

Mini-reto: Navegación y control de un robot terrestre

H Morales-Villalobos, J Acaro-Andrade, J Padilla-Marquez, L Moreno-Castro, M Borjon-Arriola, M Robles-Garcia, N Muñoz-Durán, A Ojeda-Zazueta
Tecnológico de Monterrey Campus Guadalajara
Guadalajara, México

a01637304@tec.mx
a01636269@tec.mx
a01746712@tec.mx
a01638100@tec.mx
a01637961@tec.mx
a01636143@tec.mx
a01741080@tec.mx
a01741448@tec.mx

Resumen- El presente documento tiene como finalidad exponer el estudio y resultados obtenidos en la actividad de navegación y control de un robot terrestre. Dicha actividad trató sobre implementar los conocimientos adquiridos en esta unidad de formación sobre los robot móviles y la aplicación de las distintas tool box que contiene el software de Matlab para el generador de mapeo y de este modo crear un trayectoria que lleve al robot móvil de un punto A a un punto B.

Abstract - The purpose of this document is to expose the study and the results obtained in the navigation and control activity of a terrestrial robot. This activity dealt with implementing the knowledge acquired in this training unit on mobile robots and the application of the different tool boxes that the Matlab software contains for the mapping generator and thus create a trajectory that will take the mobile robot from one point to another. point A to point B.

Palabras Clave: Robot móvil, Mapeo, Trayectoria, Algoritmo, Planificador, Robot diferencial, Controlador.

INTRODUCCIÓN

Actualmente la robótica móvil juega un papel muy importante en la sociedad, ya que cada día es posible apreciar un incremento de robots móviles

en nuestro entorno, siendo estos una constante en el día a día de millones de personas alrededor del mundo. El incremento de robots móviles en nuestra sociedad ha traído consigo grandes retos, como la generación de trayectorias eficientes, para de esta forma optimizar el desempeño en la operación de estos sistemas robóticos.

DESCRIPCIÓN DEL PROBLEMA

En esta actividad se procederá a configurar un algoritmo que con ayuda de este se pueda programar a un robot de tipo diferencial, que siga una trayectoria propuesta desde un punto inicial a un punto final al haber generado un mapa (Mapa diseñado como el salón de clases) mediante el uso de un planificador de tipo PRM.

DESARROLLO

La primer etapa consistió en obtener las ecuaciones que describen el modelo del robot diferencial, dichas ecuaciones pueden ser visualizadas a continuación:

$$V = \frac{V_R + V_L}{2} \quad (1)$$

$$\omega = \frac{V_R - V_L}{2l} \quad (2)$$

$$V = K_p \sqrt{(x - x_d)^2 + (y - y_d)^2} \quad (3)$$

$$\omega = kp(\theta - \theta_d) \quad (4)$$

Posteriormente, se realizaron las mediciones del entorno a simular, con el objetivo de obtener un entorno de simulación lo más apegado posible al espacio de trabajo del robot móvil.

A partir de las mediciones realizadas, se utilizó la función makemap para crear una matriz binaria que representa el espacio libre y los obstáculos presentes en el laboratorio, por medio de la cual fue posible obtener un mapa de ocupación binaria, en el cual se ajustó la resolución para obtener una escala de metros.

Posteriormente, se utilizó el método PRM para encontrar rutas factibles para recorrer el espacio creado. Esta técnica consiste en la creación de una cantidad específica de puntos aleatorios en el mapa, entre los cuales se generan conexiones, después de haber eliminado cualquier punto que fuera colocado dentro de un obstáculo, y siempre y cuando los puntos conectados se encuentren dentro de cierto radio entre sí y la conexión no atravesase ningún obstáculo.

Por último, para definir el camino a seguir, se debe seleccionar punto inicial y final, a partir de los cuales, se aplica el algoritmo A*, el cual busca una ruta factible para realizar el recorrido entre los puntos por medio de las conexiones generadas anteriormente, considerando en qué medida es posible acercarse a la meta por medio de cada conexión y las adyacentes.

Como salida de la ejecución de este algoritmo, se obtiene un conjunto de puntos coordinados, que demarcan el camino a seguir dentro del espacio definido para llegar desde el punto de inicio hasta el punto de salida establecidos.

Para la implementación del seguimiento del camino en el robot diferencial, se desarrolló un código que determinara la velocidad lineal y angular deseada del centro geométrico del robot.

