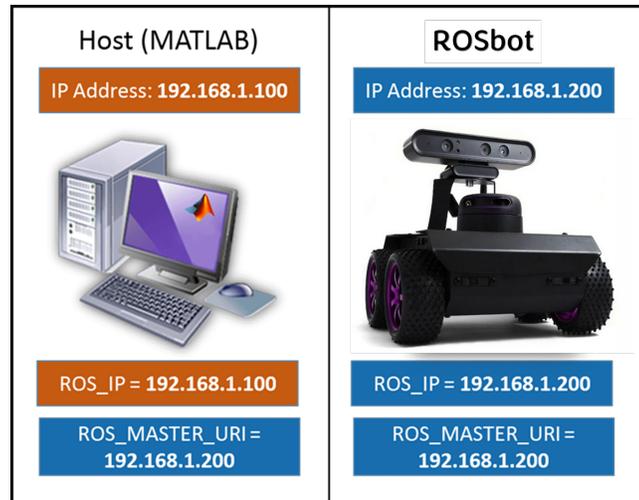


Figura 44. Configuración para una red ROS entre una laptop de control y ROSbot.

Para establecer una red ROS entre el ROSbot y la laptop de control, hay que formar una red modelo Maestro-Eslavo como se muestra en la Figura 44. Esta red puede ser a través de una red LAN, WLAN o internet. En este caso se utiliza una red que es establecida por el modem y conexión a internet. Una vez conectando ambos dispositivos a la red Wifi, se obtienen sus direcciones IP y se realizan los ajustes al entorno ROS de cada dispositivo. Suponiendo las siguientes direcciones IP sean las mostradas en la figura anterior.

laptop (Host MATLAB) IP Address: 192.168.1.100

ROSbot (ROS Master) IP Address: 192.168.1.200

Se puede empezar configurando el entorno ROS del ROSbot abriendo una terminal via VPN desde la laptop de control. Se abre el archivo `/.bashrc`, y se localizan las líneas:

```
export ROS_MASTER_URI=http://master:11311
```

```
export ROS_IPV6=on
```

y se cambian por:

```

export ROS_MASTER_URI=http://192.168.1.200:11311

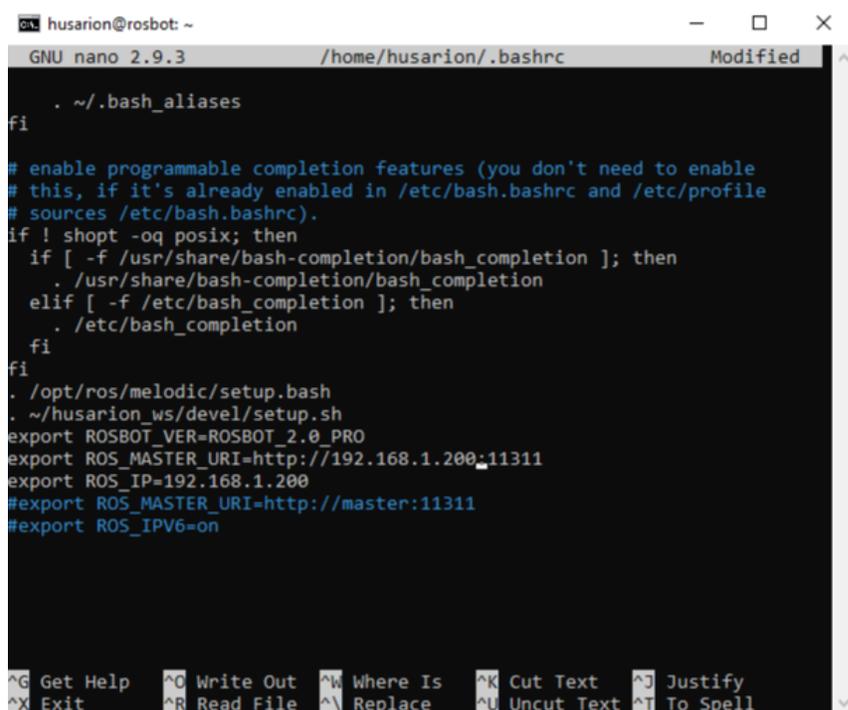
export ROS_IP=192.168.1.200

#export ROS_MASTER_URI=http://master:11311

#export ROS_IPV6=on

```

En la Figura 45 se muestra un ejemplo de esta operación.



