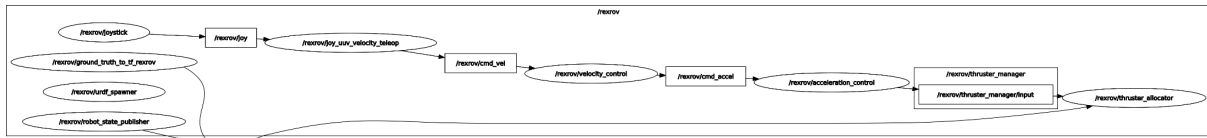


Table of Contents

Task 1	2
Rqt_graph	2
Nodes	2
/rexrov/acceleration_control	3
/rexrov/ground_truth_to_tf_rexrov	4
/rexrov/robot_state_publisher	4
/rexrov/joy_uuv_velocity_teleop	5
/rviz	5
/rosout	6
/rexrov/velocity_control	6
/rexrov/urdf_spawner	7
/rexrov/joystick	7
/rexrov/thruster_allocator	7
Task 2	8
Part A	8
Part B	8
Task 3	9
Part A	9
Part B	9
Task 4	9
Part B	9
Point Division	10

Task 1

Rqt_graph



In order to retrieve information about the nodes, topics, services, and messages from “[uuv_gazebo/rexrov_default.launch](#)”, I had to first launch the file by using the following command:
`roslaunch uuv_gazebo rexrov_default.launch.`

By using the command `roslaunch rqt_graph rqt_graph` we are able to have a graph visualization of /rexrov and its current active nodes and topics.

Nodes

If we would like to get specific information about every node's topic we can use `rostopic list` to list the topics and `rostopic info [topic]`, and the same can be done for services, we would only need to change topic by service. However, I mostly used `roslaunch` `list` and `roslaunch info [node]` to better understand the graph.

By using `roslaunch list` I was able to get a list of all the nodes:

1. /rexrov/acceleration_control
2. /rexrov/ground_truth_to_tf_rexrov
3. /rexrov/joy_uuv_velocity_teleop
4. /rexrov/joystick
5. /rexrov/robot_state_publisher
6. /rexrov/thruster_allocator
7. /rexrov/urdf_spawner
8. /rexrov/velocity_control
9. /rosout
10. /rviz

/rexrov/acceleration_control

```
vsancnaj@aguacateubuntu:~/catkin_ws/src/uuv_simulator$ rosnode info /rexrov/acceleration_control
-----
Node [/rexrov/acceleration_control]
Publications:
* /rexrov/thruster_manager/input [geometry_msgs/Wrench]
* /rosout [roscpp_msgs/Log]

Subscriptions:
* /rexrov/cmd_accel [geometry_msgs/Accel]
* /rexrov/cmd_force [unknown type]

Services:
* /rexrov/acceleration_control/get_loggers
* /rexrov/acceleration_control/set_logger_level

contacting node http://aguacateubuntu:38491/ ...
Pid: 29630
Connections:
* topic: /rosout
  * to: /rosout
  * direction: outbound (38061 - 10.82.57.126:40206) [9]
  * transport: TCPROS
* topic: /rexrov/thruster_manager/input
  * to: /rexrov/thruster_allocator
  * direction: outbound (38061 - 10.82.57.126:40216) [16]
  * transport: TCPROS
* topic: /rexrov/cmd_accel
  * to: /rexrov/velocity_control (http://aguacateubuntu:33137/)
  * direction: inbound
  * transport: TCPROS
```

This node has two publishers of message type `geometry_msgs` and `roscpp_msgs`. A message defines the structure of the data that is passed between nodes. The `/rexrov/thruster_manager/input` topic is a publisher to this node and it is a subscriber to the active node `/rexrov/thruster_allocator`. Along with services `/rexrov/acceleration_control/get_loggers` and `/rexrov/acceleration_control/set_logger_level` of type `roscpp/GetLoggers` and `roscpp/SetLoggerLevel`.

