



UNIVERSIDAD NACIONAL AUTÓNOMA DE
MÉXICO

FACULTAD DE INGENIERÍA



ROBOTS MÓVILES Y AGENTES INTELIGENTES

Grupo: 2

Examen Parcial

Profesor: Marco Antonio Negrete Villanueva

Alumno: Gutiérrez Martínez Rodrigo

Fecha entrega: 04 de junio de 2020

Semestre 2020-2

Examen Parcial

Construcción de Robots Móviles, FI, UNAM, 2020-2

Nombre: Rodrigo Gutiérrez Martínez

Instrucciones: Subir la solución en un solo archivo PDF con el nombre Examen.pdf, al repositorio en GitHub en la carpeta Entregables.

1. Explique qué es la configuración, espacio de configuraciones y grados de libertad de un robot móvil.

Configuración: Se trata de definir cómo estarán distribuidos los principales elementos que lo componen: ruedas, motores, encoders además de indicar la posición y la orientación. La precisión del robot dependerá en gran medida de la configuración que se le da. Se representa como un punto en el espacio n-dimensional, en donde el número de dimensiones n, corresponde al número de grados de libertad de un robot.

Espacio de configuraciones: denominado también (c-space), es el área en la cual el robot puede posicionarse. Para un robot móvil, se define como el conjunto de todas las transformaciones posibles, este espacio de trabajo corresponde al área libre de colisiones con obstáculos y consigo mismo (también conocido como free configuration space). Por ejemplo para un robot que se desplaza sobre una superficie, manteniendo su orientación fija $p=(x,y)$ es una configuración válida y R^2 es un espacio de configuraciones debido a que para describir el estado del robot basta con dos parámetros que indiquen su posición en XY

Grados de libertad: Por sus siglas en inglés DOF (degree of freedom) hacen referencia al número de movimientos independientes que se pueden realizar. En otras palabras, un grado de libertad es la capacidad de moverse a lo largo de un eje (movimiento lineal) o de rotar a lo largo de un eje (movimiento rotacional).

2. Investigue dos métodos basados en grafos para planeación de rutas.

Algoritmo de Dijkstra:

Su objetivo, es determinar la ruta más corta, desde el nodo origen, hasta cualquier nodo de la red. Surgió por el problema de construir un árbol de longitud total mínima entre "n" nodos y de encontrar la ruta de longitud total mínima entre dos nodos.

Su metodología se basa en iteraciones, de manera tal que, en la práctica, su desarrollo se dificulta a medida que el tamaño de la red aumenta, dejándolo en clara desventaja, frente a métodos de optimización basados en programación matemática.

La metodología es la siguiente:

1.-A partir del vértice inicial el algoritmo selecciona los vecinos directos con su respectivo costo para llegar ahí.

- 2.-Desde el vértice con el menor costo procede a sus vértices adyacentes y los marca con el costo para llegar a ellos por si mismo si este costo es menor.
- 3.-Despues de la comprobación el algoritmo procede al vértice siguiente con el costo más bajo.
- 4.- Una vez llegado al vértice meta termina y el robot puede seguir los bordes que apuntan hacia el costo del borde más bajo

La razón principal para elegir esta técnica es porque se puede realizar una búsqueda rápida de la ruta óptima en tiempo real, considerando la comodidad, la seguridad y las limitaciones del vehículo, una desventaja es que puede desperdiciar tiempo analizando nodos que no son necesarios o que se encuentran analizando nodos que no son necesarios o que se encuentran lejos del nodo de llegada.

En la conducción automatizada, se ha sido implementado en simulaciones de múltiples vehículos, en el vehículo de Little Ben (entrada del equipo de carreras de Ben Franklin a Darpa Urban Challenge) y también el equipo VictorTango.

Algoritmo de estrella (A *): En robótica, el algoritmo permite una búsqueda rápida de nodos debido a la implementación de la función heurística (es una extensión del algoritmo de búsqueda de gráficos de Dijkstra), dicha función seleccionada como función de distancia nos brinda la distancia entre el nodo y la meta. Algunas de sus características más importantes son:

- La función de evaluación de la ruta viene dado por la sumatoria del costo del camino recorrido y el valor de la heurística par encontrar la ruta optima.
- Evita caminos que no necesitan ser evaluados.
- Su búsqueda es más rápida.
- Su aspecto de diseño más importante es la determinación de la función de costo, que define los pesos de los nodos.

$$f(n) = g(n) + h(n)$$

Donde $n = (x_n, y_n)$ es el nodo en expansión actual

$f(n)$ es el costo mínimo estimado de la celda entre todos los caminos desde el nodo de inicio S al nodo de meta G.

