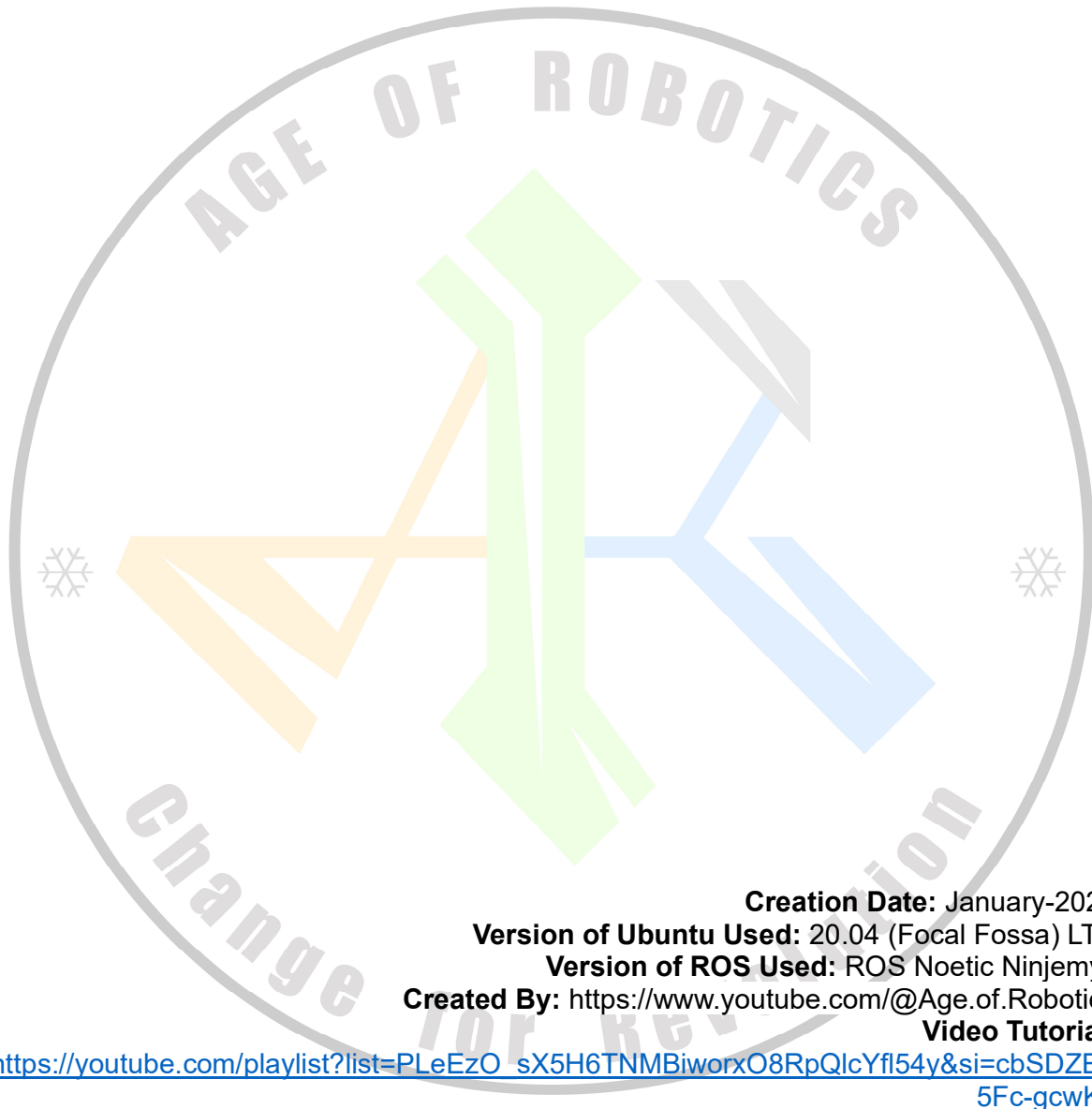


Simulate Your Custom Robotic Arm in ROS Noetic Using Moveit



Creation Date: January-2024

Version of Ubuntu Used: 20.04 (Focal Fossa) LTS

Version of ROS Used: ROS Noetic Ninjemys

Created By: <https://www.youtube.com/@Age.of.Robotics>

Video Tutorial:

https://youtube.com/playlist?list=PLeEzO_sX5H6TNMBiworxO8RpQlcYfl54y&si=cbSDZEq5Fc-gcwK2

This Tutorial is for those who want to simulate their own Robotic Arm in “Robot Operating System” from Scratch.

