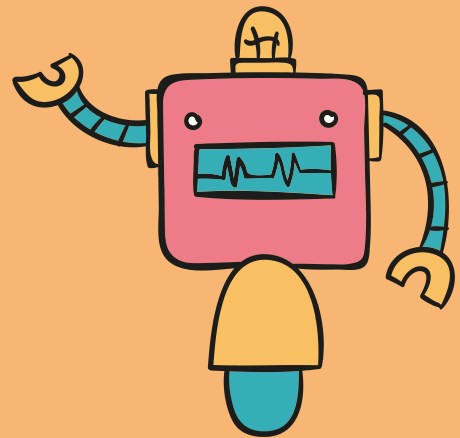
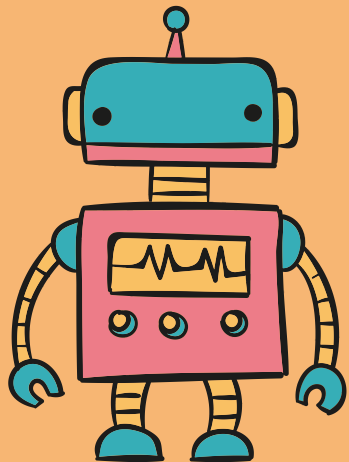
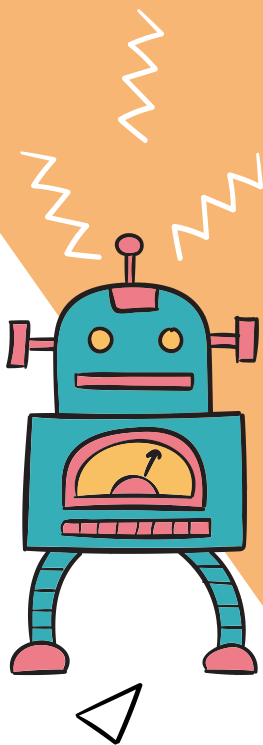


WaiterBot





01

Creating the Package

Creating a package for the robot

02

URDF

Creating the robot using a urdf file

03

Launching the Robot

Loading the URDF in rviz

04

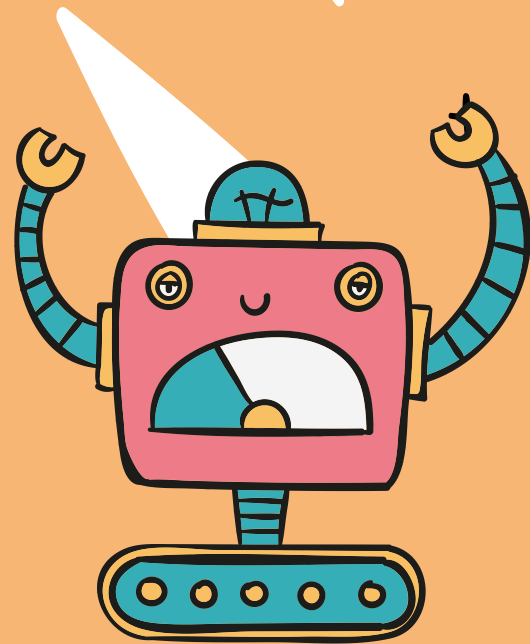
Output

The final robot



01.

Creating the Package



Inside the src folder of catkin workspace, create a package called “waiterbot” using the command

catkin_create_pkg waiterbot urdf rviz

Go to the catkin workspace and run the catkin_make command to build the workspace

catkin_make

Inside the waiterbot package, create launch and URDF folder for the files

mkdir launch

mkdir urdf

+

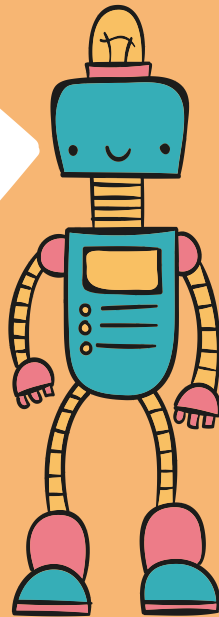
```
virtual-machine:~$ cd catkin_ws
virtual-machine:~/catkin_ws$ cd src
virtual-machine:~/catkin_ws/src$ \
> catkin_create_pkg waiterbot urdf rviz
Created file waiterbot/package.xml
Created file waiterbot/CMakeLists.txt
```

```
virtual-machine:~/catkin_ws/src$ cd ..
virtual-machine:~/catkin_ws$ catkin_make
Base path: /home/[redacted]/catkin_ws
Source space: /home/[redacted]/catkin_ws/src
Build space: /home/[redacted]/catkin_ws/build
Devel space: /home/[redacted]/catkin_ws/devel
Install space: /home/[redacted]/catkin_ws/install
```

