

Desarrollo y Teleoperación de un Robot Móvil para la navegación y asistencia en espacios cerrados



Colaboración

José Miguel Hernández Paredes; Jesús Arturo Escobedo Cabello Jesús Arturo; Benigno Muñoz Barrón Benigno; Luis Felipe Serna Hernández, Instituto Tecnológico Superior de Huichapan

RESUMEN: En el presente artículo se muestra el diseño y desarrollo de un robot móvil terrestre teleoperado, el robot tiene una arquitectura diferencial y utiliza un lidar para la percepción de obstáculos y la generación del mapa de su entorno. El prototipo puede realizar retroalimentación visual a través del sensor Hokuyo LIDAR para su teleoperación. El dispositivo de control del robot consiste en una Raspberry PI la cual ejecuta ROS (Robot Operating System) para los algoritmos de navegación y planeación, además ejecuta las leyes de control, lectura de sensores y maneja la interfaz con los usuarios. La principal contribución es el sistema de navegación en interiores a través de la estimación de la posición y orientación del robot. Las pruebas se llevaron a cabo en el departamento de Mecatrónica del Instituto Tecnológico Superior de Huichapan para mostrar la funcionalidad del sistema.

PALABRAS CLAVE: Mapeo, SLAM, Navegación semi-autónoma, Robot, ROS.

ABSTRACT: This article shows the design and development of a teleoperated mobile robot. This is a differential drive mobile robot that uses a lidar for the obstacle perception and mapping of its environment. The prototype can perform visual feedback through a Hokuyo LIDAR sensor for teleoperation. The robot control device consists of a Raspberry PI which executes ROS (Robot Operating System) for the navigation and planning algorithms, also executes the control laws, sensors' reading and manages the users interface. The main contribution is the indoor navigation system through the estimation of the robot position and orientation. Tests were carried out in the Mechatronic's department of the Technological Institute of Huichapan to show the functionality of the system.

KEYWORDS: Mapping, SLAM, semi-autonomous navigation, Robot, ROS.

INTRODUCCIÓN

La investigación en robótica móvil actualmente se enfoca tanto en el desarrollo de robots autónomos como teleoperados, y en ambos casos se utilizan una gran cantidad de sensores, donde las señales de retroalimentación y el procesamiento de datos

se llevan a la gran capacidad de las redes de super-computadores, y solamente la capacidad de movimiento se ubica en el robot.

Los robots móviles se pueden construir con base en diferentes diseños de plataformas, que se diferencian por los diversos sistemas de tracción que utilizan. Las plataformas más comunes utilizan el sistema de tracción diferencial, las cuales ocupan motores independientes para cada una de las ruedas que se sitúan sobre el mismo eje, además utilizan ruedas locas (de castor) o puntos de sostenimiento para proporcionar estabilidad mecánica a la plataforma [1].

Una forma de estimar la posición y orientación de un robot móvil es a través de la técnica de odometría, este método es el que se utiliza con mayor frecuencia en el control de movimiento en este tipo de sistemas robóticos, como principal ventaja es que solo se utilizan sensores de cuadratura digitales, lo que permite tener la posición y orientación con bajo recurso computacional, sin embargo, la desventaja principal de este método es que si por algún motivo una o más ruedas pierden contacto con el suelo el cálculo odométrico pierde precisión y exactitud, lo que se traduce en un error acumulativo en la posición y orientación de robot [2-4], es por ello que a estos sistemas se les agrega un sensor LIDAR que genera un plano con puntos equivalentes cada uno de ellos a las distancias de objetos de tal manera que el robot puede conocer su pose actual con referencia a un marco fijo [5-7].

Con el fin de poder realizar estudios teóricos sobre un robot móvil, se requiere disponer de un modelo matemático que describa el comportamiento real del sistema. Esto permite observar, a través de simulaciones, los efectos de diferentes eventos sobre el robot, y así planear estrategias de control para afrontar una tarea cualquiera [8].

Respecto a la tarea de teleoperación y autonomía, desde hace más de una década, los investigadores usan redes inalámbricas como un medio de transmisión de comandos que permiten manipular robots y obtener señales de retroalimentación. Aunque el WiFi tiene muchas ventajas en una gran variedad de campos, usándolo para teleoperar robots, también tiene algunas limitaciones, tales como el problema del retardo de tiempo incierto, la pérdida de datos y la seguridad en la transmisión [9]. Un estudio de los retardos presentes en las comunicaciones de robots teleoperados desde WiFi, se describe en [10], donde se propone un modelo del retardo de tiempo de Internet, el cual se verifica experimentalmente a través de datos reales. Pese a las dificultades, los robots teleoperados inalámbricamente, es un campo de investigación que crece en su desarrollo debido a la aparición de nuevos avances tecnológicos que facilitan la implementación de estos sistemas, como los disposi-

tivos transceptores, la masificación de cobertura o de acceso a Internet, el ancho de banda disponible a los usuarios actuales, el mejoramiento y el abaratamiento del hardware computacional, etc. Todas estas características aumentan las prestaciones y aplicabilidad de este tipo de sistemas [11].

