SYMBIOSIS INSTITUTE OF TECHNOLOGY, PUNE
DEPARTMENT OF ROBOTICS AND AUTOMATION ENGINEERING

ACADEMIC YEAR: 202425 SEM: V

ASSIGNMENT NO: 6

NAME:Akshaj Chainani
PRN: 22070127008
BATCH: RA1

Aim:

ROS Basics: Understanding of nodes, topics, services, actions, parameters, rqt, recording and playing back data with trurtlesim.

Apparatus:

PC with ROS2 installed (Foxy, Galactic, or equivalent)
ROS2 environment set up with workspace
Turtlesim package installed
RQT package installed
Terminal access for executing ROS commands

Theory

1. Nodes

In ROS2, a node is a fundamental process that performs computation. Each node is designed to perform a specific task, such as publishing sensor data or controlling a robot. Nodes communicate with each other through topics, services, or actions.

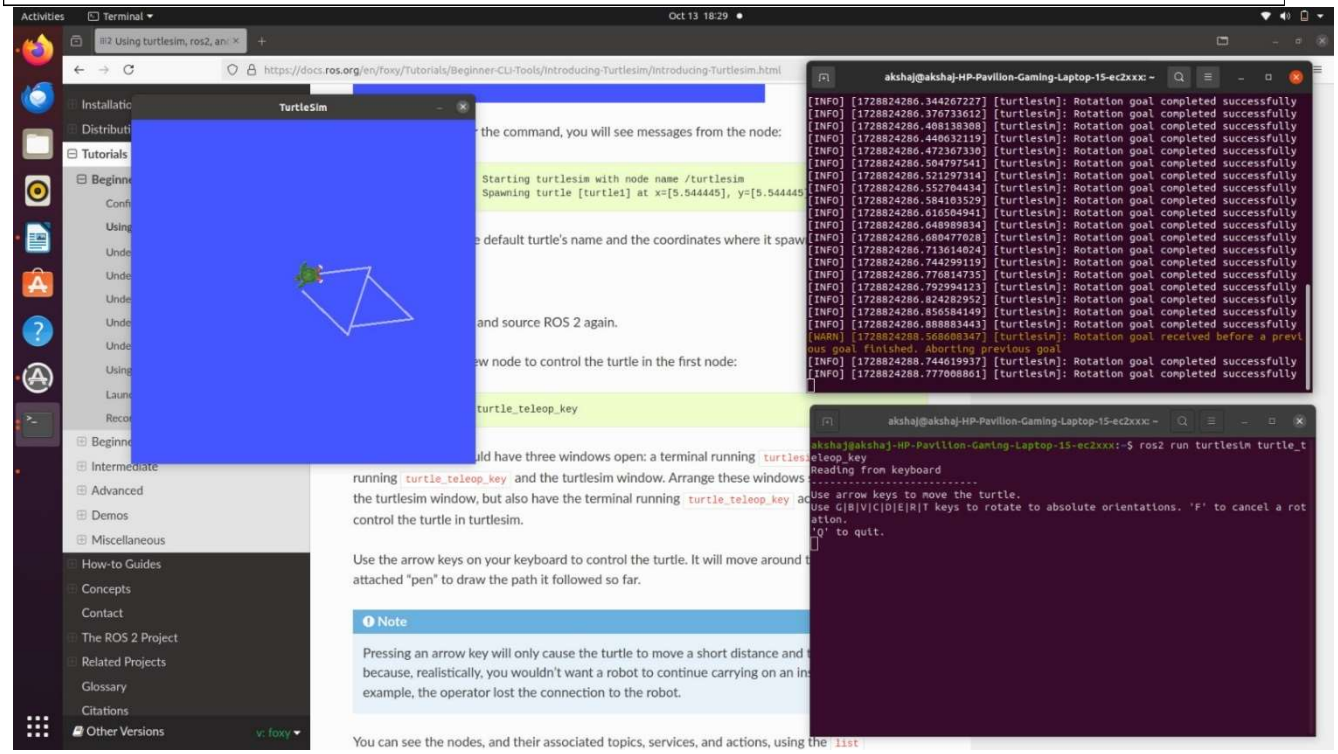
Node Management Commands:
To list active nodes:
bash
ros2 node list

