

# Rviz simulation in ros2 environment

[Rviz simulation in ros2 environment](#)  
[Rviz simulation in ros2 environment](#)  
[Quickly use](#)  
[Configure rviz2 from the model](#)

## Rviz simulation in ros2 environment

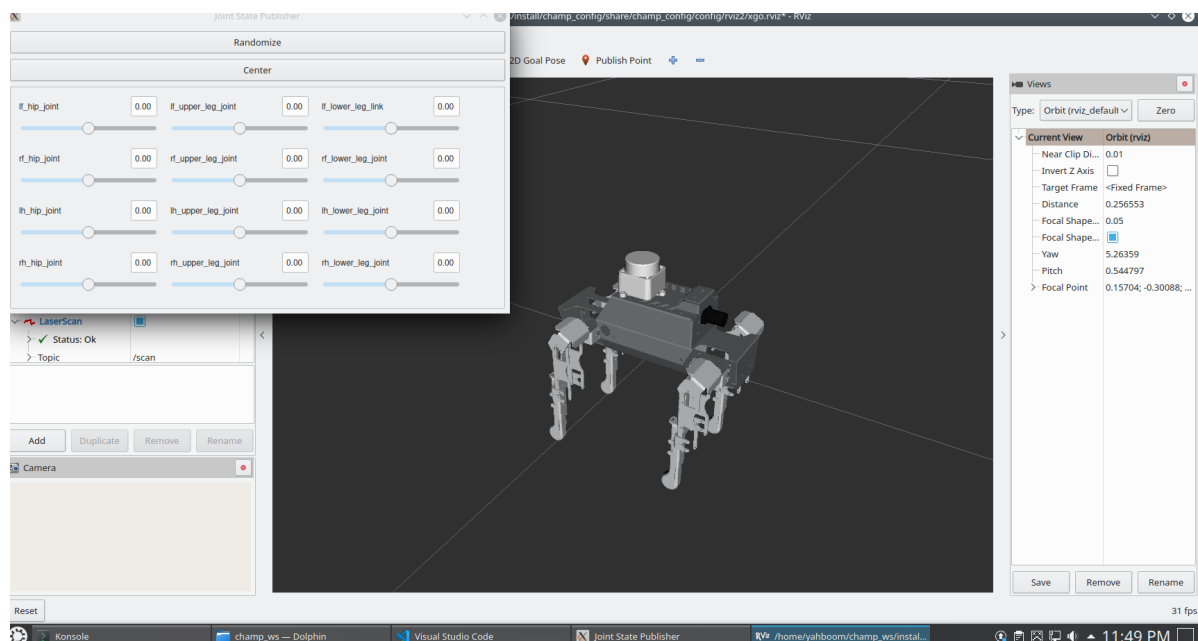
Strictly speaking, rviz is not a simulation software, but it can visually display the state of the robot and pass the relevant joint\_states\_Publisher controls the displayed joint posture of the robot. Please see section 9.0 first. Some instructions will not be explained in detail.

### Quickly use

This section describes how to use

```
ros2 launch champ_config display.launch.py
```

The display is as follows:



The posture of each joint can be adjusted through the slider in the small window. The figure shows the initial position (the initial position is determined by STL and URDF files exported by SOLIDWORK. The initial position is as shown above, which can facilitate the kinematic calculation of subsequent simulation)

The corresponding relationship of each joint is as follows:

The corresponding relationship of each joint is as follows:

```
/**:  
  ros__parameters:  
    joints_map:
```

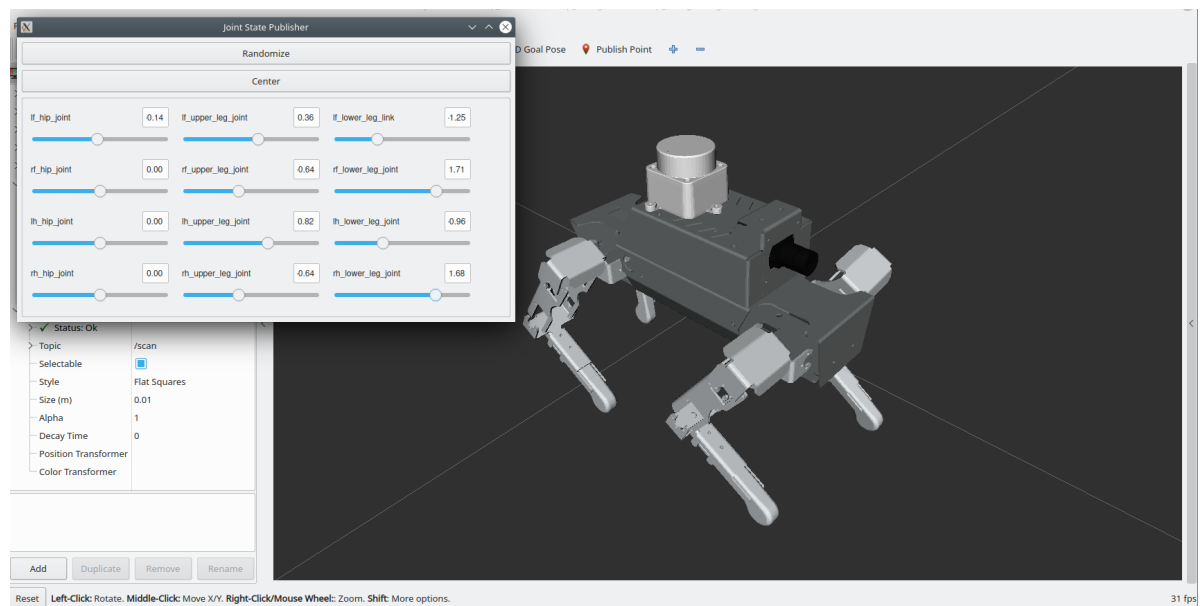
```

left_front:      #Left front leg
\-- lf_hip_joint    #shoulder
\-- lf_upper_leg_joint  #Boom
\-- lf_lower_leg_joint  #Forearm
\-- lf_foot_joint    #Toe (virtual joint)
right_front:
\-- rf_hip_joint
\-- rf_upper_leg_joint
\-- rf_lower_leg_joint
\-- rf_foot_joint

left_hind:
\-- lh_hip_joint
\-- lh_upper_leg_joint
\-- lh_lower_leg_joint
\-- lh_foot_joint
right_hind:
\-- rh_hip_joint
\-- rh_upper_leg_joint
\-- rh_lower_leg_joint
\-- rh_foot_joint

