

Analisis Implementasi dan Simulasi PRM & RRT

1. Implementasi dan Simulasi Probabilistic Roadmap (PRM) dengan Visualisasi Matplotlib From Scratch

Deskripsi: Tugas ini melibatkan implementasi algoritma Probabilistic Roadmap (PRM) dengan menggunakan Python untuk menghasilkan node acak, membentuk graf, dan mencari jalur terpendek di antara posisi awal dan tujuan. Visualisasi dilakukan menggunakan Matplotlib.

Proses:

- **File Lingkungan:** Sebuah file environment.txt berisi data tentang posisi awal, tujuan, dan lokasi serta ukuran hambatan (obstacles) yang mempengaruhi jalur yang dapat diambil.
- **Algoritma PRM:** Node acak dihasilkan di ruang bebas, dan setiap node dihubungkan ke node lainnya yang paling dekat, membentuk graf yang memungkinkan pencarian jalur.
- **Visualisasi:** Jalur PRM divisualisasikan menggunakan Matplotlib, dengan jalur yang menghubungkan node yang berhasil dibangun, serta visualisasi hambatan dan jalur terpendek yang ditemukan.

Hasil:

- **Keberhasilan:** Algoritma berhasil membangun peta probabilistik berdasarkan titik acak, menghubungkan node dengan efisien, dan menemukan jalur terpendek dalam ruang yang terbatas oleh hambatan.
- **Visualisasi:** Peta yang dihasilkan menunjukkan jalur terbaik dari titik awal menuju tujuan, dengan hambatan digambarkan sebagai area yang tidak dapat dilalui. Hasil ini juga memperlihatkan jaringan hubungan antar node dan jalur yang menghindari hambatan.

Tantangan: Pengaturan parameter yang tepat seperti jumlah sampel (numSamples) dan jarak antar node sangat penting untuk memastikan kualitas roadmap yang dibangun.

2. Implementasi dan Simulasi Probabilistic Roadmap (PRM) dengan Visualisasi RViz

Deskripsi: Tugas ini melibatkan penerapan algoritma PRM dalam konteks ROS, di mana visualisasi dilakukan di RViz untuk mempermudah pemantauan jalur yang dibangun secara real-time.

Proses:

- **Pengaturan ROS:** Menjalankan ROS dengan memanfaatkan paket rrt-planning dan node PRM yang dihubungkan dengan RViz untuk memvisualisasikan peta dan jalur yang dibangun.
- **Visualisasi:** Di RViz, pengguna dapat melihat visualisasi graf PRM yang menggambarkan jaringan node acak, serta jalur yang menghubungkan titik awal dan tujuan.
- **Parameter yang Dapat Disesuaikan:** Pengguna dapat mengubah jumlah node yang diambil dan jumlah tetangga yang dipertahankan untuk membangun PRM.

Hasil:

- **Keberhasilan:** Proses ini berhasil menunjukkan bagaimana PRM berfungsi dalam konteks ROS, dengan visualisasi interaktif yang memberikan gambaran yang jelas tentang peta, hambatan, dan jalur terpendek.
- **Visualisasi:** RViz menampilkan peta dengan marker yang mewakili node PRM dan jalur yang dihasilkan, memungkinkan visualisasi jalur yang aman dan efisien antara titik awal dan tujuan.

Tantangan: Penyesuaian parameter dalam RViz untuk mendapatkan hasil visual yang optimal, serta pengaturan koordinat dunia yang tepat agar marker dan jalur terlihat dengan jelas.

3. Implementasi Algoritma Rapidly-Expanding Random Trees (RRT) di ROS dengan Visualisasi RViz

Deskripsi: Tugas ini berfokus pada implementasi algoritma RRT di dalam ROS, dengan visualisasi di RViz untuk menunjukkan pertumbuhan pohon secara real-time.

Proses:

- **Pengaturan ROS:** Paket rrt-ros di-clone dan dikompilasi menggunakan catkin_make. Selanjutnya, node RRT dijalankan melalui ROS dan terhubung dengan RViz untuk visualisasi.
- **Proses RRT:** Algoritma RRT bekerja dengan memperluas pohon secara acak dari titik awal menuju titik tujuan, berusaha untuk menghindari hambatan yang ada.
- **Visualisasi:** Di RViz, pengguna dapat melihat bagaimana pohon RRT berkembang, dengan setiap iterasi visualisasi menunjukkan cabang pohon yang terbentuk. Marker ditambahkan untuk menggambarkan jalur yang sudah terbentuk.

Hasil:

- **Keberhasilan:** Algoritma RRT berhasil membuat pohon yang tumbuh dari titik awal menuju tujuan, meskipun terkadang algoritma membutuhkan beberapa percobaan untuk menemukan jalur terbaik.
- **Visualisasi:** Di RViz, jalur yang dibangun oleh RRT dapat dilihat dengan marker yang menunjukkan setiap langkah ekspansi pohon. Jalur akhir yang ditemukan adalah jalur terpendek yang menghindari hambatan.

Tantangan: Salah satu tantangan utama adalah mengonfigurasi parameter yang sesuai dalam RRT, seperti jumlah iterasi yang dibutuhkan untuk menemukan jalur yang optimal. Kualitas visualisasi juga bergantung pada pengaturan RViz yang tepat.

Kesimpulan

- **Implementasi PRM dengan Matplotlib** memberikan pemahaman yang lebih mendalam tentang bagaimana probabilistic roadmap bekerja untuk path planning, terutama dalam konteks ruang

bebas dan hambatan. Meskipun visualisasinya terbatas pada Matplotlib, hasilnya cukup jelas dan memberikan gambaran yang baik mengenai jalur yang dibangun.

- **Implementasi PRM dengan RViz** menggunakan ROS memungkinkan visualisasi yang lebih dinamis dan interaktif. Dengan mengubah parameter dalam params.yaml, kita dapat melihat perubahan peta dan jalur secara langsung dalam lingkungan 3D, yang memudahkan pemahaman tentang bagaimana PRM beroperasi di dunia nyata.
- **Implementasi RRT dengan RViz** memberikan perspektif tambahan tentang algoritma RRT dalam ROS. Visualisasi yang lebih real-time memberikan pemahaman yang baik tentang cara RRT mengeksplorasi ruang dan memperluas pohon secara efisien untuk menemukan jalur.