EN4594 Autonomous Systems
Laboratory Sheet
Pre-Lab Session

Title: Robot Operating System (ROS) Basics

1. ROS Structure

An executable in ROS is called a node. ROS provides the framework for nodes to communicate with other nodes.

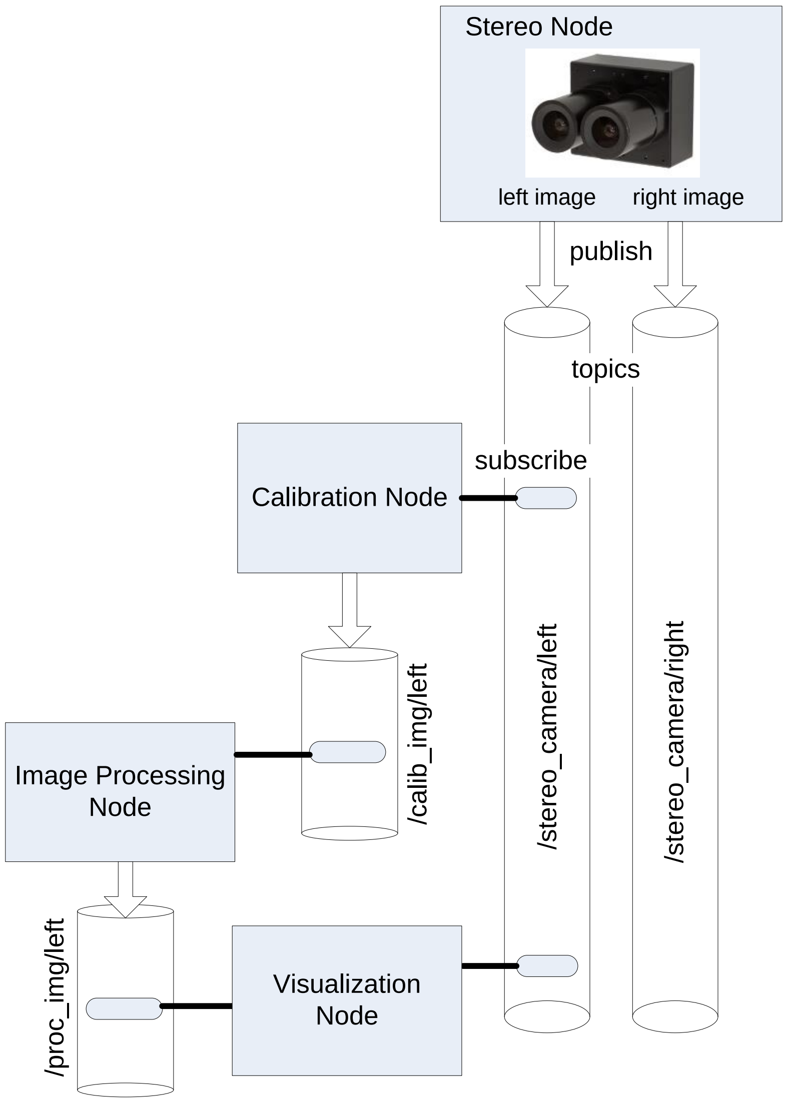


Fig. 1: ROS Structure

2. ROS Communication

ROS nodes can communicate with each other using *topics*, *services*, or *actions*, depending on the requirement, where the most common mode being *topics*.

2.1. ROS Topics

A ROS topic acts as a data channel to send messages to nodes. When a node outputs data into a topic, it is known as *publishing* of data. A node may publish data into multiple topics. Nodes can *subscribe* to a topic to obtain data published by a publisher node. This is also referred to as listening to a topic.

For example, a stereo camera node can publish the left camera image stream into one channel called `/stereo_camera/left` and publish the right camera image stream into another channel called `/stereo_camera/right`. A visualization node can subscribe to the left camera image to display it (Fig. 1).

When the ROS system is running, you can use `ros2 topic list` command in a new terminal to get details about all the topics that have been advertised in the system by various nodes. Moreover, to see the data being published on a topic, use:

```
ros2 topic echo <topic_name>
```

e.g.

```
ros2 topic echo /zumo/sensors
```

2.2. ROS Services

Nodes can also provide or use a service. Unlike topics, services are one-to-one data communication channels between two nodes.

3. Running ROS Nodes

You can run a ROS node using two methods.

3.1. Using `run`

```
ros2 run <package_name> <node_name> <parameters if any>
```

e.g.

```
ros2 run my_robot_pkg robot_node --ros-args -p max_speed:=1.0
```

Here, `--ros-args -p max_speed:=1.0` sets a parameter at runtime.

3.2. Using `launch`

```
ros2 launch <package_name> <launch_file>
```

e.g.

```
ros2 launch zumo_keyboard zumo_keyboard_no_calib.launch
```

The preferred method is the launch method. `ros2 launch` allows you to set a collection of parameters for the launching node. Furthermore, it can be used to launch multiple nodes using a single script.

4. Visualization

There are several tools available to visualize data in ROS, where `rqt` and `rviz2` are the most frequently used tools.

4.1. `rqt`

`rqt` provides a collection of plugins that enable users to visualize and interact with various aspects of a ROS system, such as displaying topics, visualizing TF (transform) data, plotting messages, and more.

To run `rqt`, open up a new terminal and type:

```
rqt
```

Using Plugins → Visualization → Plot you can visualize numeric values in a 2D plot by providing the topic name (Fig. 2).