En el caso particular de los robots móviles, se cuentan varias aplicaciones, tales como robots conectados en línea bajo condiciones estándar para que los usuarios los utilicen como plataforma para la prueba de algoritmos telemanufactura de piezas, la exploración de sitios distantes o de difícil acceso, por ejemplo, la NASA ha usado el Internet para el control de vehículos en el espacio, desde su centro local de operaciones y en colaboración con científicos distribuidos geográficamente en diferentes lugares del mundo [12] y [13]. También se aplica en robots destinados a tareas de seguridad y vigilancia, que son monitoreados desde Internet. Otra de tareas que se puede asignar a un robot móvil es servir como asistente en tareas repetitivas tales como el iRobot ROOMBA que aspira polvo de forma autónoma. En el presente artículo se muestra el modelo matemático que define a un robot móvil diferencial, el control de velocidad de las ruedas, el telecontrol, desarrollo de mapas en tiempo real utilizando ROS [14].

MATERIAL Y MÉTODOS

Este trabajo se dividirá en 5 partes principales: diseño mecatrónico de la plataforma experimental, control, navegación, resultados y conclusiones.

En la parte de diseño mecatrónico se presentará el diseño, manufactura y ensamble de la estructura física del robot móvil, posteriormente se presentará el diseño y conexión de dispositivos electrónicos, así como la parte de instrumentación de sensores y actuadores del robot.

En la parte de control se explicará la implementación de los controladores PID de velocidad para cada elemento de tracción del robot, así como la obtención de las ganancias a través de la metodología de ZN (Ziegler y Nichols). En la sección de navegación se mostrará el diseño e implementación del sistema de telecontrol en ROS, explicando los tópicos y mensajes para la transferencia de datos a través de dicha arquitectura.

Finalmente se abordará la interconexión de las áreas mecánica, electrónica, control e informática para realizar la teleoperación, se presentarán los resultados de las pruebas de operación y de funcionalidad y finalmente se darán las conclusiones.

Diseño mecatrónico de la plataforma experimental
La plataforma experimental se trata de un robot móvil diferencial que será utilizado como prototipo para

apoyar en la entrega de documentación dentro de un espacio con múltiples oficinas.

Esta plataforma cuenta con dos motores de corriente directa de alto torque con sensores de cuadratura incrementales, una unidad de control digital a bajo nivel Arduino Due y una unidad Raspberry Pi 3 con ROS y batería de respaldo, dos amplificadores de potencia para estimular la tracción diferencial, batería de 12 VDC de ácido-plomo, llantas de goma de 130 mm de diámetro, un sensor Kinect de Microsoft y un sensor láser para el mapeo del espacio de trabajo.

El proceso de diseño empieza con el desarrollo de un concepto original del prototipo, que además de ser agradable, que satisfaga las necesidades para las cuales se diseña, y se mejoran los prototipos actuales que se encuentran en la comunidad científica.

Para el presente trabajo se utilizó el Software de diseño SolidWorks [15] y [16], en el cual se dibujaron los componentes de robot, con el fin de conocer la estructura del mismo, las dimensiones necesarias, así como los cambios necesarios por obtener un prototipo funcional, la Figura. 1, muestra el diseño final ya armado.

Se eligió un diseño de tracción diferencial, debido a que es un sistema ampliamente utilizado y adecuado para realizar tareas de navegación autónomas o semiautónomas dada la sencillez del modelo.

La distribución de cada elemento, se realizó buscando mantener el centro de gravedad a la mitad de la distancia entre las ruedas de tracción como lo establece el modelo matemático de un sistema de este tipo. El robot fue diseñado de 3 niveles, uno para cada área de control, en la planta inferior se encuentran los elementos de potencia y el control de los drivers quien manipulará los motores los cuales cuentan con una reducción de 100:1; además cada motor cuenta con un sensor de cuadratura digital el cual genera un total de 3584 pulsos.

La implementación de los controladores propios de cada motor son programados en Arduino Due [17], la cual fue elegida por las capacidades computacionales de esta, dentro la cual sobresale la velocidad de reloj de 84 MHz con un CPU de 32 bits ARM, en dicha placa de desarrollo se programan los controladores de velocidad PID para cada rueda, esto porque el manejo y control de dirección se hace mediante la diferencia de velocidades de los elementos de tracción, para mover las ruedas se utilizan los drivers VN-H2SP30 [18].