$g(n)$ es la función de costo real desde el nodo de inicio $S(x_s, y_s)$ hasta el nodo de expansión actual n

$h(n)$ es la estimación heurística del costo mínimo de una trayectoria desde el nodo de expansión actual n al nodo objetivo $G(x_g, y_g)$

Entre dos nodos $h(n)$ se puede calcular como la distancia euclidiana entonces $h(n)$ puede estimarse como:

$$h(n) = \sqrt{(x_n - x_g)^2 + (y_n - y_g)^2}$$

$h(n)$ es la distancia euclidiana entre el nodo objetivo y el nodo en expansión actual.

3. Investigue dos métodos basados en muestreo para planeación de rutas.

Probabilistic Roadmap Method (PRM): Es un método de planeación de rutas para robots con n grados de libertad en ambientes estáticos. Funciona de forma análoga a un robot explorador en un ambiente desconocido, cuya tarea es construir un mapa de dicho ambiente. Se divide en dos fases:

-Fase de aprendizaje: Aquí se construye un mapa del ambiente en que se desenvuelve el cual se representa como un grafo donde los nodos representan configuraciones en las que el robot se encuentra libre de colisión y las aristas representan rutas factibles entre dos configuraciones.

Fase de consulta: Dada una configuración inicial y una configuración final, el robot explorador trata de encontrar una ruta entre estas configuraciones, utilizando el mapa que construyó en la fase de aprendizaje.

Cabe destacar que este método encontrara una ruta con una alta probabilidad, siempre y cuando dicha ruta exista.

Rapidly-exploring Random Tree (RRT): Árbol aleatorio de exploración rápida (RRT) pertenece a los algoritmos basados en muestreo aplicables a la planificación de rutas en línea. Permite una planificación rápida en espacios semiestructurados mediante la ejecución de una búsqueda aleatoria a través del área de navegación. También tiene la capacidad de considerar restricciones no holonómicas (como el radio de giro máximo y el momento del móvil).

Consiste en un método de planificación probabilístico basado en modelo parecido al de PRM pero con la diferencia de que destaca a la vista es que el grafo se expande en forma de árbol desde el punto inicial, el cual es la raíz. El algoritmo RRT se basa en la construcción de un árbol de configuraciones que crece explorando a partir de un punto origen.

El árbol se construye de forma incremental a partir de muestras aleatorias del espacio y tiende a crecer hacia grandes areas poco exploradas de forma inherente. Una muestra es una celda del espacio, elegida de forma aleatoria, que no representa un obstáculo. Después de obtener cada muestra, se intenta realizar una conexión entre esta muestra y el nodo más cercano del árbol. Si esta conexión es factible, la muestra se añade como un nuevo nodo al árbol. Si la conexión al nodo más cercano no es posible, la conexión se realiza con el nodo del árbol con menor coste de entre los posibles candidatos que permiten una conexión libre de obstáculos. Para detectar obstáculos entre dos puntos del espacio se utiliza el algoritmo de Bresenham.

4. Explique en qué consiste el proceso de SLAM (Simultaneous Localization and Mapping).

Se plantea si es posible que un vehículo autónomo arranque en una ubicación desconocida, en un entorno desconocido y luego construya un mapa de este entorno mientras usa este mapa simultáneamente para calcular la ubicación absoluta del vehículo. Para ello, se identifican o conocen previamente unos puntos característicos que serán usados de referencia por el sistema. Se hará referencia a SLAM como “el sistema” y no como “el algoritmo”, ya que son varias etapas las que componen el proceso, cada una con sus propios algoritmos.

5. Explique en que consiste la localización mediante filtros de partículas, sus ventajas sobre el Filtro de Kalman y los paquetes de ROS que lo implementan.

El filtro de partículas es un filtro bayesiano basado en la metodología Monte Carlo, y parte del supuesto de que el proceso es una cadena de Markov; en este filtro se propone generar M muestras aleatorias del estado actual a partir de la función de probabilidad a priori, llamadas partículas, que representan espacios de estado artificiales del sistema; estas partículas son agrupadas en conjuntos que se irán actualizando conforme los sensores tomen medidas y el algoritmo estime posibles nuevos estados del sistema.

