Unit 4: XACRO files

As you might have experienced already, very simple robots can have really big and complex URDF files.  
For this reason and to be able make URDF files more flexible for change, XACROS are used in Robot Model definitions even more than URDF files.

XACROS also give some tools for making simple Math operations inside the URDF files, as well as having Constants and MACROS.

1. Basics on using XACRO

Xacro is based on a macro language. All Xacro files end up being a single URDF file, but this way, you will have much cleaner and easier-to-maintain robot model definitions.

You can generate a URDF from a XACRO file by just executing:

In [ ]:



rosrun xacro xacro model.xacro **>** model.urdf

But the best way is to generate the URDF files on the fly each time you execute a launch file. This way, you will occupy less memory. The only drawback is the fact that it has to be generated each time the launch is executed, which is slower than using a URDF directly.

In [ ]:



**<**param name**=**"robot\_description"

command**=**"$(find xacro)/xacro '$(find my\_robot\_description)/robots/myrobot.urdf.xacro'" **/>**

It's common to put the XACRO files inside the "robots" folder.

2. Create your own XACRO

You are going to make a XACRO VERSION OF the Gurdy Robot.  
So, the first line in a XACRO file is:

In [ ]:



**<?**xml version**=**"1.0"**?>**

**<**robot xmlns:xacro**=**"http://www.ros.org/wiki/xacro" name**=**"gurdy"**>**

So, let's create a "robot" folder inside the my\_gurdy\_package; and inside that, create **gurdy.urdf.xacro**.

Let's start optimizing the original URDF file of Gurdy:

Constants

* One of the easiest ways to optimize a URDF is by creating constants to avoid having to change the same value in different places. So, let's look for values that are shared and could be simplified. And because Gurdy is made up of the same repeated elements, it's highly probable that a lot of elements can be made of shared constants.

Here you have an example:

In [ ]:



**<**link name**=**"head\_link"**>**

**<**inertial**>**

**<**origin xyz**=**"0 0 0.02" rpy**=**"0 0 0"**/>**

**<**mass value**=**"0.01" **/>**

**<**inertia ixx**=**"7.58333333333e-07" ixy**=**"0.0" ixz**=**"0.0" iyy**=**"7.58333333333e-07" iyz**=**"0.0" izz**=**"1.25e-06"**/>**

**</**inertial**>**

**<**collision**>**

**<**origin xyz**=**"0 0 0.02" rpy**=**"0 0 0"**/>**

**<**geometry**>**

**<**cylinder radius**=**"0.05" length**=**"0.04"**/>**

**</**geometry**>**

**</**collision**>**

**<**visual**>**

**<**origin rpy**=**"0.0 0 0" xyz**=**"0 0 0"**/>**

**<**geometry**>**

**<**mesh filename**=**"package://my\_gurdy\_description/models/gurdy/meshes/gurdy\_head\_v2.dae"**/>**

**</**geometry**>**

**</**visual**>**

**</**link**>**

**<**link name**=**"upperleg\_M1\_link"**>**

**<**inertial**>**

**<**origin xyz**=**"0 0 0.03" rpy**=**"0 0 0"**/>**

**<**mass value**=**"0.01" **/>**

**<**inertia ixx**=**"3.015625e-05" ixy**=**"0.0" ixz**=**"0.0" iyy**=**"3.015625e-05" iyz**=**"0.0" izz**=**"3.125e-07"**/>**

**</**inertial**>**

**<**collision**>**

**<**origin xyz**=**"0 0 0.03" rpy**=**"0 0 0"**/>**

**<**geometry**>**

**<**cylinder length**=**"0.06" radius**=**"0.0025"**/>**

**</**geometry**>**

**</**collision**>**

**<**visual**>**

**<**origin rpy**=**"0.0 0 0" xyz**=**"0 0 0"**/>**

**<**geometry**>**

**<**mesh filename**=**"package://my\_gurdy\_description/models/gurdy/meshes/gurdy\_higherleg\_v2.dae"**/>**

**</**geometry**>**

**</**visual**>**

**</**link**>**

These two links share the same exact mass, so why don't we create a constant that stores the mass that is shared by all the links of the Gurdy Robot?  
There are also repeated values in each link because the origin of all its sub-elements is the same, so that's another candidate for XACRO simplification.