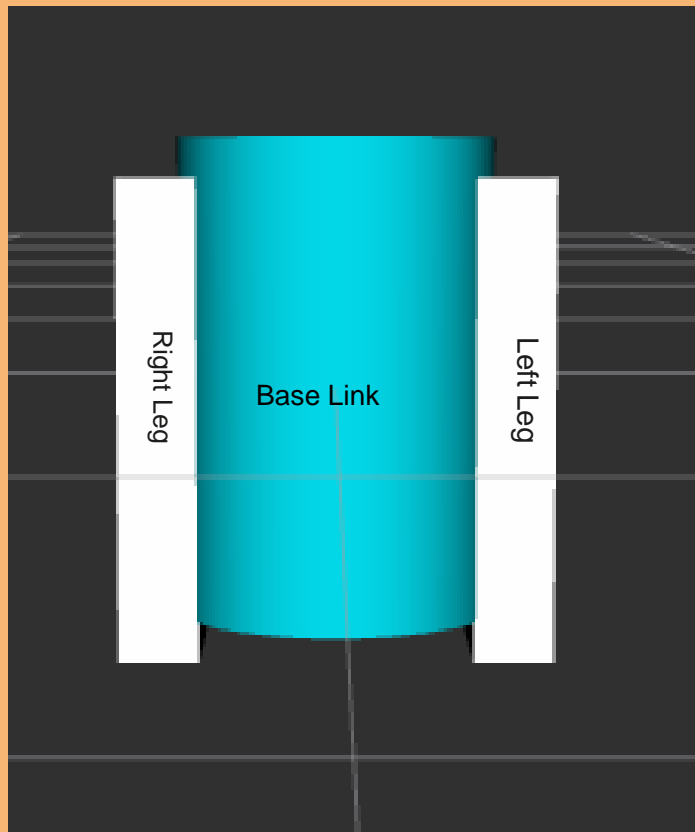
```
virtual-machine:~/catkin_ws$ . devel/setup.bash
virtual-machine:~/catkin_ws$ roscd waiterbot
virtual-machine:~/catkin_ws/src/waiterbot$ \
> mkdir launch
virtual-machine:~/catkin_ws/src/waiterbot$ \
> mkdir urdf
```

02.

