ROS: Unified Robot Description Format (URDF)

The objective of this lab is to learn how to describe a robot by using Unified Robot Description Format (URDF) in ROS.

SETTING UP YOUR COMPUTER

Open terminal window.

Clone your git repository to your home folder:

```
$ git clone URL_for_<yourname>-rtech_repository
```

Use `ls` to confirm that ***<yourname>-rtech*** has been downloaded to your home folder**.**

Next, create a new ROS package to your catkin workspace, name this package **`my_r2d2`**. Do you remember how to create a new ROS package?

Now **`cd`** to **`my_r2d2`** and create a new folder called **`urdf`**. In the **`urdf`** folder, use the following command to create a new empty file

```
$ touch r2d2.urdf
```

ROS TUTORIALS ABOUT UNIFIED ROBOT DESCRIPTION FORMAT (URDF)

In this lab you will be following a selection of ROS URDF tutorials available online. Please note that in the online tutorials, new URDF-file is created for each example but throughout this lab (except the 4th tutorial), only use the newly created **`r2d2.urdf`** as you complete the step-by-step instructions.

Once you have started to add meaningful content to **`r2d2.urdf`**, you can visualize the robot by typing the following command while in your **`my_r2d2/urdf`** folder

```
$ roslaunch urdf_tutorial display.launch model:=r2d2.urdf
```

Or from any path

```
$ roslaunch urdf_tutorial display.launch model:='$(find my_r2d2)/urdf/r2d2.urdf'
```

Here is the list of ROS tutorials and tasks to complete:

1. Building a Visual Robot Model with URDF from Scratch

   Learn how to build a visual model of a robot that you can view in Rviz.
2. Building a Movable Robot Model with URDF

   Learn how to define movable joints in URDF.
3. Adding Physical and Collision Properties to a URDF Model

   Learn how to add collision and inertial properties to links, and how to add joint dynamics to joints.
4. Using Xacro to Clean Up a URDF File

   Learn some tricks to reduce the amount of code in a URDF file using Xacro.
5. Modify the xacro-file to add a black lense (Figure 1) to your R2D2.

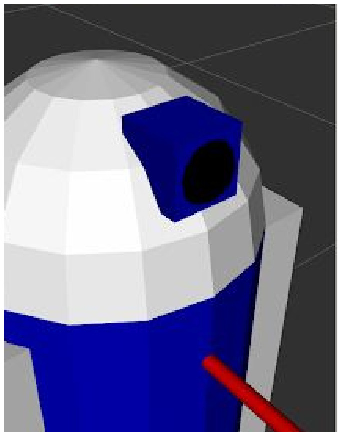

Figure 1. *R2D2 with a lense*

CLEAN UP YOUR WORKSPACE

NB! Before you leave the lab, make sure you have pushed all the files in your catkin workspace to your git cloud service.

In terminal, `cd` to **<yourname>-rtech**

Type

```
git config user.email "youremail@example.com"
```

Type

```
git status
```

You should now see all the new and modified files in red.

Prepare the relevant files for the commit.

```
git add file_name_in_red1 file_name_in_red2
```

When you now type

```
git status
```

you should see all the added files in green. You are now ready to commit changes. Type

```
git commit –m “Insert a brief explanation”
```

Your changes have now been committed but not yet uploaded to the cloud. To upload your files, type

```
git push
```

In your web browser, **verify that all the files** have been uploaded to the **<yourname>-rtech** repository.

Delete the **<yourname>-rtech** folder and any other files you created from the lab’s computer.