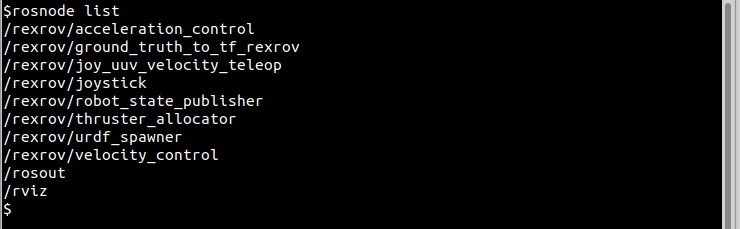
**Fall 2021 RIS Lab I project sheet**

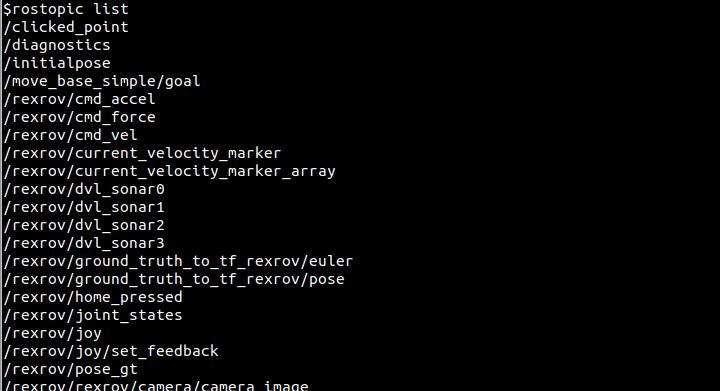
1. Launched the uuv\_gazebo/rexrov\_default.launch and inspected the nodes, topics, services it launches and the message/service types they use to communicate.

a. Short report describing these nodes, topics, services and messages including screenshots of rqt or the terminal output from which you collected this information.



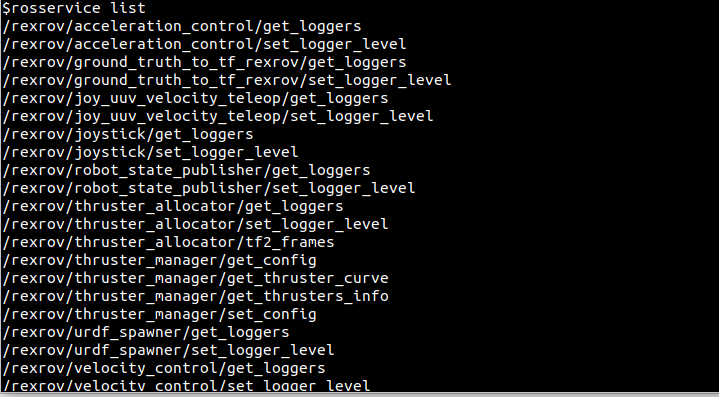
**List of Nodes:-**

* /rexrov/acceleration\_control
* /rexrov/ground\_truth\_to\_tf\_rexrov
* /rexrov/joy\_uuv\_velocity\_teleop
* /rexrov/joystick
* /rexrov/robot\_state\_publisher
* /rexrov/thruster\_allocator
* /rexrov/urdf\_spawner
* /rexrov/velocity\_control
* /rosout
* /rviz



