

Examen Parcial

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

Nombre: Luis Ignacio Viñas Petriz

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

Fecha límite de entrega: 04 de junio de 2020.

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

La configuración de un robot ~~móvil~~ es la configuración es la posición y orientación de un cuerpo, en este caso un robot en el espacio. *Esto es solo un caso particular.*

El espacio de configuraciones es el espacio "n" - dimensional donde se ubica cada grado de libertad del robot, el robot (órgano terminal) se puede ~~ver~~ como un punto en este espacio. Los grados de libertad de un robot móvil se refiere a los posibles movimientos de un robot (x, y, z y rotaciones). Para los manipuladores del robot, ~~cada articulación~~ provee un grado de libertad (se requieren 6 para ubicar un objeto rígido en cualquier posición y orientación)

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

Diagramas de Voronoi:

Formalmente se definen como una retracción con preservación de la continuidad. Si el conjunto C_i define las posiciones libres de obstáculos de un entorno, la función retracción RT construye un subconjunto C_v continuo de C_i .

$$RT(q): C_l \rightarrow C_v / C_v \subset C_l$$

De esta forma, se dice que existe un camino desde una configuración inicial q_a hasta otra final q_f , supuestas ambas libres de obstáculos, si ~~y~~ solo si existe una curva continua desde $RT(q_a)$ hasta $RT(q_f)$.

La idea fundamental de los Diagramas de Voronoi es ampliar al máximo la distancia entre el camino del robot y los obstáculos. Por ello, el diagrama de Voronoi resulta del lugar geométrico de las configuraciones que se encuentran a igual distancia de los dos obstáculos más próximos del entorno. El diagrama estará formado por dos tipos de segmentos: rectilíneos y parabólicos.

La elección de la modalidad de segmento corresponde con la clase de elementos de los obstáculos más cercanos que se encuentren enfrentados entre sí. De esta forma, el lugar

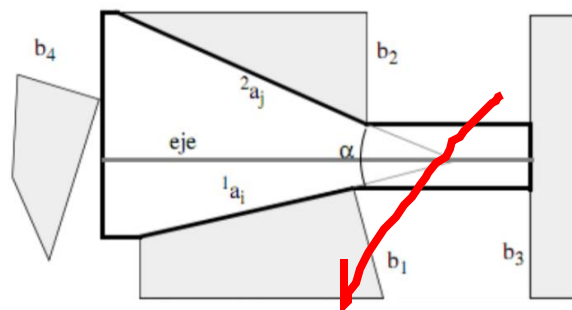
geométrico de las configuraciones que se hallan a igual distancia de dos aristas de dos obstáculos diferentes es una línea recta, mientras que en el caso de tratarse de un vértice y una arista resulta una parábola.

Modelado del espacio libre:

Igualmente, en este método los obstáculos se los representa como polígonos. La planificación se lleva a cabo a través de los CRG (cilindros rectilíneos generalizados) y al igual que Voronoi, con el uso de los CRG se pretende que el robot se mueva lo más alejado de los obstáculos. La ruta será una configuración de CRG interconectados, tal que la configuración inicial o de partida se encuentre en el primer cilindro de la sucesión y la configuración final en el último cilindro.

El proceso para construir un CRG será el siguiente:

Cálculo del eje del CRG, se define como la bisectriz del ángulo α formado por el corte de las rectas que contienen las aristas.



Por ambos lados de dichas aristas se construyen rectas paralelas al eje, con origen en los vértices de las aristas implicadas y con extremo señalado por la proyección del primer obstáculo que corta el eje.

3. Investigue dos métodos basados en muestreo para planeación de rutas. PRM es Probabilistic RoadMap
Hoja de ruta probabilística, por sus siglas en inglés PRM (Rapidly-exploring Random Tree):

Es un algoritmo de planificación de movimiento en robótica, que resuelve el problema de determinar un camino entre una configuración inicial del robot y una configuración de objetivos, evitando colisiones.

