Technical University of Crete
School of Electrical and Computer Engineering

INF 412 – Autonomous Agents – 2023

Instructor: M. G. Lagoudakis **3rd Laboratory Exercise** Deadline: 09.12.2023, 23:59

Introduction

The aim of this laboratory exercise is to familiarize yourself with one of the most widespread robot simulators, the **Gazebo** simulator [http://gazebosim.org/], which was initially developed at the University of Southern California (USC) in the USA in 2002 by Nate Koenig (PhD candidate) and Andrew Howard (supervising Professor). After several phases of development, in 2012 it was brought under the auspices of the aforetime Open Source Robotics Foundation (OSRF), now Open Robotics [www.openrobotics.org], and is now being developed at a steady pace by an experienced team of developers. In 2019 a new robot simulator project started by the same team under the name *Ignition*; in 2022 the new project was renamed simply *Gazebo*, leaving the old project with the commonplace name *Gazebo Classic*.

The Gazebo simulator (new or classic) was developed to be an essential tool in every robotics researcher's toolbox. It allows rapid testing of robotic designs and algorithms in realistic scenarios. Inside Gazebo, there is a powerful physics engine (actually, a total of four physics engines are supported), which combined with high-quality graphics and a friendly programming interface, allows accurate and efficient simulation of multi-robot systems in complex, indoor and outdoor environments in three dimensions. Most importantly, Gazebo is offered free of charge to interested users and is already supported by a large, active community of users.

Installation

To work with the latest version of the Gazebo Classic[1] simulator, **11.14.0**, you will need to secure access to a computer with Linux operating system, because at the present there is no full version for Windows available[2]. For those who only have access to a Windows machine, the easy solution is to install Virtual Box (freeware) [www.virtualbox.org] and create a virtual machine on which you will install Linux. Prefer Long-Term Support (LTS) distributions such as Ubuntu 22.04 LTS or Ubuntu 20.04 LTS [www.ubuntu.com]. Last, but not least, you can just sign into the TUC Virtual Desktop Infrastucture (VDI) service; the Ubuntu virtual machines on VDI have Gazebo Classic already installed for you! I take it for granted that all these terms don't confuse you; if you are wondering what a virtual machine is and what Linux is, then be sure to find out immediately! As an ECE student, you have to know! Once you ensure the above, follow the appropriate instructions:

- **Download** https://classic.gazebosim.org/download
- **Installation** https://classic.gazebosim.org/tutorials?cat=install (look for Ubuntu)
- **Releases** https://classic.gazebosim.org/distributions/gazebo/releases

From this point onward we assume that the installation has been completed without errors or problems.

Gazebo Tutorials

The tutorials for Gazebo Classic include detailed instructions for learning how to use it, as well as technical details for each supported robotic system by the simulator. These texts are accessible from the website https://classic.gazebosim.org/tutorials. In the Documents/Lab Material section of eClass you will also find some additional material from the robotics companies Willow Garage and The Construct that support the development of Gazebo.

Procedure

Start the Gazebo Classic simulator on your computer by typing `gazebo` in a terminal. Actually, two programs will be running: `gzserver`, which is the simulator itself, and `gzclient`, which is the graphical interface; they communicate with each other based on a protocol that uses Google Protocol Buffers. These two programs can be run independently by typing their names in separate terminals. In fact, it is possible to have several clients for different graphical presentations of the simulation from the same server. At the top right of the graphic window you will see a small camera; by clicking there, we can save a screenshot of the main graphical window. In the folder `/usr/share/gazebo-11/worlds` there are several ready-made worlds. Open some world by giving `gazebo worlds/<someWorldName>.world`. Play a bit with the options given at the top of the window (change perspective, lighting, move/rotate objects, etc.). Don't worry if you "ruin" the world, you can always start over.

[1] We stick to Gazebo Classic for the moment, due to existing accumulated experience; next year, we will switch!
[2] There are some instructions for installation on Windows, but this solution is not suitable for the time being.

Gazebo Models
Any physical entity used in a Gazebo simulation, from a simple geometric object to a complex robotic system, is called a model. Each world consists of one or more such models. Even the terrain in the simulation is such a model! Each model includes information about the dynamic, kinetic, graphical properties of the entity it describes, as well as any plugins that affect its behavior. Examples of models (from worlds you have opened) can be found in the folder `$HOME/.gazebo/models/`. Each `model` has its own subfolder containing the following:

- `model.config` (required): xml file with basic information about the model
- `model.sdf` (required): description in Simulator Description Format [http://sdformat.org]
- `model.sdf.erb` (alternatives): sdf file with embedded Ruby code for dynamic generation
- `meshes` (optional): folder with COLLADA and STL files for the model
- `materials` (optional): folder with textures, images and OGRE scripts for the model
- `plugins` (optional): folder with the plugins for the model

The main file `.sdf` describes with xml tags the elements we are interested in: solid sections, geometry, masses, sizes, joints, inertial properties, collision geometry, etc.

Experiment
Practice building robot models through the tutorial Build a Robot, specifically performing step-by-step the procedures described in the sections only Make a Mobile Robot, Attach Meshes, Add a Sensor to a Robot to model a simple two-wheeled robot. You can copy from the tutorial the code sections needed at each step. Proceed to practicing with the graphical tool to create models by following the step-by-step instructions described in Model Editor. Some `.dae` or `.stl` files for additional shapes can be found in the existing models in the `$HOME/.gazebo/models/` folder. When you create your own model and save it, a new folder will be created with all the related files. Be sure to put it together with existing models to keep them at one place. Finally, practice building worlds by following step-by-step the procedures described in Build a world and using the graphical tool for creating virtual buildings, following the step-by-step instructions described in the section Building Editor. In the world you will create, be sure to include the robot model you created earlier, in as many copies as you wish. Of course, you can also include any of the ready-made robot models you prefer. Each world is saved in its own folder, wherever you want. To open it again you have to give the line command `gazebo <path-to-my-world>/<my-world-name>.sdf`. In building creation, there is a ready-made floorplan given, but you should prefer to introduce and create something simpler, for example your dream room with doors, windows, maybe even two floors and a staircase. Not something complex, just something original! Save your building in the recommended folder `$HOME/building_editor_models/`. Then, add the building you constructed to the world you created to get a complete final creation. You might notice that all the above sections of the tutorial are at the Beginner level of difficulty, so they won't be too difficult for you; also the instructions given in the text and in the videos are quite detailed.

Exercises
What??? Do we have to do exercises now? Is the above not enough? Of course they are enough!!! And I hope you will do all of them! But to give a more personal character to the above Gazebo learning process, make sure to insert small variations in some steps of the tutorial at your discretion (different numbers, different dimensions, different geometry, different world features, different building elements, ...), so that you put your own touch on the model, the world, and the building you will create. Whenever you complete a section of the tutorial, be sure to take one (or more) screenshot(s) of the result, so that you can present what you did.

Report/Delivery/Grading
Gather and compress the folders with the xml code files you created for your model, world, and building. Also, compile a series of screenshots from your work into a short report (in PDF), where you will also describe what you did and especially any variations from the tutorial you adopted. Finally, submit code and report as two separate files via eClass through the Assignments section. Your grade will be determined by the completeness of your work.