Second Contact with Gazebo

Gazebo free simulator

Fragkoulis Logothetis
Electrical and Computer Enginnering
Chania
AM:2013030016

Abstract

The target of this exercise is to exercise ourselves on Gazebo Tools .It's important during learning a Simulator to build some useful plug-in like physics ,sensors etch.

Gazebo Plug-in

• ***world:*** It is incorporated in Gazebo Worlds and it can motivates every World dynamically

• ***model***: It is incorporated in a model and it controls the motions and the joints

• ***sensor:*** It is incorporated in a model and the Developer can easily take its measurements.

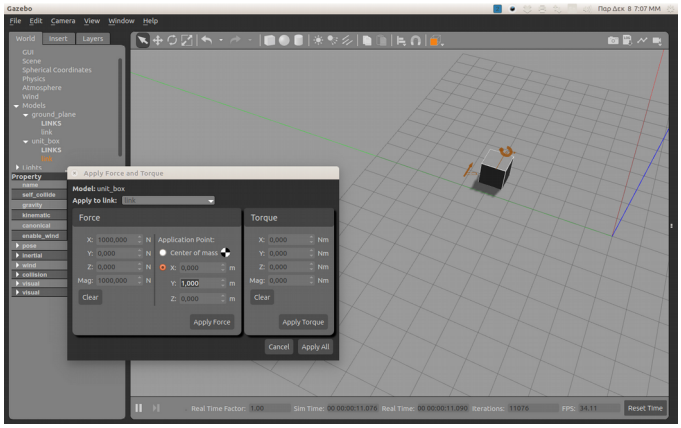
• ***visual:*** It is used to visualize Words and Objects

• ***GUI***: Graphical Interface of the Simulator

I. Analyze Parts of the Project

A. Applying Force/Torque

Gazebo supports a GUI to apply force and/or torque to models during simulation using the graphical user interface.

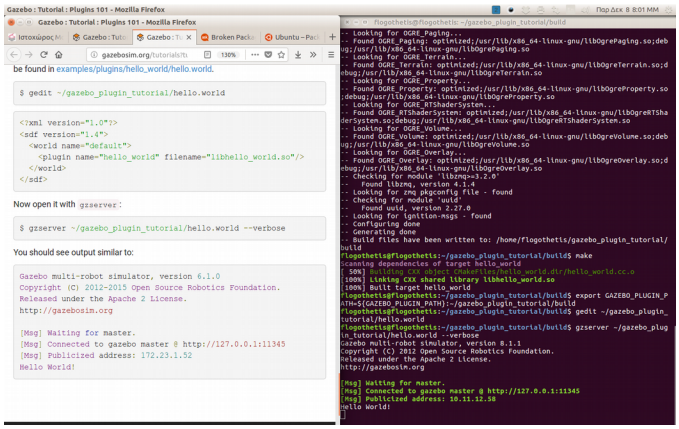


II. Plugins 101(Hello World Plugin)

A plugin is a chunk of code that is compiled as a shared library and inserted into the simulation. The plugin has direct access to all the functionality of Gazebo through the standard C++ classes.

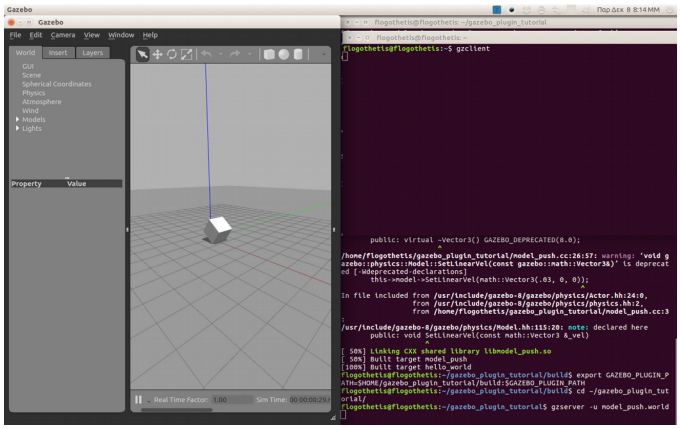
Plugins are useful because they:

- let developers control almost any aspect of Gazebo
- are self-contained routines that are easily shared
- can be inserted and removed from a running system



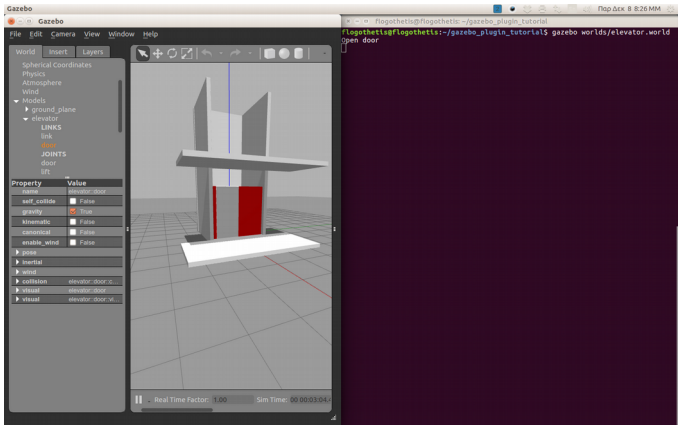
III. Model Plugins

Plug -ins allow complete access to the physical properties of models and their underlying elements (links, joints, collision objects). The following plugin will apply a linear velocity to its parent model.



