

TECHNICAL REPORT : Penggunaan ROS2, Gazebo untuk Pengendalian Robot 7 arm DOF

Dinda Rahma 1103204051

I. Pendahuluan

Pendahuluan: Laporan teknis ini bertujuan untuk menjelaskan penerapan contoh-contoh dalam buku "Mastering ROS2" dengan menggunakan simulasi Gazebo dan Webot. Dalam pengembangan sistem robotika yang kompleks, ROS2 telah menjadi kerangka kerja yang populer, sementara Gazebo dan Webot merupakan lingkungan simulasi yang umum digunakan. Dalam laporan ini, kami akan memberikan pemahaman yang komprehensif tentang struktur direktori, file-file yang relevan, dan langkah-langkah yang diperlukan untuk menjalankan contoh-contoh dalam simulasi Gazebo dan Webot. Analisis hasil implementasi dan perbandingan antara kedua simulasi akan dijelaskan, serta laporan ini akan disimpulkan dengan ringkasan temuan dan rekomendasi untuk pengembangan selanjutnya.

II. Deskripsi Buku "Mastering ROS"

a. ROS2 dan Fitur Utamanya

ROS2 (Robot Operating System 2) adalah sebuah kerangka kerja yang dirancang untuk pengembangan sistem robotika yang kompleks. ROS2 adalah evolusi dari ROS yang sebelumnya, dengan fokus pada peningkatan performa, skalabilitas, dan keandalan. Fitur-fitur utama ROS2 mencakup arsitektur yang modular, dukungan untuk komunikasi dan pertukaran data yang efisien, manajemen sumber daya yang lebih baik, dan peningkatan kemampuan real-time. ROS2 memungkinkan pengembang untuk mengembangkan sistem robotika yang lebih kompleks dengan lebih mudah dan lebih terstruktur.

b. Deskripsi Buku "Mastering ROS2"

Buku "Mastering ROS2" adalah sumber panduan yang komprehensif untuk mempelajari dan menguasai ROS2. Ditulis oleh penulis yang berpengalaman, buku ini memberikan wawasan mendalam tentang konsep, alat, dan teknik yang digunakan dalam pengembangan dengan ROS2. Buku ini mencakup berbagai topik penting seperti pengaturan lingkungan ROS2, penggunaan paket dan modul, komunikasi antar-node, navigasi, perencanaan gerakan, penginderaan, penggunaan simulasi, dan masih banyak lagi. Dengan penjelasan yang jelas dan contoh-contoh yang komprehensif, buku ini memberikan pembaca dengan pemahaman yang mendalam tentang penggunaan ROS2 dalam pengembangan sistem robotika.

c. Topik dan Bab dalam Buku

Buku "Mastering ROS2" mencakup sejumlah topik dan bab yang membahas berbagai aspek penting dalam pengembangan dengan ROS2.

Beberapa topik dan bab yang termasuk dalam buku ini antara lain: pengenalan ROS2, pengaturan lingkungan ROS2, komunikasi menggunakan ROS2 middleware, penggunaan paket dan modul, pengembangan aplikasi multi-robot, pengenalan MoveIt! framework untuk perencanaan gerakan, navigasi menggunakan ROS2, penggunaan sensor dan penginderaan dalam ROS2, penggunaan simulasi dengan Gazebo dan Webot, dan masih banyak lagi. Setiap bab memberikan penjelasan rinci, panduan langkah demi langkah, dan contoh-contoh yang memadai untuk membantu pembaca memahami dan menguasai konsep dan teknik yang terkait dengan pengembangan dengan ROS2.

