

Nama : Faiz Hibatullah
NIM : 1103210172
Kelas : TK-45-G09

Analisis

Chapter 2: Memulai Pemrograman ROS

Pada chapter ini, dasar-dasar pemrograman ROS dipelajari melalui pembuatan package `mastering_ros_demo_pkg`. Langkah awal dilakukan dengan menyiapkan workspace ROS menggunakan `catkin_make` untuk mengelola paket-paket. Workspace ini disiapkan sebagai tempat pengembangan semua file ROS yang nantinya digunakan untuk membangun aplikasi robotika. Proses ini dimulai dengan memastikan struktur folder telah sesuai, seperti menempatkan file di dalam direktori `src`, dan diikuti dengan membangun workspace menggunakan `catkin_make`.

Setelah workspace siap, langkah berikutnya melibatkan pengoperasian ROS Master menggunakan perintah `roscore`. ROS Master berfungsi sebagai pusat koordinasi yang menghubungkan semua node dalam sistem. Komunikasi antar-node diuji menggunakan publisher dan subscriber pada topic yang sama. Node publisher digunakan untuk mengirimkan pesan string secara periodik, sementara node subscriber menerima pesan tersebut dan mencetaknya di terminal.

Penggunaan pesan kustom ditambahkan untuk meningkatkan fleksibilitas dalam mendesain komunikasi. Dengan membuat file pesan kustom di dalam folder `msg` dan memperbarui file `CMakeLists.txt`, format data yang lebih kompleks dapat didefinisikan. Tantangan utama dalam tahap ini adalah memastikan semua dependensi telah terinstal dengan benar dan file konfigurasi disesuaikan agar kompatibel. Hasil akhir menunjukkan struktur dan komunikasi dalam ROS yang lebih mendalam, serta kemampuan untuk mengembangkan aplikasi robotika yang lebih kompleks di masa depan.

Chapter 3: Bekerja dengan Pemodelan 3D di ROS

Chapter ini berfokus pada pemodelan robot menggunakan URDF dan xacro, dua format yang digunakan untuk mendeskripsikan robot dalam ROS. URDF (Unified Robot Description Format) digunakan untuk mendefinisikan struktur dasar robot, seperti link, joint, serta properti fisik seperti massa dan inersia. Sementara itu, xacro (XML Macros) memungkinkan penggunaan parameter dan modularitas yang mempermudah pembuatan model robot yang lebih kompleks.

Proses pemodelan dimulai dengan membuat model robot sederhana, seperti robot diferensial atau lengan robot 7-DOF. File URDF dibuat dengan mendefinisikan elemen-elemen robot seperti link dan joint, serta menambahkan properti fisik dan visual agar robot terlihat realistis saat divisualisasikan di RViz. File xacro digunakan untuk mempermudah pengelolaan parameter yang berulang, seperti dimensi link atau properti material.

Tantangan utama adalah memastikan file URDF atau xacro tidak mengandung kesalahan sintaksis, karena ini dapat menyebabkan kegagalan saat model diluncurkan di RViz. Hasil akhir menunjukkan kemampuan untuk memvisualisasikan robot dengan tepat, memahami hubungan antar-komponen robot, dan mengaplikasikan desain ini pada simulasi yang lebih realistis.

Chapter 4: Mensimulasikan Robot di Gazebo

Dalam chapter ini, fokus diarahkan pada simulasi robot di Gazebo. Gazebo adalah simulator robotika yang memungkinkan pengujian model robot dalam lingkungan virtual sebelum diterapkan pada robot fisik. Dunia simulasi untuk lengan robot 7-DOF diluncurkan sebagai langkah awal. Lengan robot ini dirancang untuk memiliki kontrol penuh terhadap setiap joint, yang memungkinkan simulasi tugas-tugas manipulasi seperti memindahkan objek.

Plugin `ros_control` digunakan untuk mengintegrasikan kontrol gerakan robot ke dalam simulasi. Joint tertentu dapat digerakkan ke posisi yang diinginkan dengan mengirimkan data ke topic kontrol. Langkah ini memberikan wawasan tentang bagaimana kontrol gerakan diterapkan dalam sistem berbasis ROS.