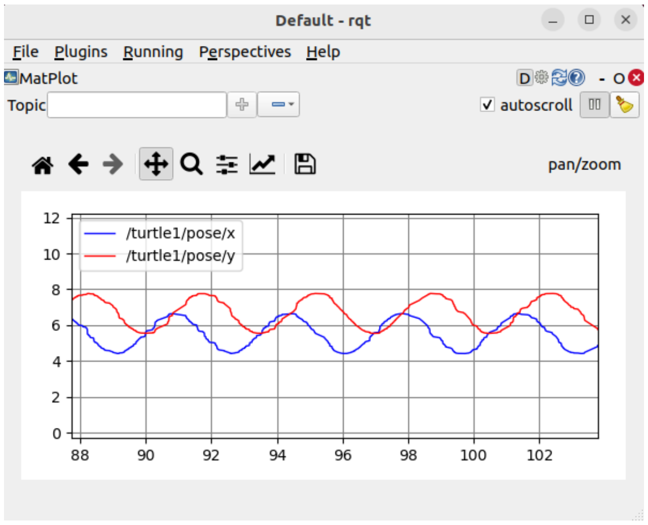


Fig. 2: An example plot in ROS `rqt`

To view all the nodes and their interconnections you can use Plugins → Introspection → Node Graph plugin (Fig. 3). The ellipses portray the nodes while the arrows portray the data flow with the topic names.

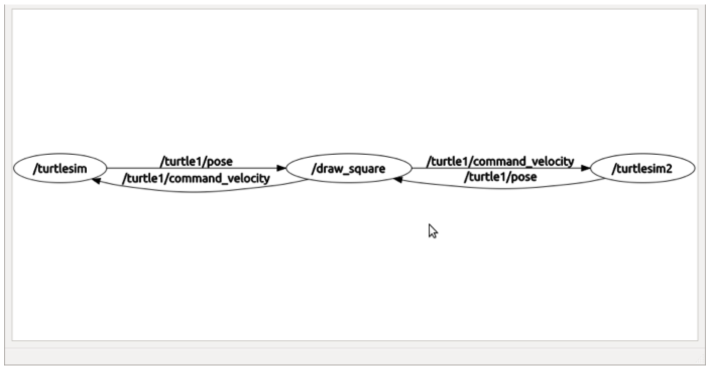
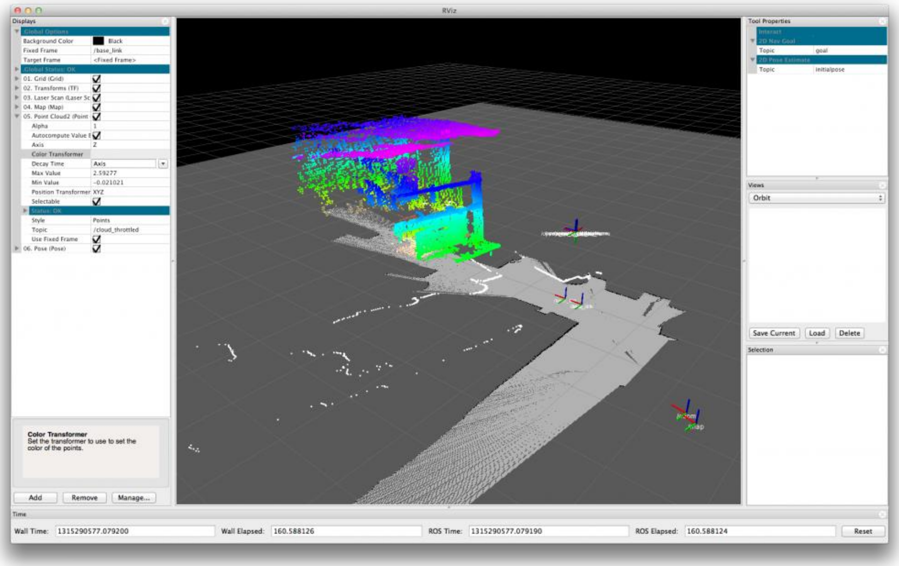


Fig. 3: A snapshot of the ROS nodes and their interconnections

4.2. `rviz2`

`rviz2` is the 3D visualization tool of ROS. You can add different types of displays into `rviz2` to visualize various kinds of data streams, including laser scans, point clouds, camera images and robot models and so on (Fig. 4).



Global Options	
Background Color	■ Black
Fixed Frame	/base_link
Target Frame	<Fixed Frame>
Global Status: OK	
01. Grid (Grid)	✓
02. Transforms (TF)	✓
03. Laser Scan (Laser Sc	✓
04. Map (Map)	✓
05. Point Cloud2 (Point	✓
Alpha	1
Autocompute Value E	✓
Axis	Z
Color Transformer	
Decay Time	Axis
Max Value	2.59277
Min Value	-0.021021
Position Transformer	XYZ
Selectable	✓
Status: OK	
Style	Points
Topic	/cloud_throttled
Use Fixed Frame	✓
06. Pose (Pose)	✓

Color Transformer
Set the transformer to use to set the color of the points.

Interact	
2D Nav Goal	
Topic	goal
2D Pose Estimate	
Topic	initialpose

Views: Orbit

Wall Time: 1315290577.079200 Wall Elapsed: 160.588126 ROS Time: 1315290577.079190 ROS Elapsed: 160.588124

Fig. 4: A snapshot of `rviz2` visualization

To run `rviz2`, open up a new terminal and type:

```
rviz2
```