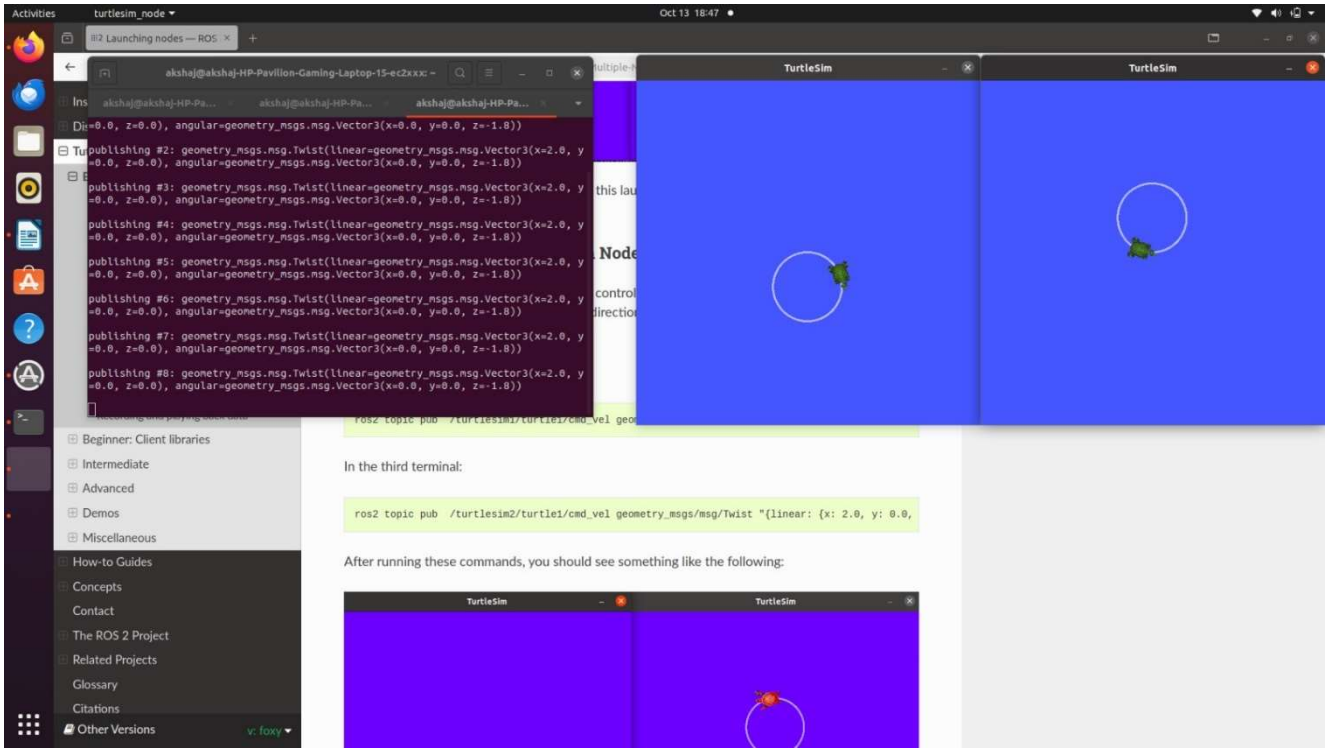
To get detailed information about a node:
bash
ros2 node info /<node_name>

SYMBIOSIS INSTITUTE OF TECHNOLOGY, PUNE

DEPARTMENT OF ROBOTICS AND AUTOMATION ENGINEERING

ACADEMIC YEAR: 202425 SEM: V





2. Topics

Topics provide a means for nodes to communicate asynchronously. One node can publish data to a topic, while others can subscribe to the topic to receive the data. Topics are used to implement the publishsubscribe model in ROS2.

Topic Management Commands:

To list all topics:

