



Actividad 1.10

SLAM de lidar

Integrantes:

A017314050 | Eleazar Olivas Gaspar

A01735696 | Azul Nahomi Machorro Arreola

A01732584 | Angel Estrada Centeno

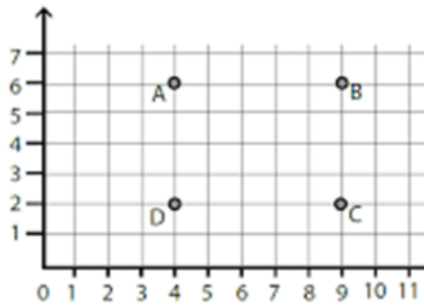
A01735692 | Arick Morelos del Campo

Profesores:

Alfredo Garcia Suarez

MAYo 2024

Waypoints 1



Solución de mapa en exampleMap:

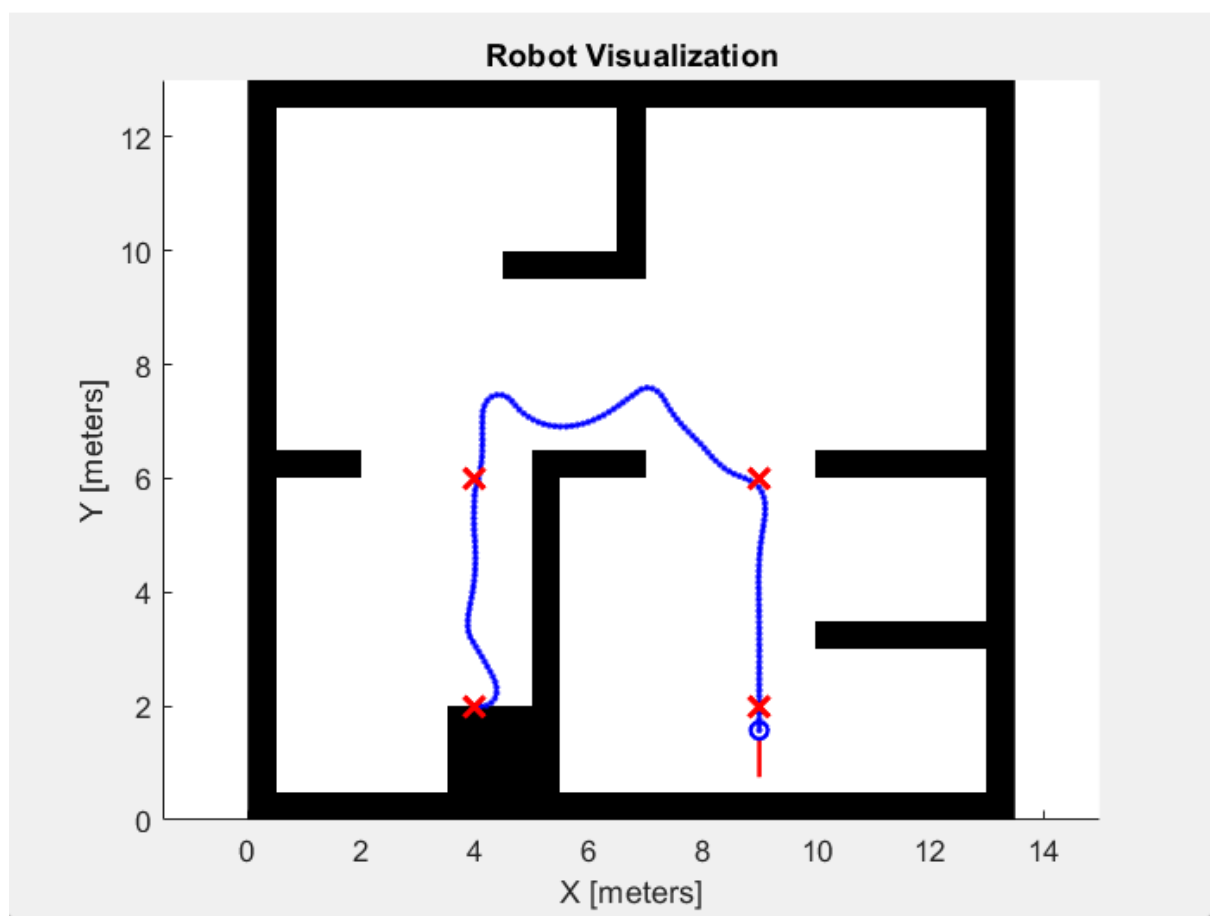


Fig 1. Resultados de la simulación de evasión de obstáculos con la implementación de LiDAR dentro de exampleMap.

