

Nama : Fadhil Dzikri Aqila

NIM : 1103213136

Kelas : TK-45-G09

Analisis 3 Algoritma Perencanaan Jalur dan ROS Motion Planning

1. Analisis 3 Algoritma Perencanaan Jalur

a. Algoritma dijkstra

Kode Python `dijkstra_planner.py` ini mengimplementasikan algoritma Dijkstra, yaitu algoritma pencarian jalur terpendek yang populer untuk menemukan jalur dengan biaya terendah antara dua node dalam sebuah graf. Fungsi ini dimulai dari node yang diberikan dan menggunakan antrian prioritas (min-heap) untuk mengeksplorasi node tetangga, terus-menerus memilih node dengan biaya terkecil yang telah terakumulasi. Algoritma ini melacak node yang sudah dikunjungi dan memperbarui jarak minimum untuk setiap tetangga saat ditemukan jalur yang lebih pendek. Algoritma berhenti ketika mencapai node tujuan, mengembalikan biaya serta jalur yang diambil. Dalam kasus ini, graf direpresentasikan sebagai daftar ketetanggaan, dan script menemukan jalur terpendek dari node 'A' ke 'D', yang menghasilkan biaya 4 dengan jalur ['A', 'B', 'C', 'D'].

b. Algoritma A*

Kode Python `a_star_planner.py` berisi implementasi algoritma pencarian A*, yang merupakan pengembangan dari algoritma Dijkstra dengan memasukkan heuristik untuk memperkirakan jarak yang tersisa ke tujuan. A* menggunakan dua metrik utama: g, yang melacak biaya aktual dari titik awal ke node tertentu, dan f, yang merupakan penjumlahan dari g dan estimasi heuristik ke tujuan (menggunakan jarak Manhattan dalam kasus ini). Algoritma ini mengeksplorasi node dengan nilai f terendah, memastikan bahwa ia menyeimbangkan antara mengeksplorasi jalur terpendek dan fokus pada node yang tampak lebih dekat dengan tujuan. Algoritma ini berhasil menghitung jalur dari pojok kiri atas ke pojok kanan bawah pada grid 2D, menghindari rintangan dan menghasilkan jalur optimal.

c. Algoritma Cell Decomposition

Kode Python `cell_decomposition.py` berisi implementasi dasar dari cell decomposition, teknik perencanaan jalur yang sering digunakan dalam robotika. Metode ini membagi ruang dengan rintangan menjadi sel-sel kecil yang bebas dari rintangan, yang menyederhanakan perencanaan jalur. Script ini mendefinisikan dua fungsi utama: `create_cells`, yang menghasilkan sel berdasarkan rintangan, dan `find_path_through_cells`, yang menemukan jalur sederhana antara titik awal dan tujuan dengan melintasi sel-sel tersebut. Meskipun implementasinya masih sederhana dan belum mencakup logika pembangkitan sel atau navigasi yang kompleks, script ini memberikan kerangka dasar untuk pengembangan lebih lanjut ke dalam perencanaan jalur berbasis sel yang lebih canggih. Outputnya, meskipun sederhana, menunjukkan koneksi langsung dari titik awal ke tujuan.

2. Analisis ROS Motion Planning

ROS (Robot Operating System) Motion Planning adalah proses penting dalam robotika yang digunakan untuk merencanakan gerakan robot dari titik awal ke tujuan, sambil menghindari rintangan dan memenuhi batasan kinematik serta dinamik robot. Salah satu komponen utama dalam ROS untuk perencanaan gerakan adalah MoveIt!, yang banyak digunakan untuk perencanaan gerakan lengan robotik dengan berbagai derajat kebebasan (DOF). MoveIt! menggabungkan algoritma pencarian jalur seperti RRT (Rapidly-exploring Random Tree), PRM (Probabilistic Roadmap), dan A*, yang membantu menemukan jalur optimal sambil menghindari tabrakan. Setelah jalur ditemukan, proses perencanaan lintasan dilakukan untuk memastikan jalur tersebut dapat diikuti dengan gerakan yang halus, mempertimbangkan kecepatan, percepatan, dan dinamika gerakan robot. ROS juga menggunakan alat pemeriksaan tabrakan seperti FCL (Flexible Collision Library) untuk memastikan robot tidak menabrak objek selama bergerak. Selain itu, untuk robot lengan, kinematika invers digunakan untuk menghitung posisi sendi yang tepat agar robot dapat mencapai posisi ujung lengan yang diinginkan. Pada robot bergerak, lokalisasi dan pemetaan menjadi kunci, dengan metode seperti SLAM (Simultaneous Localization and Mapping) memungkinkan robot membangun peta dan melokalisasi dirinya di lingkungan. Seluruh proses ini diakhiri dengan kontroler gerakan seperti PID controllers yang menerjemahkan jalur yang direncanakan ke dalam perintah eksekusi aktual oleh aktuator robot, memastikan robot bergerak sesuai rencana dengan halus dan presisi.