In [ ]:



**<**xacro:property name**=**"m" value**=**"0.01" **/>**

**<**xacro:property name**=**"head\_z\_origin" value**=**"0.02" **/>**

**<**xacro:property name**=**"upperleg\_M1\_z\_origin" value**=**"0.02" **/>**

**<**xacro:property name**=**"head\_radius" value**=**"0.05" **/>**

**<**xacro:property name**=**"head\_length" value**=**"0.04" **/>**

**<**xacro:property name**=**"upperleg\_radius" value**=**"0.0025" **/>**

**<**xacro:property name**=**"upperleg\_length" value**=**"0.06" **/>**

​

**<**link name**=**"head\_link"**>**

**<**inertial**>**

**<**origin xyz**=**"0 0 ${head\_z\_origin}" rpy**=**"0 0 0"**/>**

**<**mass value**=**"${m}" **/>**

**<**inertia ixx**=**"7.58333333333e-07" ixy**=**"0.0" ixz**=**"0.0" iyy**=**"7.58333333333e-07" iyz**=**"0.0" izz**=**"1.25e-06"**/>**

**</**inertial**>**

**<**collision**>**

**<**origin xyz**=**"0 0 ${head\_z\_origin}" rpy**=**"0 0 0"**/>**

**<**geometry**>**

**<**cylinder radius**=**"${head\_radius}" length**=**"${head\_length}"**/>**

**</**geometry**>**

**</**collision**>**

**<**visual**>**

**<**origin rpy**=**"0.0 0 0" xyz**=**"0 0 0"**/>**

**<**geometry**>**

**<**mesh filename**=**"package://my\_gurdy\_description/models/gurdy/meshes/gurdy\_head\_v2.dae"**/>**

**</**geometry**>**

**</**visual**>**

**</**link**>**

**<**link name**=**"upperleg\_M1\_link"**>**

**<**inertial**>**

**<**origin xyz**=**"0 0 ${upperleg\_M1\_z\_origin}" rpy**=**"0 0 0"**/>**

**<**mass value**=**"${m}" **/>**

**<**inertia ixx**=**"3.015625e-05" ixy**=**"0.0" ixz**=**"0.0" iyy**=**"3.015625e-05" iyz**=**"0.0" izz**=**"3.125e-07"**/>**

**</**inertial**>**

**<**collision**>**

**<**origin xyz**=**"0 0 ${upperleg\_M1\_z\_origin}" rpy**=**"0 0 0"**/>**

**<**geometry**>**

**<**cylinder length**=**"${upperleg\_length}" radius**=**"${upperleg\_radius}"**/>**

**</**geometry**>**

**</**collision**>**

**<**visual**>**

**<**origin rpy**=**"0.0 0 0" xyz**=**"0 0 0"**/>**

**<**geometry**>**

**<**mesh filename**=**"package://my\_gurdy\_description/models/gurdy/meshes/gurdy\_higherleg\_v2.dae"**/>**

**</**geometry**>**

**</**visual**>**

**</**link**>**

Math

You can perform the basic operations (+,-,\*,/), the unary minus, and use parenthesis. This allows you to, for example, calculate the inertias based on the dimensions directly in the XACRO file. Another example would be symmetrically-positioned links or joints.

In [ ]:



**<**xacro:property name**=**"m" value**=**"0.01" **/>**

**<**xacro:property name**=**"head\_z\_origin" value**=**"0.02" **/>**

**<**xacro:property name**=**"r" value**=**"0.05" **/>**

**<**xacro:property name**=**"h" value**=**"0.04" **/>**

**<**xacro:property name**=**"head\_ixx" value**=**"${(m/12.0)\*(3\*r\*r+h\*h))}" **/>**

**<**xacro:property name**=**"head\_iyy" value**=**"${head\_ixx}" **/>**

**<**xacro:property name**=**"head\_izz" value**=**"${(m\*r\*r)/2.0}" **/>**

​

**<**link name**=**"head\_link"**>**

**<**inertial**>**

**<**origin xyz**=**"0 0 ${head\_z\_origin}" rpy**=**"0 0 0"**/>**

**<**mass value**=**"${m}" **/>**

**<**inertia ixx**=**"${head\_ixx}" ixy**=**"0.0" ixz**=**"0.0" iyy**=**"${head\_iyy}" iyz**=**"0.0" izz**=**"${head\_izz}"**/>**