La idea básica detrás de PRM es tomar muestras aleatorias del espacio de configuración del robot, probarlas para determinar si están en el espacio libre y utilizar un planificador local para intentar conectar estas configuraciones a otras configuraciones cercanas. Se agregan

las configuraciones de inicio y objetivo, y se aplica un algoritmo de búsqueda de gráfico al gráfico resultante para determinar una ruta entre las configuraciones de inicio y objetivo. La idea básica detrás de PRM es tomar muestras aleatorias del espacio de configuración del robot, probarlas para determinar si están en el espacio libre y utilizar un planificador local para intentar conectar estas configuraciones a otras configuraciones cercanas. Se agregan las configuraciones de inicio y objetivo, y se aplica un algoritmo de búsqueda de gráfico al gráfico resultante para determinar una ruta entre las configuraciones de inicio y objetivo.

Árbol aleatorio de exploración rápida, por sus siglas en inglés RRT (Rapidly-exploring Random Tree):

Es un algoritmo diseñado para buscar eficientemente espacios no convexos y de alta dimensión mediante la construcción aleatoria de un árbol que llena espacios. El árbol se construye de forma incremental a partir de muestras extraídas al azar del espacio de búsqueda y está sesgado inherentemente para crecer hacia grandes áreas no buscadas del problema. Los RRT fueron desarrollados por Steven M. LaValle y James J. Kuffner Jr. Manejan fácilmente los problemas con obstáculos y restricciones diferenciales (no holonómicas y kinodinámicas) y han sido ampliamente utilizados en la planificación autónoma del movimiento robótico.

Los RRT pueden verse como una técnica para generar trayectorias de bucle abierto para sistemas no lineales con restricciones de estado. Un RRT también se puede considerar como un método de Monte-Carlo para sesgar la búsqueda en los diagramas de Voronoi más grandes de un gráfico en un espacio de configuración. Algunas variaciones incluso pueden considerarse fractales estocásticos.

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

El mapeo y localización simultáneos o SLAM (por sus siglas en inglés Simultaneous Localization and Mapping), es una técnica usada por robots y vehículos autónomos para construir un mapa de un entorno desconocido en el que se encuentra, a la vez que estima su trayectoria al desplazarse dentro de este entorno.

SLAM investiga los problemas que plantea la construcción de modelos matemáticos, geométricos o lógicos de entornos físicos, empleando como herramienta un robot móvil —

en ocasiones varios de ellos— y el conjunto de sensores y actuadores que lo conforman. Dicho de otra manera, el SLAM busca resolver los problemas que plantea el colocar un robot móvil en un entorno y posición desconocidos, y que él mismo sea capaz de construir incrementalmente un mapa consistente del entorno al tiempo que utiliza dicho mapa para determinar su propia localización.

Desde un punto de vista conceptual se impone la necesidad de razonar en un mundo a veces confuso, en ocasiones dinámico y cambiante, aprehendido mediante sensores que distan mucho de ser perfectos. En estas condiciones se busca la manera de obtener y manipular datos acerca del entorno, extraer aquel conocimiento que sea sustancial para la tarea de su representación, e integrar la información así obtenida del modo más conveniente. Así pues, en la base de cualquier solución al problema del SLAM se encuentra siempre con la necesidad de trabajar con cantidades progresivamente crecientes de información contaminada en mayor o menor medida por ruido y manipulada mediante modelos que, la mayoría de las veces, no son sino meras aproximaciones a la realidad. No es de extrañar, por lo tanto, que las soluciones más exitosas hasta el momento hayan estado basadas en la utilización de técnicas probabilísticas.

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

La localización mediante filtros de partículas es un método empleado para estimar el estado de un sistema que cambia a lo largo del tiempo. Más concretamente, es un método de Montecarlo(secuencial) usado comúnmente en visión artificial para el seguimiento de objetos en secuencias de imágenes.

Fue propuesto en 1993 por N. Gordon, D. Salmond y A. Smith como filtro bootstrap para implementar filtros bayesianos recursivos. Básicamente, el filtro de partículas se compone de un conjunto de muestras (las partículas) y unos valores, o pesos, asociados a cada una de esas muestras. Las partículas son estados posibles del proceso, que se pueden representar como puntos en el espacio de estados de dicho proceso.

