

Test de Planificación

1. El resultado de un planificador en robótica puede ser:
 - a. Puntos geométricos que ha de seguir el robot para alcanzar un objetivo.
 - b. Tareas a realizar por el robot.
 - c. Secuencia de acciones y percepciones que debe encontrar el robot.
 - d. Todas las anteriores

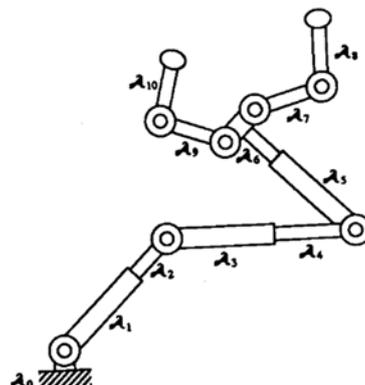
2. La planificación en robótica:
 - a. Se creó para resolver el problema del piano, pero no sirve para casos 3D
 - b. Es aplicable y a la vez utiliza conceptos de Magnetic Oddities Radiation Therapy
 - c. Es aplicable y a la vez utiliza conceptos de inteligencia artificial y teoría de control
 - d. Utiliza la distancia Euclídea como métrica, y por tanto no puede tener en cuenta otras métricas

3. Tipos de modelos de robots y obstáculos:
 - a. Matrices de rotación, ángulos de Euler y de Bryan
 - b. Cinemáticos, geométricos, híbridos
 - c. Observador, controlador, acción
 - d. ROS, Gazebo, URDF

4. En la representación de sólidos:
 - a. La geometría constructiva de sólidos se representan los objetos partiendo de primitivas
 - b. La aproximación por esferas sucesivas permite diferentes niveles de esferas
 - c. El modelado Voxel permite operar con resoluciones diferentes
 - d. Todas las afirmaciones anteriores son correctas

5. El C-Space es:
 - a. Un espacio donde la configuración del robot en un instante dado es un plano
 - b. Un espacio donde la configuración del robot en un instante dado es un segmento
 - c. Un espacio $SE(3)$ donde el robot es un punto dado por las dimensiones del robot
 - d. Un espacio donde se planifica y que difiere según la configuración del robot

6. Especifica el C-Space del siguiente mecanismo:



- a. $T^7 \times R^3$
- b. $SO(3)$
- c. $SE(3)$
- d. R^{10}

7. El problema de la planificación:

- a. Es en general NP-hard y en general se resuelve en tiempo exponencial
- b. Es en general P y en general se resuelve en tiempo polinomial (eficiente)
- c. Es en general P y en general se resuelve en tiempo lineal (muy eficiente)
- d. Es en general P y en general se resuelve en tiempo constante

8. Respecto a algoritmos tipo insecto:

- a. Tangent Bug rodea obstáculos completamente, y Bug 1 trata de seguir una recta
- b. Tangent Bug rodea obstáculos completamente, y Bug 2 trata de seguir una recta
- c. Bug 1 rodea obstáculos completamente, y Bug 2 trata de seguir una recta
- d. Bug 2 rodea obstáculos completamente, y Bug 1 trata de seguir una recta

9. Los grafos de visibilidad:

- a. Uniformizan distancias entre nodos al ajustarlos a la “retícula de visibilidad”
- b. Incrementan la visibilidad de grafos aumentando los nodos que lo componen
- c. Reducen la cantidad de caminos entre nodos si hay obstáculos
- d. Generan nodos a través del muestreo del espacio visible en el C-Space

10. Metodología general de planificación de robots:

- a. Búsqueda en Grafo → Discretización → Paso a C-Space
- b. Búsqueda en Grafo → Paso a C-Space → Discretización
- c. Paso a C-Space → Discretización → Búsqueda en Grafo
- d. Ninguna de las anteriores

11. Los diagramas de Voronoi

- a. Forman parte de la familia “roadmap”, el origen y la meta se “conectan” a éste
- b. Forman parte de la familia “probabilística”, y la métrica determina la estadística
- c. Únicamente pueden generarse utilizando la métrica L1
- d. Únicamente pueden generarse utilizando la métrica L2