Para la velocidad lineal, se decidió establecer un valor constante, ya que resulta inconveniente definirla en función de la distancia a la meta, mientras que la velocidad angular se definió como una constante k_a multiplicada por un ángulo alfa, el cual representa la diferencia entre la orientación del robot y el ángulo del vector de distancia a la meta.

$$\omega = k_a \alpha \quad \alpha = \theta_d - \theta$$

$$\theta_d = \text{atan2}\left(\frac{y}{x}\right)$$

Posteriormente, estos valores de velocidad se introducen al modelo cinemático descrito anteriormente, para obtener la velocidad lineal correspondiente a cada rueda, de forma que

$$v_r = v + d\omega \quad v_l = v - d\omega$$

Donde d representa la distancia entre cada llanta y el eje central del robot. En este caso $d = 0.1m$

Para poder realizar el control de la velocidad de cada motor, es necesario primeramente realizar una conversión de la velocidad identificada a un valor de 8 bits que pueda ser enviado por bluetooth y convertido en una señal de PWM por la tarjeta arduino.

Para esto, se llevaron a cabo distintas pruebas de velocidad, a partir de lo cual fue posible determinar una relación entre la velocidad lineal del robot y los valores de PWM utilizados.

Con base en esto, se estableció la velocidad lineal constante de la simulación como $V = 0.78 \frac{m}{s}$, correspondiente a la mitad de la velocidad máxima identificada en el robot, y se sintonizó la velocidad angular con una constante $k_a = 5$, de tal forma que un ángulo $\alpha = \frac{\pi}{2}$ resultaría en velocidades $v_r = 1.56 \frac{m}{s}$, $v_l = 0 \frac{m}{s}$

Por lo tanto, se utilizaron los siguientes valores de conversión para la obtención de los valores necesarios de PWM

$$PWM_r = 196.92v_r \quad PWM_l = 164.1v_l$$

La diferencia entre los factores se debe a que el motor derecho se debe colocar en reversa para que gire en la misma dirección que el motor izquierdo con respecto al frente del carrito. No obstante, las velocidades en reversa de los motores utilizados son menores ante la misma señal PWM en una relación identificada como aproximadamente $\frac{5}{6}$, por lo que, se compensa correspondientemente.

Antes del envío de las señales, se realizó una normalización de los valores calculados de PWM, asegurando que estos no sobrepasaran el valor máximo de 8 bits. Igualmente, se identificó que valores menores a 40 de PWM no generaban

movimiento en los motores, por lo tanto, se estableció este como valor mínimo y, por último, en caso de obtener valores negativos, se tomó únicamente la magnitud para el valor de PWM y se invirtió la dirección del motor correspondiente.

Para terminar, se utilizó la función write para el envío de la información al módulo bluetooth, el cual se definió como un objeto antes de la ejecución del código para el establecimiento de la conexión. La información se mandó en forma de un vector, como se muestra a continuación.

```
write(bt, [motor, dirección, velocidad])
```

Donde motor tiene valores de 0 y 1, correspondiente al motor derecho e izquierdo respectivamente, dirección tiene valores 1 para avance y 0 para reversa y velocidad es el valor calculado de PWM.

Construcción del robot a utilizar

Un punto importante a destacar es que nuestro robot no cuenta con un sistema reactivo para su movilidad, es decir, no cuenta con ningún tipo de sensor que le ayude a recolectar datos del ambiente para tomar acciones a partir de esos datos. El robot se moverá con ayuda de una trayectoria previamente definida.

Dentro de los elementos que se utilizaron en la construcción del robot se encuentra una batería constituida por 4 celdas de litio de 3.7v cada una, lo que nos da una alimentación total de 14.8v cc, un regulador de voltaje a 5v cc, un probador para la batería, dos motores con sus respectivas llantas, drivers para el motor, un módulo bluetooth hc-05 y como elemento de control se utilizó un arduino uno debido a su practicidad para lograr los objetivos de la actividad. A continuación se presenta un diagrama de conexiones entre los componentes:

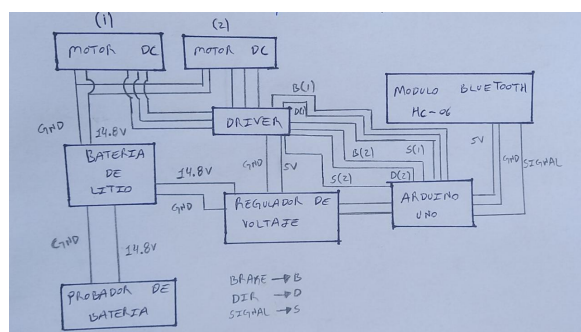


Figura 1. Diagrama de conexiones del robot

Cabe mencionar que también se utilizaron 2 botones para interrumpir la alimentación de los dos motores, además de integrar un kill switch de todo el sistema.