Posee cuatro etapas principales:

Inicialización.

Actualización.

Estimación.

Predicción.

Para realizar el seguimiento de un objeto sobre una secuencia de imágenes, el filtro de partículas "lanza" al azar un conjunto de puntos sobre la imagen (etapa de inicialización, se crea un conjunto de partículas con un estado aleatorio), realizando cálculos se le asignará un valor, o valores, a cada uno de esos puntos (etapa de actualización). A partir de estos valores, se creará un nuevo conjunto de puntos que reemplazará al anterior. Esta elección también será al azar, pero los valores que se han adjudicado a cada uno de los puntos provocarán que sea más probable de elegir aquellos puntos que hayan capturado al objeto sobre el que quiere realizar el seguimiento (etapa de estimación). Una vez que se crea el nuevo conjunto de puntos, se realiza una leve modificación al estado (posición) de cada uno de ellos, con el fin de estimar el estado del objeto en el instante siguiente (etapa de predicción).

Al terminar la etapa de predicción, se obtiene un nuevo conjunto de puntos al que se le vuelve a aplicar la etapa de actualización, repitiéndose este bucle hasta que termine la secuencia o desaparezca el objeto, caso en el cual se volvería a la etapa de inicialización.

La localización mediante filtros de partículas se utiliza para resolver el Hidden Markov Model (HMM) y los problemas de filtrado no lineal, cosa que no es posible resolver con los modelos de observación de señal lineal-gaussiana (filtro de Kalman) o clases más amplias de modelos (filtro de Benes).

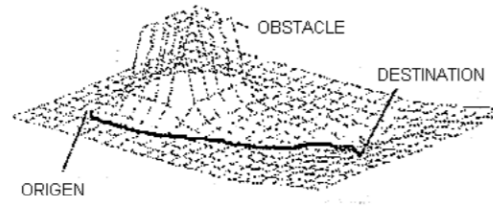
Para implementar el método de localización mediante filtros de partículas se requiere de la librería bfl, este filtro de partículas se usará para rastrear la pose de un robot contra un mapa conocido. Igualmente se requerirá crear un paquete que dependerá de `ros_cpp`, `std_msgs`, `nav_msgs`, `geometry_msgs`, `ardrone_autonomy`, y finalmente `bfl`.

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

En física, un campo de fuerza es un campo vectorial que describe una fuerza sin contacto que actúa sobre una partícula en varias posiciones en el espacio. Específicamente, un campo de fuerza es un campo vectorial \vec{F} , donde $\vec{F}(\vec{x})$ es la fuerza que sentiría una partícula si estuviera en el punto (\vec{x}) .

Se modela el robot como una partícula que está bajo la influencia de dos campos potenciales, uno lo atrae hacia el destino, mientras que otro lo aleja de los obstáculos. Los obstáculos ahora son como montañas que ejercen fuerzas de repulsión, mientras que el destino es

como si fuera un hoyo. El robot se mueve a través de un campo potencial por la pendiente más pronunciada hasta que lo lleva al destino.



El robot está bajo la influencia de un campo potencial el cual es la suma de dos campos potenciales, el destino determina un campo potencial de atracción y los obstáculos generan campos potenciales repulsivos. Dado que no se conoce la ecuación del campo potencial total, entonces se utiliza un método iterativo. Empezando con un punto x_{n-1} cualquiera, el objetivo es llegar al punto mínimo de la ecuación x_0 utilizando una función recurrente:

$$x_n = f(x_{n-1})$$

Específicamente para la posición del robot en el medio ambiente:

$$q_n = [x_n, y_n]$$

El campo potencial ejercido sobre el robot en ese punto es:

$$U(q) = U_{\text{atracción}}(q) + U_{\text{repulsión}}(q)$$

El gradiente del campo potencial

$$\bar{F}(\bar{q}) = \nabla U(\bar{q}) = \left(\frac{\partial U}{\partial x} \hat{i} + \frac{\partial U}{\partial y} \hat{j} \right)$$

Usando la técnica de "steepest descent" la siguiente posición del robot está dada por:

$$\bar{q}_n = \bar{q}_{n-1} - \delta_i \bar{f}(\bar{q}_{n-1})$$

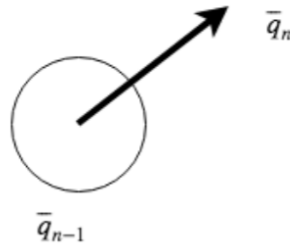
Donde δ_i son constantes que determinan el tamaño de avance del robot, en ocasiones se mantiene fija.

