Chapter - 2

Mastering_ROS_for_Robotics_Program

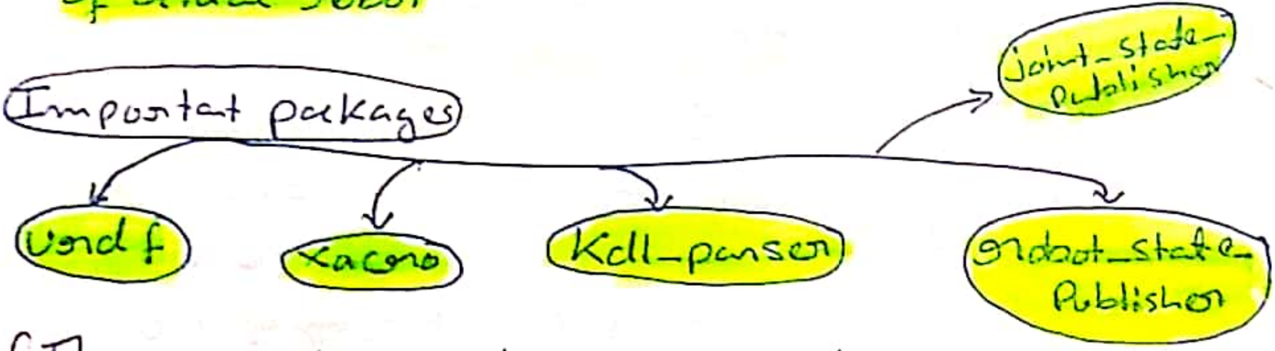
by → Lentin_Joseph

⇒ The first phase in robot manufacturing is its design and modeling.

→ {Blender}
{Solid works}

→ {One of the main purposes of modeling is simulation}

"The Virtual robot model must have all the characteristics of real hardware, the shape of robot may or may not look like the actual robot but it must be an abstract, which has all the physical characteristics of actual robot"



{These pakages help us create the 3D robot model description with the exact characteristics of real hardware.}

→ We can define a robot model, Sensors, and a working environment using URDF.

→ Flexible links can't be represented using URDF.

* joint_state_publisher

⇒ This package contains a node called joint_state_publisher, which reads the robot model description, find all joints, and publishes joint values to all nonfixed joints using GUI slider.

* Kdl_parser

⇒ Kinematic and Dynamic library (KDL) is an ROS package that contains parser tools to build a KDL tree from the URDF representation.

↳ used to publish the joint states and also to forward and inverse kinematics of the robot.

* robot_state_publisher

⇒ This package reads the Current robot joint states and publishes the 3D poses of each robot link using the Kinematics tree build from the urdf.

↳ 3D pose of the robot is published as ROStf.

→ Publishes the relationship between coordinates frames of a robot.

gazebo tag in urdf

-> Used when we Include the simulation Parameters of the gazebo simulator inside URDF.

-> We Can use this tag to include gazebo plugins, gazebo material properties etc.

=> The collision and inertia parameters are required in each link; otherwise, Gazebo will not load the robot model properly.

Conversion of xacro to URDF

```
rosrun xacro xacro.py filename.xacro ~~[illegible]~~ > filename.urdf
```

=> We can use the following line of code in the ROS launch file for converting xacro to URDF and use it as a robot-description parameter:

```
<Param name="robot_description" command="$(find
xacro)/xacro.py $(find pakagename)/urdf/filename.xacro"
/>
```

* transmission tag

=> Transmission tag relates a joint to an actuator.

=> It defines the type of transmission that we are using in a particular joint and type of motor and its parameters.

=> It also defines the type of hardware interface we use when we interface with the ROS controllers.

★ Understanding joint state publisher

⇒ It finds the nonfixed joints from the URDF model and publishes the joint state values of each joint in the sensor_msgs/JointState message format.

⇒ If we set use_gui to true, the joint_state_publisher node displays a slider based Control window to control each joint.

↳ The lower and upper value of a joint will be taken from the lower & upper values associated with the limit tag used inside the joint tag.

★ Understanding the robot state publisher

⇒ Robot state publisher package helps to Publish the state of the robot to tf.

⇒ This package subscribes to joint states of the robot and publishes the 3D pose of each link using the kinematic representation from the URDF model

———×———×———