Después de conectar todo el robot final se mira de la siguiente manera:

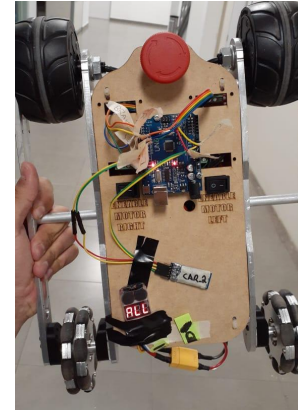


Figura 2. Solución implementada

DIAGRAMAS DE SOLUCIÓN IMPLEMENTADA

Para la solución del problema de control se propuso el algoritmo PRM (Probabilistic Roadmap) para implementarlo en nuestro robot, este algoritmo se define como una generación de puntos aleatorios sobre un mapa en forma de entrada, este algoritmo genera una serie definida de puntos sobre el mapa plano de los cuales solo se toman en cuenta los puntos que están en áreas accesibles del diseño (esto con el fin de no pasar dentro de estructuras), una vez colocados los puntos se hacen conexiones entre estos con el fin de encontrar una ruta entre el punto de partida del robot hasta el destino final. A continuación se presentan los pasos más significativos del algoritmo al implementarlo a nuestro control.

Mapa creado del laboratorio

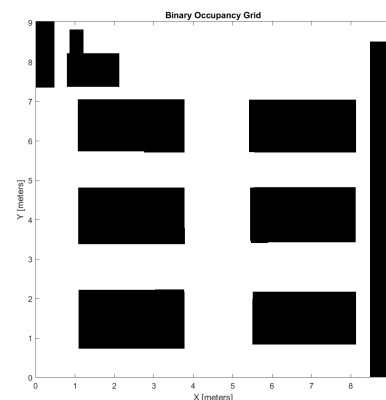


Figura 3. Definición de mapa

El punto principal a seguir es hacer la definición de nuestro mapa, siendo en nuestro caso el laboratorio de la universidad, para esto se deben indicar las coordenadas de los puntos, las estructuras, las unidades en las que se mide el mapa y las dimensiones máximas que este maneja. Una vez definidos estos valores es que se diseña el mapa dentro de una variable `map()` como se vio anteriormente.

Con el mapa generado pasamos a ejecutar el algoritmo, introduciendo nuestra variable `map()`, el número de nodos a generar, radio de conexión entre puntos, punto de partida y destino final, con esto entonces el algoritmo PRM nos genera la imagen que se observa a continuación.

Camino generado

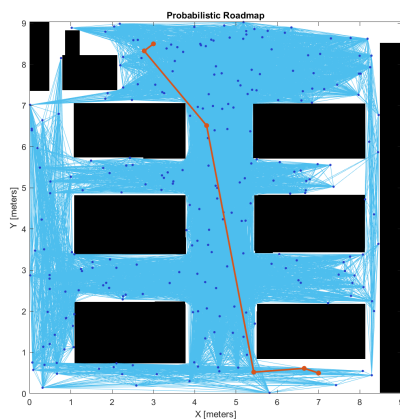


Figura 4. Creación de camino

El algoritmo nos arroja como resultado una variable llamada `path()`, la cual contiene las coordenadas de los puntos a seguir para llegar desde el punto de partida hasta el destino final. junto a esto también tenemos el resto de puntos que no se utilizarán siendo graficados y la figura donde se aprecia todo el proceso del algoritmo.

Con el algoritmo ejecutado podemos rescatar la variable `path()` que nos indica la serie de puntos del camino a seguir, teniendo el camino y el mapa lo podemos graficar en una sola imagen como se observa en la figura 5.

Simulación de seguimiento del camino

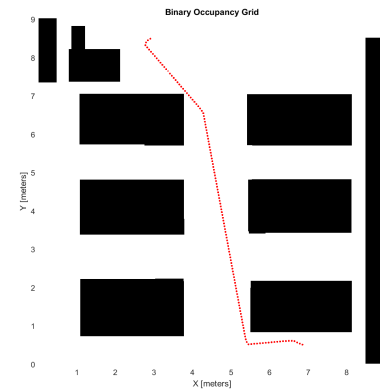


Figura 5. Simulación final del camino

Como se puede observar, el algoritmo PRM logro encontrar una ruta exitosa para el robot, sin embargo, este método puede llegar a tener fallas, la primera es que no pueda encontrar ninguna ruta posible por la forma que tiene el mapa o por la cantidad de nodos con la que fue ejecutado el algoritmo, en nuestro caso esto no fue un error significativo ya que nuestro laboratorio y por ende el mapa generado tiene formas simples y sencillas para el procesamiento del algoritmo, junto a esto también introducimos una cantidad de nodos suficientes para que el factor de probabilidad no influya tanto en nuestra contra.

