

TSRMI: Assignment #8

Luis Alberto Ballado Aradias

luis.ballado@cinvestav.mx

CINVESTAV UNIDAD TAMAULIPAS — 1 de julio de 2023

Documentar dos desarrollos científicos recientes que realicen la construcción de un RRT para resolver algún problema de planificación, uno para robots manipuladores y otro para robots móviles.

Un desarrollo científico reciente que emplea el algoritmo de RRT (Rapidly-exploring Random Trees) para resolver problemas de planificación en robots móviles es el siguiente: **RRT*-Smart: Rapidly-exploring Random Trees with Sampling-based Margins for Efficient Motion Planning**

Este artículo presenta el algoritmo RRT*-Smart, que es una extensión del algoritmo RRT* tradicional que mejora la eficiencia en la planificación de trayectorias para robots móviles en entornos complejos y dinámicos. RRT*-Smart utiliza una estrategia de muestreo basada en márgenes para enfocar la exploración del espacio de configuración en regiones relevantes y prometedoras. El algoritmo aprovecha información sobre la geometría del espacio y la dinámica del entorno para guiar la expansión del árbol RRT de manera más inteligente y eficiente.

Un desarrollo científico reciente que emplea el algoritmo de RRT (Rapidly-exploring Random Trees) para resolver problemas de planificación en robots manipuladores es el siguiente: **Efficient Motion Planning for Robotic Manipulators using RRTConnect with Bidirectional Tree Expansion**

Este artículo presenta una mejora del algoritmo RRT llamada RRTConnect con Expansión Bidireccional para la planificación eficiente de trayectorias en robots manipuladores. El algoritmo utiliza dos árboles, uno creciendo desde el estado inicial y otro creciendo desde el estado objetivo, y realiza una conexión incremental entre ellos. Esto permite una búsqueda más eficiente de trayectorias y una mejor exploración del espacio de configuración. Además, se introducen heurísticas y estrategias de expansión específicas para robots manipuladores, teniendo en cuenta las restricciones cinemáticas y los límites de los actuadores.

- ¿Cuáles son las consideraciones particulares para cada una de estas implementaciones?

RRT*-Smart: Rapidly-exploring Random Trees with Sampling-based Margins for Efficient Motion Planning

- RRT*-Smart utiliza una representación basada en árboles para generar trayectorias óptimas en entornos complejos.
- El algoritmo emplea técnicas de muestreo basadas en márgenes para dirigir la exploración del espacio de configuración hacia regiones más prometedoras y evitar la exploración innecesaria de áreas menos relevantes.
- Considera las restricciones cinemáticas y dinámicas del robot, así como la presencia de obstáculos en el entorno durante la generación de trayectorias.
- RRT*-Smart utiliza una estrategia de optimización para mejorar las trayectorias generadas, buscando minimizar la longitud y el costo de las mismas.

Efficient Motion Planning for Robotic Manipulators using RRTConnect with Bidirectional Tree Expansion

- El algoritmo RRTConnect con Expansión Bidireccional se centra en la planificación de trayectorias para robots manipuladores, que tienen múltiples grados de libertad y restricciones cinemáticas.
 - La planificación se realiza en un espacio de configuración que tiene en cuenta tanto las posiciones como las orientaciones de los eslabones del robot.
 - Se emplean técnicas de expansión bidireccional para conectar el árbol que crece desde el estado inicial con el árbol que crece desde el estado objetivo, permitiendo una búsqueda más eficiente de trayectorias y una exploración más completa del espacio de configuración.
 - Se consideran las restricciones cinemáticas del robot y los límites de los actuadores durante la expansión y conexión de los árboles.
- ¿Qué características comparten?
- Ambos desarrollos científicos utilizan el algoritmo RRT como base para la planificación de trayectorias en robots móviles.
 - Ambos consideran las restricciones cinemáticas y dinámicas del robot, así como la presencia de obstáculos en el entorno durante la generación de trayectorias.
 - Se centran en mejorar la eficiencia y optimización de las trayectorias generadas para minimizar la longitud y el costo de los movimientos del robot.

- ¿Qué técnicas emplean para acelerar el proceso de búsqueda de soluciones?

En **Efficient Motion Planning for Robotic Manipulators using RRTConnect with Bidirectional Tree Expansion**, se emplean varias técnicas para acelerar el proceso de búsqueda de soluciones en el algoritmo RRTConnect con Expansión Bidireccional.

1. Heurísticas de muestreo: Se utilizan heurísticas de muestreo inteligente para generar muestras en el espacio de configuración que se enfocan en áreas relevantes y prometedoras. Estas heurísticas permiten una exploración más eficiente del espacio de configuración y ayudan a encontrar soluciones más rápidamente.
2. Estrategias de expansión específicas para manipuladores: Se emplean estrategias de expansión adaptadas a las características de los manipuladores, teniendo en cuenta las restricciones cinemáticas y los límites de los actuadores. Estas estrategias permiten una expansión más inteligente de los árboles RRT, evitando movimientos inválidos y mejorando la eficiencia de la búsqueda.
3. Búsqueda bidireccional: El uso de dos árboles que crecen desde el estado inicial y el estado objetivo permite una búsqueda bidireccional de soluciones. Esto implica que la exploración del espacio de configuración se realiza desde dos direcciones diferentes, acelerando el proceso de búsqueda al encontrar rápidamente una conexión entre los árboles y reduciendo la longitud total de las trayectorias encontradas.
4. Conexión incremental: En lugar de intentar conectar directamente los árboles crecientes, se utiliza una estrategia de conexión incremental. Esto implica realizar conexiones parciales entre los árboles a medida que se expanden, lo que permite encontrar soluciones intermedias más rápidamente y mejorar la exploración del espacio de configuración.

