TIAGO Tutorials

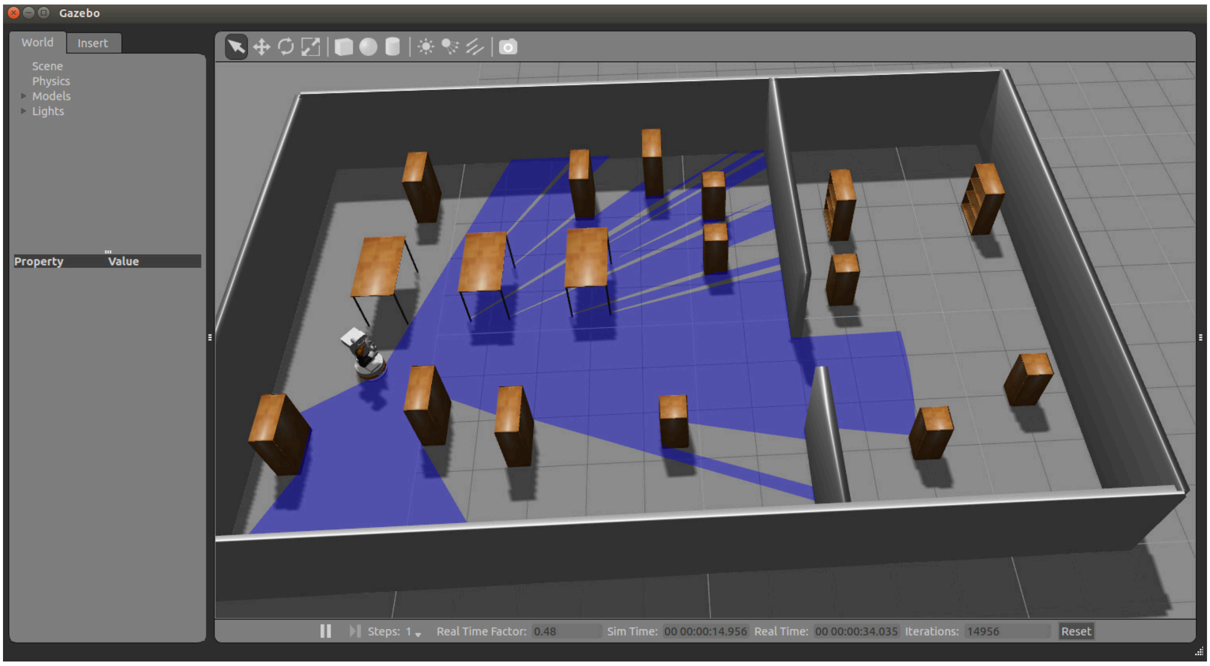
Standard Install Procedure

Introduction

The following guide will install on your Ubuntu machine all the simulation environments that will be the basis of all the Homeworks for the Intelligent Robotics 2022-23 course.

NB. This guide has been extracted from the official Wiki available at Robots/TIAGo/Tutorials so please, if you have any problem during the following steps, please take a look also at the official documentation.

TIAGO is a service robot produced by PAL Robotics. It is essentially a humanoid robot with the kinematic model of a simple mobile manipulator (differential mobile base + 6dof anthropomorphic manipulator). At the end of this install procedure you will be able to reproduce a very complex simulation that replicates all the features and capability of the real TIAGO robot. Below the results obtained at the end of this install guide.



Example picture of the robot simulated on Gazebo.

Installation Steps

1. Create Tiago workspace (optional, remind that you have to use this workspace for the assignments):

```
$> mkdir ~/tiago_public_ws
```

