

Hasil Analisis Simulasi Perencanaan Jalur dan Motion Planning di ROS

Pendahuluan

Dalam tugas ini, beberapa algoritma perencanaan jalur, termasuk Dijkstra, A*, dan Cell Decomposition, telah berhasil diimplementasikan untuk menemukan jalur terpendek dari titik awal ke titik tujuan. Selain itu, simulasi ROS Motion Planning juga berhasil dijalankan, yang mencakup aspek Path Searching dan Trajectory Optimization. Algoritma perencanaan jalur ini memiliki perbedaan karakteristik dan performa dalam menemukan jalur, yang dianalisis di bawah ini.

1. Algoritma Dijkstra

Dijkstra adalah algoritma klasik yang menemukan jalur terpendek dengan mengeksplorasi semua node secara sistematis tanpa memperhatikan arah atau tujuan. Dalam simulasi, algoritma ini terbukti efektif dalam menemukan jalur optimal, tetapi membutuhkan waktu yang lebih lama karena memeriksa semua node secara menyeluruh. Dijkstra juga tidak efisien untuk digunakan pada graf atau grid besar karena kurangnya heuristic yang mengarahkan pencarian ke tujuan.

2. Algoritma A*

A* adalah algoritma yang lebih canggih yang menggunakan heuristic (jarak Manhattan) untuk mempercepat pencarian ke arah tujuan. Dalam simulasi, A* menunjukkan kinerja yang lebih cepat dibandingkan dengan Dijkstra, karena heuristic memungkinkan algoritma untuk langsung menuju titik tujuan tanpa perlu mengeksplorasi semua node. Ini menghasilkan jalur yang sama optimalnya dengan Dijkstra, tetapi dengan waktu eksekusi yang lebih singkat, terutama pada grid besar atau graf yang kompleks.

3. Cell Decomposition

Metode Cell Decomposition membagi ruang menjadi cell-cell kecil, yang memungkinkan algoritma untuk mencari jalur yang aman di antara cell-cell yang bebas dari rintangan. Metode ini lebih sederhana dan bekerja baik dalam lingkungan dengan rintangan yang dapat dikelompokkan ke dalam cell. Namun, metode ini kurang efektif dalam lingkungan

yang sangat kompleks karena fragmentasi cell dapat menyebabkan jalur menjadi tidak optimal.

4. Simulasi ROS Motion Planning

Simulasi ROS Motion Planning menggunakan berbagai algoritma path planning seperti Greedy Best-First Search (GBFS), Dijkstra, A*, dan algoritma optimasi lintasan lainnya. Dari hasil simulasi animasi untuk setiap planner, dapat diamati bahwa:

- GBFS memiliki waktu eksekusi yang cepat namun tidak selalu memberikan jalur terpendek. Ini dikarenakan algoritma ini hanya mengandalkan heuristic tanpa mempertimbangkan total biaya jalur.
- Dijkstra, meskipun lambat, memberikan jalur terpendek karena mengeksplorasi setiap node secara menyeluruh.
- A* menunjukkan keseimbangan antara kecepatan dan akurasi jalur, memberikan hasil yang optimal dengan waktu yang lebih singkat dibandingkan Dijkstra.
- Algoritma optimasi lintasan dalam ROS, seperti Trajectory Optimization, menghasilkan lintasan yang lebih halus dan efisien untuk pergerakan robot. Hal ini penting dalam aplikasi nyata di mana jalur yang mulus dibutuhkan untuk pergerakan yang efisien dan hemat energi.