



UNIVERSIDAD DE BUENOS AIRES
FACULTAD DE INGENIERÍA
Año 2022 - 1.º trimestre

(86.49) ROBÓTICA MÓVIL
TRABAJO PRÁCTICO ESPECIAL

Alumnos:

Ochagavía, Lara

100637

lochagavia@fi.uba.ar

Hirschmann, Juan Ignacio

100755

jhirschmann@fi.uba.ar

1. Objetivo

El presente trabajo busca resolver un problema de vigilancia en un entorno conocido y uno de exploración en un entorno por conocer aplicando los conocimientos adquiridos en la materia, seleccionando con criterio los métodos a utilizar para conseguir soluciones óptimas y adecuadas a los desafíos que se presentan.

2. Desafío 1: Vigilancia

El robot debe vigilar una parte de la Facultad de Ingeniería, visitando dos puntos dentro de un entorno conocido. El robot comienza su tarea en una posición sin determinar y debe visitar los puntos dados por las siguientes coordenadas, en orden de aparición:

Punto A	(1,5 ; 1,3)
Punto B	(4,3 ; 2,1)

Una vez que se llegue al primer punto, el robot deberá permanecer con velocidad nula por tres segundos, y al llegar al punto final deberá detenerse. Se cuenta con tres minutos para realizar la tarea y no puede chocar con ningún obstáculo mientras se la lleva a cabo.

2.1. Implementación propuesta

Se abordó el problema separándolo en dos partes: la estimación de la posición actual de robot y el planeamiento de trayectoria. Para la localización del robot se implementó un filtro de partículas de 500 partículas y para el planeamiento de la trayectoria se implementó el algoritmo A*.

2.1.1. Estimación del *belief*

A grandes rasgos, el filtro de partículas consiste en generar un cierto número de partículas que representan una posible hipótesis acerca de la posición de robot. Cada hipótesis es actualizada según el modelo de movimiento y ponderadas según algún criterio. Luego, se eligen las partículas que sobreviven con probabilidad proporcional a su verosimilitud. Este procedimiento requiere definir el criterio de ponderación y la forma de remuestreo de las partículas.

Como criterio de ponderación se realizaron mediciones virtuales desde cada partícula y se comparó con la medición real utilizando el error cuadrático medio, asignando una verosimilitud inversamente proporcional. Si bien se evaluó calcular la verosimilitud como el producto de probabilidades gaussianas de acuerdo al modelo de medición basado en haces, este criterio resultaba demasiado restrictivo y podía hacer converger el *belief* del robot erróneamente:

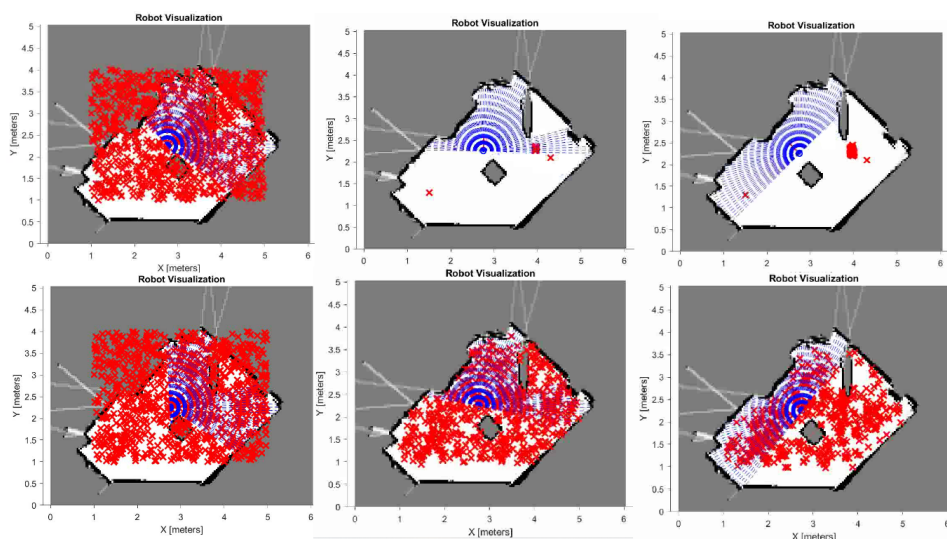


Figura 2.1: Remuestreo de partículas bajo distintos criterios: arriba considerando el modelo basado en haces y abajo con error medio cuadrático (intervalos de 2 segundos)

Se observa que en el caso del modelo basado en haces las partículas colapsan rápidamente alrededor de una hipótesis errada y lo hacen lentamente en el caso del error cuadrático medio. Dado que el error cuadrático medio reduce la variabilidad de las hipótesis de una manera lenta, existe un riesgo menor de que el *belief* converja alrededor de una posición errónea. Si bien esto reduce el tiempo necesario para localizarse, se comprobó que el *belief* resultante representaba mejor la posición real. Como medida adicional, se optó por generar un porcentaje de partículas aleatorias independientes del remuestreo realizado. Esto mitiga el problema del robot secuestrado y favorece la variabilidad de hipótesis.

