

## Examen Parcial

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

**Nombre: Sebastián Reyes Flores**

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

### Examen Parcial

1. Explique qué es la configuración, espacio de configuraciones y grados de libertad de un robot móvil. ....1
2. Investigue dos métodos basados en grafos para planeación de rutas. ....2
3. Investigue dos métodos basados en muestreo para planeación de rutas. ....3
4. Explique en qué consiste el proceso de SLAM (Simultaneous Localization and Mapping). ....4
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. ....5
6. Investigue qué son los campos potenciales y explique los pasos generales para implementarlos. ....6
7. Explique qué es una transformación homogénea y para qué se utiliza en robots móviles. ....7
8. Investigue qué es un robot con restricciones no holonómicas de movimiento. ....7

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

Una configuración, es una descripción completa del estado de un robot, que indica su posición, orientación y entorno en un cierto momento. Además, describe el estado interno de cada articulación (en caso de tener) dentro de un cierto espacio de trabajo. El concepto de configuración se puede representar como un punto en el espacio  $n$ -dimensional, en donde el número de dimensiones  $n$ , corresponde al número de grados de libertad del robot.

El espacio de configuraciones se define como el conjunto de todas las transformaciones posibles que pueden ser aplicadas a un robot, independiente de si existiera alguna obstrucción, y esta más relacionado a la cantidad de estados en los que se puede encontrar un cierto robot. Hoy en día, este concepto, también conocido como  $C$ -Space, es ampliamente utilizado en el contexto de planeación de rutas. En general, un espacio de configuraciones  $n$ -dimensional, puede usarse para modelar cualquier sistema que pueda describirse mediante  $n$  parámetros. En el caso de un robot, estos  $n$  parámetros describen su configuración.

Se denominan grados de libertad a cada una de las coordenadas independientes que son necesarias para describir el estado del sistema mecánico del robot (posición y orientación). Generalmente cada par eslabón-articulación de un robot le corresponde únicamente un grado de libertad, pero como tal una sola articulación podría tener 2 o más grados de libertad. Dentro del espacio tridimensional, para que exista un grado de libertad es necesario que, a lo largo de cada uno de los ejes, el movimiento sea independiente a los otros e independiente a la rotación de cualquiera de estos. Los seis grados de libertad del movimiento tridimensional son: delante/atrás (x), arriba/abajo (z), izquierda/derecha (y), cabecear (pitch), guñar (yaw) y rodar (roll).

Finalmente, dentro de la clasificación de los robots móviles, generalmente, se usan el movimiento en el plano x-y y se rota en el eje z.

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

### Dijkstra

Probablemente sea el algoritmo más conocido y del cual se derivan la mayoría de los algoritmos de planeación de rutas. Este algoritmo, también llamado algoritmo del caminos mínimos, es un algoritmo basado en grafos que nos permite conocer el camino más corto a cada punto dado un origen.

Para la utilizar este algoritmo se debe tener un grafo ponderado donde cada arista sea la distancia entre los nodos adyacentes. Posteriormente, en un vector del tamaño del número de nodos totales se guardarán las distancias finales a cada nodo a partir del nodo de origen en donde se le asignara un valor infinito relativo a cada uno de dichas distancias dado que es desconocido al inicio. Se recorrerá de forma iterativa cada nodo adyacente al actual para conocer la distancia mínima a partir de dicho nodo origen, se guardará la distancia mínima y se procederá a moverse al nodo que tenga la distancia más corta y marcando el nodo anterior como visitado. En cada paso se sumará la distancia que recorrió para llegar al nodo en el que se encuentra más la distancia que le recorrerá al nodo siguiente y lo compara con el valor que se encuentra en la matriz de distancia finales y guardará la distancia mínima como nuevo valor. Este proceso se repetirá hasta que el algoritmo haya visitado cada nodo y ya no haya nodos adyacentes que verificar y es aquí donde se encontrará la distancia mínima a cada nodo a partir del nodo de origen.

Cabe destacar que en este algoritmo no existen distancias o costes negativos y el nivel de complejidad del algoritmo aumentara conforme aumente la cantidad de nodos, es decir, el nivel de complejidad es igual al número de nodos al cuadrado. Además, se puede usar hasta llegar a un nodo destino o hasta recorrer todos los nodos.

