

Nama: Muhammad Fariq Taqi Pasai
NIM: 1103204193

Technical Report

1. Introduction to ROS

Robot Operating System (ROS) adalah kerangka kerja perangkat lunak yang dirancang khusus untuk mengembangkan aplikasi robotika. ROS menyediakan berbagai fungsi dan alat yang memungkinkan pengembang untuk mengendalikan robot dengan mudah. Penggunaan ROS direkomendasikan karena sejumlah alasan yang membuatnya menjadi pilihan yang baik untuk pengembangan robotika. Berikut adalah beberapa alasan utama:

- Open Source: ROS dirilis sebagai perangkat lunak open source, yang berarti dapat diakses dan digunakan secara gratis oleh siapa saja. Ini membuka peluang kolaborasi dan partisipasi aktif dari komunitas pengembang di seluruh dunia. Dengan akses sumber terbuka, pengguna dapat mengadaptasi, memodifikasi, dan berbagi kode dengan mudah.
- Fleksibilitas Platform: ROS dapat digunakan di berbagai platform, termasuk Linux dan macOS. Fleksibilitas ini memungkinkan pengembang untuk memilih perangkat keras dan sistem operasi yang sesuai dengan kebutuhan mereka.

ROS (Robot Operating System) memiliki beberapa tujuan utama yang membimbing pengembangannya. Tujuan-tujuan tersebut mencakup berbagai aspek dan mencerminkan filosofi desain ROS. ROS dirancang untuk memudahkan penggunaan kembali (code reusability) dengan menyediakan struktur modular dan paket-paket yang dapat diintegrasikan secara independen. Hal ini memungkinkan pengembang untuk membangun atas kerja orang lain dan menggabungkan komponen-komponen yang telah ada dalam proyek-proyek mereka sendiri.

2. Getting Started with ROS Programming

Langkah pertama adalah menginstal ROS di sistem Anda. Saat ini, ROS mendukung beberapa distribusi Linux seperti Ubuntu.

- Instalasi ROS: Langkah pertama adalah menginstal ROS di sistem Anda. Saat ini, ROS mendukung beberapa distribusi Linux seperti Ubuntu.
- Membuat dan Mengatur Workspace: Setelah instalasi selesai, membuat workspace untuk proyek ROS. Workspace adalah direktori tempat akan menyimpan dan mengembangkan paket-paket ROS.
- Membuat Paket ROS: Paket adalah unit dasar pengorganisasian di ROS. Gunakan perintah `catkin_create_pkg` untuk membuat paket baru.
- Mengompilasi dan Membangun Paket: Setelah menulis kode, kembali ke workspace dan jalankan perintah `catkin_make` untuk mengompilasi dan membangun paket-paket Anda.
- Menjalankan ROS Master: ROS Master adalah inti dari sistem komunikasi ROS. Jalankan ROS Master dengan perintah `roscore`

3. Working with ROS for 3D Modelling

Berikut adalah tahapan penggunaan ROS (Robot Operating System) untuk pemodelan 3D sebagai berikut:

- Persiapan dan Instalasi
- Pembuatan Model Robot

- Menggunakan RViz untuk Visualisasi: Inisialisasikan environment ROS dengan menjalankan perintah `source ~/catkin_ws/devel/setup.bash` pada terminal. Menggunakan RViz untuk visualisasi model robot, data sensor, dan elemen lainnya dalam ruang 3D.
- Simulasi dengan Gazebo: Inisialisasikan ROS Master dengan menjalankan perintah `roscore`. Mengintegrasikan model robot, sensor, dan pengontrol ke dalam Gazebo untuk simulasi 3D yang realistis.

4. Simulating Robots using ROS dan Gazebo

Simulasi robot menggunakan ROS dan Gazebo merupakan pendekatan yang sangat berguna dalam pengembangan robotika. Gazebo adalah simulator robot yang kuat dan terintegrasi dengan ROS, memungkinkan pengembang untuk menguji dan mengoptimalkan performa robot dalam lingkungan virtual sebelum menerapkannya di dunia nyata.

- Instalasi Ros dan Gazebo
- Pembuatan Workspace
- Pengaturan Model Robot: Menempatkan model robot di dalam direktori `~/catkin_ws/src` atau direktori yang sesuai dengan struktur workspace Anda.
- Mengintegrasikan ROS dengan Gazebo

5. Simulating Robots using ROS, CoppeliaSim, and webots

Menjalankan simulasi pada kedua simulator, CoppeliaSim dan Webots, untuk mengamati perilaku robot dalam berbagai lingkungan virtual. Menggunakan antarmuka yang disediakan oleh CoppeliaSim dan Webots untuk memantau simulasi. memvisualisasikan data dari robot simulasi menggunakan alat-alat visualisasi ROS seperti RViz. Mengoptimalkan parameter simulasi, model robot, atau algoritma kontrol sesuai kebutuhan untuk mencapai performa yang diinginkan. Menggunakan ROS untuk mengembangkan dan menguji algoritma kontrol robot, dan implementasikan node. Simulasi robot menggunakan ROS, CoppeliaSim, dan Webots memberikan lingkungan yang fleksibel untuk mengembangkan, menguji, dan mengoptimalkan sistem robotik sebelum diimplementasikan di dunia nyata.