2. Install the TIAGO packages (inside your workspace folder)

```
$> wget
https://raw.githubusercontent.com/pal-robotics/tiago_tutorials/kinetic-devel/
tiago_public-melodic.rosinstall

$> rosinstall src /opt/ros/melodic tiago_public-melodic.rosinstall

$> rosdep install --from-paths src --ignore-src -y --rosdistro melodic
   --skip-keys="opencv2 opencv2-nonfree pal_laser_filters speed_limit_node
   sensor_to_cloud hokuyo_node libdw-dev python-graphitesend-pip
   python-statsd pal_filters pal_vo_server pal_usb_utils pal_pcl
   pal_pcl_points_throttle_and_filter pal_karto pal_local_joint_control
   camera_calibration_files pal_startup_msgs pal-orbbec-openni2
   dummy_actuators_manager pal_local_planner gravity_compensation_controller
   current_limit_controller dynamic_footprint dynamixel_cpp tf_lookup
   opencv3 joint_impedance_trajectory_controller
   cartesian_impedance_controller omni_base_description
   omni_drive_controller"
```

3. Build the workspace

```
$> source /opt/ros/melodic/setup.bash

$> catkin build -DCATKIN_ENABLE_TESTING=0

$> source devel/setup.bash
```

4. Source the ROS workspace using ~/.bashrc (Recommended but not mandatory)

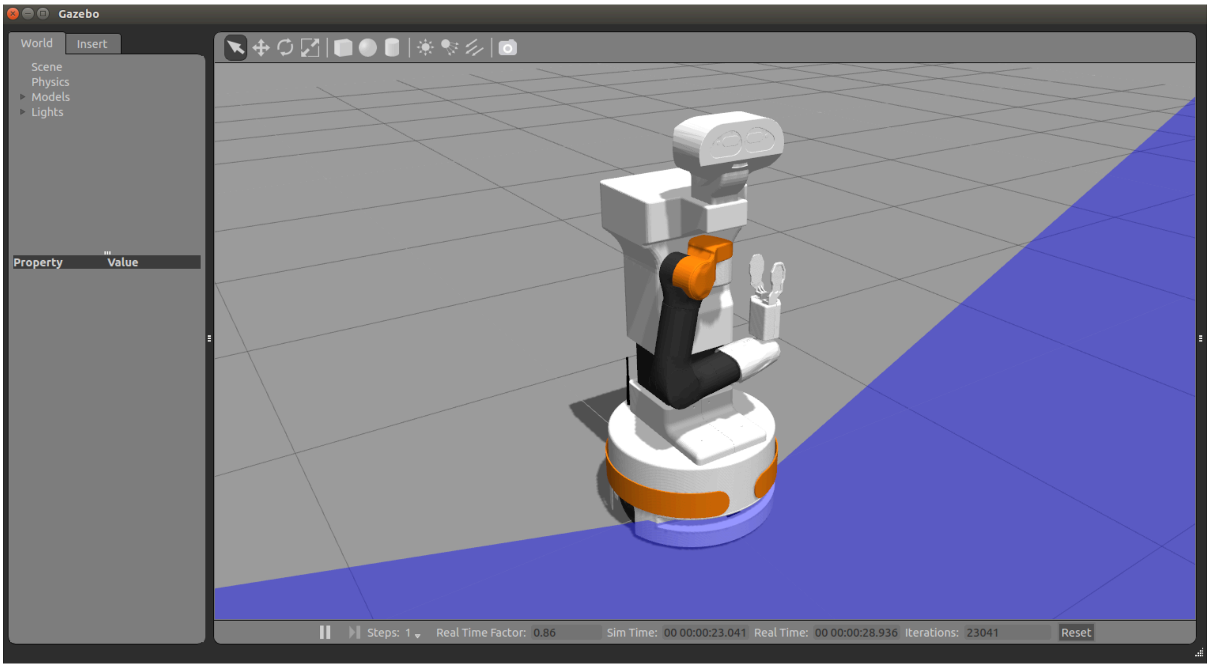
```
$> echo "source /opt/ros/melodic/setup.bash" >> ~/.bashrc

$> echo "source ~/tiago_public_ws/devel/setup.bash" >> ~/.bashrc
```

Now you can close and reopen your shell. From this moment on, your shell will be always updated and pointing to your ROS workspace.

5. Test your simulation

To launch the simulation of the TIAGo **Steel**, execute:

```
$> roslaunch tiago_gazebo tiago_gazebo.launch public_sim:=true robot:=steel
```



The **Titanium** version can be launched as follows:

```
$> roslaunch tiago_gazebo tiago_gazebo.launch public_sim:=true
robot:=titanium
```