12. Respecto a los algoritmos de búsqueda en grafo:

- a. Breadth-first y Dijkstra realizan una búsqueda exhaustiva (en anchura)
- b. Breadth-first y best-first realizan una búsqueda exhaustiva (en anchura)
- c. Depth-first y Dijkstra realizan una búsqueda “greedy” (en profundidad)
- d. Depth-first y breadth-first realizan una búsqueda “greedy” (en profundidad)

13. En una matriz 3x3 sin obstáculos, con salida y meta en extremos opuestos (casilla “1,1” vs casilla “3,3”), y sólo se permiten desplazamientos de 1 paso horizontal o vertical:

- a. Un algoritmo “greedy” (con una implementación que permite repetir casillas ya visitadas) puede acabar creando más o menos nodos que un algoritmo “exhaustivo”
- b. Un algoritmo “greedy” (con una implementación que permite repetir casillas ya visitadas) acabará creando siempre más nodos que un algoritmo “exhaustivo”
- c. Un algoritmo “greedy” (con una implementación que permite repetir casillas ya visitadas) acabará creando siempre menos nodos que un algoritmo “exhaustivo”
- d. Un algoritmo “greedy” (con una implementación que permite repetir casillas ya visitadas) jamás pasará por la casilla central “2,2”.

14. En una matriz 3x3 sin obstáculos, con salida y meta en extremos opuestos (casilla “1,1” vs casilla “3,3”), y sólo se permiten desplazamientos de 1 paso horizontal o vertical:
- Un algoritmo “exhaustivo” jamás pasará por la casilla central “2,2”.
 - Un algoritmo “exhaustivo” no siempre pasará por la casilla extrema “1,3”.
 - Un algoritmo “exhaustivo” puede utilizar menos de 5 nodos para hallar solución.
 - Un algoritmo “exhaustivo” necesita utilizar más de 5 nodos para hallar solución.
15. Los algoritmos de muestreo (probabilísticos):
- No son probabilísticamente completos, sino totalmente completos; y a cambio, pierden eficiencia con respecto a los deterministas
 - No son probabilísticamente completos, sino totalmente completos; y a cambio, ganan eficiencia con respecto a los deterministas
 - No son totalmente completos, sino probabilísticamente completos; y a cambio, pierden eficiencia con respecto a los deterministas
 - No son totalmente completos, sino probabilísticamente completos; y a cambio, ganan eficiencia con respecto a los deterministas
16. Uno de los mayores problemas de los algoritmos de muestreo (probabilísticos)
- Pueden no terminar si no existe solución
 - No se pueden aplicar en el caso de robots no holónomos
 - Resultan ineficientes en general
 - No son válidos para un número grande de grados de libertad
17. Probabilistic RoadMaps (PRM) y Rapidly-exploring Random Tree (RRT)
- PRM trabaja en el espacio cartesiano mientras que RRT lo hace en C-space
 - RRT es un algoritmo completo mientras que PRM no lo es
 - RRT utiliza la métrica euclídea mientras que PRM utiliza L_∞
 - El espacio se explora en preprocesado (“offline”) en PRM, y “online” en RRT
18. Fast Marching Method (FMM) tiene como ventaja frente a otros métodos
- Trayectorias derivables
 - Trayectorias en el C-Space
 - Trayectorias rectilíneas
 - Trayectorias en $SO(3)$
19. La diferencia entre Fast Marching Method (FMM) y Fast Marching Square (FM²)
- FM² es más óptimo que FMM
 - FM² es completo pero FMM no
 - FM² calcula un camino más suave que FMM
 - FM² no tiene mínimos locales, FMM puede tenerlos
20. En Fast Marching Learning
- Se puede no encontrar una solución aunque exista
 - Necesita al menos 3 experiencias enseñadas para poder encontrar soluciones
 - No tiene en cuenta las velocidades de las experiencias enseñadas
 - Se ejecuta sin necesidad de configurar ningún parámetro