III. Simulasi

a. Gazebo

Gazebo adalah sebuah simulator robotika yang sangat populer dan digunakan secara luas dalam pengembangan sistem robotika. Gazebo menyediakan lingkungan simulasi yang realistis dan mendetail, yang memungkinkan pengembang untuk menguji dan memvalidasi desain robotika sebelum mengimplementasikannya dalam dunia fisik. Dalam Gazebo, pengembang dapat membuat model robot, mensimulasikan interaksi fisik dengan lingkungan, mengatur sensor dan aktuator, serta menjalankan kontrol dan perencanaan gerakan. Gazebo juga mendukung pemodelan lingkungan yang kompleks, termasuk keadaan alam, objek, dan hambatan, sehingga memungkinkan pengembang untuk memperoleh pengalaman yang lebih realistis dalam pengembangan sistem robotika.

b. Kelebihan dan Kegunaan Simulasi Dalam Pengembangan Robotika

Penggunaan simulasi dalam pengembangan robotika memiliki sejumlah kelebihan dan kegunaan yang signifikan. Pertama, simulasi memungkinkan pengembang untuk menghemat biaya dan waktu dengan mengurangi ketergantungan pada perangkat keras fisik. Dengan simulasi, pengembang dapat menguji dan memvalidasi desain robotika tanpa memerlukan perangkat keras yang sebenarnya. Selain itu, simulasi memungkinkan pengembang untuk mengulang percobaan dengan mudah dan mengamati perilaku robot dalam berbagai skenario, sehingga memungkinkan peningkatan desain dan algoritma. Simulasi juga memberikan lingkungan yang aman untuk menguji perilaku robot dalam situasi yang mungkin berbahaya atau mahal dalam dunia fisik. Dengan menggunakan simulasi seperti Gazebo, pengembang dapat mempercepat proses pengembangan, mengoptimalkan kinerja sistem, dan memastikan keandalan robotika sebelum mengimplementasikannya dalam dunia nyata.

IV. Penggunaan Gazebo

a. Config

Pada tahap ini Robot 7 Arms DOF akan dikontrol. Dengan mengkonfigurasinya guna memungkinkan pengendalian dan pergerakan yang presisi pada Robot 7 Arms DOF. Berikut penjelasan mengenai konfigurasinya:

1. Bagian "joint_state_controller":
 - Kontroler ini bertanggung jawab untuk menerbitkan semua status sendi robot.
 - `publish_rate`: Kecepatan publikasi status sendi, dalam Hertz.
2. Bagian "Position Controllers":
 - Konfigurasi untuk masing-masing kontroler posisi sendi.
 - Setiap kontroler memiliki jenis yang ditentukan (misalnya, `JointPositionController`) dan sendi yang dikendalikan.
 - Parameter PID (p, i, d) digunakan untuk mengatur pengendalian posisi pada masing-masing sendi.
3. Bagian "seven_dof_arm":
 - Bagian ini mengatur kontroler pergerakan sendi lengan robot.
 - Menggunakan kontroler `JointTrajectoryController` untuk mengontrol pergerakan sendi berdasarkan trayektori yang ditentukan.
 - Daftar sendi yang dikendalikan dijelaskan dalam "joints" (`shoulder_pan_joint`, `shoulder_pitch_joint`, dll.).
 - Parameter "constraints" mengatur batasan pergerakan sendi, seperti waktu tujuan, toleransi kecepatan berhenti, dan batasan trayektori untuk masing-masing sendi.

b. Launch

Dalam buku tersebut terdapat beberapa file Launch. Berikut penjelasan dari file-file nya:

- `seven_dof_arm_bringup_moveit.launch`
File launch ini digunakan untuk menjalankan contoh yang berkaitan dengan pergerakan dan perencanaan gerakan menggunakan MoveIt! framework. Anda perlu menggunakan file ini jika Anda ingin mempelajari dan menjalankan contoh-contoh yang berkaitan dengan MoveIt!.
- `seven_dof_arm_bringup_obstacle_moveit.launch`
File launch ini digunakan untuk menjalankan contoh yang berkaitan dengan pergerakan dan perencanaan gerakan menggunakan MoveIt! framework dengan adanya hambatan (obstacle). Anda perlu menggunakan file ini jika Anda ingin mempelajari dan menjalankan contoh-contoh yang berkaitan dengan MoveIt! dan hambatan.