Las ruedas son un elemento que debe analizarse ya que estas deben tener tracción necesaria para evitar que el prototipo pierda contacto con el suelo y el

cálculo de posición y orientación sea lo más preciso posible, atendiendo a esta problemática fueron seleccionados neumáticos de las siguientes características: llantas de caucho, de 130 mm de diámetro y 60 mm de ancho, haciendo énfasis en el ancho del neumático.

En el mismo primer nivel se sitúa la batería de ácido-plomo de 12 VDC, además se busca que los centros de masa de la batería y del robot empaten.

El segundo nivel del robot, es un espacio designado a la colocación del sensor laser URG-04LX-UG01 [19]. Este sensor es el encargado de enviar los datos pertinentes a la tarjeta Raspberry pi 3 [20], la cual tendrá una comunicación a través de una conexión Wi-Fi hasta otra computadora conectada a la misma red, la tarjeta mencionada establecerá la comunicación con el usuario quien ejecutará los comandos de dirección los cuales se envían a la tarjeta Arduino, siendo esta quien calcule la potencia en los controladores PID para manipular la dirección y orientación del robot. El sensor URG-04LX-UG01, es quien mapea el espacio de trabajo, esto con el fin de realizar el croquis que describe las posibles trayectorias.

En el tercer nivel se incorpora un segundo sistema de visión, para lo cual se incorporó un Kinect, el cual puede ser observado en la parte superior del diseño en la Figura. 1, dicho diseño fue tomado de [21].

Una vez establecido el diseño y el material a utilizar se exportó el archivo en formato CorelDraw para cortar las piezas de acrílico usando un cortador CNC laser. Las piezas fueron manufacturadas en acrílico transparente de 6 mm de espesor.

Cabe denotar que el robot diferencial mantiene las ruedas de tracción en la parte frontal, esto con la finalidad de que los elementos de visión mantengan su centro de gravedad de manera colineal con el del robot [3]. En la Figura 1 se muestra el robot ensamblado con todos los componentes que lo integran.



Figura 1 Propuesta de prototipo terminada

Control de un robot móvil diferencial

En un robot móvil de ruedas del tipo diferencial la velocidad y dirección del móvil dependen de manera directa de la velocidad actual de cada una de las ruedas que se definen como ω_r y ω_l , si estas velocidades se igualan de tal manera que si $\omega_r = \omega_l$ el robot avanzará en línea recta [8] y [22], el desplazamiento total del robot se define como:

$$D = \frac{D_r + D_l}{2} \quad \text{Ec. (1)}$$

Donde: D_r es desplazamiento de la rueda derecha y D_l es el desplazamiento de la rueda izquierda.

La Figura 2 los segmentos denotados como D_r son parte de una circunferencia de radio $d+b$ y el perímetro de esta circunferencia está dada por la ecuación (2).

$$C_l = 2\pi(d + b) \quad \text{Ec. (2)}$$

Donde: C_l es el perímetro de la circunferencia por la rueda izquierda, d es la distancia entre las ruedas, b define el giro con respecto al eje del robot y θ es el ángulo de giro que define la orientación de móvil.

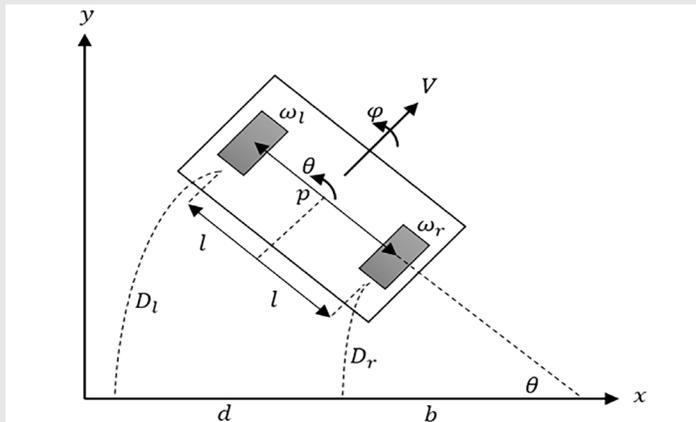


Figura 2 Vista superior del avance geométrico del robot

La ecuación (3) relaciona la distancia que recorre la rueda izquierda con el perímetro y el ángulo de giro con toda la circunferencia.

$$\frac{D_l}{C_l} = \frac{\theta}{2\pi} \quad \text{Ec. (3)}$$

Despejando θ de (3) y sustituyendo (2):

$$\theta = \frac{D_l}{d + b} \quad \text{Ec. (4)}$$

La distancia D_r representa una porción de la circunferencia de radio b y $C_r = 2\pi b$ define el perímetro de la rueda derecha y θ el ángulo de giro con toda la circunferencia:

$$\frac{D_r}{C_r} = \frac{\theta}{2\pi} \quad \text{Ec. (5)}$$

Si se despeja b y se sustituye en (5) se obtiene el radio de giro con respecto al eje del robot, ecuación (6).

$$b = \frac{D_r}{\theta} \quad \text{Ec. (6)}$$

Sustituyendo b en (4) se puede definir el ángulo de giro del robot como:

$$\theta = \frac{D_l}{d + \frac{D_r}{\theta}} = \frac{D_l - D_r}{d} \quad \text{Ec. (7)}$$

La ecuación (7) define la orientación de robot en función de los desplazamientos de las ruedas izquierda y derecha, además, es independiente al camino que el robot tome.