URDF



BASE & LEGS



```
<link name="base_link">
  <visual>
    <geometry>
      <cylinder length="0.6" radius="0.2"/>
    </geometry>
    <material name="blue"/>
  </visual>
</link>
<link name="right_leg">
  <visual>
    <geometry>
      <box size="0.6 0.1 0.2"/>
    </geometry>
    <origin rpy="0 1.57075 0" xyz="0 0 -0.3"/>
    <material name="white"/>
  </visual>
</link>

<joint name="base_to_right_leg" type="fixed">
  <parent link="base_link"/>
  <child link="right_leg"/>
  <origin xyz="0 -0.22 0.25"/>
</joint>

<link name="left_leg">
  <visual>
    <geometry>
      <box size="0.6 0.1 0.2"/>
    </geometry>
    <origin rpy="0 1.57075 0" xyz="0 0 -0.3"/>
    <material name="white"/>
  </visual>
</link>

<joint name="base_to_left_leg" type="fixed">
  <parent link="base_link"/>
  <child link="left_leg"/>
  <origin xyz="0 0.22 0.25"/>
</joint>
```

```

<link name="right_base">
  <visual>
    <geometry>
      <box size="0.4 0.1 0.1"/>
    </geometry>
    <material name="white"/>
  </visual>
</link>

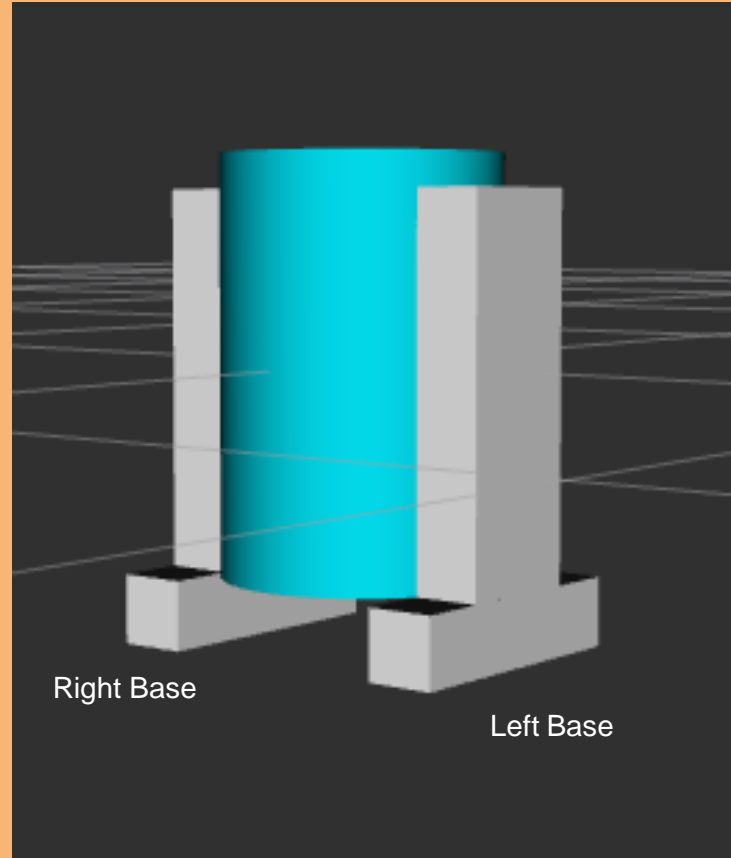
<joint name="right_base_joint" type="fixed">
  <parent link="right_leg"/>
  <child link="right_base"/>
  <origin xyz="0 0 -0.6"/>
</joint>

<link name="left_base">
  <visual>
    <geometry>
      <box size="0.4 0.1 0.1"/>
    </geometry>
    <material name="white"/>
  </visual>
</link>

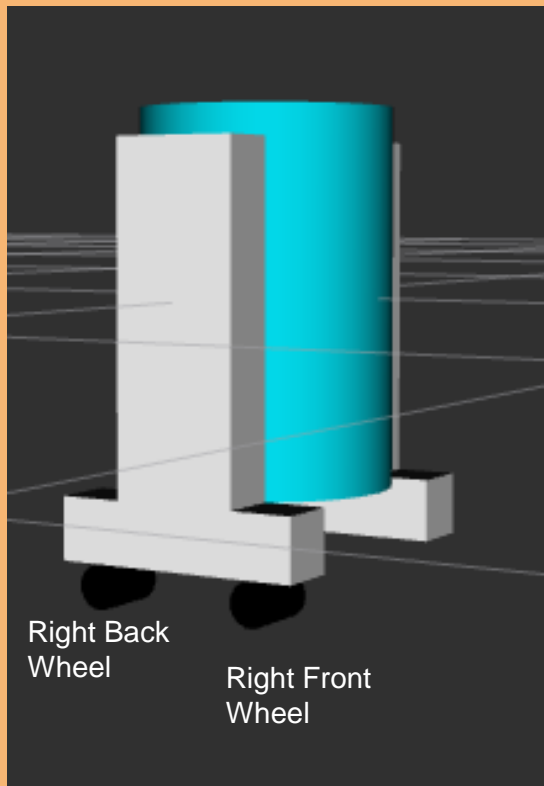
<joint name="left_base_joint" type="fixed">
  <parent link="left_leg"/>
  <child link="left_base"/>
  <origin xyz="0 0 -0.6"/>
</joint>

```

RIGHT AND LEFT BASE LINKS



RIGHT FRONT AND BACK WHEEL



```
<link name="right_front_wheel">
  <visual>
    <origin rpy="1.57075 0 0" xyz="0 0 0"/>
    <geometry>
      <cylinder length="0.1" radius="0.035"/>
    </geometry>
    <material name="black"/>
    <origin rpy="0 0 0" xyz="0 0 0"/>
  </visual>
</link>

<joint name="right_front_wheel_joint" type="continuous">
  <axis rpy="0 0 0" xyz="0 1 0"/>
  <parent link="right_base"/>
  <child link="right_front_wheel"/>
  <origin rpy="0 0 0" xyz="0.133333333333 0 -0.085"/>
</joint>

<link name="right_back_wheel">
  <visual>
    <origin rpy="1.57075 0 0" xyz="0 0 0"/>
    <geometry>
      <cylinder length="0.1" radius="0.035"/>
    </geometry>
    <material name="black"/>
    <origin rpy="0 0 0" xyz="0 0 0"/>
  </visual>
</link>

<joint name="right_back_wheel_joint" type="continuous">
  <axis rpy="0 0 0" xyz="0 1 0"/>
  <parent link="right_base"/>
  <child link="right_back_wheel"/>
  <origin rpy="0 0 0" xyz="-0.133333333333 0 -0.085"/>
</joint>
```

