Index document

What do you need to download?

I would assume you will have the first three downloaded anyway. My recommendation would be download ROS Noetic as this will be supported until May 2025 whereas ROS Melodic will not be supported from May 2023 onwards. When I upload my code to Github it will contain the 4,5 and 6 located in the folders.

1. ROS

2. Python

3. Arduino

4. The Robotiq gripper packages to be able to control its movement

If you are using ros noetic these packages can be downloaded from

[GitHub - TAMS-Group/robotiq: Robotiq package that maintained by community](https://github.com/TAMS-Group/robotiq)

If you are using ros kinetic or older these packages can be downloaded from

<https://github.com/ros-industrial/robotiq>

5. rosserial Arduino

If you are using rosserial noetic then download it from the link below

[GitHub - Aranyaa-k/Rosserial\_noetic: Set up rosserial noetic for Arduino](https://github.com/Aranyaa-k/rosserial_noetic)

if you are using any later versions then download it from this link instead.

[http://wiki.ros.org/rosserial\_arduino/Tutorials/Arduino%20IDE%20Setup](http://wiki.ros.org/rosserial_arduino/Tutorials/Arduino IDE Setup)

6. rosbag\_to\_csv

rosbag\_to\_csv was downloaded to convert my rosbag files to csv’s by which I could then use these csv’s in excel to generate my graphs. I downloaded the rosbag\_to\_csv folder from:

[GitHub - AtsushiSakai/rosbag\_to\_csv: Converter from ros bag to csv](https://github.com/AtsushiSakai/rosbag_to_csv)

In the zip file I have submitted there is a lot of files. In this document I will tell you how to access the relevant files.

The **Arduino** folder:

Arduino -> FSR\_reading -> FSR\_reading.ino

Upload the file FSR\_reading.ino to the Arduino board to get readings from the FSRs.

The **bagfiles** folder:

If you click on the bagfiles folder it will show you all the bagfiles that I recorded when the gripper was in motion.

The **boxplots\_for\_FSRs** folder:

The boxplots for FSR1, FSR2 and FSR3 are in this file as well as the mean, median, Q1, Q3 and IQR for the boxplots.

The **catkin\_ws** folder:

This is where the code that controlled my gripper is located.

The code which was used when the feedback of the gripper was obtained at a speed of 25 is found by the following path:

catkin\_ws -> src -> robotiq -> robotiq\_3f\_gripper\_control -> nodes -> repeating\_speed\_25.py

The code which was used when the feedback of the gripper was obtained at a speed of 150 is found by the following path:

catkin\_ws -> src -> robotiq -> robotiq\_3f\_gripper\_control -> nodes -> repeating\_speed\_150.py

For my slip detection algorithm, the code is found by following the path:

catkin\_ws -> src -> robotiq -> robotiq\_3f\_gripper\_control -> nodes -> Robotiqcontroltester.py

At the following path:

catkin\_ws -> src -> rosbag\_to\_csv

rosbag\_to\_csv was downloaded to convert my rosbag files to csv’s by which I could then use these csv’s in excel to generate my graphs.

The **Feedback\_with\_gripper** folder:

The excel graphs which were made from the feedback data from the gripper are found in this folder. The respective graphs at a speed of 25 and 150 are found in their respective folders. The parameters in these graphs are the current and position of the gripper’s fingers.

The **force\_readings\_of\_objects** folder:

This folder shows the variation of the force on the FSRs as well as the position and current of the robotic grippers fingers when the slip detection algorithm is used.

The **FSR\_calibration\_readings** folder:

In this folder are the excel graphs that were used in the calibration of FSR Sensor 1, 2 and 3

**All the graphs** shown in the **boxplots\_for\_FSRs**, **Feedback\_with\_gripper**, **force\_readings\_of\_objects** and **FSR\_calibration\_readings** folders were included in my dissertation write up, either in the main section or the appendix with the axes labelled correctly.

Commands to type into your linux terminal once everything has been downloaded

For each command I opened a separate terminal. However, when looking back this could have been done instead by making a roslaunch file. The command I typed in each terminal were as follows:

1.      roscore

2.      rosrun robotiq\_3f\_gripper\_control Robotiq3FGripperTcpNode.py 10.0.1.12

This is the driver node. The gripper is driven by the node Robotiq3FGripperTcpNode.py contained in the package ‘robotiq\_3f\_gripper\_control’ and the IP address of the gripper is provided as an argument.

3.      rosrun rosserial\_python serial\_node.py \_port:=/dev/ttyACM0 \_baud:=9600

ROS establishes communication with Arduino. Note you need to upload the Arduino code to the Arduino before these 6 commands are typed into the Linux terminal.

4.      rosrun robotiq\_3f\_gripper\_control Robotiqcontroltester.py

The driver node listens (subscribes) for messages on ‘Robotiq3FGripperRobotOutput’ using the ‘Robotiq3FGripperRobotOutput’ msg type. The messages are interpreted, and commands are sent to the gripper. The controller node publishes messages on ‘Robotiq3FGripperRobotOutput’ using the ‘Robotiq3FGripperRobotOutput’ msg type.

5.      rosbag record /force3 /Robotiq3FGripperRobotInput

Once you have finished recording whatever topic you decide, in 5., the topic I was recording was /force3, press CTRL + C. This command will stop recording the topic. Now you can convert the rosbag to csv which is one by typing in the following command:

6.      rosrun rosbag\_to\_csv rosbag\_to\_csv.py

Follow the instructions that appear on screen. The rosbag\_to\_csv video I used to help me is found online here: <https://www.youtube.com/watch?v=AnaLC0A4g7E&t=39s>