The image shows a terminal window titled 'husarion@rosbot: ~' running the GNU nano 2.9.3 editor on the file /home/husarion/.bashrc. The file content includes the following lines:

```

. ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
  if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
fi

. /opt/ros/melodic/setup.bash
. ~/husarion_ws/devel/setup.sh
export ROSBOT_VER=ROSBOT_2.0_PRO
export ROS_MASTER_URI=http://192.168.1.200:11311
export ROS_IP=192.168.1.200
#export ROS_MASTER_URI=http://master:11311
#export ROS_IPV6=on

```

The terminal window also shows a status bar at the bottom with various keyboard shortcuts for nano editor operations like Get Help, Write Out, Where Is, Cut Text, Justify, Exit, Read File, Replace, Uncut Text, and To Spell.

Figura 45. Configuración del ROSbot como dispositivo maestro.

Para configurar la laptop de control, se debe iniciar MATLAB y configurar la información de ambiente ROS con la herramienta -Configure ROS Network Adresses, tal como se ejemplifica en la Figura 46, en la sección ROS Master (ROS), se agrega la dirección IP del Robot. En la sección Node Host (ROS), se agrega la dirección IP del ordenador. De esta manera, cuando se ejecutan los programas en Simulink en la laptop de control, el entorno ROS interpreta que se ha ejecutado un nodo dentro de la red ROS del ROSbot.

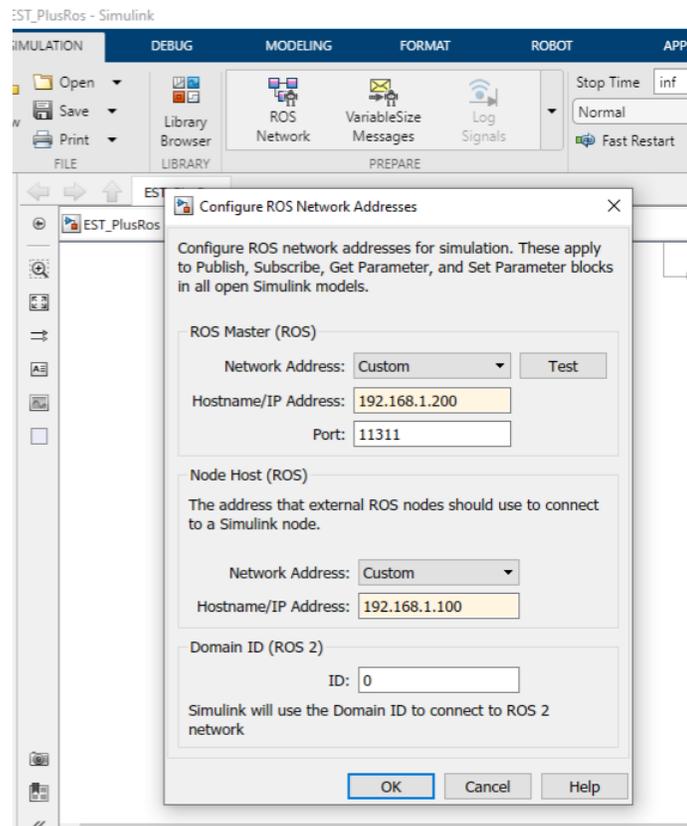


Figura 46. Ventana de configuración para el ambiente ROS en MATLAB.

Control de velocidades con retroalimentación por odometría

Como se mencionó en la sección anterior, ahora los programas que se desarrollen en Simulink pueden ser interpretados como nodos en el entorno ROS. Estos nodos programados en Simulink pueden comunicarse con otros nodos del ROSbot. De esta manera, se pueden intercambiar mensajes para el control o lectura de parámetros del robot.