**</**inertial**>**

**<**collision**>**

**<**origin xyz**=**"0 0 ${head\_z\_origin}" rpy**=**"0 0 0"**/>**

**<**geometry**>**

**<**cylinder radius**=**"${r}" length**=**"${h}"**/>**

**</**geometry**>**

**</**collision**>**

**<**visual**>**

**<**origin rpy**=**"0.0 0 0" xyz**=**"0 0 0"**/>**

**<**geometry**>**

**<**mesh filename**=**"package://my\_gurdy\_description/models/gurdy/meshes/gurdy\_head\_v2.dae"**/>**

**</**geometry**>**

**</**visual**>**

**</**link**>**

MACROS

This is by far the most useful feature of XACROS. There are two main types of MACROS:

**Simple Macro:**

In [ ]:



**<**xacro:property name**=**"m" value**=**"0.01" **/>**

**<**xacro:property name**=**"head\_z\_origin" value**=**"0.02" **/>**

**<**xacro:property name**=**"r" value**=**"0.05" **/>**

**<**xacro:property name**=**"h" value**=**"0.04" **/>**

**<**xacro:property name**=**"head\_ixx" value**=**"${(m/12.0)\*(3\*r\*r+h\*h))}" **/>**

**<**xacro:property name**=**"head\_iyy" value**=**"${head\_ixx}" **/>**

**<**xacro:property name**=**"head\_izz" value**=**"${(m\*r\*r)/2.0}" **/>**

​

**<**xacro:macro name**=**"default\_origin"**>**

**<**origin xyz**=**"0 0 0" rpy**=**"0 0 0"**/>**

**</**xacro:macro**>**

​

​

​

**<**link name**=**"head\_link"**>**

**<**inertial**>**

**<**origin xyz**=**"0 0 ${head\_z\_origin}" rpy**=**"0 0 0"**/>**

**<**mass value**=**"${m}" **/>**

**<**inertia ixx**=**"${head\_ixx}" ixy**=**"0.0" ixz**=**"0.0" iyy**=**"${head\_iyy}" iyz**=**"0.0" izz**=**"${head\_izz}"**/>**

**</**inertial**>**

**<**collision**>**

**<**origin xyz**=**"0 0 ${head\_z\_origin}" rpy**=**"0 0 0"**/>**

**<**geometry**>**

**<**cylinder radius**=**"${r}" length**=**"${h}"**/>**

**</**geometry**>**

**</**collision**>**

**<**visual**>**

**<**xacro:default\_origin **/>**

**<**geometry**>**

**<**mesh filename**=**"package://my\_gurdy\_description/models/gurdy/meshes/gurdy\_head\_v2.dae"**/>**

**</**geometry**>**

**</**visual**>**

**</**link**>**

Here, the expression:

In [ ]:



**<**origin xyz**=**"0 0 0" rpy**=**"0 0 0"**/>**

Is substituted with a smaller expression:

In [ ]:



**<**xacro:default\_origin **/>**

**Parametrised MACRO**

Macros can have input parameters. This input can be values or even entire blocks:

In [ ]:



**<**xacro:property name**=**"m" value**=**"0.01" **/>**

**<**xacro:property name**=**"head\_z\_origin" value**=**"0.02" **/>**

**<**xacro:property name**=**"r" value**=**"0.05" **/>**

**<**xacro:property name**=**"h" value**=**"0.04" **/>**

**<**xacro:property name**=**"head\_ixx" value**=**"${(m/12.0)\*(3\*r\*r+h\*h))}" **/>**

**<**xacro:property name**=**"head\_iyy" value**=**"${head\_ixx}" **/>**

**<**xacro:property name**=**"head\_izz" value**=**"${(m\*r\*r)/2.0}" **/>**

​

**<**xacro:macro name**=**"default\_origin"**>**

**<**origin xyz**=**"0 0 0" rpy**=**"0 0 0"**/>**

**</**xacro:macro**>**

​

**<**xacro:macro name**=**"head\_geometry" params**=**"head\_dae\_path"**>**

**<**geometry**>**

**<**mesh filename**=**"${head\_dae\_path}"**/>**

**</**geometry**>**

**</**xacro:macro**>**

​

**<**link name**=**"head\_link"**>**

**<**inertial**>**

**<**origin xyz**=**"0 0 ${head\_z\_origin}" rpy**=**"0 0 0"**/>**

**<**mass value**=**"${m}" **/>**

**<**inertia ixx**=**"${head\_ixx}" ixy**=**"0.0" ixz**=**"0.0" iyy**=**"${head\_iyy}" iyz**=**"0.0" izz**=**"${head\_izz}"**/>**

**</**inertial**>**

**<**collision**>**

**<**origin xyz**=**"0 0 ${head\_z\_origin}" rpy**=**"0 0 0"**/>**

**<**geometry**>**

**<**cylinder radius**=**"${r}" length**=**"${h}"**/>**

**</**geometry**>**

**</**collision**>**

**<**visual**>**

**<**xacro:default\_origin **/>**

**<**xacro:head\_geometry head\_dae\_path**=**"package://my\_gurdy\_description/models/gurdy/meshes/gurdy\_head\_v2.dae"**/>**

**</**visual**>**

**</**link**>**

Here we create a MACRO that inputs the path to the mesh, and inserts all of the geometry tags with the path input.

And here is an example using an entire block as input. To tell that a variable is a block input, an asterisk is set before the variable name:

In [ ]:



**<**xacro:macro name**=**"blue\_shape" params**=**"mass \*shape"**>**

**<**inertial**>**

**<**origin xyz**=**"0 0 0.03" rpy**=**"0 0 0"**/>**

**<**mass value**=**"${mass}" **/>**

**<**inertia ixx**=**"3.015625e-05" ixy**=**"0.0" ixz**=**"0.0" iyy**=**"3.015625e-05" iyz**=**"0.0" izz**=**"3.125e-07"**/>**

**</**inertial**>**

**<**collision**>**

**<**origin xyz**=**"0 0 0.03" rpy**=**"0 0 0"**/>**

**<**geometry**>**

**<**xacro:insert\_block name**=**"shape" **/>**

**</**geometry**>**

**</**collision**>**

**<**visual**>**

**<**origin rpy**=**"0.0 0 0" xyz**=**"0 0 0"**/>**

**<**geometry**>**

**<**mesh filename**=**"package://my\_gurdy\_description/models/gurdy/meshes/gurdy\_higherleg\_v2.dae"**/>**

**</**geometry**>**

**</**visual**>**

**</**xacro:macro**>**

​

​

**<**link name**=**"upperleg\_M1\_link"**>**

**<**xacro:upper\_leg mass**=**"0.01"**>**

**<**cylinder length**=**"0.06" radius**=**"0.0025"**/>**

**</**xacro:blue\_shape**>**

**</**link**>**

Here the whole link is substituted. Inside the mass is changed, and the block of the cylinder is created.

Normally, the .xacro files are saved in a folder called "robot" for historical reasons.

Warning

It's very important that you place the plugin for the control before the MACROS; otherwise, it won't be loaded, and could give you the warning: **"Controller Spawner couldn't find the expected controller\_manager ROS interface."**

Exercise U4-1

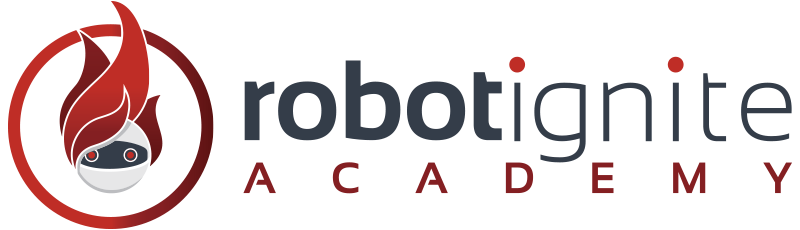
Try to clean up the Gurdy URDF file by:

1. Creating a MACROS that removes the repeated elements, like the three upper and three lower leg and foot links that are identical, because there are very few changes, apart from their names.
2. Doing the same procedure with the transmissions.
3. Doing the same thing with the Gazebo friction coefficient tags.