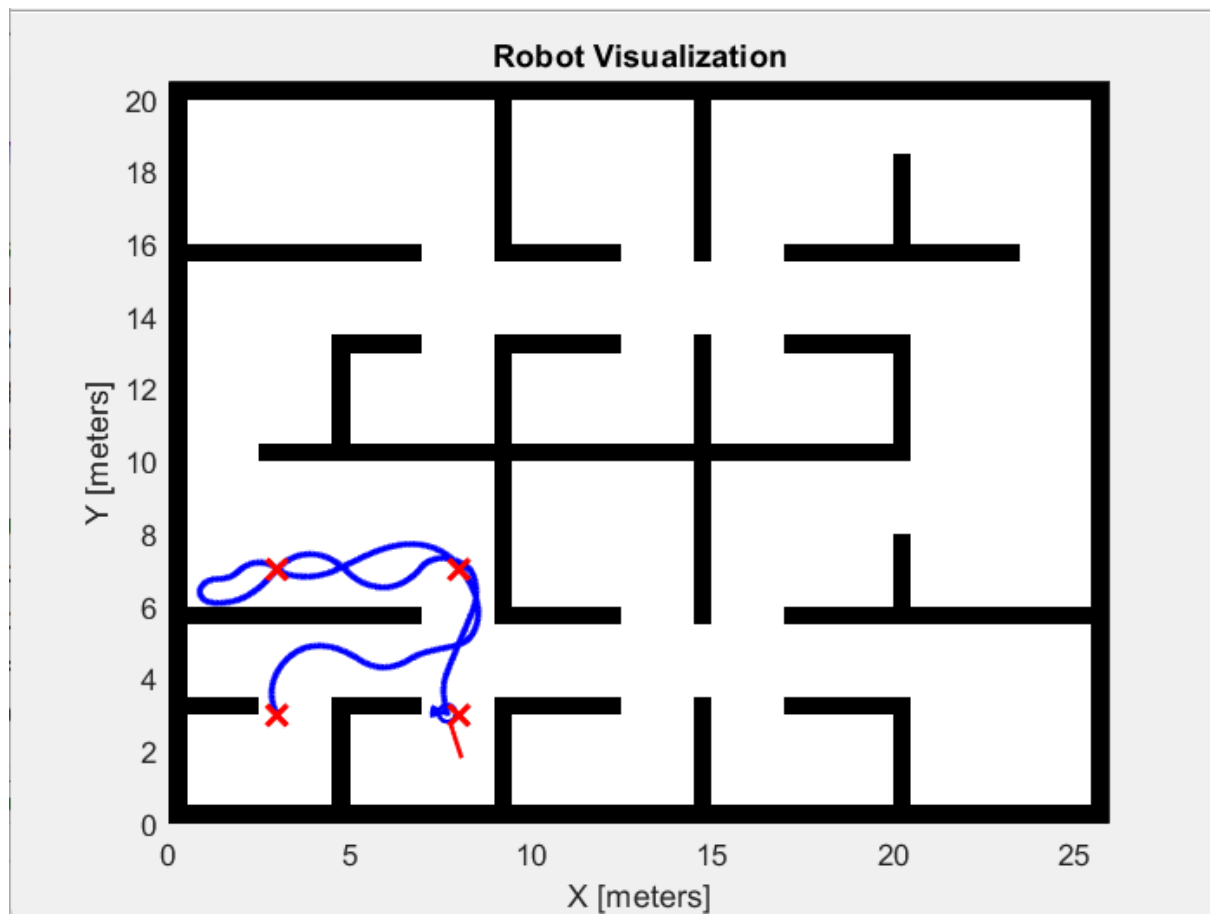
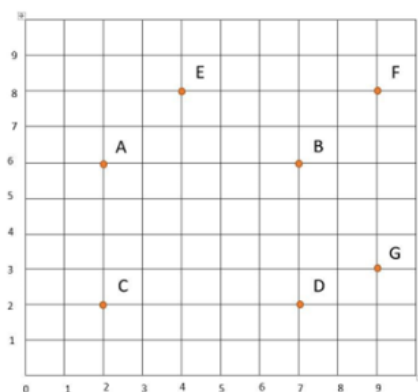