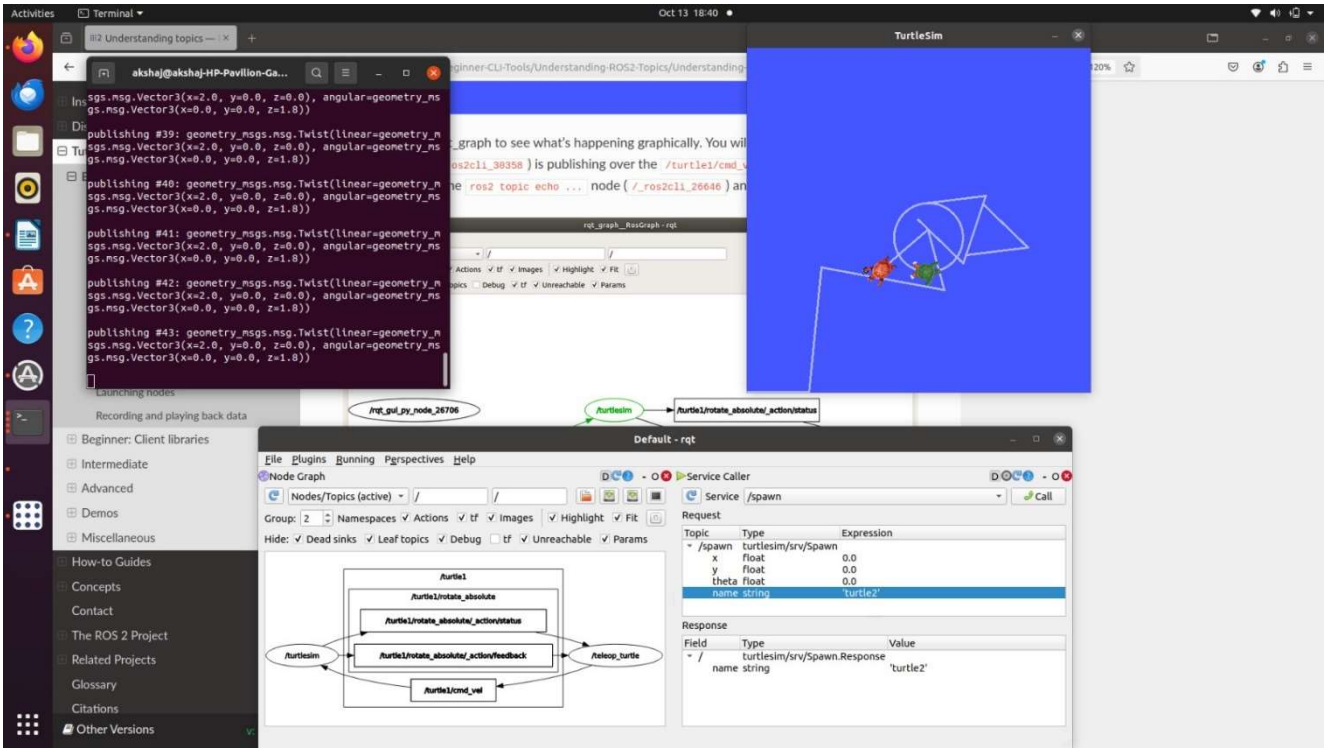
bash

ros2 topic list

To display messages published to a specific topic:

bash

ros2 topic echo <topic_name>



3. Services

Services in ROS2 follow a clientserver architecture. A client node sends a request to the server node, and the server processes the request and sends a response back. Services are ideal for tasks that need immediate responses and are not continuous.

Service Management Commands:

To list available services:

bash

ros2 service list

To call a specific service:

bash

ros2 service call <service_name> <service_type>

4. Actions

Actions are similar to services but are designed for longrunning tasks. They provide feedback to the client during execution and allow the client to cancel or preempt the goal if necessary.

Action Management Commands:
To list available actions:
bash
ros2 action list

To send a goal to an action server:
bash
ros2 action send_goal <action_name> <action_type> <values>

