Gazebo and ROS

ROS-GAZEBO Interactions and Sample Applications
ROS Practical Trainings 2019, Eskişehir

Supported by ROSIN - ROS-Industrial Quality-Assured Robot Software Components.
More information: rosin-project.eu

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 732287.

Annex_3_3_Gazebo_eng.pptx

Content

- **What is Gazebo?**
- **Gazebo Components**
  - World Files
  - Model Files
  - Environment Variables
  - Gazebo Server
  - Gazebo Client
  - Plugins
- **Gazebo Applications**
  - Gazebo Setup
  - Running Gazebo
  - Preparing Workplace
  - Creating World
  - Creating Model
  - Importing Meshes
  - Adding Sensors
  - **Control with ROS**

What is Gazebo?

Robust physics engine for simulating robots in indoor and outdoor environments, provides the necessary infrastructure with high quality graphics and graphic interface.

Gazebo;

- has high performance physic engines like ODE, Bullet, Simbody ve DART
- creates a realistic environment with OGRE
- enables sensor ve sensor data usage,
- usage of existing robot models or the possibility to create special robot models

Gazebo Components

A simulation performed with Gazebo includes following items:

- World Files
- Model Files
- Environment Variables
- Gazebo server
- Gazebo client
- Plugins

World Files

World Files, define environment models in simulation environment like;

- robot,
- sensor,
- light,
- object

These files are created using the definition format named SDF and generally have the .world extension.

World Files

Various world files comes with Gazebo setup and these are located in

<install_path>/share/gazebo-<version>/worlds

It is also possible to create user-defined world files using the SDF definition format.

World Files

An example world file "empty.world" has the following content.

```
<?xml version="1.0" ?>
<sdf version="1.5">
        <world name="default">
        <!-- A global light source -->
                <include>
                        <uri>model://sun</uri>
                </include>
        <!-- A ground plane -->
                <include>
                        <uri>model://ground_plane</uri>
                </include>
        </world>
</sdf>
```

Model Files

model files use SDF definition format like World files and only contain

```
<model>
        ...
</model>
```

style tags.

The purpose of using these files is to make models reusable and to simplify world files.

Model Files

A model created can be used in the world file as follows.

```
<include>
        <uri>model://model_file_name</uri>
</include>
```

For the simulation environment, model files that are included in the online database or included with the installation in old Gazebo versions can be used, or user-defined model files can be used.

Environment Variables

Gazebo uses various environment variables to locate and access files and establish the relationship between server and client. The environment variables come compiled with the installation.

The default environment variables are located in the shell script (.sh file) below.

<install_path>/share/gazebo/setup.sh

To change environmental variables that Gazebo will consider flollowing code

source <install_path>/share/gazebo/setup.sh

should be referenced and should be modified optionally.

Environment Variables

An example script file content is shown below.

```
export GAZEBO_MASTER_URI=http://localhost:11345
export GAZEBO_MODEL_DATABASE_URI=http://gazebosim.org/models
export GAZEBO_RESOURCE_PATH=/usr/share/gazebo-7:/usr/share/gazebo_models:${GAZEBO_RESOURCE_PATH}
export GAZEBO_PLUGIN_PATH=/usr/lib/x86_64-linux-gnu/gazebo-7/plugins:${GAZEBO_PLUGIN_PATH}
export GAZEBO_MODEL_PATH=/usr/share/gazebo-7/models:${GAZEBO_MODEL_PATH}
export LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:/usr/lib/x86_64-linux-gnu/gazebo-7/plugins
```

Gazebo Server

The Gazebo server is the part of Gazebo that takes over the entire processing load. It parses a world file given to it and then simulates the world using a physics and sensor engine.

Gazebo server does not contain any graphic interface.

Gazebo Server

Gazebo server can be started simply with command

```
gzserver
```

and also

```
gzserver worlds/empty.world
```

command can also start a world file as well(in this example, a world file that comes with Gazebo)

Gazebo Client

The Gazebo client takes over the task of connecting to a working Gazebo server and visualizing the elements of the environment.

It also offers the user the opportunity to make changes to a working simulation.

Gazebo server can simply be run with

```
gzclient
```

command in terminal.