END Exercise U4-1

Solution Exercise U4.1

Please try to do it by yourself, unless you get stuck or need some inspiration. You will learn much more if you fight for each exercise.



Here you have an example of how it could be simplified:

In [ ]:



**<?**xml version**=**"1.0"**?>**

**<**robot xmlns:xacro**=**"http://www.ros.org/wiki/xacro" name**=**"gurdy"**>**

**<**gazebo**>**

**<**plugin name**=**"gazebo\_ros\_control" filename**=**"libgazebo\_ros\_control.so"**>**

**<**robotNamespace**>/**gurdy**</**robotNamespace**>**

**</**plugin**>**

**</**gazebo**>**

**<!--** MACROS **-->**

**<**xacro:macro name**=**"upperleg\_link" params**=**"number"**>**

**<**link name**=**"upperleg\_M${number}\_link"**>**

**<**inertial**>**

**<**origin xyz**=**"0 0 0.03" rpy**=**"0 0 0"**/>**

**<**mass value**=**"0.01" **/>**

**<**inertia ixx**=**"3.015625e-05" ixy**=**"0.0" ixz**=**"0.0" iyy**=**"3.015625e-05" iyz**=**"0.0" izz**=**"3.125e-07"**/>**

**</**inertial**>**

**<**collision**>**

**<**origin xyz**=**"0 0 0.03" rpy**=**"0 0 0"**/>**

**<**geometry**>**

**<**cylinder length**=**"0.06" radius**=**"0.0025"**/>**

**</**geometry**>**

**</**collision**>**

**<**visual**>**

**<**origin rpy**=**"0.0 0 0" xyz**=**"0 0 0"**/>**

**<**geometry**>**

**<**mesh filename**=**"package://my\_gurdy\_description/models/gurdy/meshes/gurdy\_higherleg\_v2.dae"**/>**

**</**geometry**>**

**</**visual**>**

**</**link**>**

**</**xacro:macro**>**

**<**xacro:macro name**=**"lowerleg\_link" params**=**"number"**>**

**<**link name**=**"lowerleg\_M${number}\_link"**>**

**<**inertial**>**

**<**origin xyz**=**"0 0 0.03" rpy**=**"0 0 0"**/>**

**<**mass value**=**"0.01" **/>**

**<**inertia ixx**=**"3.005625e-05" ixy**=**"0.0" ixz**=**"0.0" iyy**=**"3.005625e-05" iyz**=**"0.0" izz**=**"1.125e-07"**/>**