Selain simulasi lengan robot, simulasi robot bergerak diferensial juga dilakukan dengan meluncurkan dunia Gazebo menggunakan `diff_wheeled_gazebo.launch`. Robot ini dirancang untuk bergerak dalam lingkungan simulasi dan dapat dikontrol secara manual menggunakan keyboard melalui `keyboard_teleop.launch`. Dengan teleop, robot dapat diarahkan untuk bergerak ke berbagai posisi, memungkinkan eksplorasi lingkungan secara langsung.

Tantangan utama adalah memastikan bahwa semua plugin yang diperlukan telah terinstal dan file konfigurasi telah diatur dengan benar, seperti `controller.yaml` dan file peluncuran. Hasil simulasi menunjukkan bahwa robot dapat dikendalikan secara akurat dalam lingkungan virtual, baik untuk tugas manipulasi maupun navigasi, memberikan peluang untuk eksperimen dengan tugas robotik yang lebih kompleks. Dalam chapter ini, fokus diarahkan pada simulasi robot di Gazebo. Gazebo adalah simulator robotika yang memungkinkan pengujian model robot dalam lingkungan virtual sebelum diterapkan pada robot fisik. Dunia simulasi untuk lengan robot 7-DOF diluncurkan sebagai langkah awal. Lengan robot ini dirancang untuk memiliki kontrol penuh terhadap setiap joint, yang memungkinkan simulasi tugas-tugas manipulasi seperti memindahkan objek.

Plugin `ros_control` digunakan untuk mengintegrasikan kontrol gerakan robot ke dalam simulasi. Joint tertentu dapat digerakkan ke posisi yang diinginkan dengan mengirimkan data ke topic kontrol. Langkah ini memberikan wawasan tentang bagaimana kontrol gerakan diterapkan dalam sistem berbasis ROS.

Tantangan utama adalah memastikan bahwa semua plugin yang diperlukan telah terinstal dan file konfigurasi telah diatur dengan benar, seperti `controller.yaml` dan file peluncuran. Hasil simulasi menunjukkan bahwa robot dapat dikendalikan secara akurat dalam lingkungan virtual, memberikan peluang untuk eksperimen dengan tugas robotik yang lebih kompleks, seperti navigasi atau manipulasi objek.

Chapter 6: Menggunakan MoveIt! dan Navigation Stack

Chapter ini mendalami kemampuan manipulasi robot dengan MoveIt! dan navigasi otonom menggunakan Navigation Stack. MoveIt! adalah framework yang memungkinkan perencanaan gerakan robot dengan menghindari tabrakan, sementara Navigation Stack memungkinkan robot bergerak secara otonom di lingkungan yang telah dipetakan.

Simulasi diawali dengan peluncuran robot diferensial di Gazebo, diikuti dengan proses pemetaan menggunakan SLAM (Simultaneous Localization and Mapping). SLAM memungkinkan robot memanfaatkan data sensor seperti LIDAR untuk membuat representasi peta lingkungan secara real-time. Robot digerakkan secara manual menggunakan teleop untuk menjelajahi lingkungan, dan peta hasil pemetaan disimpan menggunakan map_saver. Peta ini kemudian digunakan untuk navigasi otomatis, di mana robot bergerak menuju tujuan yang ditentukan sambil menghindari rintangan.

Tantangan terbesar adalah memastikan parameter SLAM, seperti resolusi peta dan frekuensi sensor, diatur dengan benar agar proses pemetaan akurat. Selain itu, integrasi AMCL (Adaptive Monte Carlo Localization) dengan Navigation Stack memerlukan konfigurasi yang cermat untuk mencapai hasil navigasi yang optimal. Hasilnya menunjukkan bahwa robot dapat bergerak secara efisien menuju tujuan yang diinginkan, memberikan gambaran nyata tentang potensi aplikasi robotika otonom di dunia nyata.