

Implementasi dan Simulasi Probabilistic Roadmap (PRM) dengan Visualisasi Matplotlib

Tujuan dan Pendekatan

- Algoritma PRM bertujuan untuk menemukan jalur antara titik awal dan titik tujuan dengan membangun graf yang terdiri dari node-node acak di ruang bebas.
- Node-node ini dihubungkan berdasarkan jarak tertentu dan membentuk peta jalan (roadmap), yang memungkinkan pencarian jalur terpendek dengan metode seperti algoritma Dijkstra atau algoritma jalur terpendek berbobot lainnya.

Implementasi dan Simulasi

- **Pengambilan Sampel dan Koneksi Node:** Node acak dibangkitkan di ruang bebas, lalu dihubungkan jika berada dalam jarak tertentu, dengan menggunakan struktur data seperti KDTree untuk efisiensi.
- **Pencarian Jalur:** Jalur dari titik awal ke titik tujuan diperoleh menggunakan algoritma pencarian jalur pada graf.
- **Visualisasi dengan Matplotlib:** Matplotlib digunakan untuk menampilkan roadmap dan jalur yang ditemukan. Visualisasi ini berguna untuk memeriksa distribusi node, konektivitas, serta validitas jalur dalam ruang bebas.

Analisis Kelebihan dan Kekurangan

- **Kelebihan:** Implementasi PRM ini relatif mudah dilakukan dan cocok untuk lingkungan dengan ruang bebas yang cukup luas. Matplotlib memungkinkan pengujian cepat tanpa memerlukan ROS.
- **Kekurangan:** PRM dengan Matplotlib kurang fleksibel dalam lingkungan dinamis dan tidak mendukung visualisasi interaktif yang umumnya diperlukan dalam robotika.

Implementasi dan Simulasi Probabilistic Roadmap (PRM) dengan Visualisasi di Rviz pada ROS

Tujuan dan Pendekatan

- Implementasi PRM di ROS bertujuan untuk membawa roadmap ke dalam lingkungan ROS, memungkinkan integrasi dengan sensor dan sistem navigasi yang lebih kompleks.
- Node dan jalur PRM dipublikasikan sebagai Marker di ROS, dan divisualisasikan di Rviz.

Implementasi dan Simulasi

- **Integrasi dengan ROS:** Node PRM dipublikasikan dalam format ROS sehingga roadmap dan jalur dapat divisualisasikan di Rviz sebagai Marker.

- **Visualisasi di Rviz:** Rviz memungkinkan tampilan yang dinamis dan interaktif, dengan opsi untuk menambahkan layer informasi lain seperti peta lingkungan dan data sensor.

Analisis Kelebihan dan Kekurangan

- **Kelebihan:** Implementasi ini memungkinkan aplikasi langsung di lingkungan robotika dengan ROS, memungkinkan integrasi dengan sensor. Rviz memberikan visualisasi yang interaktif dan dinamis, membantu memantau interaksi roadmap PRM dengan lingkungan nyata.
 - **Kekurangan:** PRM cenderung kurang efisien dalam lingkungan yang berubah-ubah atau memiliki banyak rintangan bergerak, karena roadmap statis perlu diperbarui secara berkala. Setup di ROS juga memerlukan waktu dan kompleksitas lebih dibandingkan dengan visualisasi sederhana.
-

Implementasi Algoritma Rapidly-Expanding Random Trees (RRT) di ROS dengan Visualisasi di Rviz

Tujuan dan Pendekatan

- Algoritma RRT bertujuan untuk mencari jalur dari titik awal ke titik tujuan dengan pendekatan eksploratif, memperluas pohon dari titik awal ke arah node acak baru hingga mencapai tujuan.
- Node RRT dan jalur akhir dipublikasikan di ROS dan divisualisasikan di Rviz.

Implementasi dan Simulasi

- **Sampling dan Ekspansi Pohon:** Setiap iterasi, RRT melakukan sampling acak dan memperluas pohon menuju node acak tersebut. Jika node baru berada dekat dengan tujuan, pencarian berhenti dan jalur dari titik awal ke tujuan dikembalikan.
- **Integrasi ROS dan Visualisasi di Rviz:** Node dipublikasikan sebagai Marker dengan jalur akhir ditampilkan sebagai LINE_STRIP di Rviz.

Analisis Kelebihan dan Kekurangan

- **Kelebihan:** RRT sangat cocok untuk lingkungan dengan rintangan padat, karena tidak memerlukan keseluruhan roadmap. Visualisasi di Rviz memungkinkan pemantauan proses eksplorasi, node yang dihasilkan, dan jalur akhir.
 - **Kekurangan:** RRT sering kali menghasilkan jalur yang kurang optimal karena pohon yang dihasilkan tidak sepenuhnya mengeksplorasi ruang bebas. Algoritma ini juga dapat menjadi lambat jika jarak antar node terlalu kecil atau lingkungan sangat kompleks.
-

Perbandingan dan Rekomendasi Implementasi PRM vs RRT di ROS

- **Kesesuaian Lingkungan:** PRM lebih sesuai untuk lingkungan statis dengan ruang bebas yang cukup besar, karena roadmap dapat digunakan berulang tanpa perlu regenerasi. Sebaliknya, RRT lebih cocok untuk lingkungan dinamis dan penuh rintangan, karena jalur dapat dicari secara langsung tanpa memerlukan roadmap.
 - **Visualisasi di Rviz:** Kedua algoritma mendapatkan manfaat dari visualisasi di Rviz, memungkinkan pemantauan proses pencarian jalur secara real-time. RRT cenderung lebih menarik untuk divisualisasikan karena menampilkan proses eksplorasi dari titik awal ke tujuan.
 - **Fleksibilitas dan Integrasi:** Kedua algoritma dapat diintegrasikan dengan data sensor robot melalui ROS. Namun, RRT lebih fleksibel dalam beradaptasi dengan lingkungan yang berubah, sementara PRM memerlukan regenerasi roadmap jika terdapat perubahan signifikan pada peta atau rintangan.
-

Kesimpulan

Kedua algoritma memiliki keunggulan tergantung pada sifat lingkungan dan kebutuhan aplikasi:

- **PRM dengan Rviz** cocok untuk lingkungan statis atau semi-statis, di mana roadmap dapat diandalkan untuk waktu lama tanpa regenerasi.
- **RRT dengan Rviz** lebih direkomendasikan untuk lingkungan dinamis atau penuh rintangan, karena pendekatan eksploratifnya memungkinkan pencarian jalur tanpa perlu roadmap awal.

Visualisasi di Rviz membantu memahami dan mengawasi cara kerja algoritma dan eksekusi hasil pencarian jalur dalam skenario nyata di ROS.