Mastering ROS2 with Webots

Harits Maulana Muzakki
1103204166
TK-44-04

Latar Belakang

Robot merupakan salah satu alat yang dapat membantu pekerjaan manusia mulai dari tahun 1950an hingga saat ini. Bidang Robotika ini banyak digunakan di berbagai industri termasuk pengembangan simulasi program robotika seperti salah satunya adalah Webots. Untuk menjalankan suatu program Webots, diperlukan ROS2 yang menjadi tujuan pada tugas ini. Menguasai ROS2 yang menjadi syarat agar program-program yang berada di Webots dapat berjalan dengan baik.

ROS Basic

ROS (Robot Operating System) adalah kerangka kerja sumber terbuka yang menyediakan kumpulan perpustakaan perangkat lunak dan alat untuk membantu pengembang membangun aplikasi robot. ROS bukan sistem operasi yang sebenarnya, tetapi merupakan meta-sistem operasi yang berjalan di atas sistem operasi yang ada (seperti Linux) dan menyediakan sekumpulan layanan yang khusus dirancang untuk robotika.

Berikut adalah beberapa konsep dan komponen dasar ROS:

1. Node: Aplikasi ROS diorganisir dalam node, yaitu program eksekusi individu yang melakukan tugas tertentu. Node dapat saling berkomunikasi dengan menerbitkan dan berlangganan pesan.

2. Topik (Topic): Node berkomunikasi dengan mengirim dan menerima pesan melalui topik. Topik adalah saluran yang diberi nama yang memungkinkan beberapa node

untuk bertukar data. Node dapat menerbitkan pesan pada topik atau berlangganan pesan pada topik yang menarik bagi mereka.

3. Pesan (Message): Pesan adalah struktur data yang digunakan untuk komunikasi antara node. Mereka mendefinisikan format dan tipe data yang akan dipertukarkan. ROS menyediakan berbagai jenis pesan yang telah ditentukan sebelumnya, seperti data sensor, perintah kontrol, dan informasi status.

4. Layanan (Service): Layanan memungkinkan node untuk meminta dan menerima respons dari node lain. Berbeda dengan topik yang memungkinkan streaming data secara kontinu, layanan menyediakan pola komunikasi permintaan-respons.

5. Aksi (Action): Aksi memungkinkan perilaku yang lebih kompleks dan umpan balik selama eksekusi tugas. Mereka menyediakan cara untuk memecah tugas yang berjalan lama menjadi sub-tugas yang lebih kecil dan memberikan umpan balik tentang kemajuan.

6. Berkas Peluncuran (Launch Files): Berkas peluncuran adalah file XML yang digunakan untuk memulai beberapa node ROS secara bersamaan. Mereka mendefinisikan node mana yang harus diluncurkan dan dengan parameter apa, menyederhanakan proses memulai sistem ROS yang kompleks.

7. Paket (Packages): ROS mengorganisir kode ke dalam paket, yaitu direktori yang berisi node, perpustakaan, file konfigurasi, dan sumber daya lain yang terkait dengan fungsionalitas atau tugas tertentu.

8. Catkin: Catkin adalah sistem pembangunan yang digunakan dalam ROS. Ini mengelola kompilasi dan dependensi paket ROS.

ROS menyediakan kerangka kerja yang fleksibel dan modular untuk mengembangkan aplikasi robotika. Ini menyederhanakan proses pengembangan dengan menyediakan

mekanisme komunikasi yang terstandarisasi, memudahkan integrasi komponen dan perpustakaan yang berbeda. ROS memiliki komunitas yang besar dan aktif, yang berkontribusi pada berbagai perpustakaan dan alat yang tersedia, sehingga membuatnya menjadi pilihan populer bagi para ahli robotika di seluruh dunia.

Webots Basic

Webots adalah lingkungan simulasi robot yang kuat dan serbaguna. Dengan Webots, dapat memodelkan dan mensimulasikan robot serta lingkungan fisik di dalam komputer . Berikut adalah beberapa konsep dan komponen dasar Webots:

1. Lingkungan Simulasi: Webots menyediakan lingkungan simulasi virtual di mana dapat membuat objek dan lingkungan fisik. dapat menambahkan objek robot, sensor, aktuator, dan objek lingkungan lainnya, seperti dinding, lantai, atau hambatan.