Plugins

The plugins are compiled as common library files and are treated as necessary code snippets. The plugins are available in all Gazebo over standard C ++ classes.

The plugins allow to control the direction of Gazebo and can be added to and removed from a working system.

Plugins

Plugins can be used when something is desired to be programmatically modified in the simulation (moving models, adding new models when certain conditions are met, etc.), when a fast interface is requested for Gazebo, and the created plug-in is desired to be useful for others.

Currently there are six types of plugins exist.

- World
- Model
- Sensor
- System
- Visual
- Interface

Plugins

Each plugin type is managed by a different Gazebo component. For example, a model plugin is added to a specific model in Gazebo and controls that model.

The plug-in type should be selected according to the desired functionality. For example;

- A pyhsic plugin for controlling world properties engine environment limunation etc. dünya özelliklerini kontrol etmek için bir dünya eklentisi,
- A model plugin to check the condition of joints and model,
- A sensor plugin to get sensor information and check sensor properties

Plugins

Plugins can be loaded from the terminal command line as follows;

```
gzserver -s <plugin_filename>
```

Also, it can be loaded in the SDF file by specifying as follows;

```
<plugin name="gazebo_ros_control" filename="libgazebo_ros_control.so">
        <robotNamespace>/MYROBOT</robotNamespace>
</plugin>
```

Setting up Gazebo

Gazebo is an independent project used by ROS.

Usually, the newest Gazebo version available at the beginning of each ROS release cycle is officially selected to be fully integrated and supported.

Setting up Gazebo

Packages can be used to install Gazebo on Ubuntu. There are two main sources (packages.ros.org and packages.osrfoundation.org) that host Gazebo packages.

- on packages.ros.org
  - ROS Indigo: Gazebo 2.x
  - ROS Kinetic: Gazebo 7.x
  - ROS Lunar: Gazebo 7.x
  - ROS Melodic: Gazebo 9.x
- on packages.osrfoundation.org
  - gazebo 7.x (gazebo7 package name)
  - gazebo 8.x (gazebo8 package name)
  - gazebo 9.x (gazebo9 package name)

packages are available. any of these sources can be used to install Gazebo.

Setting up Gazebo

For users who need to run a specific version of ROS and want to use all the packages related to Gazebo ROS ready to use, it is recommended to use the Gazebo version available at packages.ros.org. For example, ROS Kinetic hosts and uses the 7.x version of Gazebo.

Setting up Gazebo

If needed, it is possible to use a certain Gazebo and ROS version together, apart from the recommended options.

However, in this case, any ROS Ubuntu package related to Gazebo is not available from the ROS distribution source.

Equivalent of packages can be loaded from OSRF source, but all other software must be created from the source using catkin_workspaces.

Also, if there is a Gazebo version that comes with the current ROS installation, it may be necessary to uninstall this version first when a different Gazebo version is desired.

Running Gazebo

From Terminal

```
gazebo
```

command can start Gazebo.

As can be seen in the graphic interface, it is possible to modify the running simulation to a certain extent thanks to the Gazebo client.

Preparing Workplace

In order to perform ROS-Gazebo studies, Catkin working environment can be prepared.

Creating World

While creating the world to be used in the simulation environment, the ready world files in the Gazebo database can be used as well as the user can create their own world.

World File can be created;

- via graphical interface,
- manually with SDF format compatible descriptions

Model Creation

While creating the world for models to be used in the simulation environment, the model file;

- via graphical interface provided “Model Editor”
- manually by proper descriptions compatible with SDF, URDF or URDF.XACRO

Files created using the SDF format can be used directly with Gazebo, while files in URDF and URDF.XACRO format may need extra processing.

Importing Mesh

While creating the models, a model file can be defined on the <visual> tag and the model can be printed on the model.

If the <visual> tag is defined, only the visual feature of the model is affected.

Adding Sensor

While creating models, a sensor can be added to the model by defining <link>, <joint> and if necessary, <gazebo reference> for the model file for .urdf files.

Control with ROS

To be able to control a model with ROS in a simulation environment, the related packages (gazebo_ros_pkgs) must be installed first.

When the Catkin environment is created and ROS and Gazebo installations and configurations are performed properly

```
rosrun gazebo_ros gazebo
```

command will successfully run Gazebo with ROS integration.