Configuring a MultiUAV Ardupilot Simulation in Gazebo

Written by Jonathan Lopes Florêncio
Santa Catarina State University
Joinville/SC - Brazil
Contact: reddy.lopes@gmail.com

1. Install ROS Kinetic Desktop Full

Instructions in this link. Note that this package already comes with Gazebo7 installed, but it will not be used.

2. Install Gazebo8

Instructions in this link. It is important to note that the installation should be done step by step, and not via curl (found at the top of the referenced link page). The `libgazebo8-dev` library, described as optional in the tutorial, is mandatory in this work.

If the installer acknowledges the lack of a package (`sdformat5` and `libsdformat5-dev`, for example), simply add the names of these packages together with the `apt-get install` command. Example:

```
sudo apt-get install gazebo8 libgazebo8-dev sdformat5 libsdformat5-dev
```

3. Install ROS-Gazebo8 packages

Just run the following line in the shell:

```
sudo apt-get install ros-kinetic-gazebo8-ros ros-kinetic-gazebo8-ros-pkgs
ros-kinetic-gazebo8-ros-control
```

4. Make sure you run Gazebo at least once

Run

```
gazebo --verbose /usr/share/gazebo-8/worlds/iris_arducopter_demo.world
```

in the command line. The program should normally open with a lane and a quadricopter. If all goes well, you may close it.

5. Install ardupilot_gazebo plugin

Instructions in this link. Note that this plugin already comes with Gazebo8, but in an outdated version.

IMPORTANT! Follow the tutorial only until the `sudo make install` statement. It is not recommended to set the Gazebo template path.

6. Modify the model files

Go to the Iris template folder and edit the configuration file.

```
cd ~/.gazebo/models/iris_with_standoffs_demo
gedit model.config
```

Modify the `<name>` tag of `<model>` to Iris Ardupilot 9002 (or other name that identifies it). Save and close the model.config file

Now, edit the template file.

```
gedit model.sdf
```

Look inside the file for the word **libArduCopterPlugin.so** (possibly **libArdupilotPlugin.so**). You'll find the <plugin> tag opening that imports the Iris control plugin via Ardupilot. Edit it by entering the following tags:

```
<fdm_addr>127.0.0.1</fdm_addr>
<fdm_port_in>9002</fdm_port_in>
<fdm_port_out>9003</fdm_port_out>
```

This part of the file should look like this:

```
<plugin name="arducopter_plugin" filename="libArduCopterPlugin.so">
  <fdm_addr>127.0.0.1</fdm_addr>
  <fdm_port_in>9002</fdm_port_in>
  <fdm_port_out>9003</fdm_port_out>
  <imuName>iris_demo::iris::iris/imu_link::imu_sensor</imuName>
  <connectionTimeoutMaxCount>5</connectionTimeoutMaxCount>
```

This means that this model will communicate with Ardupilot via 9002 and 9003 ports.

7. Clone the models

In order to clone the models, simply create copies of the model folder. In this case, copies of the `iris_with_standoffs_demo` folder located in `~/.gazebo/models` are created. My suggestion is to use the name of the input port as the identifier, as follows:

In each of these folders, edit the configuration file (`model.config`) to change the name of the models, as done before (Iris Ardupilot 9012 and Iris Ardupilot 9022, for example).

In each of these folders, edit the template file (`model.sdf`) and change the `<fdm_port_in>` and `<fdm_port_out>` tags. Each new instance of SITL is created in ports spaced by 10 units to each other. So, the rule is always to add 10 in every port to each cloned model. If the first model had ports 9002 and 9003, the following will have ports 9012 and 9013, the next will be 9022 and 9023, and so on. These modifications are shown in next step.

8. Modify the World File

Go to the Gazebo Worlds folder and make a copy of the `iris_arducopter_demo.world` file.

```
cd /usr/share/gazebo-8/worlds
sudo cp iris_arducopter_demo.world iris_multiuav.world
```

Edit the newly created file.

```
sudo gedit iris_multiuav.world
```

At the end of the file, you will find the inclusion of the UAV model, as shown below.

```
<model name="iris_demo">
  <include>
    <uri>model://iris_with_standoffs_demo</uri>
  </include>
</model>
```

Edit this piece of code to insert the other models. An example is shown below.

```
<model name="iris_demo_9002">
  <pose>0 -1 0 0 0 0</pose>
  <include>
    <uri>model://iris_with_standoffs_demo_9002</uri>
  </include>
</model>
<model name="iris_demo_9012">
  <pose>0 0 0 0 0 0</pose>
  <include>
    <uri>model://iris_with_standoffs_demo_9012</uri>
  </include>
</model>
<model name="iris_demo_9022">
  <pose>0 1 0 0 0 0</pose>
  <include>
    <uri>model://iris_with_standoffs_demo_9022</uri>
  </include>
</model>
```

Save and close the world file.