**List of Topics:-**

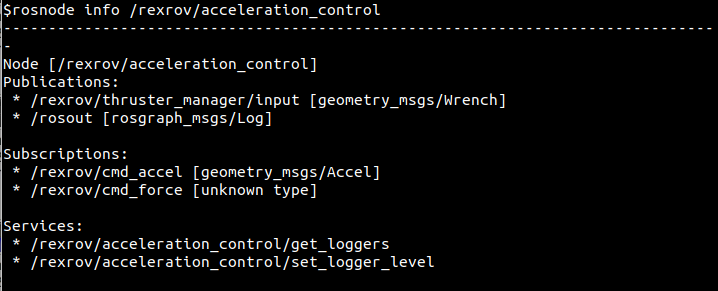
* /clicked\_point (geometry\_msgs/PointStamped)
* /diagnostics (diagnostic\_msgs/DiagnosticArray)
* /initialpose (geometry\_msgs/PoseWithCovarianceStamped)
* /move\_base\_simple/goal (geometry\_msgs/PoseStamped)
* /rexrov/cmd\_accel (geometry\_msgs/Accel)
* /rexrov/cmd\_force (geometry\_msgs/Accel)
* /rexrov/cmd\_vel (geometry\_msgs/Twist)
* /rexrov/current\_velocity\_marker (visualization\_msgs/Marker)
* /rexrov/current\_velocity\_marker\_array (visualization\_msgs/MarkerArray)
* /rexrov/dvl\_sonar0 (sensor\_msgs/Range)
* /rexrov/dvl\_sonar1 (sensor\_msgs/Range)
* /rexrov/dvl\_sonar2 (sensor\_msgs/Range)
* /rexrov/dvl\_sonar3 (sensor\_msgs/Range)
* /rexrov/ground\_truth\_to\_tf\_rexrov/euler (geometry\_msgs/Vector3Stamped)
* /rexrov/ground\_truth\_to\_tf\_rexrov/pose (geometry\_msgs/PoseStamped)
* /rexrov/home\_pressed (std\_msgs/Bool)
* /rexrov/joint\_states (sensor\_msgs/JointState)
* /rexrov/joy (sensor\_msgs/Joy)
* /rexrov/joy/set\_feedback (sensor\_msgs/JoyFeedbackArray)
* /rexrov/pose\_gt (nav\_msgs/Odometry)
* /rexrov/rexrov/camera/camera\_image (sensor\_msgs/Image)
* /rexrov/thruster\_manager/input (geometry\_msgs/Wrench)
* /rexrov/thruster\_manager/input\_stamped (geometry\_msgs/WrenchStamped)
* /rexrov/thrusters/0/input (uuv\_gazebo\_ros\_plugins\_msgs/FloatStamped)
* /rexrov/thrusters/1/input (uuv\_gazebo\_ros\_plugins\_msgs/FloatStamped)
* /rexrov/thrusters/2/input (uuv\_gazebo\_ros\_plugins\_msgs/FloatStamped)
* /rexrov/thrusters/3/input (uuv\_gazebo\_ros\_plugins\_msgs/FloatStamped)
* /rexrov/thrusters/4/input (uuv\_gazebo\_ros\_plugins\_msgs/FloatStamped)
* /rexrov/thrusters/5/input (uuv\_gazebo\_ros\_plugins\_msgs/FloatStamped)
* /rexrov/thrusters/6/input (uuv\_gazebo\_ros\_plugins\_msgs/FloatStamped)
* /rexrov/thrusters/7/input (uuv\_gazebo\_ros\_plugins\_msgs/FloatStamped)
* /rexrov/velocity\_control/parameter\_descriptions (dynamic\_reconfigure/ConfigDescription)
* /rexrov/velocity\_control/parameter\_updates (dynamic\_reconfigure/Config)
* /rosout (rosgraph\_msgs/Log)
* /rosout\_agg (rosgraph\_msgs/Log)
* /tf (tf2\_msgs/TFMessage)
* /tf\_static (tf2\_msgs/TFMessage)