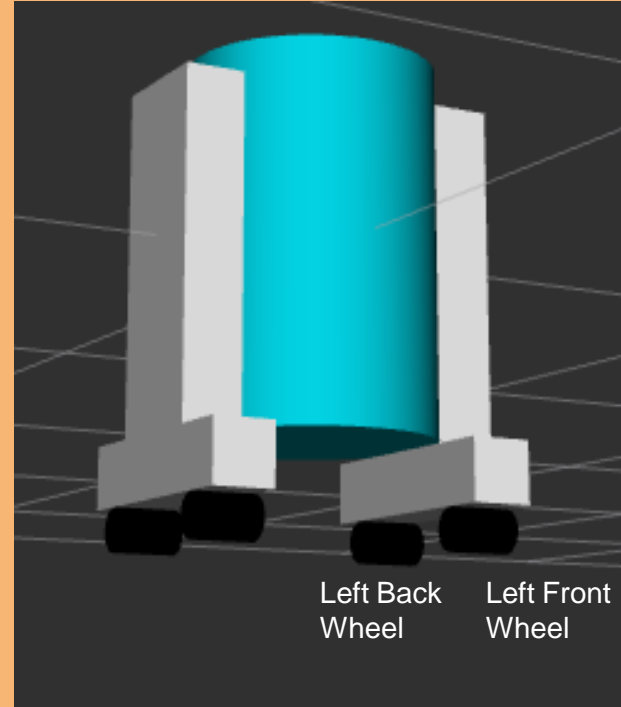

LEFT FRONT AND BACK WHEEL

```
<link name="left_front_wheel">
  <visual>
    <origin rpy="1.57075 0 0" xyz="0 0 0"/>
    <geometry>
      <cylinder length="0.1" radius="0.035"/>
    </geometry>
    <material name="black"/>
  </visual>
</link>

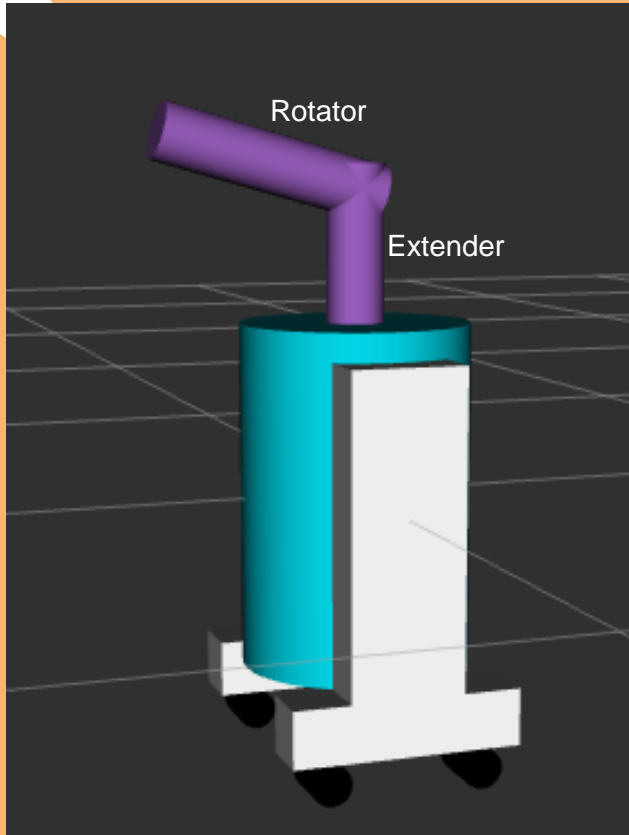
<joint name="left_front_wheel_joint" type="prismatic">
  <axis rpy="0 0 0" xyz="0 1 0"/>
  <parent link="left_base"/>
  <child link="left_front_wheel"/>
  <origin rpy="0 0 0" xyz="0.133333333333 0 -0.085"/>
  <limit effort="1000.0" lower="-0.1" upper="0.15" velocity="0.5"/>
</joint>

<link name="left_back_wheel">
  <visual>
    <origin rpy="1.57075 0 0" xyz="0 0 0"/>
    <geometry>
      <cylinder length="0.1" radius="0.035"/>
    </geometry>
    <material name="black"/>
  </visual>
</link>

<joint name="left_back_wheel_joint" type="prismatic">
  <axis rpy="0 0 0" xyz="0 1 0"/>
  <parent link="left_base"/>
  <child link="left_back_wheel"/>
  <origin rpy="0 0 0" xyz="-0.133333333333 0 -0.085"/>
  <limit effort="1000.0" lower="-0.1" upper="0.15" velocity="0.5"/>
</joint>
```