$\bar{f}(\bar{q}_{n-1})$ es un vector fuerzas unitario en la dirección del gradiente

$$\bar{f}(\bar{q}_{n-1}) = \frac{\bar{F}(\bar{q}_{n-1})}{|\bar{F}(\bar{q}_{n-1})|}$$

El robot se mueve siguiendo la pendiente más pronunciada del campo potencial, dada por las fuerzas de atracción y repulsión

$$\begin{aligned} F(\bar{q}_{n-1}) &= F_{\text{atr}}(\bar{q}_{n-1}) + F_{\text{rep}}(\bar{q}_{n-1}) \\ \bar{q}_n &= \bar{q}_{n-1} - \delta_i \bar{f}(\bar{q}_{n-1}) \end{aligned}$$



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

Una matriz de transformación homogénea es un operador que soporta la transformación rotacional y la transformación traslacional de manera integrada, además de escalado y perspectiva. Son matrices de 4x4 que transforman un vector expresado en coordenadas homogéneas desde un sistema de coordenadas hasta otro sistema de coordenadas. Consisten en 4 submatrices.

$$T = \begin{bmatrix} Rotacion_{3 \times 3} & Posicion_{3 \times 1} \\ perspectiva_{1 \times 3} & escalado_{1 \times 1} \end{bmatrix}$$

Las transformaciones homogéneas y las matrices homogéneas son necesarias para la robótica móvil para representar e identificar la posición y orientación del robot, ya que podrá identificar un sistema móvil entre el robot y el sistema fijo de su entorno, además de identificar la traslación y rotación entre cada uno de sus manipuladores.

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

Las restricciones holonómicas son las relaciones entre las variables de posición (y posiblemente de tiempo) que se pueden expresar de la siguiente forma:

$$f(q_1, q_2, q_3, \dots, q_n, t) = 0$$

Donde son las n coordenadas que describen el sistema. Por ejemplo, el movimiento de una partícula limitado a tumbarse en la superficie de una esfera está sujeta a una restricción holonómica, pero si la partícula es capaz de caer de la esfera bajo la influencia de la gravedad, la restricción se convierte en no holonómica.

Por lo tanto, un robot tiene restricciones no holonómicas cuando su estado depende del camino tomado para lograrlo. Dicho sistema se describe mediante un conjunto de

parámetros sujetos a restricciones diferenciales, de modo que cuando el sistema evoluciona a lo largo de una ruta en su espacio de parámetros (los parámetros varían continuamente en valores) pero finalmente regresa al conjunto original de valores de parámetros al comienzo de la ruta, el sistema en sí mismo puede no haber vuelto a su estado original.

Por ejemplo, se sabe que los robots móviles son no holonómicos, es decir, están sujetos a restricciones no holonómicas de igualdad no integrables que involucran la velocidad.

Fuentes:

<https://ccc.inaoep.mx/~esucar/Clases-ia/Laminas2014/intro-robotica.pdf>

<https://porprofesionalmic.files.wordpress.com/2015/09/investigacion-documental-navegacion-planificacion-rutas.pdf>

<http://wiki.ros.org/bfl/Tutorials/Example%20of%20using%20a%20particle%20filter%20for%20localization%20by%20bfl%20library>

https://biorobotics.fi-p.unam.mx/wp-content/uploads/Courses/robots_moviles/2020_2/lecciones/robots_moviles_2020_2_leccion_3.pdf

<https://web2.qatar.cmu.edu/~gdicaro/16311-Fall17/slides/16311-4-Holonomy-WheeledRobots-Kinematics.pdf>