En ocasiones el radio r_n de la rueda no es el real cuando la rueda toca el piso, es decir, el radio puede disminuir debido al peso que se ejerce en el robot, este efecto se muestra en la Figura 3.

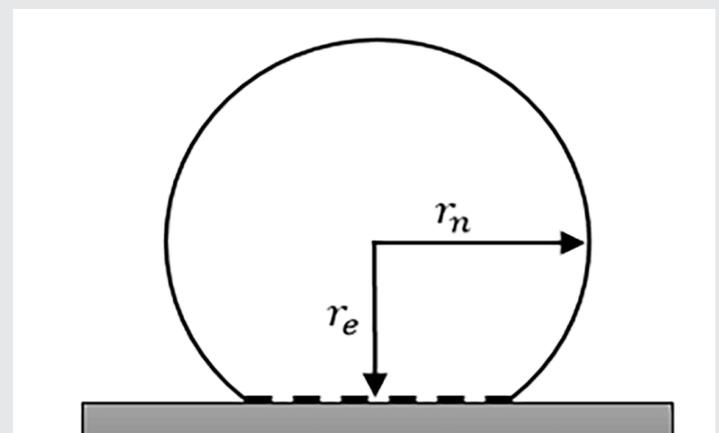


Figura 3 Compensación del radio por efectos del peso sobre la rueda

Entonces el radio efectivo r_e está en función de la compliancia de las ruedas y se puede determinar empíricamente.

Entonces la distancia que recorre la rueda derecha e izquierda al tomar en cuenta el efecto de compliancia se define en (8) y (9).

$$D_r = \frac{2\pi\theta_r}{360^\circ} \quad \text{Ec. (8)}$$

$$D_l = \frac{2\pi\theta_l}{360^\circ} \quad \text{Ec. (9)}$$

Donde: θ_r y θ_l corresponden al ángulo de giro de las ruedas derecha e izquierda respectivamente en grados.

Para calcular la posición en un plano de referencia (x,y) se utilizará la Figura 4.

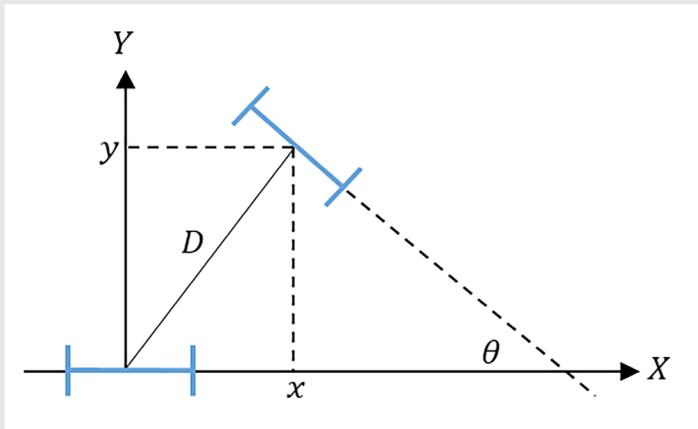


Figura 4 Estimación de la posición (x,y)

La obtención de las coordenadas en el eje x y el eje y es trivial ya que se trata de un simple triángulo rectángulo.

$$x = D * \sin(\theta) \quad \text{Ec. (10)}$$

$$y = D * \cos(\theta) \quad \text{Ec. (11)}$$

Donde: D es el desplazamiento total del robot que se define en la ecuación (1).

