

Tiago simulation guide

Jacob Nørgaard

29. april 2020

1 Getting started

Use following link to install fresh Ubuntu and ROS:

<http://wiki.ros.org/Robots/TIAGo/Tutorials/Installation/InstallUbuntuAndROS>

And the following link to install Tiago simulator:

<http://wiki.ros.org/Robots/TIAGo/Tutorials/Installation/TiagoSimulation>

You should now have a workspace at

```
~/tiago_public_ws
```

When using the public simulator you have two robot options when launching:

- TIAGo Steel: in this configuration the end-effector is a parallel gripper
- TIAGo Titanium: the wrist has a 6-axis force/torque sensor and the end-effector is the under-actuated 5-finger Hey5 hand.

To launch the simulator, run the following command in your workspace:

```
roslaunch tiago_gazebo tiago_gazebo.launch public_sim:=true  
↪ robot:=steel
```

or

```
roslaunch tiago_gazebo tiago_gazebo.launch public_sim:=true  
↪ robot:=titanium
```

You can also launch the simulation in a small office environment with people in it:

```
roslaunch tiago_gazebo tiago_gazebo.launch public_sim:=true  
↪ robot:=titanium world:=simple_office_with_people
```

2 Controlling the simulation

2.1 Controlling the robot with your keyboard

Open two terminals and direct to your workspace in both and source the workspace:

```
cd ~/tiago_public_ws
source ./devel/setup.bash
```

Launch the simulation:

```
roslaunch tiago_gazebo tiago_gazebo.launch public_sim:=true
↪ robot:=titanium world:=simple_office_with_people
```

In the second terminal, run the following command:

```
roslaunch key_teleop key_teleop.py
```

In this terminal windows you can now push your arrowkeys and see the robot move in the simulation.

2.2 Move through velocity commands

The velocity commands are sent to the simulation through the topic `mobile_base_controller/cmd_vel` of type `geometry_msgs/Twist` which is composed of:

- *geometry_msgs/Vector3 linear*
- *geometry_msgs/Vector3 angular*

and is specified by a linear and an angular velocity.

How to:

Once again direct to your workspace and source the workspace:

```
cd ~/tiago_public_ws
source ./devel/setup.bash
```

And Launch the simulation:

```
roslaunch tiago_gazebo tiago_gazebo.launch public_sim:=true
↪ robot:=steel
```

In the second terminal windows you can now send commands specifying the linear and angular velocity by either:

```
rostopic pub /mobile_base_controller/cmd_vel
↳ geometry_msgs/Twist "linear:
    x: 0.5
    y: 0.0
    z: 0.0
angular:
    x: 0.0
    y: 0.0
    z: 0.0" -r 3
```

or

```
rostopic pub /mobile_base_controller/cmd_vel
↳ geometry_msgs/Twist -r 3 -- '[0.5,0.0,0.0]' '[0.0, 0.0,
↳ 0.0]'
```

To make the robot move with a constant linear velocity of 0.5 m/s and -r denotes that the message is published 3 times per second.

The same can be done with angular velocity:

```
rostopic pub /mobile_base_controller/cmd_vel
↳ geometry_msgs/Twist "linear:
    x: 0.0
    y: 0.0
    z: 0.0
angular:
    x: 0.0
    y: 0.0
    z: 0.3" -r 3
```