Extender & Rotator



```
<link name="extender">
  <visual>
    <origin xyz="0 0 0.3" rpy="0 0 0" />
    <geometry>
      <cylinder length="0.4" radius="0.05"/>
    </geometry>
    <material name="amethyst"/>
  </visual>
</link>

<joint name="base_to_extender" type="prismatic">
  <parent link="base_link"/>
  <child link="extender"/>
  <axis xyz="0 0 1"/>
  <limit effort="1000.0" lower="-0.1" upper="0.15" velocity="0.5"/>
</joint>

<link name="rotator">
  <visual>
    <origin xyz="0.15 0 0.5" rpy="0 1.3 0" />
    <geometry>
      <cylinder length="0.4" radius="0.05"/>
    </geometry>
    <material name="amethyst"/>
  </visual>
</link>

<joint name="extender_to_rotator" type="continuous">
  <parent link="extender"/>
  <child link="rotator"/>
  <axis xyz="0 0 1"/>
</joint>
```

End Effector & Tray +

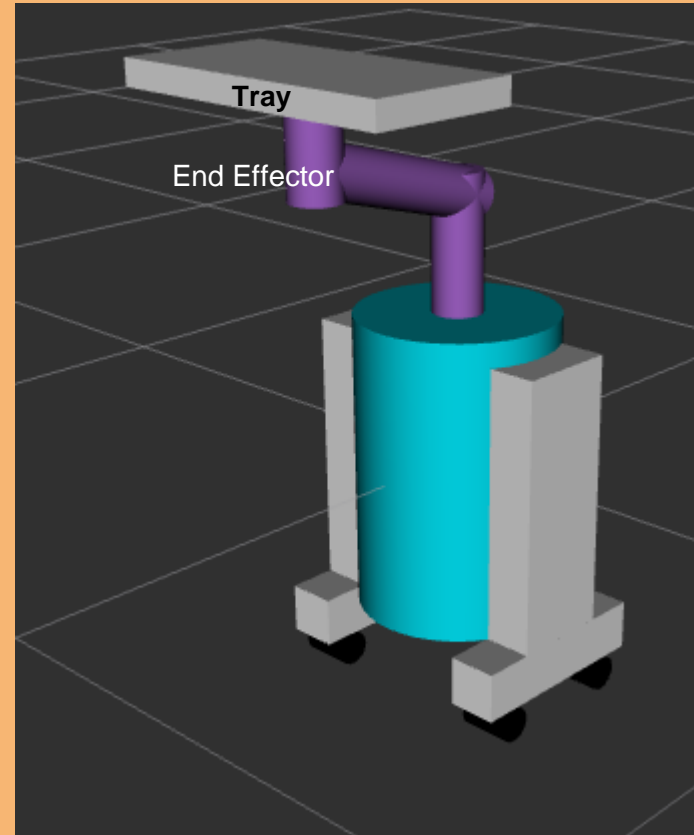


```
<link name="end_effector">
  <visual>
    <origin xyz="0.35 0 0.6" rpy="0 0 0" />
    <geometry>
      <cylinder length="0.2" radius="0.05"/>
    </geometry>
    <material name="amethyst"/>
  </visual>
</link>

<joint name="rotator_to_endeff" type="fixed">
  <parent link="rotator"/>
  <child link="end_effector"/>
</joint>

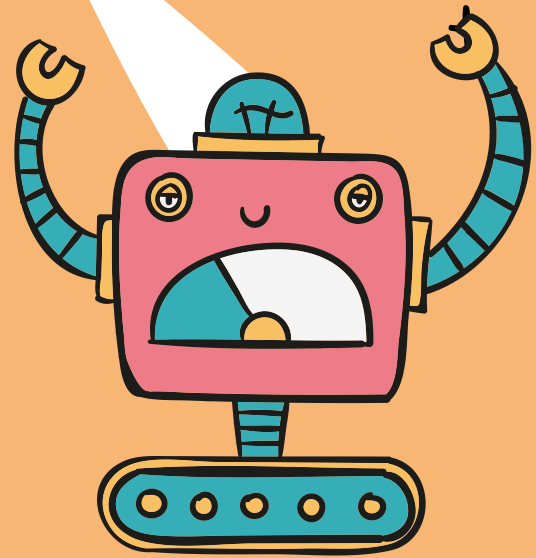
<link name="tray">
  <visual>
    <origin xyz="0.35 0 0.7" rpy="0 0 0" />
    <geometry>
      <box size="0.3 0.6 0.05"/>
    </geometry>
    <material name="white"/>
  </visual>
</link>

<joint name="end_to_tray" type="fixed">
  <parent link="end_effector"/>
  <child link="tray"/>
</joint>
```



03.