**</**inertial**>**

**<**collision**>**

**<**origin xyz**=**"0 0 0.03" rpy**=**"0 0 0"**/>**

**<**geometry**>**

**<**cylinder length**=**"0.06" radius**=**"0.0015"**/>**

**</**geometry**>**

**</**collision**>**

**<**visual**>**

**<**origin rpy**=**"0.0 0 0" xyz**=**"0 0 0"**/>**

**<**geometry**>**

**<**mesh filename**=**"package://my\_gurdy\_description/models/gurdy/meshes/gurdy\_lowerleg\_v2.dae"**/>**

**</**geometry**>**

**</**visual**>**

**</**link**>**

**<**joint name**=**"upperlegM${number}\_lowerlegM${number}\_joint" type**=**"revolute"**>**

**<**parent link**=**"upperleg\_M${number}\_link"**/>**

**<**child link**=**"lowerleg\_M${number}\_link"**/>**

**<**origin xyz**=**"0 0.0095 0.06" rpy**=**"0 0 3.14159"**/>**

**<**limit lower**=**"-2.9" upper**=**"1.5708" effort**=**"1.0" velocity**=**"0.005"**/>**

**<**axis xyz**=**"0 1 0"**/>**

**</**joint**>**

**</**xacro:macro**>**

**<**xacro:macro name**=**"foot\_link" params**=**"number colour"**>**

**<**link name**=**"footM${number}\_link"**>**

**<**inertial**>**

**<**origin xyz**=**"0 0 0" rpy**=**"0 0 0"**/>**

**<**mass value**=**"0.01" **/>**

**<**inertia ixx**=**"1.28e-06" ixy**=**"0.0" ixz**=**"0.0" iyy**=**"1.28e-06" iyz**=**"0.0" izz**=**"1.28e-06"**/>**

**</**inertial**>**

**<**collision**>**

**<**origin xyz**=**"0 0 0" rpy**=**"0 0 0"**/>**

**<**geometry**>**

**<**sphere radius**=**"0.008"**/>**

**</**geometry**>**

**</**collision**>**

**<**visual**>**

**<**origin rpy**=**"0.0 0 0" xyz**=**"0 0 0"**/>**

**<**geometry**>**

**<**sphere radius**=**"0.008"**/>**

**</**geometry**>**

**</**visual**>**

**</**link**>**

**<**gazebo reference**=**"footM${number}\_link"**>**

**<**kp**>**1000.0**</**kp**>**

**<**kd**>**1000.0**</**kd**>**

**<**mu1**>**10.0**</**mu1**>**

**<**mu2**>**10.0**</**mu2**>**

**<**material**>**Gazebo**/**${colour}**</**material**>**

**</**gazebo**>**

**<**joint name**=**"basefoot\_peg\_M${number}\_basefoot\_peg\_M${number}\_joint\_joint" type**=**"fixed"**>**

**<**parent link**=**"lowerleg\_M${number}\_link"**/>**

**<**child link**=**"footM${number}\_link"**/>**

**<**origin xyz**=**"0 0 0.06" rpy**=**"0 0 0"**/>**

**</**joint**>**

**</**xacro:macro**>**

**<**xacro:macro name**=**"upper\_transmission" params**=**"number"**>**

**<**transmission name**=**"tran${number}"**>**

**<**type**>**transmission\_interface**/**SimpleTransmission**</**type**>**

**<**joint name**=**"head\_upperlegM${number}\_joint"**>**

**<**hardwareInterface**>**EffortJointInterface**</**hardwareInterface**>**

**</**joint**>**

**<**actuator name**=**"motor${number}"**>**

**<**hardwareInterface**>**EffortJointInterface**</**hardwareInterface**>**

**<**mechanicalReduction**>**1**</**mechanicalReduction**>**

**</**actuator**>**

**</**transmission**>**

**</**xacro:macro**>**

**<**xacro:macro name**=**"lower\_transmission" params**=**"number number\_transm"**>**

**<**transmission name**=**"tran${number\_transm}"**>**

**<**type**>**transmission\_interface**/**SimpleTransmission**</**type**>**

**<**joint name**=**"upperlegM${number}\_lowerlegM${number}\_joint"**>**

**<**hardwareInterface**>**EffortJointInterface**</**hardwareInterface**>**

**</**joint**>**

**<**actuator name**=**"motor${number\_transm}"**>**

**<**hardwareInterface**>**EffortJointInterface**</**hardwareInterface**>**

**<**mechanicalReduction**>**1**</**mechanicalReduction**>**

**</**actuator**>**

**</**transmission**>**

**</**xacro:macro**>**

**<**material name**=**"blue"**>**

**<**color rgba**=**"0 0 0.8 1"**/>**

**</**material**>**

**<**material name**=**"red"**>**

**<**color rgba**=**"0.8 0 0 1"**/>**

**</**material**>**

**<**material name**=**"green"**>**

**<**color rgba**=**"0 0.8 0 1"**/>**

**</**material**>**

**<!--** **\*** **\*** **\*** Link Definitions **\*** **\*** **\*** **-->**

​

**<**link name**=**"base\_link"**>**

**<**collision**>**

**<**origin xyz**=**"0 0 0" rpy**=**"0 0 0"**/>**

**<**geometry**>**

**<**box size**=**"0.01 0.01 0.01"**/>**

**</**geometry**>**

**</**collision**>**

**<**visual**>**

**<**origin xyz**=**"0 0 0.01" rpy**=**"0 0 0"**/>**

**<**geometry**>**

**<**box size**=**"0.01 0.01 0.01"**/>**

**</**geometry**>**

**</**visual**>**

**</**link**>**

​

**<**link name**=**"head\_link"**>**

**<**inertial**>**

**<**origin xyz**=**"0 0 0.02" rpy**=**"0 0 0"**/>**

**<**mass value**=**"0.01" **/>**

**<**inertia ixx**=**"7.58333333333e-07" ixy**=**"0.0" ixz**=**"0.0" iyy**=**"7.58333333333e-07" iyz**=**"0.0" izz**=**"1.25e-06"**/>**