- `seven_dof_arm_gazebo_control.launch`

File launch ini digunakan untuk mengontrol dan menjalankan simulasi Gazebo untuk robot dengan tujuh derajat kebebasan (seven-dof arm). Anda perlu menggunakan file ini jika Anda ingin mempelajari dan menjalankan contoh-contoh yang melibatkan simulasi Gazebo.

- `seven_dof_arm_obstacle_world.launch`

File launch ini digunakan untuk menjalankan contoh yang melibatkan dunia simulasi dengan adanya hambatan (obstacle) pada bab 7. Anda perlu menggunakan file ini jika Anda ingin mempelajari dan menjalankan contoh-contoh yang melibatkan hambatan.

- `seven_dof_arm_with_rgbd_world.launch`

File launch ini digunakan untuk menjalankan contoh-contoh yang melibatkan dunia simulasi dengan penggunaan sensor RGB-D (warna dan kedalaman).

- `seven_dof_arm_world.launch`

File launch ini digunakan untuk menjalankan contoh-contoh pada dunia simulasi standar (tanpa hambatan atau sensor tambahan).

c. Worlds

Kode XML yang diberikan adalah deskripsi dunia simulasi dalam format SDF (Simulation Description Format) untuk Gazebo. Berikut adalah penjelasan singkat mengenai deskripsi tersebut:

1. Dunia Simulasi:

- Dunia simulasi diberi nama 'grasping'.
- Mendefinisikan sumber cahaya global menggunakan model 'sun' yang di-include.
- Mendefinisikan bidang tanah menggunakan model 'ground_plane' yang di-include.

2. Model 'table':

- Model ini merepresentasikan sebuah meja.
- Memiliki sebuah link dengan visual 'visual1' berbentuk kotak.
- Pose (posisi dan orientasi) dari link dan visual ditentukan.
- Model ini diatur sebagai statis (tidak bergerak).

3. Model 'ground':

- Model ini merepresentasikan permukaan tanah.

- Memiliki sebuah link dengan visual 'visual1' berbentuk kotak.
- Pose dari link dan visual ditentukan.
- Model ini diatur sebagai statis (tidak bergerak).

4. Pengaturan Fisika Simulasi:

- Menggunakan fisika tipe 'ode' (Open Dynamics Engine).
- Maksimum ukuran langkah simulasi adalah 0.001.
- Faktor waktu nyata adalah 1 (waktu simulasi berjalan seiring waktu nyata).
- Tingkat pembaruan waktu nyata adalah 1000 (kecepatan pembaruan waktu simulasi).
- Gravitasi diatur ke arah bawah dengan nilai 9.81.

Deskripsi ini menggambarkan dunia simulasi untuk Gazebo dengan meja, permukaan tanah, dan pengaturan fisika yang sesuai.

d. CMakeLists.txt

Ini adalah contoh CMakeLists.txt untuk paket ROS dengan nama 'seven_dof_arm_gazebo'. Berikut adalah penjelasan singkat mengenai CMakeLists.txt tersebut:

1. Minimum Version:

- Mengatur versi minimum dari CMake yang diperlukan (versi 3.0.2).

2. Pencarian Dependensi:

- Menggunakan `find_package` untuk mencari dependensi ROS seperti `gazebo_msgs`, `gazebo_plugins`, `gazebo_ros`, `gazebo_ros_control`, dan `mastering_ros_robot_description_pkg`.

3. Deklarasi Pesan, Layanan, dan Aksi ROS:

- Terdapat komentar yang menjelaskan cara mendeklarasikan dan membangun pesan, layanan, dan aksi ROS dari dalam paket ini.

4. Deklarasi Parameter Rekonfigurasi Dinamis ROS:

- Terdapat komentar yang menjelaskan cara mendeklarasikan dan membangun parameter rekonfigurasi dinamis ROS dari dalam paket ini.

5. Konfigurasi Spesifik catkin:

- Melakukan deklarasi menggunakan ``catkin_package`` untuk menghasilkan berkas konfigurasi CMake pada paket ini.
- Berkas konfigurasi akan mencakup informasi tentang direktori ``include``, library, dependensi, dan lainnya.