Con esto podemos pasar a implementar la ruta generada con nuestro control, y puesto que la variable `path()` de la ruta nos entrega puntos exactos es que encaja perfectamente con nuestro control de seguimiento de una trayectoria del robot diferencia.

EXPLICACIÓN Y ANÁLISIS DE LOS RESULTADOS OBTENIDOS

Con la simulación del mapa de la clase, se logró que el carrito pudiera tener la trayectoria más corta para atravesar el salón de clases. Anteriormente se probó otra trayectoria para aprender el uso de los componentes y trayectorias de Matlab. Esto se puede consultar en el video 1 anexo en la siguiente sección.

En el momento en el que se realizaron algunas pruebas con el carrito fuera de la clase, este logró hacer las curvas que se necesitaban y el código funcionó en el momento de la trayectoria. No obstante, el problema más que todo fue la falta de tiempo para probar en el salón y que algunos componentes del carro dejaron de funcionar por

problemas externos. La demostración se encuentra en el video 2.

Dentro de las dificultades que se presentaron, se encuentran irregularidades en los caminos, es decir, al principio se realizó una toma de muestras de velocidad en el robot haciéndolo que recorra una distancia definida con una velocidad PWM dada, esto con el objetivo de convertir la velocidad pwm a metros sobre segundos, desafortunadamente el suelo donde se hicieron las pruebas era irregular, además de que las llantas seleccionadas para el robot no eran las adecuadas para ese tipo de suelo.

Otra dificultad que se encontró fue de manera computacional, ya que el controlador recibe datos mediante conexión bluetooth, el problema lo tuvimos cuando nos dimos cuenta que la simulación sucedía demasiado rápido y nuestro sistema de control no es lo suficientemente poderoso para responder a simulaciones tan rápidas.

Para resolver esto, primeramente se implementó una pausa específica después del envío de la información al robot, probando con distintos valores. Sin embargo, esto resultaba en que el robot se desplaza grandes distancias o generará curvas mayores a las queridas según la simulación, por lo tanto, se decidió implementar una retroalimentación por parte del controlador, de forma que este mandara una señal de vuelta a Matlab una vez que había realizado la implementación de los comandos recibidos, a fin de evitar una saturación.

A pesar de esto, sucedía un error de comunicación, en el cual se generaba un desfase en los arreglos de datos, resultando en que los valores del motor, dirección y velocidad se recibieran en una categoría incorrecta, generando errores. Desafortunadamente, debido a las restricciones de tiempo, no fue posible realizar las pruebas necesarias para solucionar este problema.

En la sección de anexos, es posible consultar los códigos de Matlab y arduino utilizados en el proyecto.

FOTOGRAFÍAS Y VIDEOS

En el primer video es el primer acercamiento hacia el uso de trayectorias

[**Prueba 1: Carro dando vuelta a un poste \(link\)**](#)

En este video se puede evidenciar el giro de la trayectoria y cómo se desenvuelve con las coordenadas dadas.

[**Prueba 2: Intento de trayectoria \(link\)**](#)

En el siguiente video, se puede observar la ejecución del código de simulación de seguimiento de trayectoria en Matlab.

[**Simulación del seguimiento de trayectoria \(link\)**](#)

CONCLUSIÓN

Héctor Morales

A mi parecer, esta actividad fue una gran manera de poner en práctica los distintos conocimientos que se adquirieron a lo largo del módulo de robótica móvil, al igual que de integrarlos con conocimientos que fueron impartidos en bloques y semestres anteriores.

Para la integración de este reto, fue necesario primero aprender el funcionamiento y modelación del robot diferencial, la cual permite representar la relación entre las distintas velocidades presentes en los elementos del robot y la manera en la que estas se traducen en una velocidad lineal y angular sobre el centro geométrico de este, lo cual puede ser controlado para generar el desplazamiento deseado para seguir un camino.