El nodo que se desarrolla a continuación, es un nodo programado en Simulink para control de las velocidades del robot. El controlador se retroalimenta con odometría calculada por un nodo del ROSbot. Para compartir información, hay que configurar los tópicos por los cuales se comunicará el nodo en Simulink. De manera específica para el ROSbot y el controlador utilizado, se elaboran los diagramas de bloques ilustrados en la Figura 47.

En la etapa de suscripción, Simulink tiene acceso al mensaje `nav_msgs/odometry` a través del tópico `/odom/wheel`. Seguido del bloque de Suscripción, se complementa con otros bloques para adquirir los parámetros que nuestro controlador requiere. En este reporte, los parámetros de interés son $[X, Y, \theta]^T$. En

la etapa de publicación, se enmascaran las velocidades calculadas en el mensaje `geometry_msgs/Twist`, y se envían a través del tópico `/cmd_vel`.

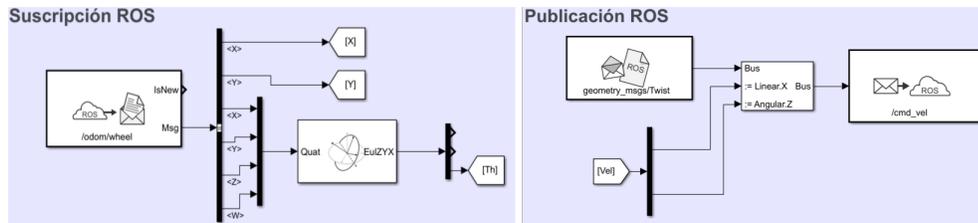


Figura 47. Suscripción y Publicación del nodo Simulink para compartir mensajes.

Para activar el nodo del ROSbot que calcula la odometría (entre otras funciones), se ejecuta el siguiente comando de manera remota:

```
roslaunch rosbot_ekf all.launch rosbot_pro:=true
```

Al ejecutar este comando, se activa el entorno ROS en el ROSbot y también se activan diferentes nodos dentro del paquete `rosbot_ekf` (paquete con el mismo nombre que el nodo). Los nodos que se habilitan con este comando se muestran en el lado izquierdo de la figura 48. Una vez que el entorno ROS se encuentra encendido y el nodo `rosbot_ekf` se ha habilitado, estamos listos para intercambiar información de odometría y enviar velocidades de rueda al robot.

```

husarion@rosbot: ~
husarion@rosbot:~$ rosnode list
/imu_publisher
/msgs_conversion
/rosbot_ekf
/rosout
/serial_node
husarion@rosbot:~$

husarion@rosbot: ~
husarion@rosbot:~$ rosnode list
/EST_PlusRos_63236
/imu_publisher
/msgs_conversion
/rosbot_ekf
/rosout
/serial_node
husarion@rosbot:~$

```

Figura 48. Nodos habilitados para el control de velocidades con retroalimentación por odometría.

Para habilitar el nodo programado en Simulink, hay que ejecutar el programa. Al hacerlo, se puede verificar su incorporación utilizando el comando `rosnode list`. En el lado derecho de la Figura 48, se aprecia la incorporación del nodo de Simulink bajo el nombre `EST_PlusRos_63236`.

Simulación Odometría

En una simulación de 19s utilizando el control de velocidades [8], fundamentado en [9]; con los siguientes parámetros: C.I. RV = [0.3; 0.8; 0], C.I. ROSbot = [0; 0; 0], $[v_{ref}; \omega_{ref}]^T = [0.2; 0]^T$, $V_{MAX} = 0.2\text{m/s}$, se obtuvieron los resultados de la Figura 49.