6. Build:

- Menentukan lokasi tambahan dari berkas header dengan menggunakan ``include_directories``.
- Ada beberapa contoh komentar yang menjelaskan cara menambahkan pustaka C++ atau eksekutor C++.

7. Install:

- Terdapat beberapa contoh komentar yang menjelaskan cara melakukan instalasi pada paket ini, seperti instalasi skrip eksekusi Python atau skrip eksekusi lainnya.

8. Testing:

- Terdapat beberapa contoh komentar yang menjelaskan cara menambahkan pengujian unit menggunakan GTest atau pengujian lainnya.

Kode tersebut merupakan contoh CMakeLists.txt yang dijelaskan dalam komentar dan beberapa bagian di dalamnya masih di-komentari. Anda dapat menghilangkan komentar dan mengonfigurasi sesuai dengan kebutuhan proyek Anda.

e. package.xml

Ini adalah contoh berkas ``package.xml`` untuk paket ROS dengan nama `'seven_dof_arm_gazebo'`. Berikut adalah penjelasan singkat mengenai isi berkas tersebut:

1. Nama, Versi, dan Deskripsi Paket:

- ``<name>``: Nama paket (`seven_dof_arm_gazebo`).
- ``<version>``: Versi paket (`0.0.0`).
- ``<description>``: Deskripsi singkat tentang paket ini.

2. Informasi Pemelihara:

- ``<maintainer>``: Informasi pemelihara paket (nama dan email).

3. Lisensi:

- ``<license>``: Lisensi yang digunakan oleh paket (dapat diganti dengan nama lisensi yang sesuai).

4. URL dan Author (Opsional):

- `<url>`: URL yang terkait dengan paket (contoh: situs web, pelacak bug, atau repositori).
- `<author>`: Informasi penulis atau kontributor lainnya (nama dan email).

5. Ketergantungan:

- Terdapat beberapa tag `<build_depend>`, `<build_export_depend>`, dan `<exec_depend>`. Di dalam tag-tag tersebut, Anda dapat menambahkan ketergantungan paket ROS atau ketergantungan sistem yang diperlukan oleh paket ini.

6. Ekspor:

- `<export>`: Bagian ini dapat digunakan untuk menambahkan informasi tambahan yang diminta oleh alat atau sistem lain terkait paket ini.

V. Kesimpulan

ROS (Robot Operating System) adalah sebuah kerangka kerja perangkat lunak yang digunakan dalam pengembangan robotika. Dengan menggunakan model komputasi terdistribusi, ROS memungkinkan pengembang untuk menghubungkan berbagai proses yang saling berkomunikasi melalui pertukaran pesan. Dengan dukungan dari Catkin sebagai sistem pembangunan, pengembang dapat mengelola dependensi, mengompilasi kode sumber, dan membangun paket ROS. Dalam konteks pengembangan robotika, ROS bekerja sama dengan Gazebo, sebuah simulator fisika open-source. Gazebo digunakan untuk mensimulasikan robot di lingkungan yang realistis sebelum implementasi fisiknya. Dengan menggunakan file launch (*.launch) dan world (*.world), pengembang dapat mengkonfigurasi simulasi Gazebo dengan memuat model robot, mengatur parameter sistem, dan menggambarkan lingkungan simulasi. Melalui kombinasi ROS dan Gazebo, pengembang dapat lebih efisien dalam mengembangkan sistem robotika yang kompleks dan terintegrasi, serta menguji dan mengoptimalkan sistem mereka sebelum melakukan implementasi fisik, yang pada akhirnya menghemat waktu dan biaya dalam proses pengembangan.

VI. Referensi

- Joseph, L., & Cacace, J. (2020). Mastering ROS for Robotics Programming: Third Edition. Packt Publishing.
- ROS Wiki. (n.d.). ROS - Robot Operating System. Diakses pada 23 Juni 2023, dari, <http://wiki.ros.org/>
- Gazebo. (n.d.). Gazebo: A dynamic multi-robot simulator. Diakses pada 23 Juni 2023, dari, from <http://gazebosim.org/>