Kalman: proporciona directamente una solución recursiva al problema de navegación y un medio para calcular estimaciones consistentes de la incertidumbre en las ubicaciones de los puntos de referencia del robot y del mapa sobre la base de modelos estadísticos para el movimiento del vehículo y observaciones relativas de puntos de referencia.

Así que algunas ventajas pueden ser:

- No paramétrico (distribuciones no lineales y multimodales)
- Es adaptativo: más partículas al aumentar la densidad probabilista.
- Tienen poco costo computacional -.
- Pueden ser aplicados en RTS con algoritmos Any-Time.
- Más fáciles de implementar.

Paquetes que lo implementan:

El paquete **gmapping** utiliza un filtro de partículas en el que cada partícula lleva un mapa individual del entorno, lo que supone un elevado número de partículas que habrá que intentar reducir. Para resolver este problema, Gmapping presenta una técnica de adaptación por la cual, mediante un filtro de partículas Rao-Blackwellize 4 se logra una disminución de la cantidad de muestras. Lo que propone, es un enfoque para calcular la distribución exacta propuesta teniendo en cuenta no sólo el movimiento del robot sino también la observación más reciente. Esta drástica disminución de la incertidumbre sobre la pose del robot se lleva a cabo en el paso de predicción del filtro. Además, se aplica un enfoque selectivo para llevar a cabo la operación de remuestreo que reduce notablemente el problema de la reducción de partículas.

La instrucción que ejecuta el paquete es:

```
$ roslaunch gmapping slam_gmapping scan:=base_scan
```

para usar slam_gmapping necesitamos un robot móvil que proporcione datos de odometría y que además esté equipado con un telémetro láser. El nodo slam_gmapping recibe un mensaje tipo sensor_msgs/laserScan y construye un mapa como mensaje tipo nav_msgs/OccupancyGrid. El mapa se recupera via ROS mediante el topic o service correspondiente.

6. Investigue qué son los campos potenciales y explique los pasos generales para implementarlos.

Consiste en calcular campos imaginarios de repulsión que emanan de los obstáculos. Los campos pueden variar de acuerdo con la distancia del obstáculo o geométricamente de acuerdo con una definición; también se pueden imponer límites de influencia de éstos para no tener que calcular los campos de los obstáculos distantes. El proceso consiste en encontrar un camino que se mantenga tan alejado de los obstáculos como sea posible.

Considera al robot como una partícula bajo la influencia de un campo potencial artificial, cuyas variaciones modelan el espacio libre. La función potencial U en un punto p del espacio euclídeo, se define sobre el espacio libre y consiste en la composición de un potencial atractivo $U_a(p)$, que atrae al robot hacia la posición destino, y otro repulsivo $U_r(p)$ que lo hace alejarse de los obstáculos, es decir:

$$U(p) = U_a(p) + U_r(p)$$

La fuerza artificial $F(p)$ a la que afecta el vehículo en la posición p , por el potencial artificial $U(p)$ resulta:

$$F(p) = -\nabla U(p)$$

Al igual que la función potencial, la fuerza artificial es el resultado de la suma de una fuerza de atracción $F_a(p)$, proveniente de la posición destino, y otra fuerza de repulsión $F_r(p)$ debidas a los obstáculos del entorno de trabajo:

$$F(p) = F_a(p) + F_r(p)$$

- i) Calcular el potencial $U(p)$ que actúa sobre el vehículo en la posición actual p según la información recabada de los sensores.
- ii) Determinar el vector fuerza artificial $F(p)$ según la expresión.
- iii) En virtud del vector calculado construir las consignas adecuadas para los actuadores del vehículo que hagan que éste se mueva según el sentido, dirección y aceleración especificadas por $F(p)$.

7. Explique qué es una transformación homogénea y para qué se utiliza en robots móviles.

Consisten en matrices de 4×4 que nos van a dar el cambio completo de orientación y de traslación relacionando dos sistemas de coordenadas, es decir Representan la transformación de un vector de coordenadas homogéneas de un sistema a otro. Algunas de sus características de estas matrices son:

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.