Por su parte, los caminos pueden ser generados de múltiples maneras, una de las cuales fue aplicada para la generación de un mapa a escala real del espacio en el cual se trabajó con el robot, donde, después de la aplicación de los métodos, se generó un camino para que el robot pudiera seguir por medio de la identificación de velocidades deseadas en función de la diferencia entre la pose del robot en cada momento dado y la pose deseada.

No obstante, considero que el aspecto más enriquecedor fue la transformación de la simulación a acciones reales, ya que, nos permitió conocer las distintas complicaciones que pueden darse a la hora de generar la comunicación entre distintos elementos del sistema, al igual que el efecto que tiene usar ciertos componentes electrónicos, al igual que las distintas acciones que se pueden aplicar para resolver estos problemas en cierta medida.

Michelle Borjon

Al realizar esta actividad se hizo una integración de todas las actividades que estuvimos desarrollando a lo largo del módulo de Robótica móvil y de los aprendizajes que obtuvimos en el módulo del reto. Si bien, en las actividades del módulo 4 nos enfocamos más en comprender el movimiento del robot y los algoritmos que le permitirían realizar tales desplazamientos, en el módulo de reto obtuvimos las herramientas para volver la simulación realidad.

Debido a que los últimos semestres estuve de manera online, fue una gran experiencia para mí el poder estar presente al momento de realizar esta actividad, ya que las experiencias más cercanas que había tenido eran realizar simulaciones y ver los videos que enviaban mis compañeros de equipo al momento de poner a prueba el robot. Por lo que, experimentar de primera mano las dificultades a las que nos enfrentamos desde el momento de realizar la actividad me ayudó a comprender el cómo debían de conectarse correctamente cada uno de los cables de señal, break y dirección en el arduino para que tuvieran coherencia con lo que se expresaba en el código, así como me permitió ver que en ocasiones es necesario hacer prueba y error hasta obtener el resultado deseado, como fue el caso para obtener las velocidades a las cuales el robot avanzaba lo más recto posible.

Una vez teniendo esto, se implementaron códigos y fórmulas que aprendimos en clase para comprender el movimiento de robots móviles. No obstante, como en toda experimentación, nos enfrentamos a retos que no pudimos solucionar por la falta de tiempo, como lo fue la recepción de datos en dirección y velocidad, que generó problemas en el desplazamiento del robot.

Melissa Robles

En cuanto lo anteriormente abordado, puedo concluir que mediante esta actividad fue posible poner en práctica los conocimientos previamente adquiridos sobre el modelo de bicicleta, la creación de mapas para que un robot móvil pueda recorrer trayectorias que comienzan desde un punto inicial hasta un punto final y del mismo modo estas trayectorias puedan optimizar rutas. El hecho de haber probado nuestro algoritmo desarrollado de PRM en un robot diferencial en físico, fue una experiencia nueva, puesto que si tuvimos

demasiados percances al momento de hacer pruebas, ya que por grupos nos teníamos que turnar para poder utilizar y hacer pruebas con el robot diferencial, por lo que en un punto fue imposible realizar pruebas de que éste siguiera la trayectoria que habíamos generado en un mapa del salón de clases, ya que el robot estaba dañado físicamente, entonces este no podía correr de la misma manera que nos hubiera gustado como equipo probar, pero en sí concluyo que fue una grata experiencia llevar a la práctica los conocimientos adquiridos durante esta unidad de formación.

Belén Muñoz

Esta actividad es uno de los primeros acercamientos que tenemos sobre el uso de la robótica móvil. A pesar de que tuvimos actividades previas, se pudo realizar de manera física el uso de planificación y trayectorias. Este acercamiento nos permitió conocer esta área que aún es de investigación y darnos cuenta de la importancia y delicado es el crear una ruta.

En estos momentos se podría decir que hicimos un carrito tipo juguete, pero en un futuro puede ser ya sea vehículos autónomos, como AGVs o rutas en un gps. Asimismo nos enfrentamos y aprendimos el uso de Matlab para controlar un dispositivo. Ya que jamás lo habíamos hecho. De esta actividad me llevo lo importante que son los algoritmos, la importancia de seguir investigando los y como actualmente se depende mucho de la robótica móvil.

David Padilla

Los algoritmos para planificación de trayectorias es un campo de estudio que se encuentran en auge, ya que actualmente existe un gran interés en optimizar el proceso, para de esta forma reducir el costo computacional que lleva consigo esta actividad y de esta forma poder reducir tiempo, así como tener rutas y trayectorias más eficientes. El tener conocimiento de algunos de los métodos empleados actualmente para resolver esta problemática como lo es el PRM, contribuye en que podamos contribuir algún día en el refinamiento de estos métodos o en la creación de un nuevo algoritmo más eficiente, en diversos rubros.