**List of Services:-**

* /rexrov/acceleration\_control/get\_loggers (roscpp/GetLoggers)
* /rexrov/acceleration\_control/set\_logger\_level (roscpp/SetLoggerLevel)
* /rexrov/ground\_truth\_to\_tf\_rexrov/get\_loggers (roscpp/GetLoggers)
* /rexrov/ground\_truth\_to\_tf\_rexrov/set\_logger\_level (roscpp/SetLoggerLevel)
* /rexrov/joy\_uuv\_velocity\_teleop/get\_loggers (roscpp/GetLoggers)
* /rexrov/joy\_uuv\_velocity\_teleop/set\_logger\_level (roscpp/SetLoggerLevel)
* /rexrov/joystick/get\_loggers (roscpp/GetLoggers)
* /rexrov/joystick/set\_logger\_level (roscpp/SetLoggerLevel)
* /rexrov/robot\_state\_publisher/get\_loggers (roscpp/GetLoggers)
* /rexrov/robot\_state\_publisher/set\_logger\_level (roscpp/SetLoggerLevel)
* /rexrov/thruster\_allocator/get\_loggers (roscpp/GetLoggers)
* /rexrov/thruster\_allocator/set\_logger\_level (roscpp/SetLoggerLevel)
* /rexrov/thruster\_allocator/tf2\_frames (tf2\_msgs/FrameGraph)
* /rexrov/thruster\_manager/get\_config (uuv\_thruster\_manager/GetThrusterManagerConfig)
* /rexrov/thruster\_manager/get\_thruster\_curve (uuv\_thruster\_manager/GetThrusterCurve)
* /rexrov/thruster\_manager/get\_thrusters\_info (uuv\_thruster\_manager/ThrusterManagerInfo)
* /rexrov/thruster\_manager/set\_config (uuv\_thruster\_manager/SetThrusterManagerConfig)
* /rexrov/urdf\_spawner/get\_loggers (roscpp/GetLoggers)
* /rexrov/urdf\_spawner/set\_logger\_level (roscpp/SetLoggerLevel)
* /rexrov/velocity\_control/get\_loggers (roscpp/GetLoggers)
* /rexrov/velocity\_control/set\_logger\_level (roscpp/SetLoggerLevel)
* /rexrov/velocity\_control/set\_parameters (dynamic\_reconfigure/Reconfigure)
* /rosout/get\_loggers (roscpp/GetLoggers)
* /rosout/set\_logger\_level (roscpp/SetLoggerLevel)
* /rviz/get\_loggers (roscpp/GetLoggers)
* /rviz/load\_config (rviz/SendFilePath)
* /rviz/reload\_shaders (/rviz/reload\_shaders)
* /rviz/save\_config (rviz/SendFilePath)
* /rviz/set\_logger\_level (roscpp/SetLoggerLevel)