Launching the Robot



Inside the launch folder of the waiterbot package, create a .launch file
nano display.launch

- The argument **gui** is used to control the robot
- The **joint_state_publisher** finds all of the non-fixed joints and publishes a JointState message with all those joints defined
- Since **gui** is present, the **joint_state_publisher** displays the joint positions in a window as sliders
- The **robot_state_publisher** uses URDF specified by **robot_description** and **joint_state_publisher** to calculate the forward kinematics of the robot and publish the results.

```
<launch>
<arg name="model" default="$(find waiterbot)/urdf/visual.urdf"/>
<arg name="gui" default="true" />
<param name="robot_description" command="$(find xacro)/xacro $(arg model)" />
  <node if="$(arg gui)" name="joint_state_publisher" pkg="joint_state_publisher_gui" type="joint_state_publisher_gui" />
  <node unless="$(arg gui)" name="joint_state_publisher" pkg="joint_state_publisher" type="joint_state_publisher" />
  <node name="robot_state_publisher" pkg="robot_state_publisher" type="robot_state_publisher" />
  <node name="rviz" pkg="rviz" type="rviz" />
</launch>
```

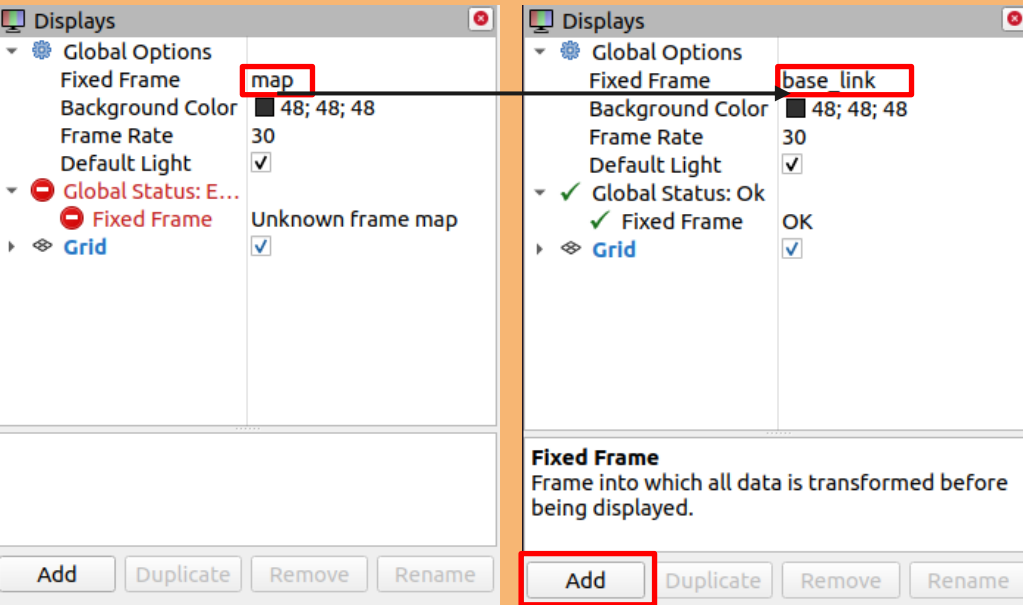
The following command, will launch rviz
roslaunch waiterbot display.launch
However, to display the robot, a few
more configurations are needed

```
-virtual-machine:~$ roslaunch waiterbot display.launch
... logging to /home/.../.ros/log/3bc345a8-be26-11eb-a65c-43486f
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

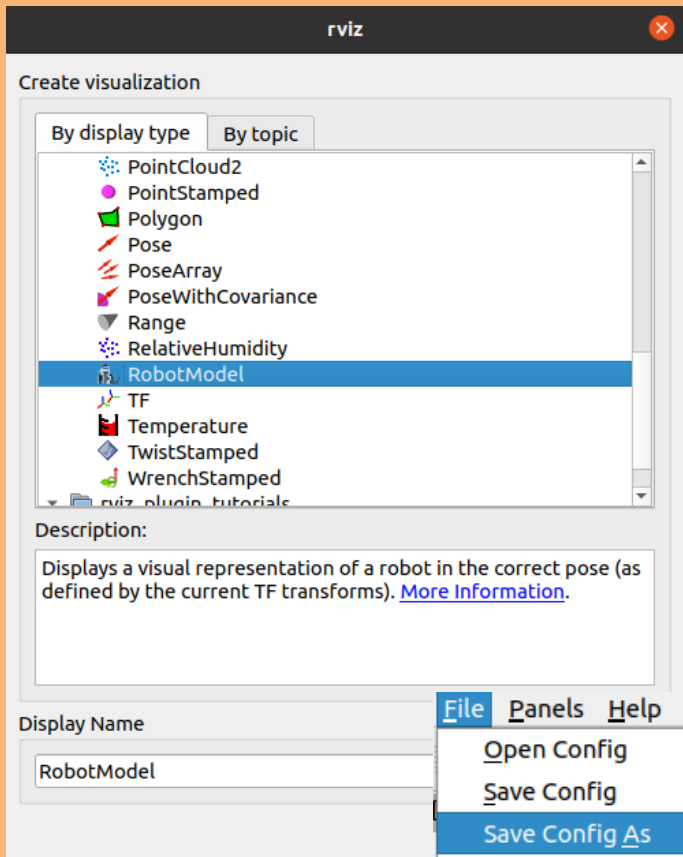
started roslaunch server http://...-virtual-machine:40915/

SUMMARY
=====

PARAMETERS
* /robot_description: <?xml version="1...
* /rostdistro: noetic
* /rosversion: 1.15.9
```