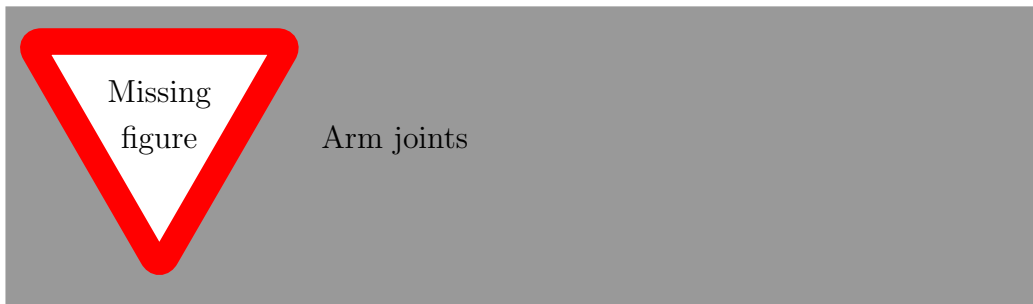
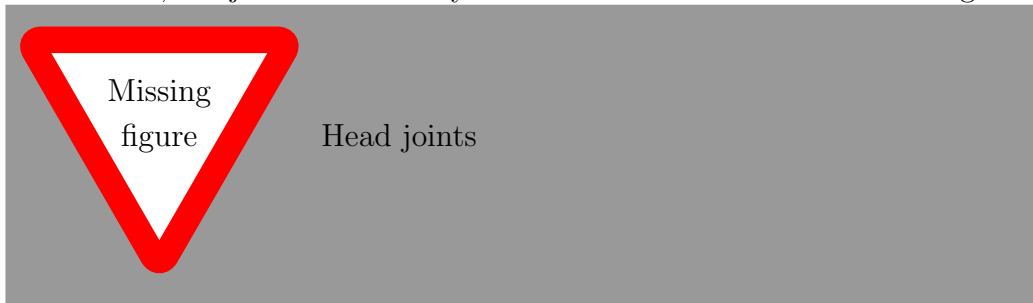
or

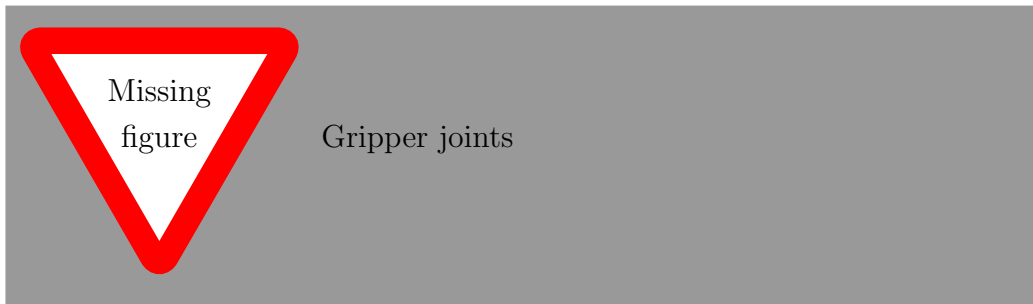
```
rostopic pub /mobile_base_controller/cmd_vel
↳ geometry_msgs/Twist -r 3 -- '[0.0,0.0,0.0]' '[0.0, 0.0,
↳ 0.3]'
```

to make the robot rotate around its own axis at 0.3 rad/s.

3 Joint trajectory controller

An example of how to move the arm by using joint trajectory action. For the following exercise it is important to understand which joints on the robot has which names. This is due to the fact that the joints in the program will only be enumerated and therefore a figure to look up the names. The head joints is shown in ??, the the arm in ??, for the torso in ?? and for the gripper in ??. Since the Tiago that is available in the lab only has the gripper attachment, the joints for the Hey5 attachment is not included in this guide.





To start off, once again source your workspace:

```
cd ~/tiago_public_ws
source ./devel/setup.bash
```

And launch your simulation:

```
roslaunch tiago_gazebo tiago_gazebo.launch public_sim:=true
↪ robot:=steel
```

The following example will move the arm through two waypoints with the joints described above configured as follows:

Waypoint 1:

```
arm_1_joint: 0.2
arm_2_joint: 0.0
arm_3_joint: -1.5
arm_4_joint: 1.94
arm_5_joint: -1.57
arm_6_joint: -0.5
arm_7_joint: 0.0
```

Waypoint 2:

```
arm_1_joint: 2.5
arm_2_joint: 0.2
arm_3_joint: -2.1
arm_4_joint: 1.9
arm_5_joint: 1.0
arm_6_joint: -0.5
arm_7_joint: 0.0
```

To run the example, run the following command:

```
roslaunch tiago_trajectory_controller run_traj_control
```