A Python library for scripting and rapid-prototyping of Gazebo simulations

Musa Morena Marcusso Manhães

Bosch Corporate Research – Renningen, Germany
Robotics Systems Department (CR/AER)

7th ROS-Industrial Conference – Stuttgart, Germany
INTRODUCTION
Introduction
The ideal process for the development of a Gazebo simulation

Models
- Geometries
- Meshes
- Kinematics
- Parameters
  - SDF / URDF
    - Gazebo model

World
- Model assets
  - World layout
  - Physics engine configuration
  - SDF
    - Gazebo world
Introduction

The actual process for the development of a Gazebo simulation

Models

- Geometries
- Meshes
- Kinematics
- Parameters

→ SDF / URDF

→ Gazebo model

World

- Model assets
- World layout
- Physics engine configuration

→ SDF

→ Gazebo world

Fixing materials for meshes for better visualization

Instability due to wrong physics parameters

Instability due to big differences in moments of inertia for connected links

Multiple iterations testing the model in Gazebo for errors in poses, meshes, physics parameters, scaling

Finding the correct physics engine configuration

Placement of models without explicit measurements on the GUI

Tuning of parameters for different physics engines
Introduction

Application-dependent difficulties

- Generation variations of worlds and models (e.g. object placement, model geometry, physics engine configuration)
- Scripting of world layouts and event-based actions
- SDF allows more control of the model and its parametrization regarding physics, but most of robot descriptions are written in URDF and don’t use SDF to its full potential
- The differences between SDF and URDF morphology
Introduction

Approach: Procedural Generation

- Technique from gaming development
- Rapid-prototyping of simulation scenarios
- Abstractions to simulation entities
- Allow scripting of Gazebo simulations (generation of models, setting/accessing parameters in runtime, interacting with simulation via script)
- Extend templating options for robot descriptions
- Improve conversion between URDF and SDF formats for better use of Gazebo’s features
FEATURES
• List all static Gazebo models found on the resources path
• Load the Gazebo model using its tag name
• Visualize visual and collision geometries
• Access and edit SDF parameters without editing the file
- Open Gazebo inside the notebook
- Create box model and spawn it in Gazebo with 5 different friction coefficients
- Apply force to each box
- Create model with chassis, 2 wheels, caster wheel with friction set to zero
- Spawn it
- Modify the model by adding IMU, contact and camera sensors
- Spawn modified model
- Export generated model as Gazebo model
• Use model factory to create single-link models (box, sphere, cylinder, from mesh, from extruded polygon)
• Use lambda functions to dynamically generate the parameters
- Load xacro files and its generated URDF model
- Parse Jinja templates with pcg extensions for generation of model and world configurations
- Call ROS processes from the notebook to interact with the robot
Features

World generation

Collision geometries

Original setup

Collision geometries
Features

Pose randomization

- Dining room
- Kitchen
- Living room
Features
Generating grid maps from Gazebo worlds via ray tracing
Conclusion

- `pcg_gazebo_pkgs` can be used for testing simulation scenarios without editing XML files.
- Scripting can be used to generate assets and interact with the simulation.
- Dynamic model and world generation allows generation of large number of assets with small effort for testing robotics systems solutions in various contexts.
- Python libraries can be used on the simulation building process, along with Jupyter notebooks.
- Model editing and inspection.
- `sdf2urdf` and `urdf2sdf` give more possibilities of ways to represent the robot description.
- Package available at [https://github.com/boschresearch/pcg_gazebo_pkgs](https://github.com/boschresearch/pcg_gazebo_pkgs) under Apache-2.0 license.
THANK YOU

CONTACT
Musa Morena Marcusso Manhães (musa.marcusso@de.bosch.com)

REPOSITORY
https://github.com/boschresearch/pcg_gazebo_pkgs