Estas técnicas combinadas permiten acelerar el proceso de búsqueda de soluciones en el algoritmo RRTConnect con Expansión Bidireccional, mejorando la eficiencia y la efectividad de la planificación de trayectorias para robots manipuladores.

En el artículo **RRT*-Smart: Rapidly-exploring Random Trees with Sampling-based Margins for Efficient Motion Planning**

1. Margen basado en muestreo: Se utiliza un margen basado en muestreo que se ajusta dinámicamente durante la expansión del árbol RRT*-Smart. Este margen permite dirigir la exploración hacia áreas prometedoras del espacio de configuración, evitando la exploración innecesaria de regiones poco relevantes. Al ajustar el margen según la distribución de las muestras, se mejora la eficiencia de la búsqueda y se reduce el tiempo de cálculo.

2. Actualización selectiva de nodos: En lugar de actualizar todos los nodos del árbol RRT*-Smart en cada iteración, se utiliza una estrategia de actualización selectiva. Solo se actualizan los nodos que se encuentran dentro del margen basado en muestreo, lo que reduce la cantidad de cálculos necesarios y mejora la eficiencia de la búsqueda.
3. Reducción de la dimensionalidad: Para espacios de alta dimensionalidad, se emplea una técnica de reducción de dimensionalidad para mejorar la eficiencia de la búsqueda. Esto implica utilizar técnicas como PCA (Análisis de Componentes Principales) o LLE (Embedding Locally Linear) para proyectar el espacio de configuración en un espacio de menor dimensión, donde la búsqueda se realiza de manera más eficiente.
4. Conexión parcial: En lugar de intentar encontrar una conexión completa entre el árbol creciente y el objetivo en cada iteración, se permite una conexión parcial. Esto implica realizar conexiones entre subconjuntos de nodos de los árboles, lo que reduce el tiempo de cálculo y mejora la eficiencia de la búsqueda.

Estas técnicas combinadas permiten acelerar el proceso de búsqueda de soluciones en el algoritmo RRT*-Smart, mejorando la eficiencia y la efectividad de la planificación de trayectorias para robots móviles.

■ ¿Qué algoritmo utilizan para refinar la solución?

En el artículo **RRT*-Smart: Rapidly-exploring Random Trees with Sampling-based Margins for Efficient Motion Planning**, el algoritmo utilizado para refinar la solución es el algoritmo de optimización basado en gradiente. Una vez que se ha encontrado una trayectoria aproximada utilizando el algoritmo RRT*-Smart, se utiliza el algoritmo de optimización para mejorar la calidad de la trayectoria.

El algoritmo de optimización basado en gradiente busca encontrar una trayectoria que minimice una función de costo, que puede estar relacionada con factores como la longitud de la trayectoria, la suavidad de los movimientos, el consumo de energía, entre otros. Se utilizan técnicas de optimización numérica para encontrar los valores de los parámetros que minimicen la función de costo.

El refinamiento de la solución se realiza mediante iteraciones sucesivas del algoritmo de optimización. En cada iteración, se evalúa la trayectoria actual y se calcula el gradiente de la función de costo con respecto a los parámetros de la trayectoria. Luego, se actualizan los parámetros utilizando el gradiente y se obtiene una nueva trayectoria refinada.

El proceso de refinamiento continúa hasta que se alcanza una convergencia satisfactoria o se cumple algún criterio de terminación predefinido. Al finalizar, se obtiene una trayectoria refinada que representa una solución mejorada en términos de la función de costo objetivo.

En el artículo **Efficient Motion Planning for Robotic Manipulators using RRTConnect with Bidirectional Tree Expansion**, se utiliza el algoritmo RRTConnect para generar una solución inicial y luego se aplica un proceso de refinamiento utilizando el algoritmo de optimización basado en gradiente.

Una vez que se obtiene una solución inicial utilizando RRTConnect, se aplica el algoritmo de optimización basado en gradiente para refinar la solución. El objetivo es mejorar la calidad de la trayectoria inicial, teniendo en cuenta criterios de optimización como la longitud de la trayectoria, la suavidad de los movimientos, la colisión con obstáculos, entre otros.

El algoritmo de optimización basado en gradiente busca ajustar los parámetros de la trayectoria para minimizar una función de costo. En cada iteración, se evalúa la trayectoria actual y se calcula el gradiente de la función de costo con respecto a los parámetros de la trayectoria. Luego, se actualizan los parámetros utilizando el gradiente y se obtiene una nueva trayectoria refinada.

El proceso de refinamiento continúa iterativamente hasta que se alcance una convergencia satisfactoria o se cumpla algún criterio de terminación predefinido. Al finalizar, se obtiene una trayectoria refinada que representa una solución mejorada en términos de la función de costo objetivo.