```

After adjust, as shown in the following figure.



## Configure rviz2 from the model

### Advanced part

#### Model modification

Change the suffix name of the URDF file exported by SolidWorks to xacro, as shown in the following figure (add a sentence after the robot name)

Check whether the joint setting is correct

```

<joint
  name="lf_hip_joint"
  type="revolute">
  <origin
    xyz="0.09332 0.022407 0.01454"
    rpy="0 0 0" />
  <parent
    link="base_link" />
  <child
    link="lf_hip_link" />
  <axis
    xyz="-1 0 0" />

```

## Create launch file

```

import os
from ament_index_python.packages import get_package_prefix
from ament_index_python.packages import get_package_share_directory
from launch import LaunchDescription
from launch.actions import DeclareLaunchArgument, ExecuteProcess,
IncludeLaunchDescription
from launch.conditions import IfCondition, UnlessCondition
from launch.launch_description_sources import PythonLaunchDescriptionSource
from launch.substitutions import Command, LaunchConfiguration, PythonExpression
from launch_ros.actions import Node
from launch_ros.substitutions import FindPackageShare

import xacro
def generate_launch_description():
    \# Constants for paths to different files and folders
    \# Set the path to different files and folders.
    \#Specify path:
    default_xacro_model_path =
os.path.join(get_package_share_directory('champ_description'), 'urdf/',
'xgo_rviz.xacro')
    default_rviz_config_path =
os.path.join(get_package_share_directory('champ_config'), 'config/rviz2/',
'xgo.rviz')
    \# Launch configuration variables specific to simulation
    \#Configuration variables
    gui = LaunchConfiguration('gui')
    rviz_config_file = LaunchConfiguration('rviz_config_file')
    \# Declare the launch arguments
    \#Declare startup parameters
    declare_use_joint_state_publisher_cmd = DeclareLaunchArgument(

```

```

name='gui',
default_value='True',
description='Flag to enable joint_state_publisher_gui')
declare_rviz_config_file_cmd = DeclareLaunchArgument(
name='rviz_config_file',
default_value=default_rviz_config_path,
description='Full path to the RVIZ config file to use')
declare_xacro_model_path_cmd = DeclareLaunchArgument(
name='xacro_model',
default_value=default_xacro_model_path,
description='Absolute path to robot xacro file')
declare_use_robot_state_pub_cmd = DeclareLaunchArgument(
name='use_robot_state_pub',
default_value='True',
description='Whether to start the robot state publisher')
declare_use_rviz_cmd = DeclareLaunchArgument(
name='use_rviz',
default_value='True',
description='Whether to start RVIZ')
robot_description_config = xacro.process_file(default_xacro_model_path)
robot_desc = robot_description_config.toxml()
\# Subscribe to the joint states of the robot, and publish the 3D pose of each
link.
\#Robot status release
start_robot_state_publisher_cmd = Node(
package="robot_state_publisher",
executable="robot_state_publisher",
name="robot_state_publisher",
parameters=[
{"robot_description": robot_desc}],
output="screen")
\# Publish the joint state values for the non-fixed joints in the URDF file.
\#Robot joint status publishing node
start_joint_state_publisher_cmd = Node(
condition=UnlessCondition(gui),
package='joint_state_publisher',
executable='joint_state_publisher',
name='joint_state_publisher')
\# A GUI to manipulate the joint state values
\#Robot joint state publishing node (with GUI)
start_joint_state_publisher_gui_node = Node(
condition=IfCondition(gui),
package='joint_state_publisher_gui',
executable='joint_state_publisher_gui',
name='joint_state_publisher_gui')
\# Launch RViz2
\#Start rviz2
start_rviz_cmd = Node(
package='rviz2',
executable='rviz2',
name='rviz2',
output='screen',
arguments=['-d', rviz_config_file])
return LaunchDescription(
[

```

```
declare_use_joint_state_publisher_cmd,  
declare_rviz_config_file_cmd,  
declare_xacro_model_path_cmd,  
declare_use_robot_state_pub_cmd,  
declare_use_rviz_cmd,  
start_robot_state_publisher_cmd,  
start_joint_state_publisher_cmd,  
start_joint_state_publisher_gui_node,  
start_rviz_cmd,  
]  
)
```