Dicha matriz se haya compuesta por 4 sub-matrices:

- 1.- $R_{3 \times 3}$, matriz de rotación.
- 2.- $p_{3 \times 1}$, vector de traslación.
- 3.- $f_{1 \times 3}$, transformación de perspectiva.
- 4.- $w_{1 \times 1}$, escalado global.

$$T = \begin{bmatrix} R_{3 \times 3} & p_{3 \times 1} \\ f_{1 \times 3} & w_{1 \times 1} \end{bmatrix} = \begin{bmatrix} \text{Rotación} & \text{Traslación} \\ \text{Perspectiva} & \text{Escalado} \end{bmatrix}$$

En robots móviles la submatriz $f_{1 \times 3}$ es nula, y la submatriz $w_{1 \times 1}$ que representa un escalado global es la unidad:

Se utilizan para obtener las relaciones cinemáticas entre las variables que componen el robot. Para que un robot pueda manipular una pieza, es necesario conocer su localización, es decir la posición y orientación de ésta con respecto a la base del robot lo cual permite un posterior modelado del Robot en su conjunto. Por ejemplo, en un brazo industrial la manipulación de la pieza llevada a cabo por el robot implica el movimiento espacial de su extremo, lo cual hace que surja una necesidad de una herramienta matemática para especificar la posición y orientación del extremo del robot respecto la base del robot

8. Investigue qué es un robot con restricciones no holonómicas de movimiento.

Las restricciones holonómicas son aquéllas que son expresadas como un sistema de ecuaciones algebraicas con las variables de posición, tanto traslacionales como rotacionales, hacen que el sistema sea integrable sin involucrar ninguna variable de velocidad. También se puede constituir un sistema de ecuaciones con variables de velocidad que logren integrarse, así de conducir el sistema de restricciones a uno con variables de posición.

La limitación de la maniobrabilidad se conoce como una restricción no holonómica, La restricción no holonómica se expresa como una restricción diferencial sobre el movimiento del móvil. Esta expresión varía según la elección del sistema de coordenadas.

Si el sistema de restricciones de velocidad no es integrable, se dice que las restricciones son no holonómicas. Por tanto, si un sistema mecánico está sujeto sólo a restricciones holonómicas, se dice, que el sistema es holonómico; de lo contrario, es no holonómico. Los manipuladores compuestos de pares prismáticos y de revolución son ejemplos de sistemas holonómicos, mientras que los robots móviles normalmente constituyen sistemas no holonómicos.

Ejemplo: Configuración Ackermann (usada en automóviles).

Debido a las restricciones no-holonomicas:

- En robótica móvil se usa cinemática diferencial
- Se hace transformaciones de velocidades (en lugar de posiciones)
- En general no se puede obtener relaciones de posición (la velocidad no es en general integrable)

Bibliografía

Martínez, S., & Sisto, R. (31 de mayo de 2020). *fing*. Obtenido de

https://www.fing.edu.uy/inco/grupos/mina/pGrado/easyrobots/doc/Campos_potenciales2_0.pdf

Acosta, I. (s.f.). Robot Móvil para investigación en algoritmos de planeamiento de rutas: sistemas de planeamiento de rutas. 1-7.

Del Pozo Quintero, A. (2012). Modelo Cinemático Dinámico del Mini Robót Móvil Ricimaf. *EAC*.

Fuentes, R. A. (octubre de 2011). Obtenido de <https://core.ac.uk/download/pdf/30045307.pdf>

Gonzalez, D., Perez, J., Milan es, V., & Nashashibi, F. (2015). A review of motion planning techniques for automated vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 17(4), 1135-1145.

Martínez Carranza, J. (mayo de 2020). *inaoep*. Obtenido de <https://ccc.inaoep.mx/~carranza/docs/introb/s3.pdf>

Mecafenix, F. (25 de febrero de 2019). *ingmecafenix*. Obtenido de [ingmecafenix](http://ingmecafenix.com).

Muñoz Ramos, O. (20 de diciembre de 2007). Obtenido de

http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/munoz_r_o/capitulo4.pdf

Murphy R.R. (2000) Introduction to AI Robotics. MIT Press.

Paden, B., Cap, M., Yong, S. Z., Yershov, D., & Frazzoli, E. (2016). A survey of motion planning and control techniques for self-driving urban vehicles. *IEEE Transactions on intelligent vehicles*, 1(1),33-55.