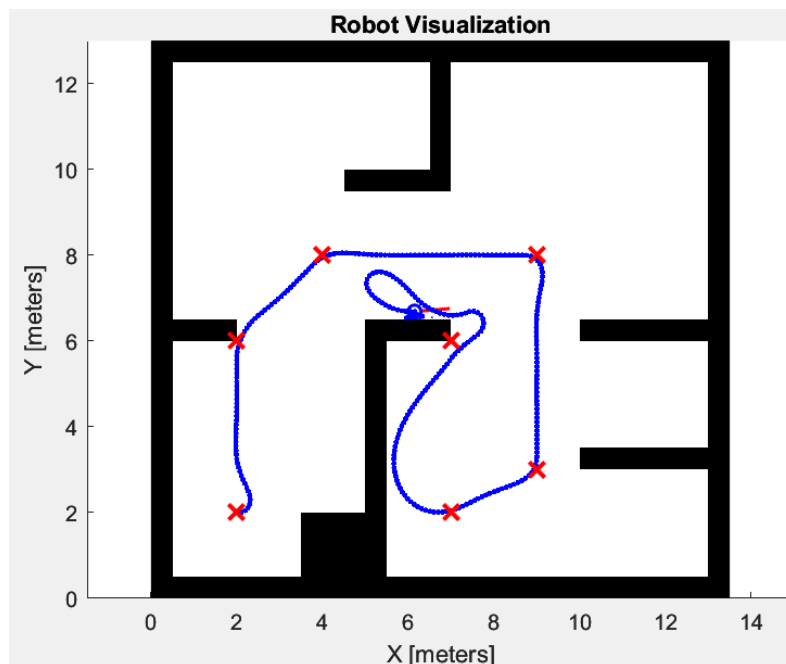
Fig 2. Esquivamiento de bordes dentro de complexMap.

Para ambos ejercicios anteriores lo que se mueve principalmente, o con mayor frecuencia es la resolución, el largo de lidar así como el orden de los puntos, de no llegar a una solución de esta forma se empieza a mover el look ahead, las velocidades lineales y angulares con el fin de moverse de mejor manera dentro de los mapas.

Waypoints 2

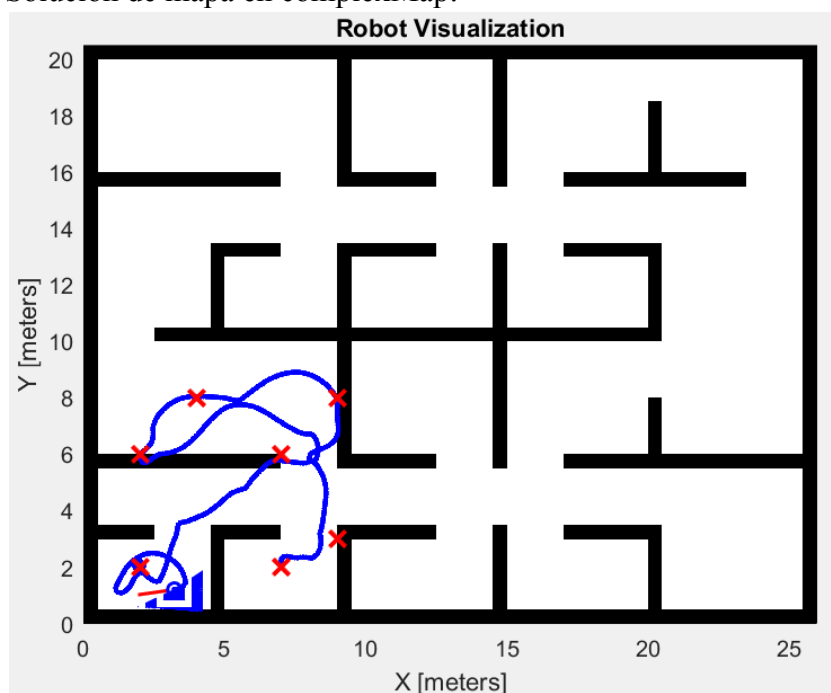


Solución de mapa en exampleMap:



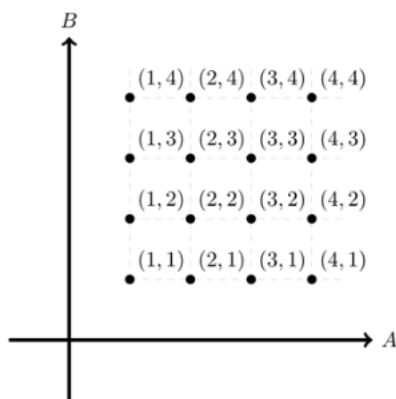
Para este recorrido, en este mapa específicamente que es más amplio, las variables modificadas más importantes fueron el rango máximo del LIDAR ya que no es necesario detectar los bordes con mucha distancia, en este caso se dejó a 0.5 metros. También se modificó la distancia de visión a un valor de 1, ya que esto le da dirección al robot de donde debe ir pero aún es bastante preciso para llegar a los puntos. Con un tiempo de simulación más corto se evitaría que el robot de ese recorrido excesivo al final al haber llegado al punto final.

Solución de mapa en complexMap:

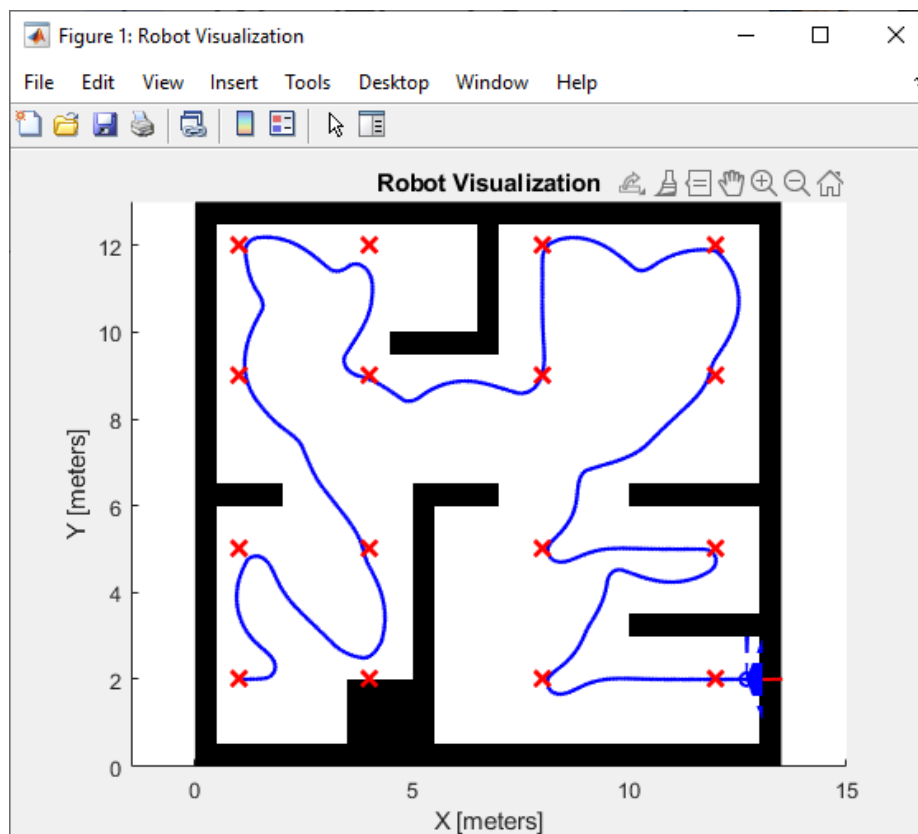


Para este mapa mucho más reducido en espacio de maniobrabilidad se ajustaron los valores del rango máximo del LIDAR a 1.5 pues así el robot puede ver los muros mucho antes de chocar con ellos. También se ajustó la distancia de visión a 0.3 metros para evitar que el robot viera un punto aún lejano y atravesara la pared. En este recorrido el orden en que se recorrieron los puntos fue clave, ya que de comenzar por el punto (2,2) que se encuentra encerrado por los muros del laberinto es sumamente difícil hacer que el robot rodee la pared para llegar al segundo punto, en cambio la atraviesa.