Si el radio de la rueda es r_e , las velocidades lineales [m/s] correspondientes son $v_l = \omega_l r_e$ y $v_r = \omega_r r_e$. En este caso la velocidad lineal y angular en el modelo se definen como [3-4] y [8-9]:

$$v = \frac{v_r + v_l}{2} = \frac{(\omega_r + \omega_l)r_e}{2} \quad \text{Ec. (12)}$$

$$\omega = \frac{v_r - v_l}{d} = \frac{(\omega_r - \omega_l)r_e}{d} \quad \text{Ec. (13)}$$

Por otro lado si se especifica la velocidad lineal v en [m/s] y angular ω del robot en [rad/s], las velocidades de giro que hay que aplicar a las ruedas izquierda y derecha son [3] y [23]:

$$\omega_l = \frac{v - \left(\frac{b}{2}\right)\omega}{r_e} \quad \text{Ec. (14)}$$

$$\omega_r = \frac{v + \left(\frac{b}{2}\right)\omega}{r_e} \quad \text{Ec. (15)}$$

La velocidad lineal y angular puede definirse en variables de control como [6]:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{(r_e \sin(\theta))}{2} & -\frac{(r_e \sin(\theta))}{2} \\ \frac{(r_e \cos(\theta))}{2} & \frac{(r_e \cos(\theta))}{2} \\ -\frac{r_e}{b} & \frac{r_e}{b} \end{bmatrix} \begin{bmatrix} \omega_l \\ \omega_r \end{bmatrix} \quad \text{Ec. (16)}$$

Cálculo de la velocidad ω_l y ω_r

El cálculo de estas dos velocidades angulares de las ruedas izquierda y derecha respectivamente es de

suma importancia puesto que si se conocen es posible controlarlas para darle ese movimiento deseado al robot. La velocidad angular de cualquiera de las ruedas se determina a través de la observación de un sensor del tipo digital que esta acoplado al eje de cada motor que permite el movimiento de la rueda, a este sensor se le conoce como encoder de cuadratura incremental y con la ecuación (17) [2] se puede determinar la velocidad de cada rueda.

$$w_{l,r} = 2\pi \frac{P_{l,r}}{P_t} \Delta t \left[\frac{\text{rad}}{\text{s}} \right] \quad \text{Ec. (17)}$$

Donde: $w_{l,r}$ es la velocidad angular en [rad/s], $P_{l,r}$ es el número de pulsos resultantes de la velocidad en la rueda izquierda y derecha respectivamente, P_t el número total de pulsos por vuelta de encoder y Δt el periodo de muestreo en [Hz].

Control PID de velocidad angular

Para la tarea de controlar la velocidad de cada una de las ruedas se implementó una técnica de control PID, entonces con la finalidad de intentar llevar a cero el error de velocidad surge la justificación de aplicar un controlador PID en el control de robots [10] y [11]. La ley de control PID se puede expresar como:

$$\tau = K_p \tilde{\omega} + K_v \dot{\tilde{\omega}} + K_i \int_0^t \tilde{\omega}(t) dt \quad \text{Ec. (18)}$$

Donde $\tilde{\omega}$ y $\dot{\tilde{\omega}}$ representan el error entre la velocidad angular deseada y la real medida ω , las constantes K_p , K_v , $K_i \in \mathbb{R}$, llamadas respectivamente ganancias proporcional, derivativa e integral, son variables definidas positivas [10].

La sintonización de las ganancias K_p , K_v , y K_i se realiza con el método de Ziegler y Nichols, estas reglas funcionan con sistemas que al aplicar un escalón unitario la respuesta es en forma de S, en la Figura 5 se muestra la respuesta típica de este tipo de sistemas.

Con los puntos L y T es posible determinar las ganancias del controlador PID con la Tabla 1 [12].

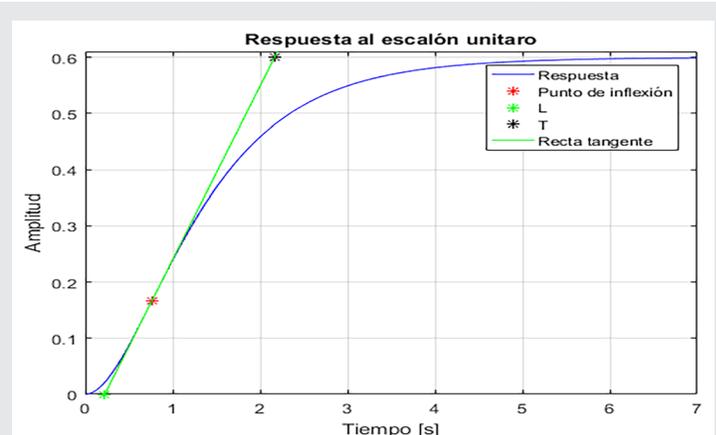


Figura 5 Respuesta de un sistema ante una entrada de escalón unitario en la que se observan las constantes de Ziegler y Nichols

Tabla 1 Criterios de Ziegler-Nichols

Tipo	Ki	Ti	Td
P	T/L	∞	0
PI	$0.9*(T/L)$	$L/0.3$	0
PID	$1.2*(T/L)$	$2*L$	$0.5*L$

Navegación y generación de mapas

ROS [14] proporciona una colección de herramientas, bibliotecas y convenciones que tienen como objetivo simplificar la tarea de crear un comportamiento robótico complejo y robusto en una amplia variedad de plataformas robóticas. Entre otras características importantes, ROS fomenta la abstracción de hardware, controladores de dispositivos, librerías, herramientas de visualización, comunicación por mensajes y administración de paquetes.