**</**inertial**>**

**<**collision**>**

**<**origin xyz**=**"0 0 0.02" rpy**=**"0 0 0"**/>**

**<**geometry**>**

**<**cylinder radius**=**"0.05" length**=**"0.04"**/>**

**</**geometry**>**

**</**collision**>**

**<**visual**>**

**<**origin rpy**=**"0.0 0 0" xyz**=**"0 0 0"**/>**

**<**geometry**>**

**<**mesh filename**=**"package://my\_gurdy\_description/models/gurdy/meshes/gurdy\_head\_v2.dae"**/>**

**</**geometry**>**

**</**visual**>**

**</**link**>**

​

**<**gazebo reference**=**"head\_link"**>**

**<**mu1**>**10.0**</**mu1**>**

**<**mu2**>**10.0**</**mu2**>**

**</**gazebo**>**

​

**<**joint name**=**"fixed" type**=**"fixed"**>**

**<**parent link**=**"base\_link"**/>**

**<**child link**=**"head\_link"**/>**

**</**joint**>**

​

**<**xacro:upperleg\_link number**=**"1"**/>**

**<**xacro:upper\_transmission number**=**"1"**/>**

​

**<**xacro:upperleg\_link number**=**"2"**/>**

**<**xacro:upper\_transmission number**=**"2"**/>**

​

**<**xacro:upperleg\_link number**=**"3"**/>**

**<**xacro:upper\_transmission number**=**"3"**/>**

​

​

**<**joint name**=**"head\_upperlegM1\_joint" type**=**"revolute"**>**

**<**parent link**=**"head\_link"**/>**

**<**child link**=**"upperleg\_M1\_link"**/>**

**<**origin xyz**=**"-0.02165 -0.0125 -0.008" rpy**=**"3.14159 0 0.523599"**/>**

**<**limit lower**=**"-1.55" upper**=**"0.0" effort**=**"1.0" velocity**=**"0.005"**/>**

**<**axis xyz**=**"0 1 0"**/>**

**</**joint**>**

**<**joint name**=**"head\_upperlegM2\_joint" type**=**"revolute"**>**

**<**parent link**=**"head\_link"**/>**

**<**child link**=**"upperleg\_M2\_link"**/>**

**<**origin xyz**=**"0.02165 -0.0125 -0.008" rpy**=**"3.14159 0 2.61799"**/>**

**<**limit lower**=**"-1.55" upper**=**"0.0" effort**=**"1.0" velocity**=**"0.005"**/>**

**<**axis xyz**=**"0 1 0"**/>**

**</**joint**>**

**<**joint name**=**"head\_upperlegM3\_joint" type**=**"revolute"**>**

**<**parent link**=**"head\_link"**/>**

**<**child link**=**"upperleg\_M3\_link"**/>**

**<**origin xyz**=**"0 0.025 -0.008" rpy**=**"3.14159 0 -1.5708"**/>**

**<**limit lower**=**"-1.55" upper**=**"0.0" effort**=**"1.0" velocity**=**"0.005"**/>**

**<**axis xyz**=**"0 1 0"**/>**

**</**joint**>**

​

​

​

**<**xacro:lowerleg\_link number**=**"1"**/>**

**<**xacro:lower\_transmission number**=**"1" number\_transm**=**"4"**/>**

​

**<**xacro:lowerleg\_link number**=**"2"**/>**

**<**xacro:lower\_transmission number**=**"2" number\_transm**=**"5"**/>**

​

**<**xacro:lowerleg\_link number**=**"3"**/>**

**<**xacro:lower\_transmission number**=**"3" number\_transm**=**"6"**/>**

​

**<**xacro:foot\_link number**=**"1" colour**=**"Red"**/>**

**<**xacro:foot\_link number**=**"2" colour**=**"Green"**/>**

**<**xacro:foot\_link number**=**"3" colour**=**"Blue"**/>**

​

