

Nama : Agung Reynaldi Avizena

NIM : 1103204044

UAS Robotic - Mastering ROS for Robotics Programming

1. Chapter 1 - Introduction to ROS

Elemen	Deskripsi
Technical Requirements	Memerlukan komputer dengan OS Ubuntu 20.04 LTS atau distribusi Debian 10 GNU/Linux.
Why Learn ROS?	Sistem Operasi Robot (ROS) adalah kerangka kerja fleksibel untuk menulis perangkat lunak robot. Tujuannya adalah membentuk cara standar memprogram robot dengan komponen perangkat lunak siap pakai, dukungan sensor dan aktuator canggih.
ROS Origins	Proyek ROS dimulai pada tahun 2007 oleh Morgan Quigley di Willow Garage. Bertujuan menetapkan standar memprogram robot dan menawarkan perangkat lunak siap pakai untuk integrasi mudah.
ROS Advantages	- Kemampuan kelas atas dengan fungsi siap pakai seperti SLAM dan AMCL. - Banyak alat debugging dan simulasi dalam ekosistem ROS. - Dukungan sensor dan aktuator kelas atas.
ROS Metapackages	Metapaket ROS mengelompokkan sejumlah paket menjadi satu paket logis. File package.xml dalam metapaket berisi tag ekspor dengan <metapackage/> untuk menunjukkan statusnya.
Running ROS Master and Parameter Server	Sebelum menjalankan node ROS, roscore harus dijalankan untuk memulai ROS master dan server parameter. Roscore juga memulai node rosout untuk pengumpulan log pesan dari node lain.

2. Chapter 2 - Getting Started with ROS Programming

Bab ini memperkenalkan proses pembuatan paket ROS (Robot Operating System), yang berfungsi sebagai unit dasar pemrograman ROS. Laporan ini berfokus pada distribusi Noetic Ninjemys dan menggunakan sistem pembangunan catkin untuk konstruksi paket.

- Setelah menginstal ROS, inisiasilah pembuatan ruang kerja catkin dengan nama 'catkin_ws' menggunakan perintah: catkin init.
- Siapkan lingkungan ruang kerja dengan menjalankan salah satu file konfigurasi menggunakan perintah: source devel/setup.bash.

c. Membuat Paket ROS:

- Perintah `catkin_create_pkg` adalah cara nyaman untuk membuat paket ROS. Jalankan perintah ini dalam direktori 'src' dari ruang kerja catkin.
- Bangun paket dengan perintah `catkin_make` dari direktori ruang kerja catkin.

d. Node ROS - Publisher:

- Jelajahi pembuatan node ROS sederhana, `demo_topic_publisher.cpp`, yang menerbitkan nilai integer pada topik `/numbers`.
- Manfaatkan tipe pesan `std_msgs/Int32` untuk menangani data integer.
- Inisialisasi node ROS dengan nama unik, buat penerbit, dan terbitkan data dengan tingkat yang ditentukan.

e. Node ROS - Subscriber:

- Perkenalkan node ROS lainnya, `demo_topic_subscriber.cpp`, yang berlangganan ke topik `/numbers`.
- Implementasikan fungsi panggilan kembali untuk memproses pesan yang diterima dan menampilkannya menggunakan `ROS_INFO`.
- Inisialisasi node ROS, buat pelanggan, dan mulai putar ROS untuk menangani pesan yang masuk.

f. Modifikasi `CMakeLists.txt`:

- Edit file `CMakeLists.txt` di direktori paket untuk mengompilasi dan membangun kode sumber.

g. Build Workspace:

- Gunakan perintah `catkin_make` untuk membangun seluruh ruang kerja atau tentukan paket menggunakan opsi `-DCATKIN_WHITELIST_PACKAGES`.

h. Running ROS nodes:

- Pastikan master ROS berjalan dengan perintah `roscore`.
- Jalankan node penerbit dan pelanggan menggunakan perintah `roslaunch`.

i. Node Communication:

- Diagram menggambarkan alur komunikasi antara node penerbit dan pelanggan. Penerbit (`demo_topic_publisher`) menerbitkan pada topik `/numbers`, dan pelanggan (`demo_topic_subscriber`) menerima dan memproses pesan.

3. Chapter 3 - Working with ROS for 3D Modeling

Bab ini, kita akan membahas beberapa paket ROS penting yang umum digunakan untuk membangun dan memodelkan robot dalam lingkungan tiga dimensi (3D). Fokusnya terutama pada paket URDF (Unified Robot Description Format) dan paket terkait dalam ekosistem ROS.

1. Paket URDF:

- `urdf`: Paket inti untuk memodelkan robot dalam ROS. Berisi parser C++ untuk URDF, yang merupakan file XML yang merepresentasikan struktur dan geometri robot.
- `urdf_parser_plugin`: Mengimplementasikan metode untuk mengisi struktur data URDF.
- `urdfdom_headers`: Header struktur data inti untuk menggunakan parser URDF.
- `collada_parser`: Mengisi struktur data dengan mem-parsing file Collada.
- `urdfdom`: Mengisi struktur data dengan mem-parsing file URDF.

2. Paket Pendukung:

- `joint_state_publisher`: Alat penting untuk merancang model robot menggunakan URDF. Mencakup node `joint_state_publisher`, yang membaca deskripsi model robot dan menerbitkan nilai sendi.
- `joint_state_publisher_gui`: Mirip dengan paket `joint_state_publisher`, tetapi juga menyediakan antarmuka pengguna (GUI) dengan slider untuk berinteraksi dengan setiap sendi robot.