### A\*

En realidad es  $\text{Nodos} * \text{Aristas}$

El algoritmo A\* es bastante parecido al algoritmo de Dijkstra, pero agrega una característica fundamental

que permite encontrar el camino más corto de manera más rápida y esto es el uso de una función heurística para ponderar que tan cercano se encuentra cierto nodo al objetivo, teniendo el objetivo un costo de 0 otorgado por dicha función heurística. En caso de que dicho valor fuera igual para todos los nodos, este algoritmo funcionaría exactamente igual que el de Dijkstra.

De igual forma, se necesita un grafo ponderado donde cada arista representara la distancia entre nodos adyacentes, pero en este caso, cada nodo contara con el valor otorgado por la función heurística previamente mencionada. Lo que realiza este algoritmo es que, a cada paso y con cada nodo adyacente, se obtiene un cierto costo que es la suma de la distancia desde el nodo de origen hasta el nodo al que se desea llegar más el costo asignado por la función heurística a dicho nodo. De esta forma se recorrerá el grafo siguiendo el camino más rápido y corto para llegar a un objetivo.

La complejidad computacional de este algoritmo está completamente ligada a la calidad en la asignación de los costos de la función heurística a cada nodo, es decir, si se asignan pobremente los costos de esta función en los nodos, esto significaría que se recorrerán una mayor cantidad de nodos y por ende se deberán realizar una mayor cantidad de sumas y comparaciones para llegar a un objetivo.

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

Este tipo de métodos busca resolver las restricciones de tiempo que los métodos deterministas conllevan tal como la planeación en grandes dimensiones del espacio. Los métodos más conocidos y usados en la robótica son los métodos PRM (Probabilistic Roadmap Method) y el RRT (Rapidly Exploring Random Tree)

#### **PRM (Probabilistic Roadmap Method)**

El método PRM (Probabilistic Roadmap Method) es un algoritmo de planificación de movimiento en robótica, que resuelve el problema de determinar una ruta entre una configuración inicial del robot y una configuración final, evitando colisiones. El PRM se construye a partir de un conjunto de configuraciones muestreadas desde el espacio de configuraciones y lleva en el nombre "mapa vial" (roadmap) porque, dado que el número de muestras tiende al infinito, la probabilidad de que la gráfica parezca un mapa vial aumentaría demasiado.

Este método, aun cuando usa grafos para su desarrollo, es considerado un método basado en el muestreo, ya que al inicio de este se generan un conjunto de  $N$  muestras aleatorias del espacio de configuraciones del robot (las cuales no generen colisiones) que posteriormente pasarán a ser los nodos en los que se realizará el algoritmo de búsqueda basado en grafos.

El PRM generalmente cuenta con 2 etapas, una etapa de aprendizaje y una etapa de consulta. En la etapa de aprendizaje se realiza el muestreo aleatorio del espacio de configuraciones mencionado anteriormente y

uniéndolas de tal forma que se construya el mapa (que posteriormente pasara a ser un grafo) por el cual se buscara moverse el robot sin colisionar. Mientras que, en la etapa de consulta, el algoritmo de cierta manera pregunta si puede existir un camino que una dos configuraciones encontradas en la etapa anterior, siempre comenzando por preguntar si existe un camino entre el punto de partida y la meta, y en caso de no existir comenzara a unir los caminos posibles hasta llegar a un objetivo. Estas dos etapas se realizarán de forma iterativa, lanzando nuevas muestras con las cuales generar un mapa más exacto con mejor opciones para llegar a un objetivo.

### **RRT (Rapidly Exploring Random Tree)**

El método RRT (Rapidly-Exploring Random Tree) está de cierta manera basado en el PRM siendo expuesto por primera vez en 1998, dos años después que el PRM. El método RRT es un algoritmo de planificación probabilístico, basado en modelo estático que construye un único grafo unidireccional en forma de árbol, este parte desde el punto inicial y se expande por todo el entorno de trabajo mediante un proceso de muestreo en el que busca puntos aleatorios hasta llegar al punto final, momento en el que se detiene.

El algoritmo recibe como entradas los parámetros del planificador: configuración inicial y final, número de nodos máximos (iteraciones máximas del bucle for) y el salto entre nodos (distancia en píxeles que medirán las ramas del árbol). Como salida devolverá el árbol en sí.

