10.0

Hernández Abad Liber Adrián
Robots móviles
Grupo 1

1. La configuración es una especificación completa de la posición de cada punto de un sistema.

   El espacio de configuración es el espacio de todas las posibles configuraciones del sistema (la configuración es simplemente un punto en el espacio de configuraciones).

   El número de grados de libertad de un robot es la dimensión del espacio de configuraciones o el número mínimo de parámetros que se necesitan para especificar la configuración.

2.
   - A* (A-estrella): Permite una búsqueda rápida de nodos debido a la implementación de heurísticas (es una extensión del algoritmo de Dijkstra), su característica más importante es la determinación de la función de costo, que define los costos de los nodos. El algoritmo es costoso en términos de memoria y velocidad para áreas muy grandes pero muy adecuado para espacios de búsqueda conocidos por el robot.

   - D* (D-estrella): Parte de una familia de algoritmos de búsqueda junto con Focussed D y D* Lite. Asume sobre las partes desconocidas del terreno (por ejemplo que no contiene obstáculos) y encuentra el camino más corto a partir de las coordenadas actuales hasta las coordenadas deseadas. Repite el proceso hasta que llega a su meta o determina que las coordenadas no pueden ser alcanzadas.

3.
   - PRM (Probabilistic Roadmap Method): Toma un muestreo aleatorio del espacio de configuración del robot, prueba si está en el espacio libre y usa el planeador local para intentar conectar estas configuraciones a otras cercanas. Se agrega la salida y la meta y después se aplica una búsqueda por método de grafo al grafo resultante de la unión de las configuraciones entre salida y meta para determinar un camino entre estas.

   - RRT (Rapidly-exploring Random Tree): Permite una planeación rápida en espacios semi estructurados mediante la ejecución de una búsqueda aleatoria a través del área a navegar. Tiene la capacidad de considerar restricciones no holonómicas.

4. SLAM (Localización y construcción del mapa simultáneamente por sus siglas en inglés) consiste en el proceso que sigue un vehículo autónomo, el cual empieza en una posición desconocida en un ambiente desconocido y después construye un mapa de su ambiente progresivamente al mismo tiempo que usa este mapa para calcular la posición absoluta del vehículo.

5. Ambos son métodos probabilísticos para calcular la posición del robot en su ambiente.

   El filtro de partículas mantiene densidad de probabilidades y su idea principal es la determinación del posterior (siguiente posición estimada) mediante un set de sampleo. Cada sampleo consiste de un par (x,?) que contiene un vector de estado x del sistema y un factor de peso ? los cuales son usados para determinar la importancia de dicha partícula. El posterior es representado por la distribución de los sampleos y su factor de importancia.

El filtro de Kalman es un método recursivo que determina el estado de un sistema dinámico en presencia del ruido. Mantiene la estimación tanto del vector de estado como la matriz de error estimado de covarianza. Esto quiere decir que la salida del filtro de Kalman es una función de densidad de probabilidad Gaussiana.

El filtro de Kalman es un caso específico de la estimación probabilística donde el estimado se asume Gaussiano (unimodal, es decir, que asume que hay un solo pico en la distribución estadística) mientras que el filtro de partículas es un método Bayesiano que puede manejar más distribuciones estadísticas en el ambiente.

En ROS el filtro de Kalman es implementado mediante el paquete *robot_pose_ekf* mientras que un caso específico de filtro de partículas llamado *Particle Intersection* puede ser implementado mediante el paquete *mcl_pi.* Además se puede implementar otro filtro de partículas usando la biblioteca *bfl*.

6. Consiste en calcular campos imaginarios de repulsión que emanan de los obstáculos. La fuerza de repulsividad varía de acuerdo a la distancia entre el robot y el obstáculo, entre más cerca esté del obstáculo mayor es la fuerza de repulsión.

   Para su implementación se puede utilizar primeramente el método de rejilla, para calcular la distancia en un mapa, el cual es una matriz compuesta de elementos llamados pixeles. Un pixel tiene el valor 0 si está completamente libre de obstáculos y 1 si está completamente o parcialmente ocupado por un obstáculo.

   Los pixeles pueden tener 4 u 8 vecinos, dependiendo de la movilidad del robot. Después se aplica un algoritmo (por ejemplo el brushfire) para aproximar la distancia y por lo tanto la función repulsiva. La entrada de este algoritmo es una matriz de pixeles que poseen el valor 0 y 1, la salida es una matriz de pixeles cuyos valores correspondientes miden la distancia al obstáculo más cercano. Estos valores se pueden usar para calcular la función de potencial repulsivo así como su gradiente.

   Entre más cerca se encuentre un pixel de un obstáculo mayor será su valor, el algoritmo se encarga de elegir el pixel vecino con el menor valor para elegir una ruta.

7. Transformación de otro sistema de coordenadas a un sistema de coordenadas homogéneo. Se aplica mediante matrices de transformación y se utilizan para:
   - Representar la posición y orientación de un sistema girado y trasladado con respecto a un sistema fijo.
   - Transformar un vector expresado en coordenadas movibles y su representación en un sistema fijo.
   - Rotar y trasladar un vector con respecto a un sistema fijo.

   La matriz de transformación está compuesta por los términos escalamiento, traslación, rotación y perspectiva.

8. Dentro de las propiedades de un robot se encuentra la capacidad de movimiento de este. Un robot no holonómico es el que está sujeto a restricciones del mismo nombre. Se dice que un robot tiene restricciones no holonómicas cuando está sujeto a restricciones de velocidad, que no se puede mover en cualquier dirección aún sin la presencia de obstáculos como por ejemplo a los lados mientras tiene una velocidad hacia enfrente. No obstante esto no significa que el robot no pueda alcanzar cualquier posición que desee, lo cual puede lograr por ejemplo con maniobras de estacionamiento en paralelo.

   En otras palabras, su posición final depende del camino que tome mientras que un omnidireccional no depende del camino que tome ya que puede moverse en cualquier dirección sin restricción.