Usando como base el framework de ROS se procede a implementar y utilizar diferentes subsistemas. En combinación de algoritmos escritos en C++ para Arduino, en Python para ROS y el robot mecánico se implementó el sistema de navegación para el prototipo propuesto (ver Figura 6).

En primer lugar, se utiliza un nodo o (programa de ROS) que escucha al teclado de la computadora remota y dependiendo de las teclas que presione el operador enviará diferentes comandos de velocidad a la computadora on-board del robot a través del protocolo SSH.

Usando SSH, la máquina cliente inicia una conexión con la máquina del servidor mediante una sesión cifrada, evitando que alguien pueda obtener una contraseña o cualquier otro tipo de información que se envíe por la red [24]. ROS utiliza el protocolo SSH para realizar redes de computadoras para compartir información propia del sistema operativo, lo que permite extender su alcance.

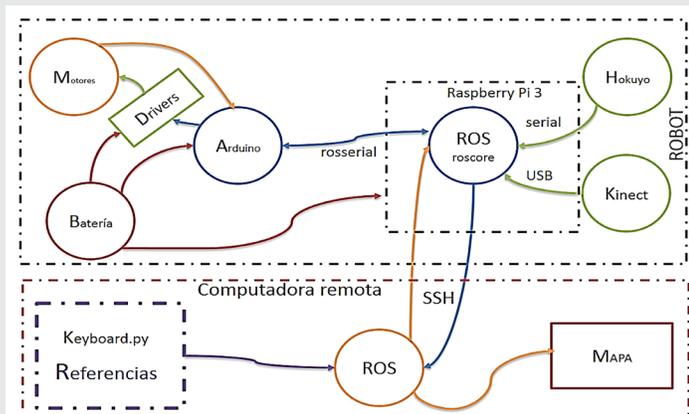


Figura 6 Esquema general de navegación y mapeo con el prototipo propuesto

Los pasos a seguir para ejecutar el sistema de teleoperación son los siguientes:

En el robot:

1. Lanzar el ROS maestro (roscore)
2. Lanzar el nodo roserial (comunicar con el firmware de Arduino) en Raspberry pi 3
3. Lanzar el nodo urg_node
4. Lanzar el nodo hector_slam

En la computadora remota:

1. Lanzar un driver traductor para compatibilidad entre mensajes del robot y los mensajes estándar de ROS.
2. Lanzar el visualizador de ROS rviz
3. Lanzar el programa de telecontrol keyboard.py.

RESULTADOS

Para validar los controladores PID de velocidad de las ruedas como el cálculo de odometría se realizaron dos pruebas que el robot debe realizar: 1) trayectoria lineal y 2) trayectoria circular.

En la Figura 7 se muestra la trayectoria del robot como resultado de haber seguido una referencia trayectoria en línea recta.

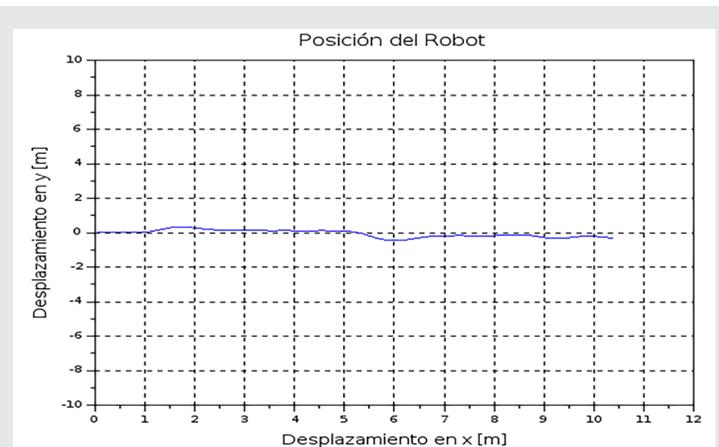


Figura 7 Prueba 1, trayectoria del robot en el plano

Para la prueba 2 (Figura 8 y 9), se siguió una trayectoria circular con un diámetro de 95 cm.