Run

```
gazebo --verbose /usr/share/gazebo-8/worlds/iris_multiuav.world
```

in the command line. You should see a world similar to the previous one, but with 3 quadricopters.

9. Clone the ArduPilot source code

If you are in an university, maybe the git protocol is blocked. So, run the following code to run the downloads under https protocol.

```
git config --global url."https://".insteadOf git://
```

Perform the remaining steps to clone the Ardupilot main code.

```
cd ~
git clone https://github.com/ArduPilot/ardupilot.git ardupilot1
cd ardupilot
git submodule update --init --recursive
alias waf="$PWD/modules/waf/waf-light"
```

IMPORTANT! All waf commands must be run from within the `ardupilot` folder.

```
waf configure --board=sitl
waf all
```

You must perform a clone for each UAV you want to simulate. In this example, with three quadricopters, three clones of the code will be needed (that is, perform the above procedure three times). I particularly believe there is a more efficient way of doing this, but I have not had time to test it yet.

10. Run the simulation

Open four different terminals.

Terminal 1 (this code will connect to port 9002):

```
cd ~/ardupilot1/Tools/autotest
./sim_vehicle.py -v ArduCopter -f gazebo-iris  --console -I0
```

Terminal 2 (this code will connect to port 9012):

```
cd ~/ardupilot2/Tools/autotest
./sim_vehicle.py -v ArduCopter -f gazebo-iris  --console -I1
```

Terminal 3 (this code will connect to port 9022):

```
cd ~/ardupilot3/Tools/autotest
./sim_vehicle.py -v ArduCopter -f gazebo-iris  --console -I2
```

Terminal 4 (run Gazebo):

```
gazebo --verbose /usr/share/gazebo-8/worlds/iris_multiuav.world
```

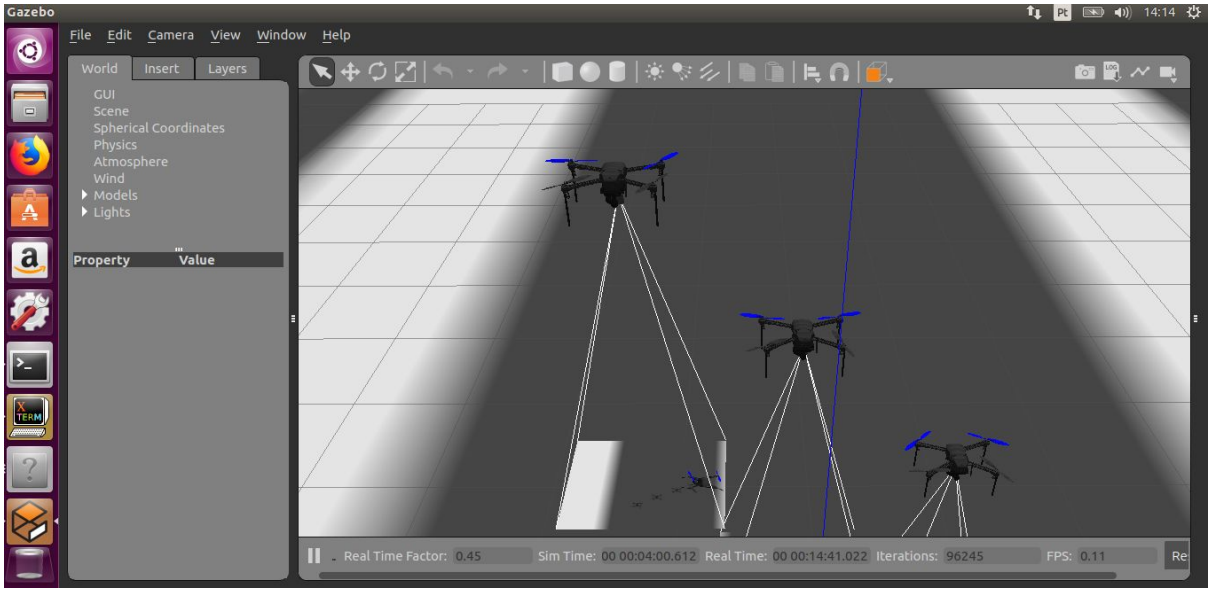
Wait until the following message appears in the SITL console:

After that, enter these instructions on terminals 1, 2 and 3:

```
rc 3 1500
mode guided
arm throttle
takeoff 2
```

The value after the word `takeoff` will determine the takeoff height. You may modify these values. The following figure shows a test with takeoff values 2, 3 and 4.

Note that instance 0 of port 9002 is farther to the right. In the initial world perspective, the X axis grows ahead in the track direction, the Y axis grows to the left, perpendicular to the track, and the Z axis grows upwards, perpendicular to the ground plane.



From now on, you can fly with the quadricopters and see the results in the Gazebo. For more information on how to control ArduCopter via MavProxy, see this link.