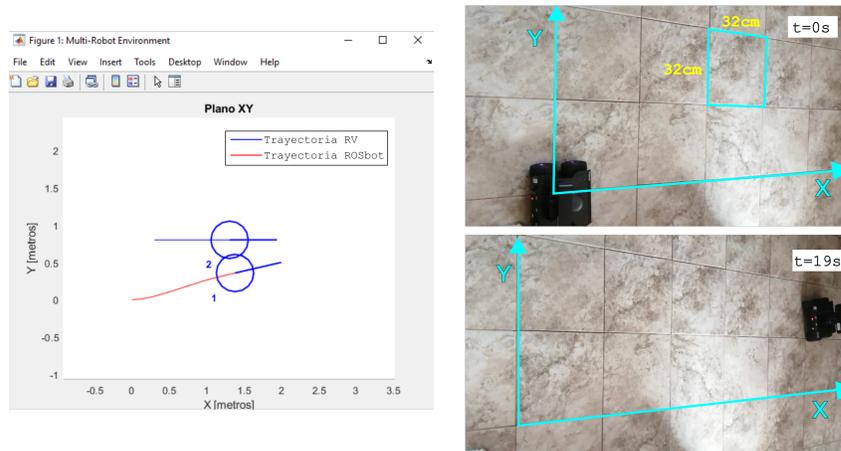


Figura 49. Resultados visualizados en Matlab, valores de posición dados por el robot.

Lectura de LiDAR Modelo A3

El ROSbot cuenta con un LiDAR A3 [10]. Para encender el LiDAR se ejecuta el nodo `rplidar_a3.launch`. Este nodo habilita el tópic `/scan`, el cuál comparte el mensaje `sensor_msgs/LaserScan`. Con base en la definición de este mensaje [11], se puede configurar un nodo en Simulink para adquirir los parámetros de interés. Podemos programar un nodo de solo lectura que se suscriba al tópic `/scan` tal como se muestra en la Figura 50.

Los datos referentes a ángulos se encuentran expresados en radianes, mientras que los datos de rangos se encuentran en metros. Hasta este momento se ha experimentado publicar diferentes valores para los parámetros: `AngleIncrement`, `AngleMin` y `AngleMax`, sin embargo, estos valores vuelven a ajustarse cada que se activa el LiDAR por lo que parecieran no ser editables utilizando el nodo `rplidar_a3.launch`. Por otra parte, la configuración actual envía 1440 mediciones (parámetro `Ranges`), mientras que el bloque de suscripción de Simulink lo trunca a 128 lecturas.

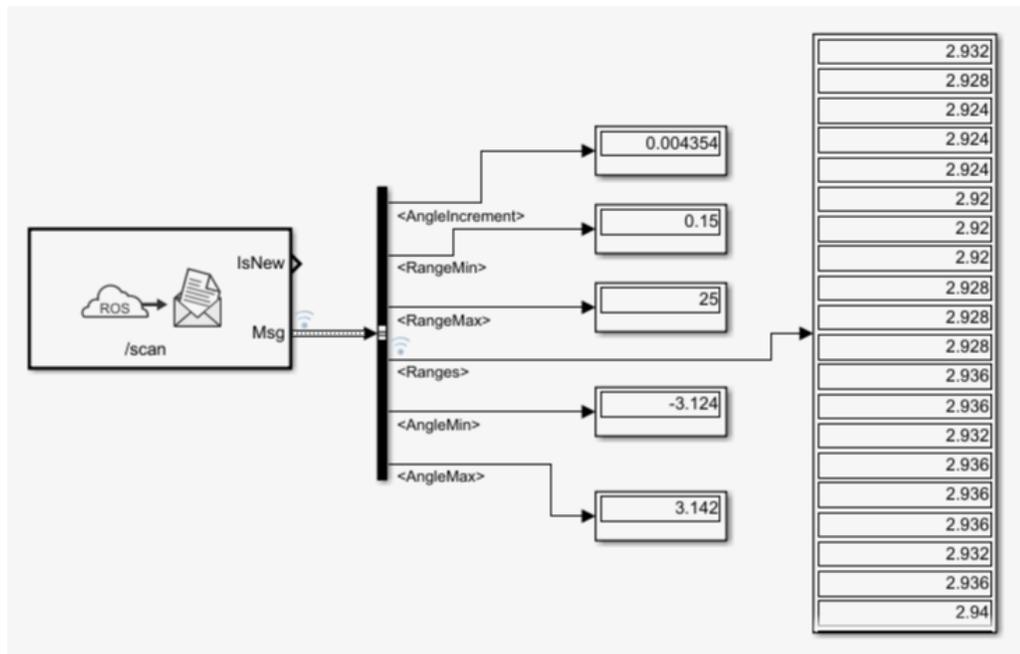


Figura 50. Lectura de LiDAR A3 en Simulink.