De igual forma es relevante recalcar la implementación de los modelos generados en un sistema físico, ya que en las simulaciones, tanto el

robot como el entorno se encuentran en condiciones ideales. Ya que en condiciones reales hay diversos factores que afectan el desempeño del mismo modelo, que representan un gran reto a la hora de la implementación de este.

Johann Acaro

Podemos identificar que en el trabajo realizado se logró aplicar los conocimientos adquiridos durante el curso estudiado. Es importante recalcar que los análisis realizados del modelo bicicleta y el robot diferencial fueron esenciales para poder entender cómo podemos mediante el análisis cinemático hacer que estos se muevan de una posición a otra basándonos en coordenadas previamente dadas. Luego de entender bien este principio mediante el software de Matlab primero tuvimos que crear una simulación de mapa del salón, con el comando Makemap. Para poder hacer esto nos basamos en medidas proporcionales tomadas directamente del salón, ya que al momento de colocar el programa en el robot sería necesario tener un ambiente real para probarlo.

Un punto importante a mencionar dentro del aprendizaje obtenido, fueron los distintos métodos de trayectorias y rutas, ya que nos dieron una claridad al momento de querer optimizar caminos por los que debe ir un robot. En nuestro caso usamos el planificador PRM, que aunque sabemos que no es de los más eficientes, puede hacer su trabajo y ocupa un trabajo informático menor. Estas aplicaciones se complementaron con lo trabajado en reto donde se entendió la comunicación que podíamos lograr entre el módulo bluetooth del arduino y matlab. De manera que la parte electrónica en donde controlamos todo lo de alimentación, break, dirección y pwm de los motores, se comunicó con la programación y fue el conjunto de todo este trabajo que nos ayudó a poner de manera física en el robot los conocimientos aprendidos. Fue un camino complejo, pero los aprendizajes se pudieron aterrizar con el proyecto realizado.

Alejandro Ojeda

En este proyecto trabajamos en la implementación de varios puntos y temas vistos durante el bloque, lo cual considero como un gran avance en el ámbito de nuestra carrera, se tuvo que poder acoplar temas de programación y control tales como la generación de mapas y trayectorias, junto a

esto uno de los puntos más difíciles que es el control del movimiento de un robot en función de un camino de puntos, pero lo más retador como tal fue hacer una comunicación bluetooth exitosa del robot a nuestra "terminal" lo cual representó la parte a la más tiempo se le dedicó del proyecto. Como tal veo este un trabajo completo y retador ya que empleamos casi por completo nuestros conocimientos como mecatrónicos.

Leopoldo Moreno

En lo personal me gustó mucho este mini reto que tuvimos casi al final del bloque porque logramos aplicar mucho conocimiento previamente analizado en clase en un robot físico donde pudimos apreciar y comprender las numerosas aplicaciones que podemos tener en el área de la robótica. En esta ocasión aplicamos un controlador a un robot diferencial, en donde se realizó un movimiento siguiendo una trayectoria de un mapa previamente elaborado por nosotros, utilizamos un *arduino UNO* como microcontrolador, sin embargo se utilizó matlab para la elaboración del programa que le indica cómo el robot debe seguir la trayectoria, esta comunicación se hizo mediante un módulo bluetooth hc-05 para transmitir información de manera inalámbrica. Se realizaron pruebas de velocidad en el robot para convertir datos PWM en información que nosotros como usuarios podamos entender, es decir, velocidad en metros sobre segundos.

Fue una práctica muy cool donde utilizamos conocimientos de electrónica para la elaboración del robot, elementos de cinemática para elaborar las ecuaciones que describen el movimiento del robot y programación para la elaboración del mapa y trayectoria del robot mediante una conexión inalámbrica para transmitir información al robot de manera remota.

REFERENCIAS

P. Corke, Robotics, Vision and Control: Fundamental algorithms in Matlab, 2nd ed., Springer International Publishing, 2017.

Y. Ding, "Simple Understanding of Kinematics Bicycle Model". [2020]. [Online]. Available: <https://dingyan89.medium.com/simple-understanding-of-kinematic-bicycle-model-81cac6420357>

ANEXOS

[Código de Matlab](#)

[Código de Arduino](#)

[Datos experimentales de velocidad](#)