

Analisis Tugas Robotika Week 6

Nama: Muhammad Nugraha Sadewa

Nim : 1103210061

Pendahuluan

Dalam tugas ini, kami mengimplementasikan dua algoritma pencarian jalur yang populer, yaitu Probabilistic Roadmap (PRM) dan Rapidly-Expanding Random Trees (RRT). Keduanya sering digunakan dalam navigasi robotika untuk menemukan jalur di lingkungan yang tidak pasti atau berisi rintangan. Kami menggunakan dua metode visualisasi berbeda: Matplotlib dan Rviz pada ROS, untuk memberikan gambaran yang lebih jelas tentang performa setiap algoritma.

1. Implementasi dan Simulasi PRM dengan Visualisasi Matplotlib

Deskripsi: Probabilistic Roadmap (PRM) membangun graf berdasarkan node acak dalam ruang konfigurasi. PRM sangat berguna untuk lingkungan statis, di mana rute dapat direncanakan sebelum robot mulai bergerak. Pada simulasi ini, kami menggunakan Matplotlib untuk menampilkan graf yang dibangun oleh PRM, termasuk node acak, koneksi antar node, serta jalur terpendek.

Analisis:

- **Kelebihan:** Implementasi PRM dengan Matplotlib memungkinkan kita untuk memahami struktur graf secara sederhana dan jelas. Matplotlib memberikan visualisasi 2D yang mudah diatur, dan cukup responsif untuk dataset yang relatif kecil.
- **Kekurangan:** PRM di Matplotlib terbatas pada visualisasi statis dan tidak berinteraksi langsung dengan ROS, sehingga kurang sesuai untuk aplikasi real-time atau simulasi yang kompleks.
- **Efisiensi dan Akurasi:** PRM dapat menghasilkan jalur optimal ketika jumlah node acak yang cukup telah dihasilkan. Namun, waktu komputasi bertambah signifikan saat lingkungan memiliki rintangan yang rumit atau membutuhkan kepadatan node yang tinggi.

Kesimpulan untuk PRM dengan Matplotlib: PRM di Matplotlib cocok untuk studi dasar dan lingkungan yang sederhana. Ini memberikan wawasan awal yang baik mengenai cara PRM membangun graf tanpa mengorbankan banyak sumber daya komputasi.

2. Implementasi dan Simulasi PRM dengan Visualisasi di Rviz

Deskripsi: Pada implementasi ini, PRM diintegrasikan ke ROS dan divisualisasikan menggunakan Rviz, memungkinkan interaksi langsung dengan ROS dan memperlihatkan roadmap secara dinamis. Setiap node dan koneksi dapat dilihat dalam Rviz, yang sangat membantu untuk simulasi di lingkungan robotika.

Analisis:

- **Kelebihan:** Rviz memberikan visualisasi yang interaktif dan memungkinkan kita memantau proses pembentukan graf PRM secara real-time. Integrasi dengan ROS juga memudahkan untuk menghubungkan PRM dengan modul navigasi lain dalam ROS.
- **Kekurangan:** Proses setup dan debugging di ROS lebih rumit dibandingkan dengan visualisasi di Matplotlib, serta memerlukan lebih banyak sumber daya komputasi.
- **Efisiensi dan Akurasi:** Dengan ROS dan Rviz, PRM menjadi lebih cocok untuk lingkungan nyata. Namun, PRM di Rviz tetap membutuhkan waktu yang lebih lama untuk membangun jalur di lingkungan yang rumit karena ketergantungannya pada jumlah node acak dan koneksi antar node.

Kesimpulan untuk PRM dengan Rviz: PRM di Rviz adalah pilihan yang baik untuk simulasi lingkungan yang lebih realistis, di mana robot harus merencanakan jalur dalam lingkungan yang relatif statis. Interaksi langsung dengan ROS memberikan fleksibilitas lebih, meski menambah kompleksitas implementasi.

3. Implementasi Algoritma RRT dengan Visualisasi di Rviz

Deskripsi: Rapidly-Expanding Random Trees (RRT) adalah algoritma yang bekerja dengan memperluas node secara acak, mencari node terdekat untuk setiap titik baru, dan membuat jalur yang menghindari rintangan. Pada simulasi ini, RRT diimplementasikan di ROS dan visualisasinya ditampilkan di Rviz, memungkinkan kita melihat perluasan node dan jalur terbentuk secara real-time.

Analisis:

- **Kelebihan:** RRT sangat efektif di lingkungan dinamis yang penuh dengan rintangan, karena algoritma ini cepat dan tidak membutuhkan seluruh graf untuk dibentuk sebelum robot bergerak. Visualisasi di Rviz menampilkan perkembangan tree yang tumbuh secara iteratif, yang memberikan insight bagus mengenai jalur yang dipilih robot.
- **Kekurangan:** RRT biasanya menghasilkan jalur yang kurang optimal dibandingkan PRM, karena jalur sering kali memiliki banyak belokan tajam dan

lebih panjang. Selain itu, implementasi RRT membutuhkan lebih banyak iterasi untuk mencapai target, terutama dalam ruang konfigurasi yang besar.

- **Efisiensi dan Akurasi:** RRT efisien dalam menemukan jalur di lingkungan yang berubah-ubah, tetapi jalur yang dihasilkan sering kali tidak optimal dalam hal panjang. RRT membutuhkan lebih sedikit waktu untuk mulai bergerak dibandingkan PRM karena tidak memerlukan seluruh graf dibangun terlebih dahulu.

Kesimpulan untuk RRT dengan Rviz: RRT di Rviz adalah pilihan yang baik untuk navigasi di lingkungan dinamis atau dengan banyak rintangan. Meski jalur yang dihasilkan mungkin tidak optimal, kecepatan dan responsivitasnya membuat algoritma ini ideal untuk aplikasi real-time.

Kesimpulan Akhir dan Rekomendasi

Setiap algoritma memiliki kelebihan dan kekurangan yang membuatnya lebih sesuai untuk aplikasi tertentu:

1. **PRM dengan Matplotlib:** Ideal untuk studi awal dan simulasi di lingkungan statis yang sederhana. Cocok untuk skenario di mana interaksi langsung dengan ROS tidak dibutuhkan.
2. **PRM dengan Rviz:** Lebih cocok untuk simulasi yang melibatkan ROS, khususnya di lingkungan yang statis atau semi-statis. Memberikan gambaran yang lebih realistis dan memungkinkan integrasi yang lebih baik dengan modul ROS lainnya.
3. **RRT dengan Rviz:** RRT di Rviz cocok untuk lingkungan dinamis, di mana robot harus mencari jalur secara real-time dengan banyak rintangan. Meskipun jalur mungkin tidak optimal, algoritma ini cepat dalam menemukan rute yang memungkinkan navigasi segera.

Rekomendasi: Untuk lingkungan yang statis, PRM memberikan hasil yang lebih optimal. Namun, di lingkungan dinamis, RRT menjadi pilihan yang lebih fleksibel karena dapat beradaptasi lebih cepat terhadap perubahan. Kombinasi kedua algoritma ini mungkin menjadi pilihan ideal untuk lingkungan nyata, di mana area statis dan dinamis dapat diproses secara berbeda.