By using `rostopic info /rexrov/thruster_manager/input`, we are able to retrieve this information.

```
vsancnaj@aguacateubuntu:~/catkin_ws/src/uuv_simulator$ rostopic info /rexrov/thruster_manager/input
Type: geometry_msgs/Wrench

Publishers:
* /rexrov/acceleration_control (http://aguacateubuntu:45023/)

Subscribers:
* /rexrov/thruster_allocator (http://aguacateubuntu:45395/)
```

/rexrov/ground_truth_to_tf_rexrov

```
vsancnaj@aguacateubuntu:~/catkin_ws/src/uuv_simulator$ rosnodetool info /rexrov/ground_truth_to_tf_rexrov
-----
Node [/rexrov/ground_truth_to_tf_rexrov]
Publications:
* /rexrov/ground_truth_to_tf_rexrov/euler [geometry_msgs/Vector3Stamped]
* /rexrov/ground_truth_to_tf_rexrov/pose [geometry_msgs/PoseStamped]
* /rosout [roscpp_msgs/Log]
* /tf [tf2_msgs/TFMessage]

Subscriptions:
* /rexrov/pose_gt [unknown type]

Services:
* /rexrov/ground_truth_to_tf_rexrov/get_loggers
* /rexrov/ground_truth_to_tf_rexrov/set_logger_level
```

This node has four publishers of message type geometry_msgs, roscpp_msgs, and tf2_msgs. The /rexrov/ground_truth_to_tf_rexrov/euler topic is only a publisher to this node. The /rexrov/ground_truth_to_tf_rexrov/pose is only a publisher to this node. This node is subscribed to only one topic and has two services of get_loggers and set_logger_level.

/rexrov/robot_state_publisher

```
vsancnaj@aguacateubuntu:~/catkin_ws/src/uuv_simulator$ rosnodetool info /rexrov/robot_state_publisher
-----
Node [/rexrov/robot_state_publisher]
Publications:
* /rosout [roscpp_msgs/Log]
* /tf [tf2_msgs/TFMessage]
* /tf_static [tf2_msgs/TFMessage]

Subscriptions:
* /rexrov/joint_states [unknown type]

Services:
* /rexrov/robot_state_publisher/get_loggers
* /rexrov/robot_state_publisher/set_logger_level
```

This node has three publishers of message type roscpp_msgs and tf2_msgs. The /tf topic has a publisher to this node and to /rexrov/robot_state_publisher. The /tf_static is only a publisher to this node. This node is subscribed to only one topic and has two services of get_loggers and set_logger_level.

/rexrov/joy_uuv_velocity_teleop

```
vsancnaj@aguacateubuntu:~/catkin_ws/src/uuv_simulator$ rosnodetool info /rexrov/joy_uuv_velocity_teleop
-----
Node [/rexrov/joy_uuv_velocity_teleop]
Publications:
* /rexrov/cmd_vel [geometry_msgs/Twist]
* /rexrov/home_pressed [std_msgs/Bool]
* /rosout [roscpp_msgs/Log]

Subscriptions:
* /rexrov/joy [sensor_msgs/Joy]

Services:
* /rexrov/joy_uuv_velocity_teleop/get_loggers
* /rexrov/joy_uuv_velocity_teleop/set_logger_level
```

This node has three publishers of message type geometry_msgs, std_msgs, and roscpp_msgs. The /rexrov/cmd_vel topic is only a publisher to this node. The /rexrov/home_pressed is only a publisher to this node. This node is subscribed to only one topic and has two services of get_loggers and set_loggers_level.

/rviz

```
vsancnaj@aguacateubuntu:~/catkin_ws/src/uuv_simulator$ rosnodetool info /rviz
-----
Node [/rviz]
Publications:
* /clicked_point [geometry_msgs/PointStamped]
* /initialpose [geometry_msgs/PoseWithCovarianceStamped]
* /move_base_simple/goal [geometry_msgs/PoseStamped]
* /rosout [roscpp_msgs/Log]

Subscriptions:
* /rexrov/current_velocity_marker [unknown type]
* /rexrov/current_velocity_marker_array [unknown type]
* /rexrov/dvl_sonar0 [unknown type]
* /rexrov/dvl_sonar1 [unknown type]
* /rexrov/dvl_sonar2 [unknown type]
* /rexrov/dvl_sonar3 [unknown type]
* /rexrov/pose_gt [unknown type]
* /rexrov/rexrov/camera/camera_image [unknown type]
* /tf [tf2_msgs/TFMessage]
* /tf_static [tf2_msgs/TFMessage]

Services:
* /rviz/get_loggers
* /rviz/load_config
* /rviz/reload_shaders
* /rviz/save_config
* /rviz/set_logger_level
```

This node has four publishers of message type geometry_msgs and roscpp_msgs. This node is subscribed to ten topics and has five services.