Contents

Introduction.....	3
List of Video Lessons on YouTube	4
1 Conventions Used.....	5
2 Install ROS Noetic and Create CATKIN Workspace	6
2.1 ROS installation	6
2.2 Setup a CATKIN Workspace.....	6
3 Overview of ROS Package Exported From SOLIDOWRKS.....	8
4 Add the URDF Package To Your Workspace and Add The Dependencies.....	10
4.1 Copy the ROS package to your catkin workspace.	10
4.2 Edit the CmakeLists.txt	10
4.3 Edit the package.xml file.....	12
4.4 Modify the URDF file to make it suitable for Simulation	14
4.5 Write a .yaml file to define ROS controllers to be used for different joints and publishing joint states.	17
4.6 Create a .launch file to spawn/open your robotic arm in gazebo	19
4.7 Build your catkin workspace.....	21
4.8 Launch your robot's launch file to load it 1st time in GAZEBO.	21
5 Creating a new Manipulator Package to control the Robotic Arm URDF using Moveit Setup Assistant.....	23
5.1 Creating the Moveit Package using Moveit Setup Assistant to control our URDF file.....	23
5.2 Testing the Moveit Package Generated using Moveit Setup Assistance.....	24
6 Configuring the Moveit Package to work properly.	25
6.1 Writing a new ROS Controller for connecting the joint_trajectory_controller with ROS manipulator or moveit package.....	25
6.2 Edit the "simple_moveit_controller_manager.launch.xml" file available in the launch folder of your moveit package.	27
6.3 Create a new launch file to launch moveit simulation in Rviz and Gazebo.....	28
6.4 Build your catkin workspace.....	30
6.5 Launch your robot's moveit launch file to load it moveit simulation with Rviz and Gazebo	30
6.6 Solving the possible issues in after launching the simulation:.....	31
7 Further Steps.....	32
8 References:	33

Introduction

In 2022, I had created a video series on simulating custom robotic arm using ROS Melodic. In that tutorial, I had explained the process of exporting the URDF file for the custom robotic arm using SOLIDWORKS and importing it in ROS Melodic and simulating it using Moveit, Rviz and Gazebo.

You can find the series for ROS Melodic here:

https://youtube.com/playlist?list=PLeEzO_sX5H6TBD6EMGgV-qdhzxPY19m12

Lot of learns used those lessons to learn the process. But, many of them using ROS Noetic and are facing issues while creating the simulation packages. So, I am creating this new tutorial series for those who are using ROS Noetic on Ubuntu 20.04

It was not possible to make a single video, as it will be very long. So, I have divided the tutorials in multiple lessons, so you can follow then step by step in small chunks.

You can find the series for ROS Noetic here:

https://youtube.com/playlist?list=PLeEzO_sX5H6TNMBiworxO8RpQlcYfl54y&si=cbSDZEq5Fc-gcwK2

GitHub Repository to get the necessary files for this this tutorial:

TBA

Note: Here, I will not provide the ready to use packages, as it is better to build them step by step to troubleshoot the missing components on your system. Rather, I will provide the necessary files to create the packages.

The process to export the URDF Package will be same. So, I will not repeat the process, I will add the same lessons for this step in the playlist.

The process from lesson

Thank you for refereeing this tutorial. Hope it will be helpful to you and you will be able to simulate your own custom robot in ROS Noetic.

List of Video Lessons on YouTube

Lessons to Export URDF File from SolidWorks are same from the series for ROS Melodic and the exported package can be used with ROS Noetic without any issues:

- **Lesson 1:** Assemble the Robotic Arm Properly in SOLIDWORKS: <https://youtu.be/Bsh3DWmQ-uM>
- **Lesson 2:** Set the zero positions of the joints and set axis systems: <https://youtu.be/g7mZp8hnIok>
- **Lesson 3:** Export the URDF file from SOLIDWORKS: https://youtu.be/I08lO_SRBbk

Below are the lessons for using the URDF Package for simulation in ROS Noetic. There are few different steps at some points than ROS Melodic Series:

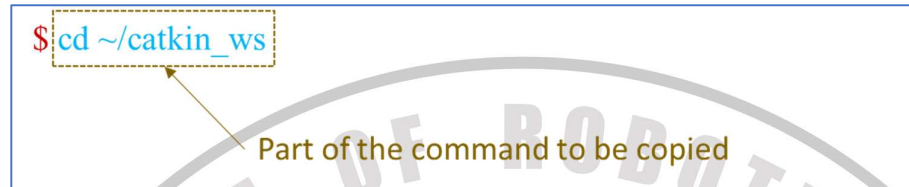
- **Lesson 4:** Create a new Catkin
- **Lesson 5:** Copy the URDF Package in your workspace and add dependencies for your custom package: TBA
- **Lesson 6:** Modify the URDF file to make it suitable for Simulation: TBA
- **Lesson 7:** Create a JointTrajectoryController to control your robotic arm's joints: TBA
- **Lesson 8:** Create a ". Launch" file to load your URDF and controllers in Gazebo: TBA
- **Lesson 9:** Build the Catkin Workspace and launch the URDF for 1st time in Gazebo: TBA
- **Lesson 10:** Create a Moveit Package to manipulate (Adding Motion) your robotic arm using "Moveit Setup Assistant": TBA
- **Lesson 11:** Testing the Moveit Package by launching the default launch files: TBA
- **Lesson 12:** Replace the Default ROS Controller file with New ROS Controller file: TBA
- **Lesson 13:** Create a ". Launch" file to load full simulation: TBA
- **Lesson 14:** Launch the simulation and plan some motion: TBA
- **Lesson 15:** Set Predefined Position of Robotic Arm Using Python Script: TBA
- **Lesson 16:** Print End Effector Position and Joint Angles Using Python Script: TBA

1 Conventions Used

In this document, some colour coding is done to distinguish some notations. They are as given below

- **Command to execute in terminal:**

To distinguish one command from another command, \$ is used at the beginning of a new command. The part after the \$ symbol is the actual command.



Reader should only copy the blue part of the command and not the \$ symbol.

- **Red coloured Code:** The lines in a code snippet displayed by RED colour are already available in the file which is asked to be edited.
- **Green Coloured Code:** The lines in a code snippet displayed by GREEN colour need to be added by the reader in the file which is asked to be edited.
- **Purple Coloured Code:** The Part of the code snippet, reader need to change before pasting in the desired file.
- **Home directory:** “~/” this part in a command means your home directory.
- **Notes:** Notes are written in *italics*.
- **Comments:**

- o Comments in YAML starts with #
#This is YAML comment
- o Comments in XML are enclosed in <!-- and -->
<!--This is XML Comment -->

2 Install ROS Noetic and Create CATKIN Workspace

Link to the full YouTube video tutorial playlist: [TBA](#)

2.1 ROS installation

If you haven't installed ROS yet, follow <https://wiki.ros.org/noetic/Installation/Ubuntu> this page.

I am using Ubuntu **20.04 (Focal Fossa)** and **ROS Noetic Ninjemys**.

Every ROS version is created for different Ubuntu versions. If you have another version of Ubuntu, download the version of ROS that is supported by your version of ubuntu. For more details:

<http://wiki.ros.org/Distributions>

Important Note: If you are using different version of ROS, the code provided in this tutorial may not work for you.

Same Tutorials for ROS Melodic:

2.2 Setup a CATKIN Workspace

Full YouTube Tutorial: [TBA](#)

If you are using ROS from long time and have you catkin workspace already created, you can just use that for this tutorial.

But, if you are new or want to create a separate workspace for this project, just copy and run the commands given below in your Ubuntu Terminal one by one.

Pre-requisites: Installing CATKIN Tools and std_msgs package

1. Update your Ubuntu packages
`$ sudo apt-get update`
1. Install CATKIN Tools for ROS Noetic.
`$ sudo apt-get install ros-noetic-catkin python3-catkin-tools`
2. Install std_msgs package
`$ sudo apt install ros-noetic-std-msgs`

Create Your CATKIN Workspace

2. Update your Ubuntu packages
`$ sudo apt-get update`
3. Go to home directory
`$ cd ~/`
4. Create a catkin workspace folder along with src folder in it. I am creating a workspace with name "moveit_ws".
`$ mkdir --parents moveit_ws/src`
You can use any name for your workspace like "catkin_ws".
5. Go to the catkin workspace you created.
`$ cd ~/moveit_ws`
6. Initialize the Catkin workspace
`$ catkin init`

<https://www.youtube.com/channel/UCNaMh2qgE3nU2euCg0EJ8NQ>

7. Go to the catkin workspace directory that you created if not already in it.

```
$ cd ~/moveit_ws
```

8. Build your workspace.

```
$ catkin build
```

After initializing and building your CATKIN workspace, it will create various folders like devel, src, .etc in your workspace folder.

9. Source the "setup.bash" file, which is automatically generated in your catkin workspace's "devel" folder

If you are already in your catkin workspace:

```
$ source devel/setup.bash
```

If you are not in your catkin workspace, then give full path:

```
$ source ~/moveit_ws/devel/setup.bash
```

Sourcing the setup.bash files makes the workspace active and ROS master aware of the files and packages available in your workspace. You need to source this file every time you launch a new terminal.