**<**gazebo**>**

**<**plugin name**=**"gazebo\_ros\_imu\_controller" filename**=**"libgazebo\_ros\_imu.so"**>**

**<**robotNamespace**>/**gurdy**</**robotNamespace**>**

**<**topicName**>**imu**/**data**</**topicName**>**

**<**serviceName**>**imu**/**service**</**serviceName**>**

**<**bodyName**>**base\_link**</**bodyName**>**

**<**gaussianNoise**>**0**</**gaussianNoise**>**

**<**rpyOffsets**>**0 0 0**</**rpyOffsets**>**

**<**updateRate**>**5.0**</**updateRate**>**

**<**alwaysOn**>**true**</**alwaysOn**>**

**<**gaussianNoise**>**0**</**gaussianNoise**>**

**</**plugin**>**

**</**gazebo**>**

​

​

​

**</**robot**>**

As you may have seen, symmetry and repetition are the best features for XACROS to make a difference and simplify quite a bit.  
In essence, you can make macros for **one full leg,** and then you only have to use it with a different number three times. That's why all the macros have this **number** variable.  
In this example, you have the macros: upperleg\_link, lowerleg\_link, foot\_link, upper\_transmission, and the lower\_transmission. It is very intuitive, and it has been relatively easy because of the naming of each link and joint.  
Try to use numbers if elements repeat in a URDF, and then afterwards, MACROS will be easier to apply.  
The only thing that could be made macro, but does not need obvious math, is the **head\_upperlegMX\_joint**, which has been left as is.

Here you have an example of how you could have done the **spawn\_gurdy\_xacro.launch,** similar to the **spawn\_urdf.launch**:

In [ ]:



**<?**xml version**=**"1.0" encoding**=**"UTF-8"**?>**

​

**<**launch**>**

​

**<**arg name**=**"x" default**=**"0.0" **/>**

**<**arg name**=**"y" default**=**"0.0" **/>**

**<**arg name**=**"z" default**=**"0.0" **/>**

**<!--<**arg name**=**"xacro\_robot\_file" default**=**"" **/>-->**

**<**arg name**=**"robot\_name" default**=**"" **/>**

​

**<!--** Load the URDF into the ROS Parameter Server **-->**

**<**param name**=**"robot\_description" command**=**"$(find xacro)/xacro '$(find my\_gurdy\_description)/robot/gurdy.urdf.xacro'" **/>**

**<!--** Run a python script to the send a service call to gazebo\_ros to spawn a URDF robot **-->**

**<**node name**=**"urdf\_spawner" pkg**=**"gazebo\_ros" type**=**"spawn\_model" respawn**=**"false" output**=**"screen"

args**=**"-urdf -x $(arg x) -y $(arg y) -z $(arg z) -model $(arg robot\_name) -param robot\_description"**/>**

**</**launch**>**

And the main launch:

In [ ]:



**<?**xml version**=**"1.0" encoding**=**"UTF-8"**?>**

**<**launch**>**

**<**include file**=**"$(find my\_gurdy\_description)/launch/spawn\_gurdy\_xacro.launch"**>**

**<**arg name**=**"x" value**=**"0.5" **/>**

**<**arg name**=**"y" value**=**"0.0" **/>**

**<**arg name**=**"z" value**=**"0.18" **/>**

**<**arg name**=**"robot\_name" value**=**"gurdy" **/>**

**</**include**>**

**<**include file**=**"$(find my\_gurdy\_description)/launch/gurdy\_control.launch"**/>**

**</**launch**>**

Congratulations! You now know all the basics of URDF model creation.

Go on to the next unit to start the microproject.