2. Robot Model: dapat membuat model robot yang akurat dalam Webots dengan menentukan bentuk, dimensi, dan properti fisiknya. juga dapat melampirkan sensor dan aktuator ke robot, memprogram perilaku robot, dan mengatur interaksi antara robot dan lingkungannya.

3. Sensor dan Aktuator: Webots menyediakan berbagai jenis sensor dan aktuator yang dapat digunakan pada robot . Sensor seperti kamera, lidar, atau sensor jarak dapat digunakan untuk mengambil informasi dari lingkungan simulasi. Sementara itu, aktuator seperti motor atau roda dapat menggerakkan robot dan memungkinkannya berinteraksi dengan lingkungan.

4. Program Kontrol: dapat menghubungkan program kontrol ke robot dalam Webots menggunakan bahasa pemrograman seperti C++, Python, atau MATLAB. Dengan program kontrol ini, dapat mengatur perilaku robot, membuat keputusan, dan mengendalikan sensor dan aktuator.

5. Interaksi dengan Robot: Webots menyediakan antarmuka yang memungkinkan berinteraksi dengan robot yang sedang disimulasikan. dapat mengontrol robot secara langsung, mengubah parameter, memulai atau menghentikan simulasi, dan melihat output dari sensor dan aktuator.

6. Skrip dan Pustaka: Webots mendukung penggunaan skrip dan pustaka eksternal untuk memperluas fungsionalitasnya. dapat menggunakan skrip untuk mengotomatisasi tugas-tugas tertentu, mengimpor pustaka eksternal, atau menggunakan perpustakaan khusus yang dikembangkan oleh komunitas Webots.

7. Integrasi ROS: Webots memiliki integrasi dengan ROS (Robot Operating System), yang memungkinkan menggunakan ROS dalam lingkungan simulasi. Ini memungkinkan mengembangkan dan menguji aplikasi ROS pada robot simulasi sebelum menerapkannya pada robot fisik.

Webots memungkinkan untuk memperoleh pemahaman mendalam tentang perilaku robot dan memvalidasi algoritma sebelum diterapkan pada robot fisik. Ini merupakan alat yang kuat untuk pengembangan dan eksperimen robotika, memungkinkan untuk merancang, menguji, dan mengoptimalkan aplikasi robot tanpa perlu menghabiskan waktu dan biaya pada perangkat keras fisik.

Code Analysis

Kode yang diberikan ditulis dalam bahasa VRML_SIM (Virtual Reality Modeling Language for Simulation) dan mewakili dunia simulasi dengan robot dan kotak kayu. Mari kita analisis kode tersebut baris per baris:

1. `WorldInfo`: Node ini memberikan informasi tentang dunia simulasi. Dalam hal ini, ia mengatur sistem koordinat menjadi "NUE".

2. `Viewpoint`: Node ini mewakili pandangan kamera dalam dunia simulasi. Ini mendefinisikan orientasi dan posisi pandangan.

3. `TexturedBackground` dan `TexturedBackgroundLight`: Node-node ini mendefinisikan latar belakang bertekstur untuk lingkungan simulasi.

4. `RectangleArena`: Node ini mewakili arena persegi panjang dalam simulasi.

5. `E-puck`: Node ini mewakili robot yang bernama E-puck. Ini menentukan translasi (posisi) robot dalam dunia simulasi dan pengontrol yang digunakan, yaitu "robot_motion".

6. `WoodenBox`: Node ini mewakili kotak kayu. Ini menentukan translasi (posisi), ukuran, dan massa kotak dalam dunia simulasi.

Secara keseluruhan, kode tersebut mengatur dunia simulasi dengan robot (E-puck) dan kotak kayu. Robot ditempatkan pada posisi (-0,22, 0, 0) dan menggunakan pengontrol "robot_motion". Kotak kayu ditempatkan pada posisi (0, 0,05, 0) dengan ukuran 0,1 x 0,1 x 0,1 dan massa 2.

Kode ini mewakili sebuah scene dasar dalam VRML_SIM, dan tergantung pada simulator atau viewer yang digunakan, kode ini akan merender dunia simulasi dan memungkinkan interaksi dan simulasi dengan robot dan kotak tersebut.