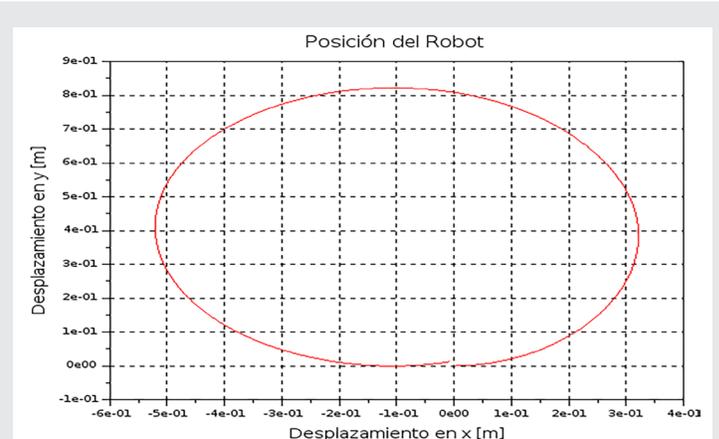


Figura 8 Prueba 2) trayectoria del robot en el plano

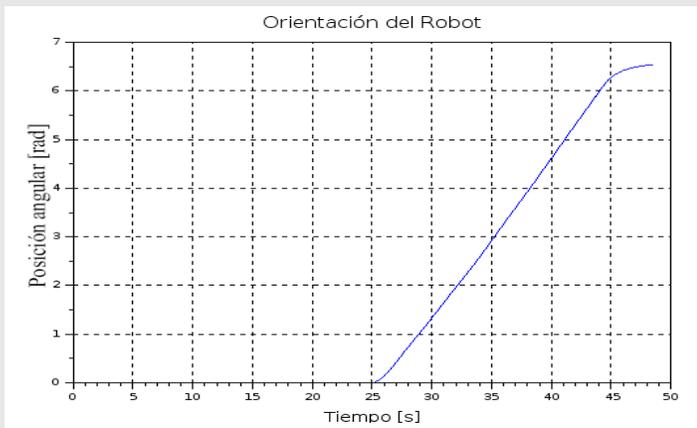


Figura 9 Prueba 2) posición angular del robot

En la Figura 10 se presenta el mapa del lugar donde se realizan las pruebas aquí mencionadas, este mapeo se realiza con el uso de la paquetería Hector SLAM escrita en ROS, que en conjunto con otras paqueterías se puede realizar navegación autónoma o semiautónoma.

Para el desarrollo del mapa se puso el robot en funcionamiento con la tele operación de tal manera que el operario lo llevará por toda la habitación para que el láser y Héctor SLAM realizarán el mapa conforme se avanzaba por todo el espacio.

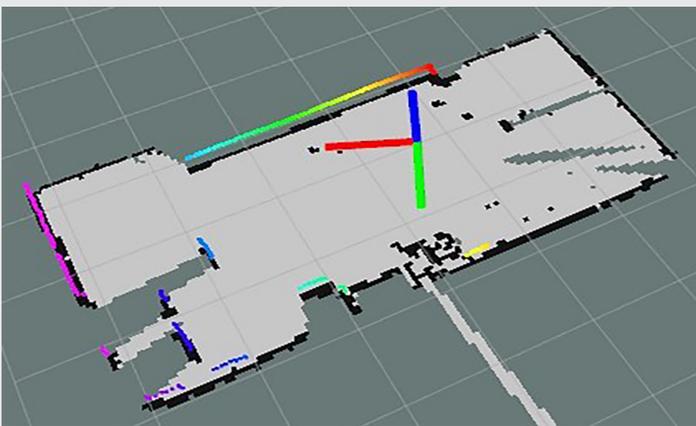


Figura 10 Mapa del espacio de trabajo del robot

CONCLUSIONES

En este trabajo se presentó el diseño e implementación de un robot móvil diferencial tele operado, basado en la arquitectura de ROS, esta configuración es muy popular sin embargo la arquitectura del robot es abierta, es decir las dimensiones y códigos de programación se pueden ajustar de acuerdo su utilización.

Uno de los principales desarrollos para este robot fue el sistema del control a bajo nivel y la tele operación en ROS, el cual consistió en un control PID, el contro-

lador fue sintonizado para reducir el error en estado estacionario de la velocidad angular de cada una de las ruedas y la implementación de protocolos de comunicación en SSH, USB y serial.

Se pudo demostrar la implementación del sistema de navegación semi-autónoma y control usando sistemas embebidos de bajo costo como Arduino y Raspberry Pi.

En las tres pruebas que se realizaron para validar la funcionalidad de robot resultaron exitosas y pueden usarse como base para realizar tareas con mayor complejidad como por ejemplo navegación autónoma con evasión de obstáculos.

Se plantea como trabajo a futuro la implementación de un sistema de navegación completamente autónomo sobre la misma plataforma para el uso en asistencia en tareas de oficina.

BIBLIOGRAFÍA

[1] Hernández-Paredes J. M., Serna-Hernández L. F., Quintanar-García S., 2017, *Control de Movimiento y Seguimiento de Trayectorias de Un Robot Móvil Diferencial a través de la Estimación Odométrica*, - REIA, VOL3, ISSN 2448- 6817.