Dado que también existen restricciones temporales, resultó prohibitivo y contraproducente aplicar el remuestreo instantáneo a instantáneo y se optó por realizarlo cada cierta cantidad de movimientos. Además, se eligió seleccionar solamente un número fijo de haces para realizar las comparaciones.

En un primer momento, para conocer la posición inicial del robot, se lo hace girar en su propio eje, sin desplazarse, para tomar las mediciones que realiza el LIDAR y obtener con el filtro la estimación de las coordenadas iniciales. Todos los movimientos realizados por el robot son realizados con una velocidad lineal fija o con una velocidad angular fija, esto indica que el robot únicamente podrá rotar sobre su eje o avanzar linealmente.

2.1.2. Planeamiento de trayectoria

Una vez obtenida la estimación de la posición inicial, se planifica una trayectoria libre de obstáculos utilizando el algoritmo A* para llegar al primer punto objetivo con el camino óptimo.

El algoritmo A* busca la mejor trayectoria posible libre de obstáculos a partir de encontrar las celdas vecinas en el mapa de ocupación cuya probabilidad de ocupación sea menor a un cierto umbral para asegurar la ausencia de obstáculos en el recorrido. Para este caso particular, para asegurar que no se elija ninguna celda ocupada, se optó por aumentar el mapa con la función *inflate*, la cual aumenta la probabilidad de ocupación de las celdas vecinas a los obstáculos para que estos últimos se vean mayores.

Al pasarlo a la función *A_star* el mapa, la posición estimada del robot y el destino deseado, se obtiene como resultado un arreglo con coordenadas que el robot debería visitar para llegar al objetivo evitando colisionar con cualquier obstáculo en el camino y con estos datos, se calculan los tiempos necesarios durante los que deben aplicarse las velocidades lineal y angular para llegar a cada punto. Sin embargo, se considera que el robot presenta errores tanto al desplazarse como al estimar su ubicación, por lo que se realizan correcciones de su ubicación cada una cierta cantidad de pasos y es por eso que se vuelve a calcular el camino óptimo luego de aplicar una cierta cantidad de correcciones a la estimación de la posición.

Al llegar al punto A, se le dan las indicaciones al robot de esperar tres segundos en dicha posición para luego planificar la trayectoria hacia el punto B, aplicando las correcciones y re-planeamiento de la misma manera que se mencionó con anterioridad.

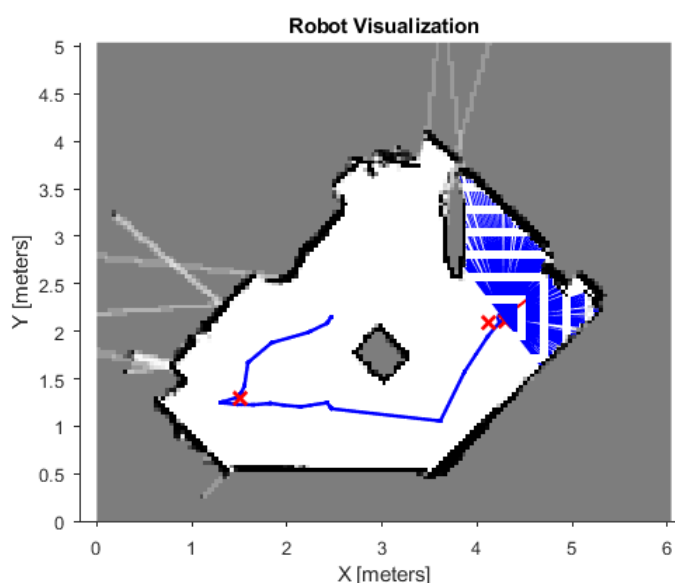


Figura 2.2: Recorrido del robot desde su posición inicial hasta el punto B

Se observa en la figura 2.2 que el recorrido del robot continúa pasadas las coordenadas del punto A, esto se puede deber a que el *belief* del filtro de partículas estima la posición del robot con cierto error o a ciertos errores en los movimientos del robot, por lo que se continúa con el desplazamiento más allá del objetivo real, pero no por gran diferencia.

3. Desafío 2: Exploración

Para este desafío se tiene al robot ubicado en algún punto de un entorno desconocido. El objetivo es generar un mapa de ocupación del entorno a medida que lo explora. No deben ocurrir colisiones, por lo que se tiene que tener en cuenta a la hora de seleccionar las direcciones de desplazamiento. El algoritmo que se eligió para solucionar este problema es el SLAM, que realiza localización del robot y mapeo del entorno en el que se desplaza en simultáneo.

Para ejecutar el algoritmo mencionado, se creó un objeto de la clase `robotics.LidarSLAM`, que aplica el algoritmo a partir de los datos de medición del sensor y la resolución que se espera para el mapa a crearse.