El algoritmo RRT hace crecer un árbol enraizado en la configuración inicial mediante el uso de muestras aleatorias del espacio de búsqueda. A medida que se extrae cada muestra, se intenta una conexión entre esta y el estado más cercano en el árbol. Si la conexión es factible (pasa completamente a través del espacio libre y obedece cualquier restricción), esto resulta en la adición del nuevo estado al árbol. Con un muestreo uniforme del espacio de búsqueda, la probabilidad de expandir un estado existente es proporcional al tamaño de su región de Voronoi (la cual es una región libre de obstáculos). Como las regiones más grandes de Voronoi pertenecen a los estados en la frontera de la búsqueda, esto significa que el árbol se expande preferentemente hacia grandes áreas no buscadas. Al utilizar el nombre de árbol, generalmente este algoritmo tiende a usar los nombres de padre e hijo refiriéndose a los nodos origen y destino respectivamente.

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

El SLAM es una técnica que tiene como propósito que un robot o vehículo autónomo pueda construir mapas en un entorno desconocido y al mismo tiempo pueda desplazarse estimando una trayectoria utilizando dicho mapa.

Para la implementación de esta técnica es necesario el uso de sensores que permitan extraer información

del entorno y así conocer la ubicación del robot en función de un conjunto de puntos característicos del entorno (landmarks) que se emplearán como referencia, estos puntos deben de ser reobservables y fácil de distinguir para el robot.

Posteriormente, la técnica implementa un algoritmo de localización del robot donde se estimará la odometría en cada instante (generalmente llamado estado). Este algoritmo debe dar una solución de forma probabilista que sea calculada en cada instante en función del instante anterior (haciendo uso del teorema de Bayes) para así plantear la trayectoria que se seguirá.

El algoritmo más estudiado y conocido es el Filtro extendido de Kalman, pero también existen otros métodos como el filtro de partículas o el filtro gráfico y generalmente cumplen con un etapa de predicción y otra de actualización de la odometría que se repetirá de manera recursiva en cada momento durante el transcurso de la técnica, generalmente hasta llegar a un objetivo.

El problema principal de estos métodos generalmente va ligado a un error en la estimación de la posición que esta ligada a las bases probabilísticas que tienen los algoritmos con los que funciona el SLAM.

## 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 que se utiliza dentro de los robots móviles y vehículos autónomos para estimar el estado de dicho robot en cierto ambiente a lo largo de un periodo de tiempo. Este método utiliza la visión artificial para identificar objetos en una secuencia de imágenes y darles un seguimiento. El filtro de partículas se basa en el método de Montecarlo el cual se encarga de resolver problemas que no tienen una solución estrictamente numérica o analítica, sino que tienen un fuerte componente aleatorio. Este método resuelve el problema mediante la simulación de la variable aleatoria y recogiendo las muestras obtenidas.

En el filtro de partículas recogemos un conjunto de muestras (partículas) a las cuales asignaremos unos valores (pesos). Las partículas representan muestras del espacio de estado mientras que los pesos serían indicadores de la importancia que cada una de estas partículas adquiere, todo esto con el fin de aproximar una función de densidad a partir de estas muestras usando ecuaciones de recurrencia bayesiana. En el caso de los robots móviles se utiliza para estimar el ambiente en el que se encuentra el robot en función de que se detecten objetos en un rango visible (o detectable) del robot.

Este método consta de 4 etapas de funcionamiento:

- Inicialización: se lanza por primera vez un conjunto de puntos o "partículas" sobre la imagen.
- Actualización: Se creará un nuevo conjunto de puntos que reemplazará al anterior.
- Estimación: Por medio de un conjunto de cálculos (basados en el método de Montecarlo) se le

asignará un nuevo valor a cada partícula siendo más probable que este segundo conjunto de puntos si pertenezcan al objeto que se está identificando.

- Predicción: 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.

Las etapas de actualización, estimación y predicción se realizarán continuamente para el funcionamiento del método.

Las principales ventajas sobre el filtro de Kalman es que da mejores resultados al localizar y predecir los objetos de un cierto entorno. Esto se debe a que el filtro de Kalman tiene la limitación de que las variables de estado se distribuyen de forma Gaussiana y da pobre resultados a variables que no sigan dicha distribución, mientras que, en el filtrado de partículas, la densidad de estado del sistema en el instante  $t$  es representada por un conjunto de muestras (partículas) con pesos (probabilidad de muestreo) lo cual no va acorde a una cierta distribución.