[2] Katsuhiko Ogata, 2010, *Ingeniería de control moderna*, Pearson Educación, S.A., Madrid, 577-595.

[3] Anibal Ollero Baturone, 2005, *Robótica - Manipuladores y Robots Móviles*, Ed. Marcombo, ISBN: 9788426713131.

[4] M.A. Zamora, L.M. Tomas-Balibrea, H. Martínez, A. G. Skarmeta, 2000, *Navegación Planificada de un Robot Móvil en Entornos Interiores Desconocidos*, Universidad de Murcia.

[5] Ulrich Weiss and Peter Biber, 2011, *Plant detection and mapping for agricultural robots using a 3D LIDAR sensor*, *Robotics and Autonomous Systems* 265-273.

[6] Joydeep Biswas and Manuela Veloso, 2010, *WiFi Localization and Navigation for Autonomous Indoor Mobile Robots*, *IEEE International Conference on Robotics and Automation*, Alaska, USA.

[7] André Araújo, David Portugal, Micael S. Couceiro and Rui P. Rocha, 2013, *Integrating Arduino-based Educational Mobile Robots in ROS*, *13th International Conference on Autonomous Robot Systems*, Lisbon, Portugal

[8] Javier Gonzales Jiménez y Anibal Ollero Baturone, 2015, *Estimación de la Posición de un Robot*

Móvil, Universidad de Málaga y Universidad de Sevilla.

[9] Ken Goldberg and Roland Siegwart, 2002, *Beyond Webcams: An introduction to online robots*. Mit press, Massachusetts, ISBN: 9780262072250.

[10] Emanuel Slawiński, José Francisco Postigo y Vicente A. Mut, 2006, *Experiencias en teleoperación bilateral de robots*, Revista Iberoamericana de Automática e Informática Industrial, Vol: 3, no: 1, ISSN: 1697-7912.

[11] Nelson David Muñoz C., Néstor Alfonso Delgado Hernández y José Alejandro Díaz C., 2006, *Diseño de un escenario en línea para robots teleoperados desde Internet*, Revista Scientia et Technica, No. 32, pp. 85-90, ISSN 0122-1701.

[12] P. G. Backes; K. S. Tso; J. S. Norris; G. K. Tharp; J. T. Slostad; R. G. Bonitz; K. S. Ali, 2000, *Internet-based operations for the Mars Polar Lander misión*, IEEE International Conference on Robotics and Automation. Proceedings. ICRA, pp: 2025-2032.

[13] Eduardo Morales, L. Enrique Sucar, 2014, *Introducción a la Robótica Móvil*, Instituto Nacional de Astrofísica, Óptica y Electrónica.

[14] ROS. 2008. *The Robot Operating System*. 2018, de ROS Sitio web: <http://www.ros.org/about-ros/>.

[15] Sarope Kalpajian and Steven R. Schmid, 2002, *Manufactura, Ingeniería y Tecnología*, Pearson Educación, ISBN: 9789702601371.

[16] Dassault Systemes, 2016, *Introducción a SolidWorks, Manual SolidWorks*.

[17] Arduino Inc., 2018, *Arduino Due, Technical Specifications*, acceso online, <https://store.arduino.cc/usa/arduino-due>.

[18] STMicroelectronics, 2008, *Automotive fully integrated H-bridge motor driver, Data Sheet*, acceso online: <http://cdn.sparkfun.com/datasheets/Dev/Arduino/Shields/10832.pdf>, revisión 8.

[19] Hokuyo Automatic CO. LTD., 2009, *Scanning Laser Range Finder URG-04LX-UG01, Data Sheet*, acceso online, <https://www.robotshop.com/media/files/pdf/hokuyo-urg-04lx-ug01-specifications.pdf>.

[20] Raspberry Pi 3 Org., 2018, *Raspberry Pi 3 Model B Technical Specifications*, acceso online: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>.

[21] Bishoy, 2014, *Kinect model assembled*, acceso: <https://grabcad.com/library/kinect-1>.

[22] Daniel Pizarro, Manuel Mazo, Enrique Santiso, Marta Marron, David Jimenez, Santiago Cobreces and Cristina Losada, 2010, *Localization of Mobile Robots Using Odometry and an External Vision Sensor*, Sensors 2010, 10, 3655 - 3680; doi:10.3390/s100403655.

[23] Aguilera Hernández Martha I., Bautista Miguel A. y Iruegas Joaquín, 2007, *Diseño y Control de Robots Móviles*, Mecamex.

[24] Cardenal Gardok, Hostalia, acceso online 2018, *Qué es el protocolo SSH y cómo configurarlo para mejorar la seguridad de acceso a los servidores Linux*, Hostaliawhitepapers.