Dado que el movimiento del robot debe permitir el mapeo de todo el espacio independientemente de sus condiciones iniciales, se optó por recorrer el espacio en sentido horario, asegurando que la trayectoria siga el obstáculo que detecte a su izquierda y esquivando obstáculos a medida que estos se presentan por delante. Una vez que el robot se encuentre lo suficientemente cerca de su posición inicial, invertirá el sentido del recorrido.

La forma en que el robot esquiva los obstáculos frontales es adoptando una orientación que sea paralela al vector que se forma entre los extremos de sus mediciones entre un rango de apertura determinado. A continuación, se insertan esquemas a modo ilustrativo, no a escala, de cómo aplica las rotaciones el robot.

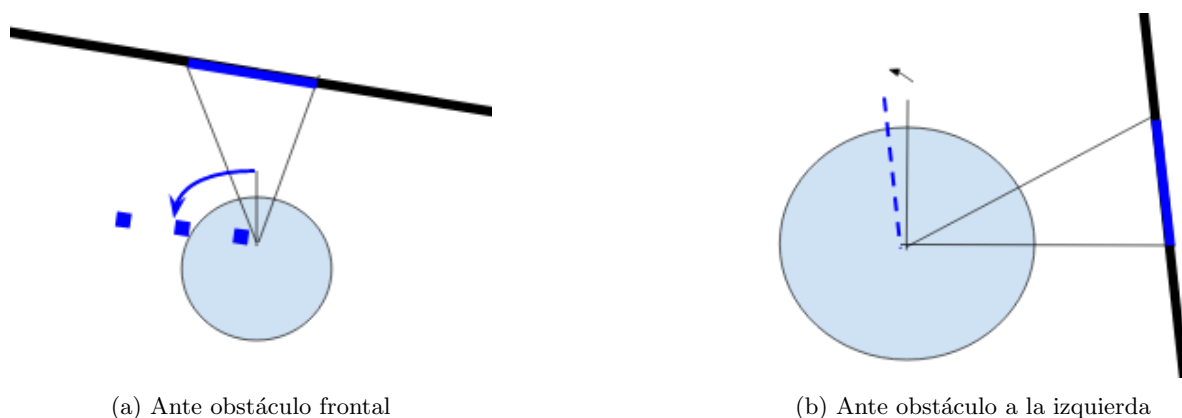
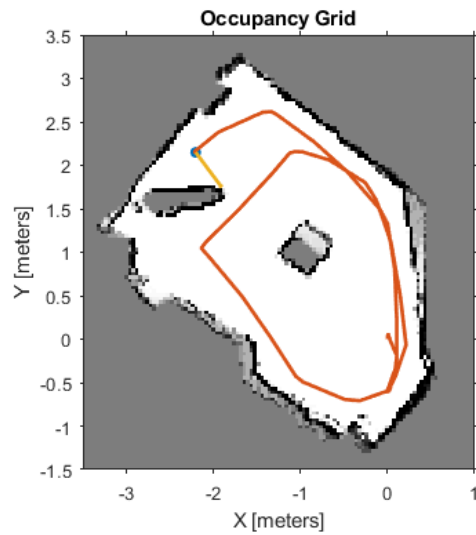


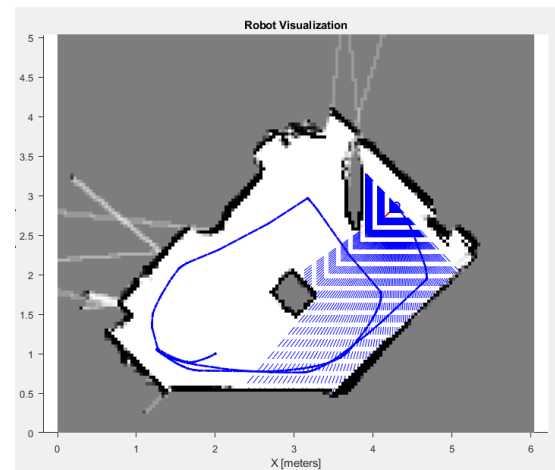
Figura 3.1: Rotación del robot

Si bien se evaluaron otros métodos de implementar el algoritmo para recorrer todo el entorno, tales como realizar movimientos aleatorios ante la detección de un obstáculo o trasladarse rebotando sobre las paredes, los resultados fueron de calidad variable y poco consistentes.

Al finalizar el recorrido, el mapa de grillas generado es el siguiente:



(a) Mapa de grillas generado



(b) Mapa original

Figura 3.2: Mapas del entorno

Puede compararse la figura 3.2a con el mapa de la figura 3.2b para observar que los resultados son semejantes. Las mayores diferencias pueden observarse en recovecos a los cuales el robot no alcanzó por estar parametrizado con grandes medidas de seguridad.