Algunas de las paqueterías de ROS que implementan los filtros de partículas son *"amcl"*, *"bfl"* o *"mcl\_pi"* aunque el más utilizado para la localización utilizando este método es *"amcl"*. Existen otros paquetes como el *"coreSLAM"* que realiza el método SLAM usando el filtro de partículas.

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

Los campos potenciales es una técnica utilizada dentro de los robots móviles donde se modela un robot como una partícula que esta bajo la influencia de dos campos, uno que lo atrae hacia un cierto objetivo y otro que lo aleja de los obstáculos. Esta técnica consiste en diseñar un campo de energía potencial con máximos en los obstáculos (montañas) y un mínimo en el punto meta (un hoyo). En esta técnica el robot se moverá a través de la pendiente más pronunciada pasando de una posición de mayor energía a uno de menor energía hasta llegar a su destino (energía nula), para lograr esto se utilizará el gradiente.

A grandes rasgos, los pasos generales para implementar un campo potencial serían los siguientes:

1. Identificar la meta creando un campo atractivo con forma de cono donde la meta será la parte de menor energía.
2. Identificar los obstáculos y crear un campo repulsivo (generalmente con forma de cono achatado invertido). Se generará una fuerza de repulsión más fuerte cuanto más cercano esté el robot al obstáculo, de manera que esta se desvanezca a cierta distancia de este.
3. Diseñar el campo de fuerzas por donde se moverá el robot conjuntando ambos campos.
4. Colocar el robot en una posición inicial e implementar con el algoritmo de movimiento utilizando el gradiente descendente. El robot se moverá en dirección de la pendiente más pronunciada donde en cada iteración se realizará la suma de los campos atractivo y repulsivo para conocer la

fuerza local en cada instante y así saber hacia qué dirección se debe mover el robot hasta llegar a una meta.

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

Debido a que los robots son cuerpos rígidos, una transformación es una función que mapea cada partícula de este a un nuevo lugar en el espacio. Una transformación homogénea sirve para representar la posición y orientación de un sistema girado y trasladado OUBW, con respecto a un sistema fijo de referencia OXYZ, refiriéndonos a la descripción de la rotación y traslación realizada sobre un sistema referencia.

Generalmente, una transformación homogénea se realiza a través de una matriz  $4 \times 4$  que representa la transformación de un vector de coordenadas homogéneas de un sistema de coordenadas a otro. Esta matriz está formada por cuatro submatrices las cuales serían la de rotación, traslación, perspectiva y escalado.

En los robots móviles es evidente la necesidad de representar cambios traslacionales, además de rotacionales, dado que en las transformaciones rotacionales únicamente permiten representar cambios de orientación entre dos sistemas de coordenadas diferentes, pero no se hace representar el cambio de posición, es ahí donde surge la importancia de las transformaciones homogéneas dentro de los robots móviles. Es por esto que se hace una simplificación en la matriz de transformación homogénea donde la submatriz que representa la transformación de perspectiva es nula mientras que submatriz de escalado no varía por lo que es la unidad, simplificando a una transformación únicamente de la rotación y la traslación.

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

Un robot con restricciones no holonómicas de movimiento es aquel que no es capaz de moverse en todas las direcciones sin necesidad de rotar (o realizar un movimiento diferente) previamente para llegar a un cierto punto, es decir, no se puede mover de manera instantánea en todas direcciones. El ejemplo más claro podría ser el de un robot de dos ruedas (convencionales) que no puede desplazarse de manera lateral sin antes tener que maniobrar y girar previamente para llegar a una cierta posición.

Esto se debe a que las restricciones no holonómicas de movimiento se refieren, de manera más matemática, a que las restricciones diferenciales de movimiento no son integrables y por ende no pueden ser expresadas como restricciones de posición u orientación. Generalmente, también pueden ser relacionadas con el número de grados de libertad de un robot y la cantidad de estos que puede ser controlada. En el caso de un robot de dos ruedas que se desplaza en un plano, existen 3 grados de libertad, 2 para la posición ( $x, y$ ) y 1 para la orientación. Como únicamente se pueden controlar el desplazamiento en un sentido (dígase  $x$ ) y la orientación en función de las velocidades de las ruedas, esto nos dejaría con un grado de libertad no controlable que sería lo que nos daría la característica de restricción no holonómica de movimiento, ya que tiene menos grados de libertad controlables que número total de grados de libertad.