Software Architectures for Robotics

Lab Session 5

Simulation environment

UNIVERSITÀ DEGLI STUDI DI GENOVA | Dibris

EMARO LAB

Table of contents

Modeling and simulations

Testing complex and delicate systems require simulations!

- Physics engines use mathematical models and laws in order to "update" the world state (pose, velocity, force)

- Physics engines are often coupled with rendering environments in order to manage graphics.

Physics + Graphics = Simulator

Available simulators

- Gazebo: official ROS simulator
- VRep: cross platform professional simulator
- Unity3d: game engine (Kerbal space program, Firewatch, Monument Valley, Temple run, Assassin creed)
- Unreal Engine: game engine (Unreal tournament, Mass effect, Batman Arkam city, Tekken 7)

Gazebo

Gazebo is composed by:

- `gzserver`: emulates physics, light, sensors...
- `gzclient`: display the graphics and tools for interacting with the simulation

The client and the server communicate using the gazebo communication library (ignition)

Gazebo II

The physics library is integrated with four open-source physics engines:

- Open Dynamics Engine (ODE)
- Bullet
- Simbody
- Dynamic Animation and Robotics Toolkit (DART)

Gazebo: installation and startup

Gazebo7 comes with ROS full desktop version.

Run the simulator with:

```
rosrun gazebo_ros gazebo
```

For any trouble please refer to: gazebo tutorials

Robot description

Gazebo represents robot configurations by the use of XML files in different format:

- URDF (Rviz)
- XACRO (Gazebo)
- SDF (Gazebo)

Configuration files are used to describe the simulated robot: number/type of joints, geometry, links, actuators, plugins ....

Robot description II

Formats have small differences between them but the concept is the same

XML description files are divided in sections describing a single element of the model

SDF is well documented here and supported by the gazebo foundation

Creating a URDF file

- File: robot1.urdf

- Links and joints descriptions (type, position ...)

- Visual field contains the geometry (box, sphere, mesh...) and set the material/color

Joints

In the joint block we define:

- Type (prismatic, revolute ...)
- Parent (base frame)
- Child (target frame)
- Limits (rotation / translation if any)

Joints: types

In the joint block we define:

- Revolute: a hinge joint that rotates along the axis - limited range
- Continuous: revolute with no limits
- Prismatic: translates along the axis
- Fixed: blocked, no degrees of freedom
- Floating: 6 degrees of freedom
- Planar: it moves along a plane

Check the format documentation for more precise information

URDF example

The ROS parameter *"robot_description"* should point to the urdf file

Check and launch `launch/display.launch`

1. joint_state_publisher: sends command from a rosnode
2. robot_state_publisher: sends commands from gui

URDF: remarks

- URDF is used for visualization in RViz, you can open them from gazebo but it is tricky..
- XACRO are used to generate URDF files
- SDF is the preferred type for simple models
- The gazebo model editor helps you with a gui during model creation

URDF: remarks

- Check URDF syntaxt: `check_urdf robot1.urdf`
- Visualize URDF: `urdf_to_graphiz robot1.urdf`

Spawn a URDF inside gazebo with:

```
rosrun gazebo spawn_model -file path/to/urdf.urdf
-urdf -model model_name
```

Let's follow a tutorial ...

http://gazebosim.org/tutorials/?tut=ros_urdf