**Determine the message/service type of topics/services**



**b. Commands to retrieve this information and write down what these commands do**

**rosnode list**

**rostopic list**

**rosservice list**

**rosnode info <node\_name>**

**rostopic info <topic\_name>**

**rosservice info <service\_name>**

**2. Controlling the robot using teleop\_twist\_keyboard**

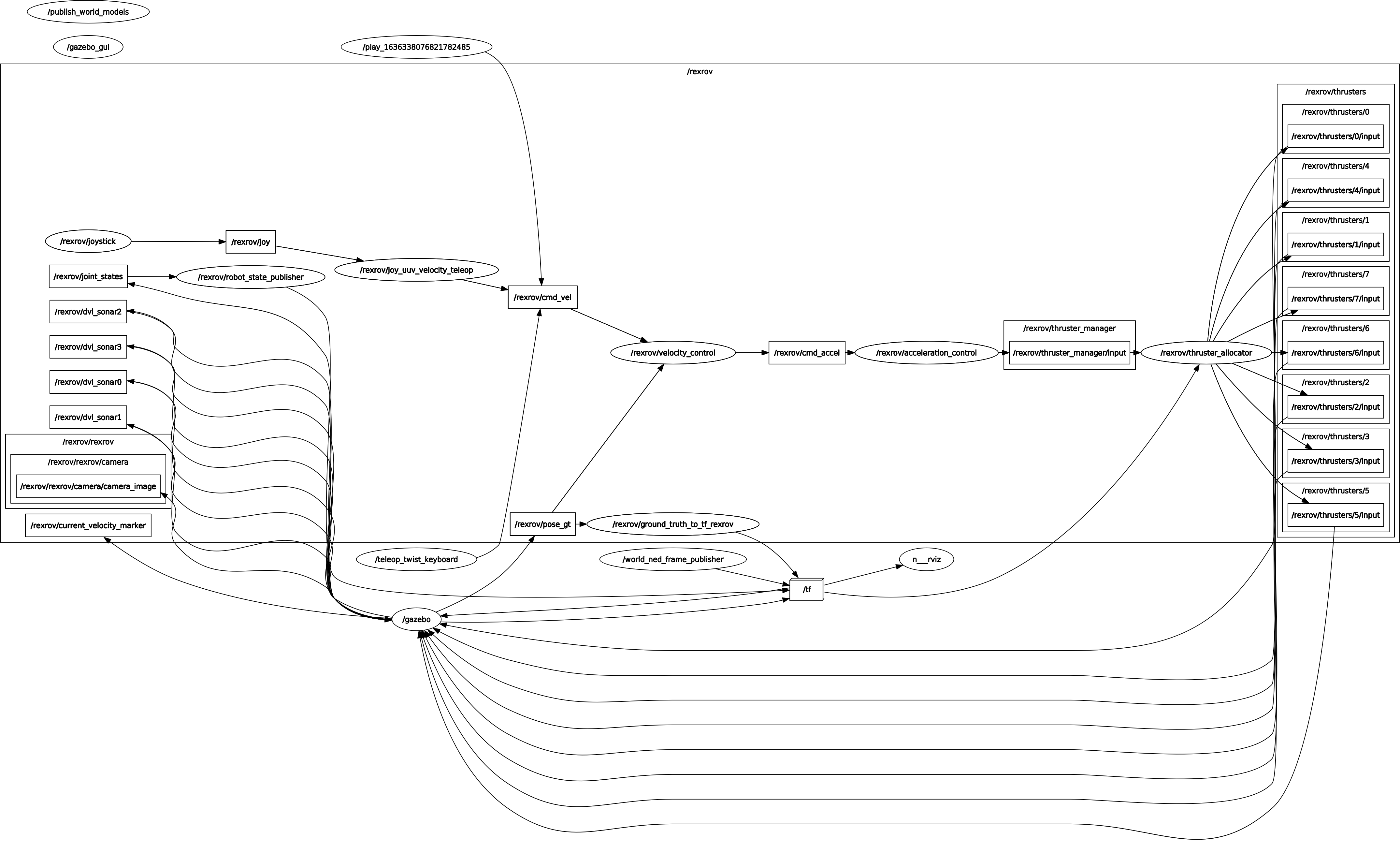
a. launch file

uuv\_gazebo/launch/rexrov\_underwater\_world.launch contains the rexrov\_default.launch, empty\_underwater\_world and teleop twist node to control the simulated ROV. Also added the comments.

b. rosbag file

bags/cmd\_vel\_rosbag.bag contains the cmd\_vel topic data, when the ROV was moving around. Also played the rosbag and it moved the ROV.

RQT Graph



**3. Forces and Torque Output**

Downloaded the rexrov\_wrench\_control.launch file

a. created the node similar to teleop twist keyboard

uuv\_simulator/uuv\_control/uuv\_thruster\_manager/scripts/wrench\_control.py

b. uuv\_gazebo/rexrov\_wrench\_control.launch is the launch file which launches AUV and my node.

**4. Custom URDF**

a. my\_rov/urdf/my\_rov.urdf created a custom rov urdf

b. the thruster positioning at the back left and right to move the robot forward/backward/left/right etc.

c. my\_rov/scripts/thruster\_control.py is the node that takes the forces and torques as input and publishes thrust commands for the ROV and uses the service for this conversion.

Created a launch file my\_rov/launch/thruster\_control.launch which executes the thurster\_control node

d.

I. my\_rov/launch/my\_rov\_underwater\_world.launch is the launch file which load the robot into the empty\_underwater\_world.launch in gazebo

ii. uuv\_simulator/uuv\_control/uuv\_thruster\_manager/scripts/wrench\_control.py started the exercise 3 node to publish force/torque data

iii. my\_rov/launch/thruster\_control.launch is used to execute my\_rov/scripts/thruster\_control.py node to publish the commands to the vehicle.

Combined all the required nodes in the launch file my\_rov/launch/my\_rov\_underwater\_world.launch