Setting Up a Ubuntu 20.04-ROS environment

Last updated by Peter Kennedy, May 2022

Table of Contents

[Part I: Setting up the Ubuntu virtual machine 1](#_Toc103782438)

[Part II: Setting up the ROS environment 2](#_Toc103782439)

[Part III: Setting up Jackal and Kinova gen-3 Lite Robots 2](#_Toc103782440)

[Part IV: Brief summary of how I went about solving problems with getting Kinova gen 3 lite working with Rviz 2](#_Toc103782441)

# Part I: Setting up the Ubuntu virtual machine

1. Assuming you are using windows…. download VirtualBox and Ubuntu 20.04 LTS
   1. It is likely preferable to just have FCS get you a ubuntu install over virtualization though. If you must virtualize over a different Linux distro, then QEMU is a much faster (but harder to setup) OS virtualization software. Skip to part II if any of these apply to you.
2. Right-click your Ubuntu disk, mount it
3. Install VirtualBox, create a new Virtual machine set to Ubuntu 64bit, 8Gb Memory, 20Gb Hard disk.
4. Start the machine, Select your mounted Ubuntu Disk
5. Select language
6. Minimal Installation, install graphics and wi-fi hardware, updates while installing ubuntu, “Erase disk and install ubuntu”
7. Make your computer name and password something simple, auto-login (its just a virtual machine)
8. Let ubuntu install, go through the restart, and finish your install.
9. Install updates, go to settings (bottom left-hand corner 9 dots to bring up apps, then search settings), raise your resolution to something comfortable under “Screen Display”.
10. When the software updater finishes, select restart later, close the Virtual Machine, select power down (this is needed for us to change settings).
11. In the Virtual Box Manager, click settings, go to display, give maybe half of your video memory, enable 3D Acceleration.
12. Go to System > Processor, give Plenty of CPUs and enable Nested VT-x/AMD-v
13. Go to Shared Folder, Create a new Shared folder, under Folder Path Select something like Documents > VBoxShared (folder I created). Check Auto Mount, and OK.
14. If you have one, plug in your Xbox or Playstation controller, go to VirtualBox Manager settings, USB, add USB filter (USB with a + icon), select your controller. When you have setup your controller, you will have to unplug and replug your controller to use it in the virtual machine.
15. Click Ok on your settings, start your virtual machine again.
16. Once logged-in, on the Virtual Machine’s toolbar, select Devices > Insert Guest CD Addon, accept everything.
17. Right-click the ubuntu desktop, click open in terminal.
18. Type ‘sudo adduser $USER vboxsf’ without the quotation marks. This sets up your shared folder (called something like “sf\_yourSharedFolderName”, this appears as a drive in Ubuntu’s file explorer.
19. Restart again for the changes to apply.
20. Under devices, you can also enable drag n drop and clipboard sharing between your main desktop and the virtual machine.

# Part II: Setting up the ROS environment

1. Search Ubuntu for “Software and Updates”
2. Enable Source code. Close the Software & Updates application. Click Reload.
3. All the following steps take place within ubuntu. To work with the Jackal and kinova robots, we will be using the latest version of ROS1 (noetic).
4. Drag and drop in the provided ROS-installer.sh and workspace-setup.sh scripts
5. Open a terminal
6. Chmod u+x ROS-installer.sh, chmod u+x workspace-setup.sh
7. Bash ROS-installer.sh
8. Yes to everything
9. Bash workspace-setup.sh

If the script doesn’t work, take a look at in a text editor, it will have some links to websites where much of this was obtained, it may be a case of not being in the right folder or not having sourced the right script.

# Part III: Setting up Jackal and Kinova gen-3 Lite Robots

1. Bash robot-setup.sh

# Part IV: Running

1. Roscore
2. New terminal

# Part V: Brief summary of how I went about solving problems with getting Kinova gen 3 lite working with Rviz

Problem 1: Warning, ….

Fixed by making sure gazebo is playing, such that the robot state is being sent

Problem 2: Fixed frame map doesn’t exist

Bad Solution:

Fixing Fixed Frame map can be done with ‘rosrun tf static\_transform\_publisher 0.0 0.0 0.0 0.0 0.0 0.0 map my\_frame 100’

Good Solution:

See error by creating axis, setting it to follow base\_link.: For frame [arm\_link]: No transform to fixed frame [map]. TF error: [Could not find a connection between 'map' and 'arm\_link' because they are not part of the same tree.Tf has two or more unconnected trees.]

Fixed: sudo apt install ros-noetic-tf2-tools then rosrun tf2\_tools view\_frames.py to see the frames tree, this led me to use this: rosrun tf static\_transform\_publisher 0.0 0.0 0.0 0.0 0.0 0.0 map world 100

Now all the transforms are linked to real world data, but still no arm model.

Now checking the text output of ‘roslaunch kortex\_gazebo spawn\_kortex\_robot.launch arm:=gen3\_lite start\_rviz:=false’

We see the following errors:

* /home/x/catkin\_ws/devel/lib/kortex\_driver/kortex\_arm\_driver: error while loading shared libraries: libfcl.so.0.6: cannot open shared object file: No such file or directory
* /opt/ros/noetic/lib/moveit\_ros\_move\_group/move\_group: error while loading shared libraries: libfcl.so.0.6: cannot open shared object file: No such file or directory
* log file: /home/x/.ros/log/a2b11d52-d6bd-11ec-a324-b7b20e0b097b/my\_gen3\_lite-urdf\_spawner-4\*.log  
  Traceback (most recent call last):  
  File "/home/x/catkin\_ws/src/ros\_kortex/kortex\_gazebo/scripts/home\_robot.py", line 7, in <module>  
  from kortex\_driver.msg import ActionNotification, ActionEvent  
  ModuleNotFoundError: No module named 'kortex\_driver'

So it seems kortex is broken, first course of action is trying to fix that libfcl.so.0.6: cannot open shared object file: no such file error

Tried <https://github.com/ros-planning/moveit/issues/786>

(run ‘export LC\_NUMERIC=en\_US.UTF-8’ in a terminal)

Then Roscore, roslaunch kortex\_gazebo …, error is gone

Now we only have the following warnings in the terminal once we run the static tf publish:

[ WARN] [1652964827.920610165]: Falling back to using the move\_group node's namespace (deprecated Melodic behavior).

[ WARN] [1652964830.319255622]: Could not identify parent group for end-effector 'end\_effector'

[ WARN] [1652964832.975486858]: Could not identify parent group for end-effector 'end\_effector'

[ WARN] [1652964833.136471996]: Resolution not specified for Octomap. Assuming resolution = 0.1 instead

At the very end it says “process has finished cleanly”

But still no arm in rviz??

WO529783-0

Follow the export master URI and such steps carefully if you want jackal\_viz to work

Roswtf is your friend (it finds node publishing errors)

I am attempting to get the Rviz simulation of ROS\_Kortex's gen3\_lite arm working. I have an issue where Gazebo can spawn the model fine, but no model appears in Rviz. There are a few other potential issues which may be related:

- There is an error of unknown frame [map] which can be solved by publishing to map using static\_transform\_publisher or by changing the fixed frame.

- There is no my\_gen3\_lite/robot\_description available.

>>>> apply the robot\_state >> robot\_state\_publisher change as seen in latest commit on github

Be sure to add a robot\_model and set the fixed\_frame to base\_link in rviz to see the robot.

For local simulation, use ros\_master\_uri=http://localhost:11311/

For the robot use <http://CPR-J100-0574:11311>

More things:

sudo apt install git-all

get pycharm (or some other python editor)

when doing git, use this to store your PAT: git config --global credential.helper store then do a push