/rosout

```
vsancnaj@aguacateubuntu:~/catkin_ws/src/uuv_simulator$ rosnodetool info /rosout
-----
Node [/rosout]
Publications:
* /rosout_agg [rosgraph_msgs/Log]

Subscriptions:
* /rosout [rosgraph_msgs/Log]

Services:
* /rosout/get_loggers
* /rosout/set_logger_level
```

/rosout is the name of the console log reporting mechanism in ROS. It can be thought as comprising several components: The `rosout` node for subscribing, logging, and republishing the messages

/rexrov/velocity_control

```
vsancnaj@aguacateubuntu:~/catkin_ws/src/uuv_simulator$ rosnodetool info /rexrov/velocity_control
-----
Node [/rexrov/velocity_control]
Publications:
* /rexrov/cmd_accel [geometry_msgs/Accel]
* /rexrov/velocity_control/parameter_descriptions [dynamic_reconfigure/ConfigDescription]
* /rexrov/velocity_control/parameter_updates [dynamic_reconfigure/Config]
* /rosout [rosgraph_msgs/Log]

Subscriptions:
* /rexrov/cmd_vel [geometry_msgs/Twist]
* /rexrov/pose_gt [unknown type]

Services:
* /rexrov/velocity_control/get_loggers
* /rexrov/velocity_control/set_logger_level
* /rexrov/velocity_control/set_parameters
```

This node has four publishers of message type geometry_msgs, dynamic_reconfigure, and rosgraph_msgs. The /rexrov/cmd_accel, /rexrov/velocity_control/parameter_updates, and /rexrov/velocity_control/parameter_descriptions topics are only a publisher to this node. This node is subscribed to two topics and has three services of get_loggers and set_logger_level.

/rexrov/urdf_spawner

```
vsancnaj@aguacateubuntu:~/catkin_ws/src/uuv_simulator$ rosnode info /rexrov/urdf_spawner
-----
Node [/rexrov/urdf_spawner]
Publications:
* /rosout [rosgraph_msgs/Log]

Subscriptions: None

Services:
* /rexrov/urdf_spawner/get_loggers
* /rexrov/urdf_spawner/set_logger_level
```

This node has one publisher of message type `rosgraph_msgs`. This node is subscribed to no topics and has two services of `get_loggers` and `set_logger_level`.

/rexrov/joystick

```
vsancnaj@aguacateubuntu:~/catkin_ws/src/uuv_simulator$ rosnode info /rexrov/joystick
-----
Node [/rexrov/joystick]
Publications:
* /diagnostics [diagnostic_msgs/DiagnosticArray]
* /rexrov/joy [sensor_msgs/Joy]
* /rosout [rosgraph_msgs/Log]

Subscriptions:
* /rexrov/joy/set_feedback [unknown type]

Services:
* /rexrov/joystick/get_loggers
* /rexrov/joystick/set_logger_level
```

This node has three publishers of message type `diagnostic_msgs`, `sensor_msgs`, and `rosgraph_msgs`. The `/diagnostics` and `/rexrov/joy` topics are only a publisher to this node. This node is subscribed to only one topic and has two services of `get_loggers` and `set_logger_level`.

/rexrov/thruster_allocator

```
vsancnaj@aguacateubuntu:~/catkin_ws/src/uuv_simulator$ rosnode info /rexrov/thruster_allocator
-----
Node [/rexrov/thruster_allocator]
Publications:
* /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]
* /rosout [rosgraph_msgs/Log]

Subscriptions:
* /rexrov/thruster_manager/input [geometry_msgs/Wrench]
* /rexrov/thruster_manager/input_stamped [unknown type]
* /tf [tf2_msgs/TFMessage]
* /tf_static [tf2_msgs/TFMessage]

Services:
* /rexrov/thruster_allocator/get_loggers
* /rexrov/thruster_allocator/set_logger_level
* /rexrov/thruster_allocator/tf2_frames
* /rexrov/thruster_manager/get_config
* /rexrov/thruster_manager/get_thruster_curve
* /rexrov/thruster_manager/get_thrusters_info
* /rexrov/thruster_manager/set_config
```

This node has nine publishers of message type `uuv_gazebo_ros_plugins_msgs`. It subscribes to four topics and has seven services.

