

# ROS Exercise 2 - URDF and XACRO

Mobile Robotics

*Prof. Dr.-Ing. Gavin Kane*

HS Emden Leer  
Fachbereich Technik  
Abteilung Elektrotechnik + Informatik



**Overview** In this practical you will:

- use URDF to model your own robot through geometric shapes,
- examine the use of RVIZ to check a robots configuration,
- use XACRO to parametise URDF files, and include multiple robots in a scene, and
- include packaged models, including 3D Data, in your scenes.

## 1 URDF Robot creation

Complete ros wiki tutorial: **Building a Visual Robot Model with URDF from Scratch** at:

<http://wiki.ros.org/urdf/Tutorials/Building%20a%20Visual%20Robot%20Model%20with%20URDF%20from%20Scratch>

Use the information you have learnt through this tutorial to create your own robot in a new package. Create a package in your `/src/` directory with the command:

```
1 $ catkin_create_pkg practical3_urdf
```

In this package, create a directory `urdf`, and create a new file in the directory called `"myrobot.urdf"`. In this file, create a Robot as shown in Figure 1. The Cylinders are to be 60cm long with a diameter of 20cm. The Spheres have a diameter of 20cm. The Cylinders are to represent the links of the robot. The Spheres are to represent the Joints of the robot. You should place the spheres with a "fixed" joint, 10cm higher than the height of the links. The center of the spheres should also be the center of rotation for the joints. The Robot is to have three joints (in addition to the fixed sphere joints):

- Joint 1: Bottom Cylinder to World. Rotation around z axis. Range= $\pm 170$  deg
- Joint 2: Bottom Cylinder to Middle Cylinder. Rotation around x axis. Range= $\pm 90$  deg
- Joint 3: Middle Cylinder to Top Cylinder. Rotation around x axis. Range= $\pm 90$  deg

Once you have generated your URDF file, use `check_urdf` to check the syntax of the file. When the file parses in `check_urdf` correctly, use the same launch file as the online tutorial, but modify the urdf filename, to load your own file into RVIZ. Using the `robot_joint_publisher` to move the joints, to confirm, that the robot moves in accordance with the given required.

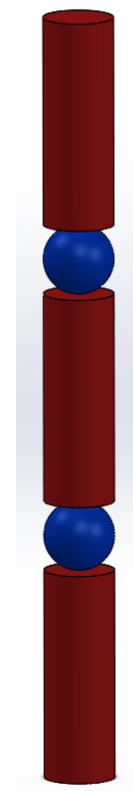


Fig. 1: Robot Structure

## 2 Parametising the URDF with XACRO

Modify the start and end of your urdf to include the following lines, and save with the new file extension `.urdf.xacro`

```
1 <?xml version="1.0"?>
2 <robot xmlns:xacro="http://www.ros.org/wiki/xacro" name="myrobotname">
3   ...
4   ...
5 </robot>
```

To parametrise the lengths and dimeters of the cylinders and spheres, use the information in online ROS Wiki tutorial:

<http://wiki.ros.org/urdf/Tutorials/Using%20Xacro%20to%20Clean%20Up%20a%20URDF%20File>

Using the same launch file as above, but with the new `.urdf.xacro` file, check that changing a paramter, changes the display of the robot in RVIZ.

## 3 URDF Reuse, with XACRO MACRO

In order to be able to reuse the robot, the XACRO must be turned into a macro. In this macro, all names will need to be prefixed, to prevent duplicate names.

```
1 <?xml version="1.0"?>
2 <robot xmlns:xacro="http://www.ros.org/wiki/xacro">
3   <xacro:macro name="myrobotname" params="prefix">
4     ...
5     ...
6   </xacro:macro>
7 </robot>
```

Note how the name of the robot has moved into the name of the macro. And the Prefix is now provided as an input parameter. To use the prefix parameter, you now need to change all link and joint names to include the prefix parameter. For example:

```
1 <link name="${prefix}_base_link">
```

A second `.urdf.xacro` file now needs to be created, that will call this macro, and insert the robot into the scene.

In this new file, start with the following template:

```
1 <?xml version="1.0"?>
2 <robot name="myscene" xmlns:xacro="http://www.ros.org/wiki/xacro">
3
4   <link name="world" />
5
6   <xacro:include filename="$(find urdf_exercise)/urdf/myrobot.urdf.xacro" />
7   <xacro:myrobot prefix="robot1_" />
8
9   ...
10  ...
11
12 </robot>
```

You will see that a package relative path is used for accessing the urdf file. This prevents absolute paths on a computer breaking simulations when moved or copied to other computers. The package name and filename will need to be modified.

You will then need to place the robot at the desired location, by providing a "fixed" joint, to place it at the desired position in the world.

By duplicating the macro call line `<xacro:myrobot...` with a different prefix name, a second instance of the robot can be placed into the scene.

The goal here is to place the two robots 1meter apart.

## 4 Including model with 3D Data in Scene

Search the Github website to find the KUKA LBR Robot, ROS Package. It is part of the Kuka\_Experimental Package. Download the package and place it as part of your src directory in the catkin workspace. Rebuild and resource your catkin workspace so that the package can be found when you attempt to use it.

In the same way that you have referenced and included your own macro file above, include the `lbr_iiwa_14_r820.xacro` file using a package relative path, and include the new robot in your scene.

Using an appropriate joint, place the robot 1 metre behind, but in the middle of the two robots.

When the final modifications are completed, and you are pleased with the result in RVIZ, generate the static urdf from the xacro files using the command:

```
1 rosrun xacro xacro --inorder urdf_exercise/urdf/myscene.urdf.xacro > myscene.urdf
```