Referencias

- [1] <https://up-board.org/wp-content/uploads/datasheets/UPDatasheetV8.5.pdf>
- [2] <https://husarion.com/downloads/>
- [3] <https://husarion.com/manuals/rosbot/#connect-rosbot-to-your-wi-fi-network>
- [4] <https://husarnet.com/docs/manual-client>
- [5] <https://husarnet.com/docs/begin-linux>
- [6] <https://husarion.com/downloads/>
- [7] <https://husarion.com/manuals/rosbot/#rosros2-api>
- [8] Controlador Enrique
- [9] Controlador Canudas
- [10] <https://www.slamtec.com/en/Lidar/A3>
- [11] http://docs.ros.org/en/api/sensor_msgs/html/msg/LaserScan.html

NASA TLX

El método NASA TLX consta de seis dimensiones o indicadores.

1. Demanda Mental (M): Mide la actividad mental y perceptiva, recordar, calcular, encontrar. También evalúa si la tarea resulta difícil, fácil o compleja. Los valores posibles van de baja a alta.
2. Demanda Física (F): Mide la cantidad de esfuerzo físico necesario como jalar, empujar, etc. Los valores van de bajo a alto.
3. Demanda Temporal (T): Mide la presión del tiempo que sienten durante el trabajo y si este puede realizarse lento o con prisa.
4. Rendimiento / Ejecución (R): Es lo bien que ejecutaron la tarea y como se sienten satisfechos con los resultados de su trabajo. Los valores van de perfecto a fracaso.
5. Esfuerzo (E): Es la cantidad de trabajo duro que los participantes requieren para lograr el nivel requerido de rendimiento. Los valores van de bajo hasta alto.
6. Frustración (NF): Es como los participantes se sienten inseguros, desesperados o molestos cuando hacen el trabajo. Los valores van de bajo a alto.

Ahora que conoce las dimensiones, marque con un círculo el aspecto del par de cada cuadro que contribuye más a la carga de la tarea.

M - F	F - T	T - E
M - T	F - R	T - NF
M - R	F - E	R - E
M - E	F - NF	R - NF
M - NF	T - R	E - NF

En las siguientes preguntas, circule el valor de la escala donde considera su respuesta.

(M) ¿Qué tan demandante mentalmente fue la tarea?

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----

(F) ¿Qué tan demandante físicamente fue la tarea?

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----

(T) ¿Qué tan apurado estaba para realizar la tarea?

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----

(R) ¿Qué tan exitoso fue para lograr lo que se pidió?

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----

(E) ¿Qué tan duro trabajó para lograr un buen rendimiento?

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----

(NF) ¿Qué tan inseguro, desesperado o molesto se sintió en esta tarea?

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----

Figura 51. Cuestionario NASA TLX.

ID:

Cuestionario 1: Características de un guía

Cuando alguna persona lo ayuda a desplazarse

1.- ¿Qué acciones puede hacer su guía que le agraden y le ayuden?

2.- ¿Qué acciones puede hacer su guía que no le agraden y dificulte la guía?

Siguiendo con el caso en que una persona lo guíe.

3.- ¿En que situación cree conveniente acelerar el paso?

4.- ¿En qué situación cree conveniente caminar despacio?

5.- ¿Qué puede hacer una persona para que usted ya no quiera su ayuda?

6.- ¿Qué puede hacer una persona para que usted la prefiera como guía?

Figura 52. Cuestionario 1.

ID:

Cuestionario 2: Desempeño del Robot

1.- Del 1 al 10 donde 10 es lo más cómodo y 1 lo más incómodo ¿Qué tan cómodo se siente utilizando el robot?

1.- Del 1 al 10 donde 10 es lo más seguro y 1 lo más inseguro ¿Qué tan seguro se siente utilizando el robot?

3.- La velocidad del robot le pareció:

- a) Lenta
- b) Adecuada
- c) Rápida

4.- Para usted ¿Cuál sería el aspecto o aspectos más importante(s) para mejorar del robot?

5.- ¿Qué tipos de experimentos o pruebas agregaría?

Figura 53. Cuestionario 2.