Change the “Fixed Frame” from
map to base_link, then click “Add”



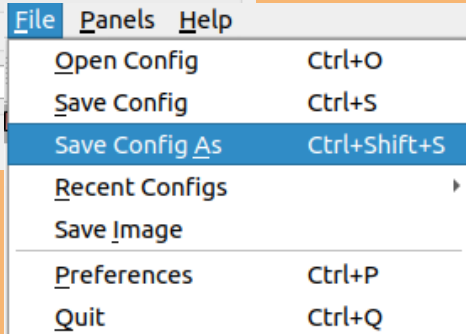
Select the robot model option
Now the robot will appear, once the robot appears:

1. Click on File
2. Click on Save configuration as
3. Save the file in the launch directory of the package as "config.rviz"
4. Add the line

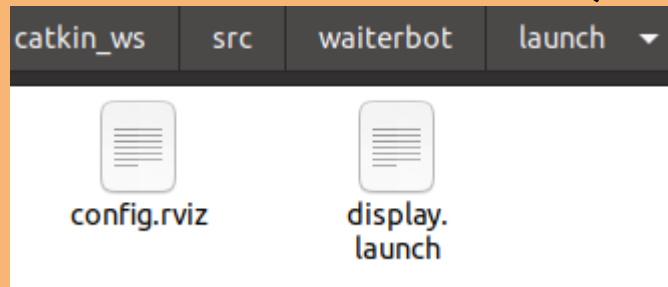
```
args="-d $(find waiterbot)/launch/config.rviz"  
required="true"
```