3. Persiapan Lingkungan:

- Buat paket ROS di ruang kerja catkin dengan dependensi pada paket seperti `urdf`, `xacro`, dan `roscpp`.
- Pastikan paket yang diperlukan terinstal menggunakan manajer paket ROS.

4. Implementasi:

- Desain dan buat file URDF untuk robot dalam paket ROS yang telah dibuat.

- Gunakan xacro untuk merancang file URDF dan konversikan ke file URDF menggunakan perintah `roslaunch xacro`.
- Tampilkan model robot dalam RViz menggunakan file peluncuran dan perintah `roslaunch`.

5. Contoh Penggunaan:

- View Demo: Gunakan file peluncuran `view_demo.launch` untuk menampilkan model robot di RViz.
- View Pan Tilt Xacro: Konversikan file xacro menjadi URDF dan tampilkan di RViz.
- View Arm: Meluncurkan model robot dengan GUI `joint_state_publisher` di RViz.

4. Chapter 4 - Simulating Robots Using ROS and Gazebo

Pada bagian ini, kita membahas simulasi robot lengan dengan tujuh derajat kebebasan di Gazebo menggunakan ROS. Instalasi paket Gazebo dan ROS dilakukan sebagai langkah persiapan. Pembuatan model simulasi melibatkan pembaruan deskripsi robot dan penambahan parameter simulasi. Simulasi ini memungkinkan pengguna untuk memodelkan pergerakan robot dan mensimulasikan berbagai jenis sensor di lingkungan Gazebo.

1. Persiapan Lingkungan:

- Instal paket-paket Gazebo dan ROS yang diperlukan menggunakan perintah `sudo apt-get install`.
- Periksa instalasi Gazebo dengan menjalankan perintah `roslaunch gazebo_ros gazebo`.

2. Pembuatan Paket ROS:

- Buat paket ROS untuk mensimulasikan lengan robot di Gazebo dengan dependensi pada paket seperti `gazebo_msgs` dan `gazebo_ros_control`.
- Klon paket lengkap dari repositori Git atau dapatkan dari sumber buku ini.

3. Simulasi Robot 7-DOF di Gazebo:

- Gunakan perintah `roslaunch` untuk meluncurkan simulasi dengan file konfigurasi `seven_dof_arm_world.launch`.
- Periksa visualisasi lengan robot di Gazebo.

4. Sensor di Gazebo:

- Simulasikan sensor RGBD di Gazebo dengan meluncurkan simulasi menggunakan `seven_dof_arm_with_rgbd_world.launch`.
- Periksa topik-topik yang dihasilkan oleh plugin sensor.

5. Visualisasi Data Sensor:

- Lihat data gambar dari sensor visi 3D menggunakan perintah `roslaunch image_view`.
- Gunakan RViz untuk melihat data awan titik dari sensor visi 3D.

6. Pengendalian Sendi Robot:

- Untuk menggerakkan sendi robot, menerbitkan nilai sendi yang diinginkan ke topik perintah pengendali posisi sendi.
- Contoh: `rostopic pub /seven_dof_arm/joint4_position_controller/command std_msgs/Float64 1.0`.
- Periksa status sendi robot menggunakan perintah `rostopic echo`.

5. Chapter 5 - Simulating Robots Using ROS, CoppeliaSim, and Webots

Sebelum memulai pekerjaan dengan CoppeliaSim dan Webots, instalasi dan konfigurasi lingkungan yang sesuai diperlukan untuk memungkinkan jembatan komunikasi antara ROS dan simulator simulasi. CoppeliaSim dan Webots adalah simulator lintas platform yang mendukung pengembangan dan simulasi robot. Panduan instalasi dan konfigurasi diberikan untuk memulai integrasi dengan ROS.

1. Instalasi dan Konfigurasi CoppeliaSim:

- Unduh versi terbaru CoppeliaSim dari situs resmi Coppelia Robotics.
- Ekstrak arsip unduhan dan konfigurasikan variabel lingkungan `COPPELIASIM_ROOT`.
- Pastikan ROScore berjalan sebelum membuka CoppeliaSim dengan perintah `./coppeliaSim.sh`.

2. Simulasi Kamera dan Pengolahan Pesan:

- Coppeliasim digunakan untuk mensimulasikan kamera pasif yang menerima data visi dari kamera aktif. Visualisasi data menggunakan paket `image_view` dengan perintah `roslaunch image_view image:=/camera/image_raw`.

3. Penerbitan Pesan dalam Skrip Lua:

- Untuk menerbitkan pesan ROS dari skrip Lua, pesan perlu dibungkus dalam struktur data yang sesuai. Demikian pula, informasi yang diterbitkan pada topik ROS harus dikumpulkan.

4. Integrasi dengan URDF:

- Model robot dengan tujuh derajat kebebasan dapat diimpor ke Coppeliasim dengan mengonversi model `xacro` menjadi file URDF.

5. Instalasi dan Penggunaan Webots:

- Instal Webots dari sumber web resmi atau menggunakan manajer paket Debian/Ubuntu APT.
- Otentikasi repositori Cyberbotics dan instal Webots dengan perintah `sudo apt-get install webots`.
- Jalankan Webots dari terminal dengan perintah `webots`.

6. Simulasi dengan Webots:

- Webots menyediakan antarmuka pengguna untuk mengontrol simulasi dan memodifikasi parameter robot.
- Simulasi dapat diatur, dimulai, dihentikan, atau dipercepat melalui menu simulasi di antarmuka Webots.