Waypoints 3

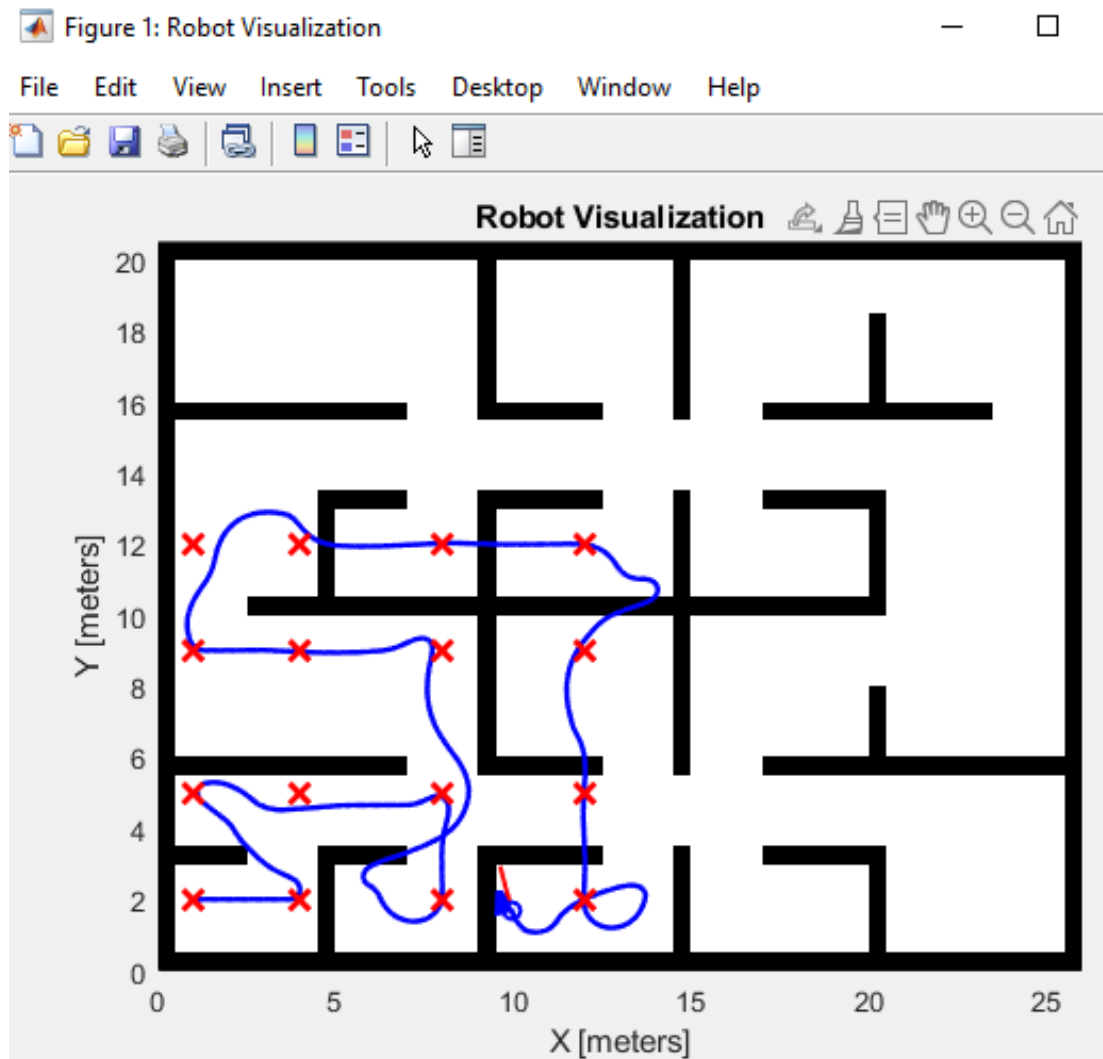


Solución de mapa en exampleMap:



En este recorrido podemos observar cómo se logró el objetivo a excepción de un pequeño error en algunos puntos, esto debido al ángulo que se genera en ciertos momentos de la trayectoria, para esta trayectoria definimos un samptime de 0.7 por un tiempo total de 60, un LookaheadDistance de 05.

Solución de mapa en complexMap:



para este tipo de mapa que es más complejo la velocidad lineal y angular debe ser más precisa ya que un error aunque sea mínimo cambia la trayectoria, en esta trayectoria el tiempo de simulación fue un poco más grande y el samptime fue el mismo que el mapa anterior, sin embargo los límites del sensor se modificaron para mejorar la trayectoria, sin embargo aun existe problemas con el cruce de algunas paredes en ciertos ángulos.