IV.Occupied Event(Elevetor)

The elevator plugin works well the Occupied Event component of the SimEventPlugin. The Occupied Event sends a message whenever a 3D region becomes occupied with a model.



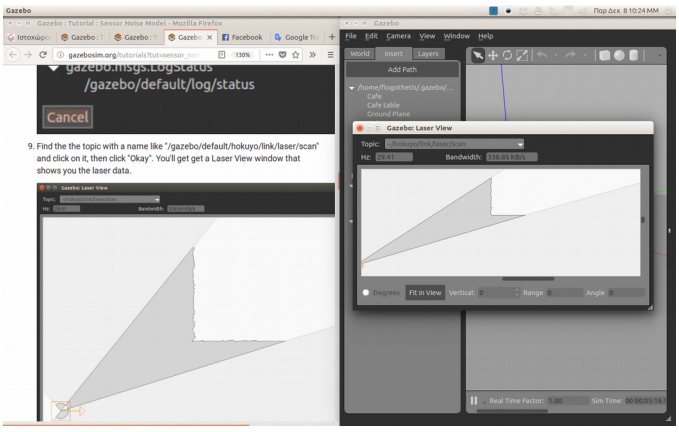
V. Sensor Noise Model

Gazebo provides models of many common sensors. In the real world, sensors exhibit noise, in that they do not observe the world perfectly. By default, Gazebo's sensors will observe the world perfectly (though not the IMU; read more below). To present a more realistic environment in which to try out perception code, we need to explicitly add noise to the data generated by Gazebo's sensors.

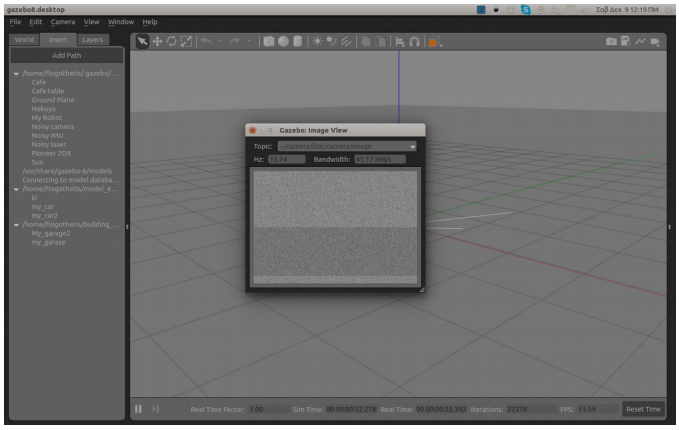
At the time of writing, Gazebo can add noise to the following types of sensors:

- Ray (e.g., lasers)
- Camera
- IMU

RAY (LASER) NOISE

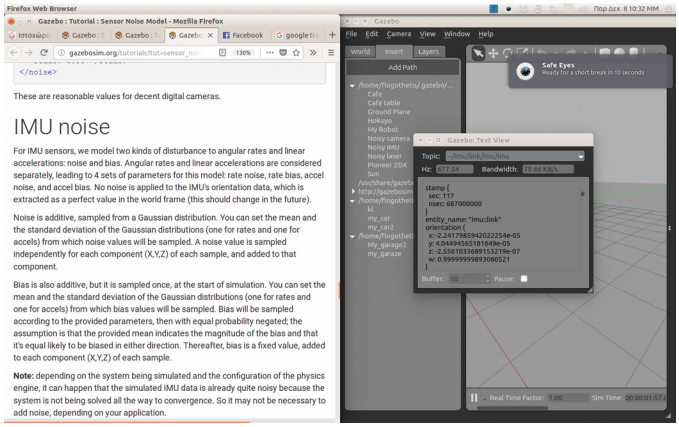


CAMERA NOISE



IMU NOISE

For IMU sensors, we model two kinds of disturbance to angular rates and linear accelerations: noise and bias. Angular rates and linear accelerations are considered separately, leading to 4 sets of parameters for this model: rate noise, rate bias, accel noise, and accel bias.



VI.CONTACT SENSOR PLUGIN

It is also possible to create a plugin for the contact sensor. This plugin can get the collision data, manipulate it, and output it to an arbitrary destination (for example a ROS topic).