to the launch file

This will help directly load the robot file without any further configurations



The launch directory

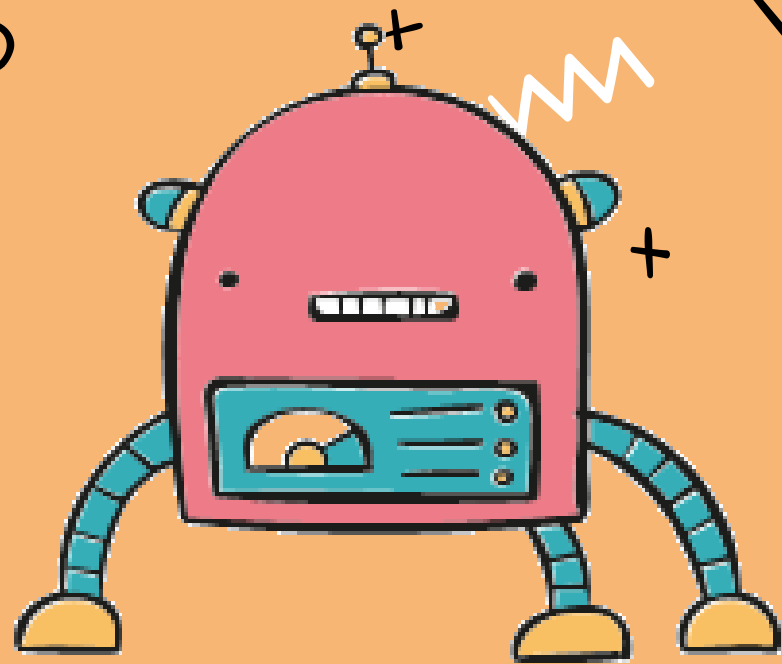


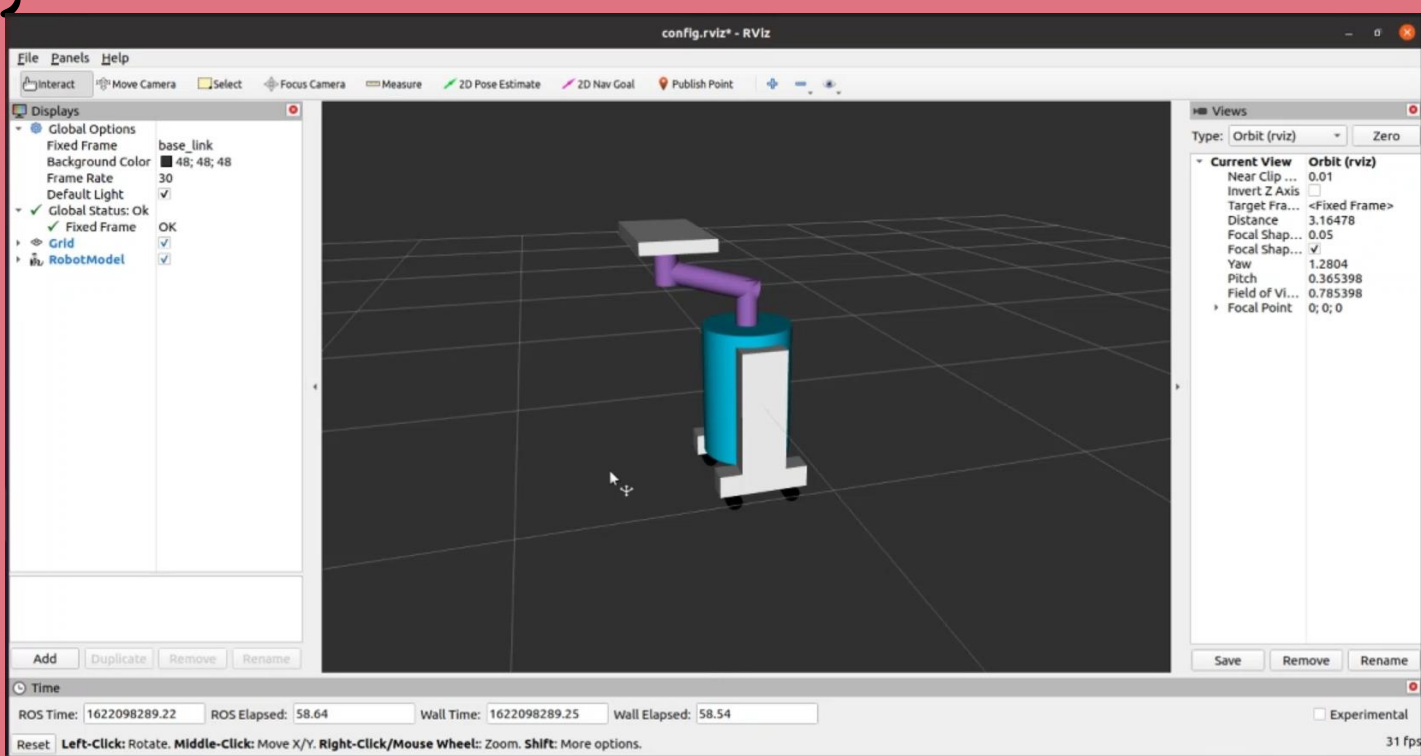
The new launch file

```
<launch>
<arg name="model" default="$(find waiterbot)/urdf/visual.urdf"/>
<arg name="gui" default="true" />
<param name="robot_description" command="$(find xacro)/xacro $(arg model)" />
  <node if="$(arg gui)" name="joint_state_publisher_gui" pkg="joint_state_publisher_gui" type="joint_state_publisher_gui" />
  <node unless="$(arg gui)" name="joint_state_publisher" pkg="joint_state_publisher" type="joint_state_publisher" />
  <node name="robot_state_publisher" pkg="robot_state_publisher" type="robot_state_publisher" />
  <node name="rviz" pkg="rviz" type="rviz" args="-d $(find waiterbot)/launch/config.rviz" required="true" />
</launch>
```


04.

Output





Rqt_graph



Nodes

```
-virtual-machine:~$ \
> rostopic list
> rosnode list
/joint_state_publisher
/robot_state_publisher
/rosout
/rviz
```



THANK YOU!