3.

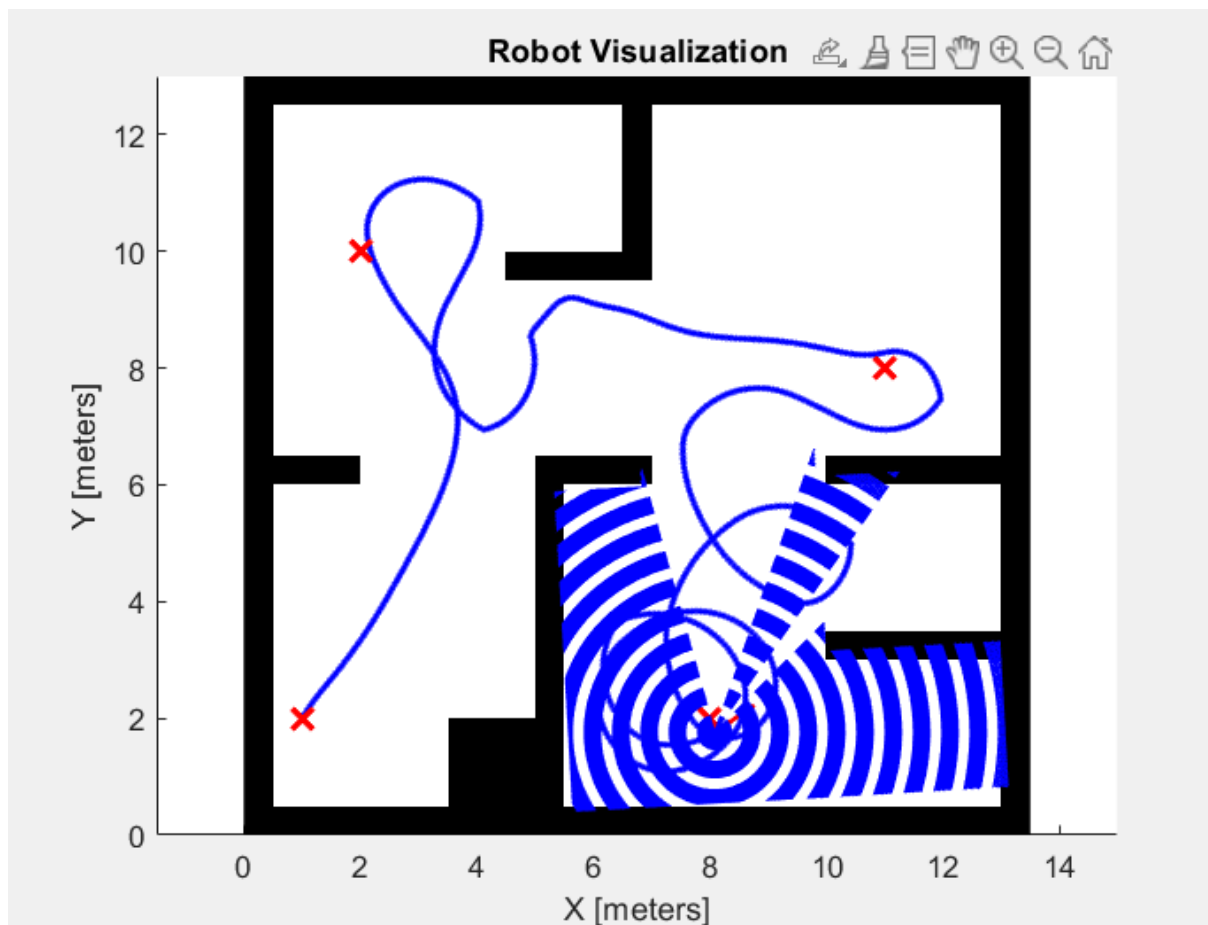
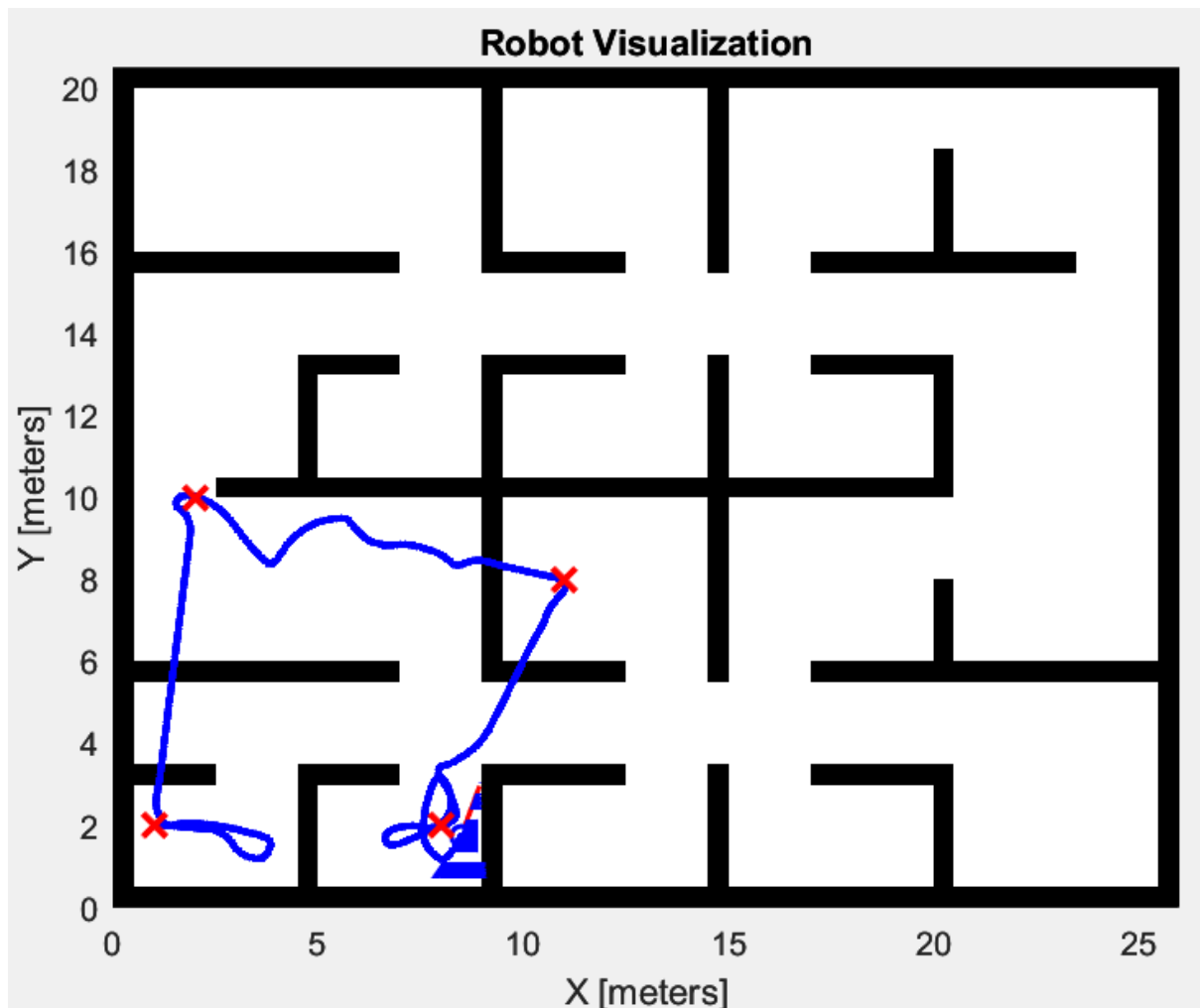


Fig 8. recorrido del punto 3 en exampleMap.

Como se puede ver a diferencia de los demás ejercicios este no pudo regresar al punto inicial, esto es debido a que el algoritmo que se usa se cicla en el último punto, esto es debido a que al dejar de percibir el obstáculo intenta ir directo al punto final pero no es capaz de darle la vuelta a todo el obstáculo por lo que se queda haciendo círculos en el penúltimo punto.



Este recorrido no tiene solución con este algoritmo de bug0 ya que sin importar todas las combinaciones que se hagan para ajustar las variables de Lidar.maxRange para la detección de los muros; o de LookaheadDistance para que el robot vea y llegue a los waypoints; incluso la cantidad de laser que manda el LIDAR o las velocidades del robot. El robot se cicla tratando de rodear el muro pero a la vez intenta alcanzar el waypoint por lo que no consigue salir del primer punto y termina ignorando las paredes y las atraviesa por completo.