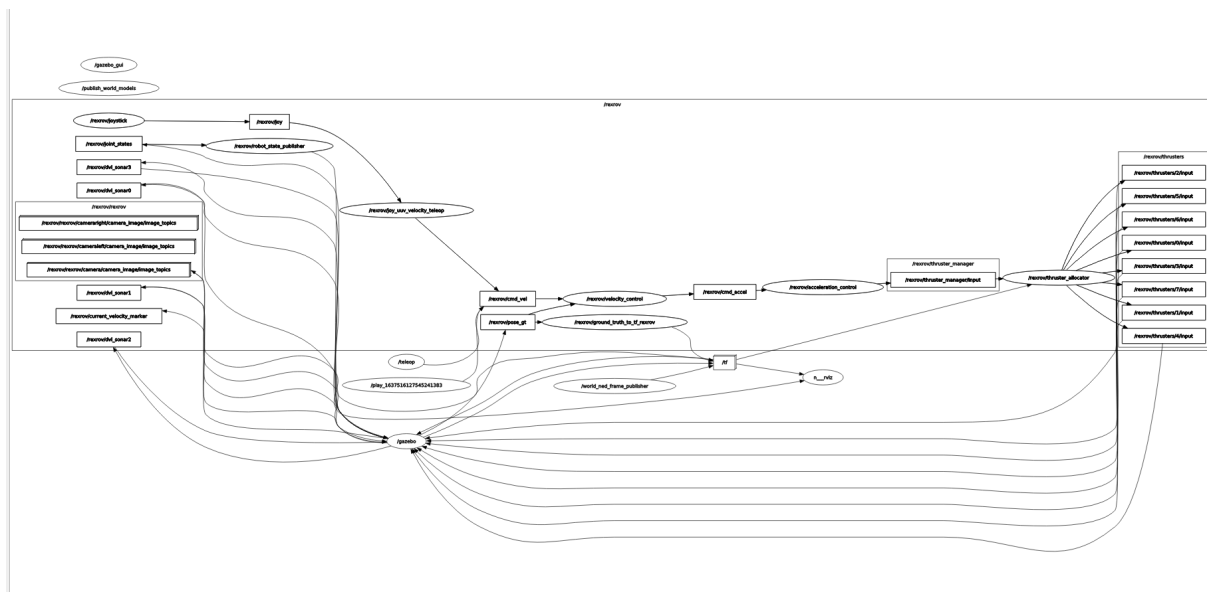
Task 2

Part A

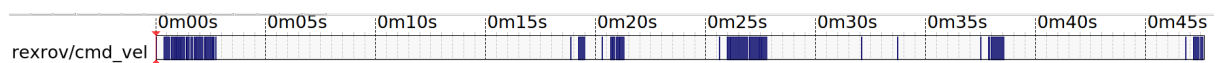
In order to see the robot launched in the gazebo world and controlled by the node `teleop_twist_keyboard`, you need to run the launch file under `uuv_gazebo/launch/rexrov_demos/task.launch`. Explanations for the functioning of the launch file are provided as comments.

Part B

Rqt_graph



Trajectory Plot



Task 3

Part A

To find the newly created node, similar to `teleop_twist_keyboard`, please look for the file called `teleop_wrench_keyboard.py` under `uuv_gazebo/src/teleop_wrench_keyboard.py`.

The file is similar to the previous node (`teleop_twist_keyboard`), but publishes a different type of messages to a different topic and contains some variables not used in the previous file.

Part B

To be able to launch the robot in the gazebo world, we had to create several files. The first file is a launch file that launches the node created in the previous part of this task. The file `launch_wrench_control.launch`, that can be found under `uuv_gazebo/launch/rexrov_demos` starts the node `teleop_wrench_keyboard`. The second file created, also found under the same directory and called `launch_rexrov_wrench`, is the main launch file that combines the control node, the gazebo world and the robot in one single launch file. Detailed explanation of the functioning of the launch files can be found as comments in the launch file itself.

Task 4

Part A

The ROV was designed using urdf and rviz and no other simulation software like blender was used.

First unzip the simulation folder and then go to workspace,
Then run `roslaunch simulation test_gazebo.launch`

The ROV has 1 control head and 5 thrusters added..

Part B

The completed URDF file has 5 thrusters.

1. One for vertical ascend and submerge function
2. Two for sideways movement left and right without tilt.
3. 3 for maximum thrust from behind and reverse movement.

This was decided for optimal mobility and easy use since the ROV in question was never intended for specialized cases.

Point Division

We divided the work by questions and for some parts of those questions

1. Valentina 25 pts
2. Yassine & Valentina
 - a. Yassine 10 pts & Valentina 5 pts
 - b. Valentina 5 pts
3. Yassine 20 pts
4. Yasar 35 pts

Name	Percentages
Valentina	35 %
Yassine	30 %
Yasar	35 %