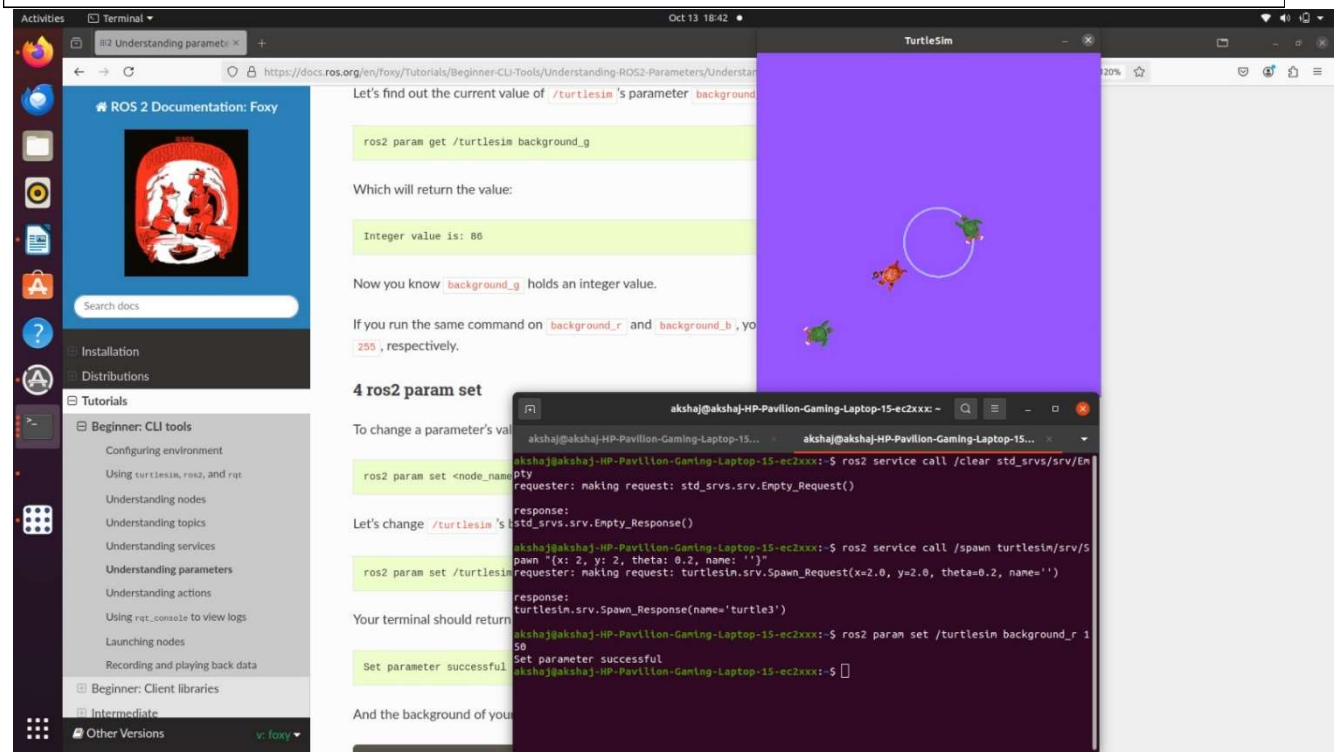
5. Parameters

Parameters are used to configure nodes dynamically. These parameters can be integers, strings, booleans, or lists, and they control the behavior of nodes without the need to restart them.

Parameter Management Commands:
To list parameters for a node:
bash
ros2 param list

To get the value of a parameter:
bash
ros2 param get <node_name> <param_name>

To set a parameter:
bash
ros2 param set <node_name> <param_name> <value>



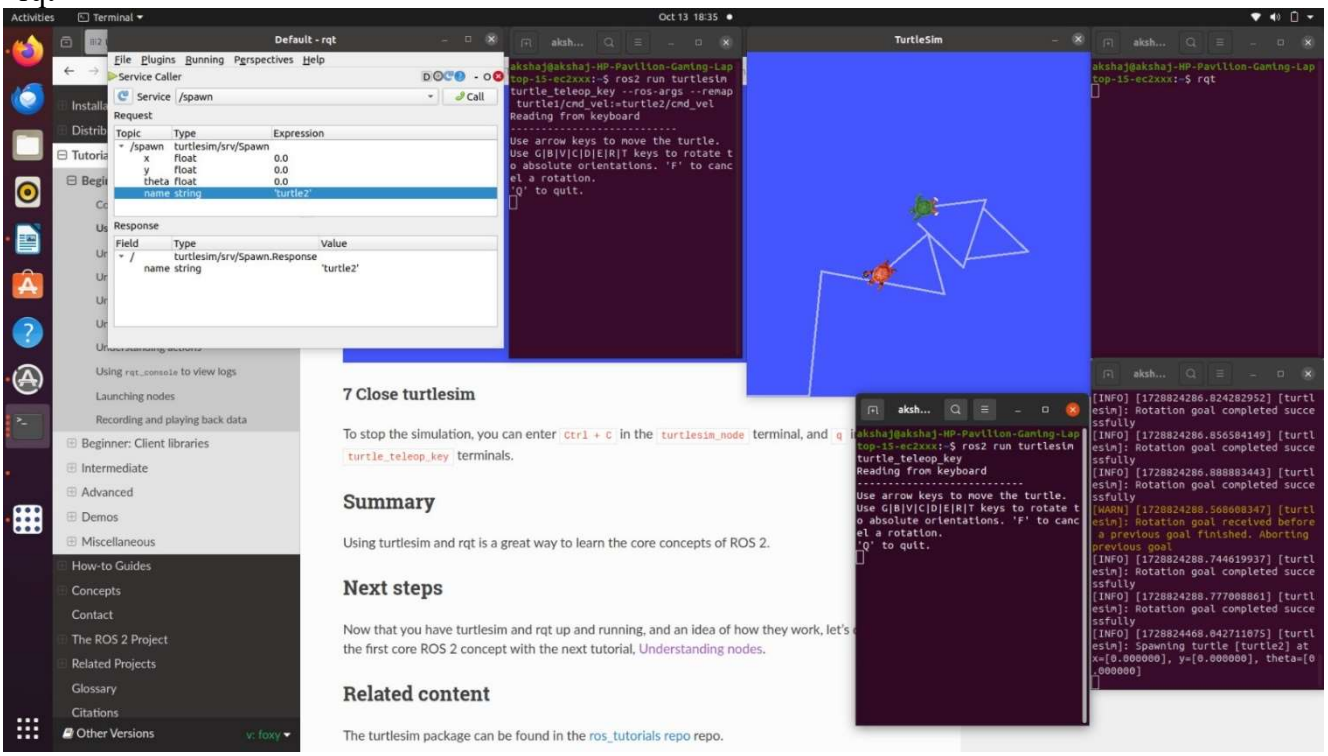
6. RQT

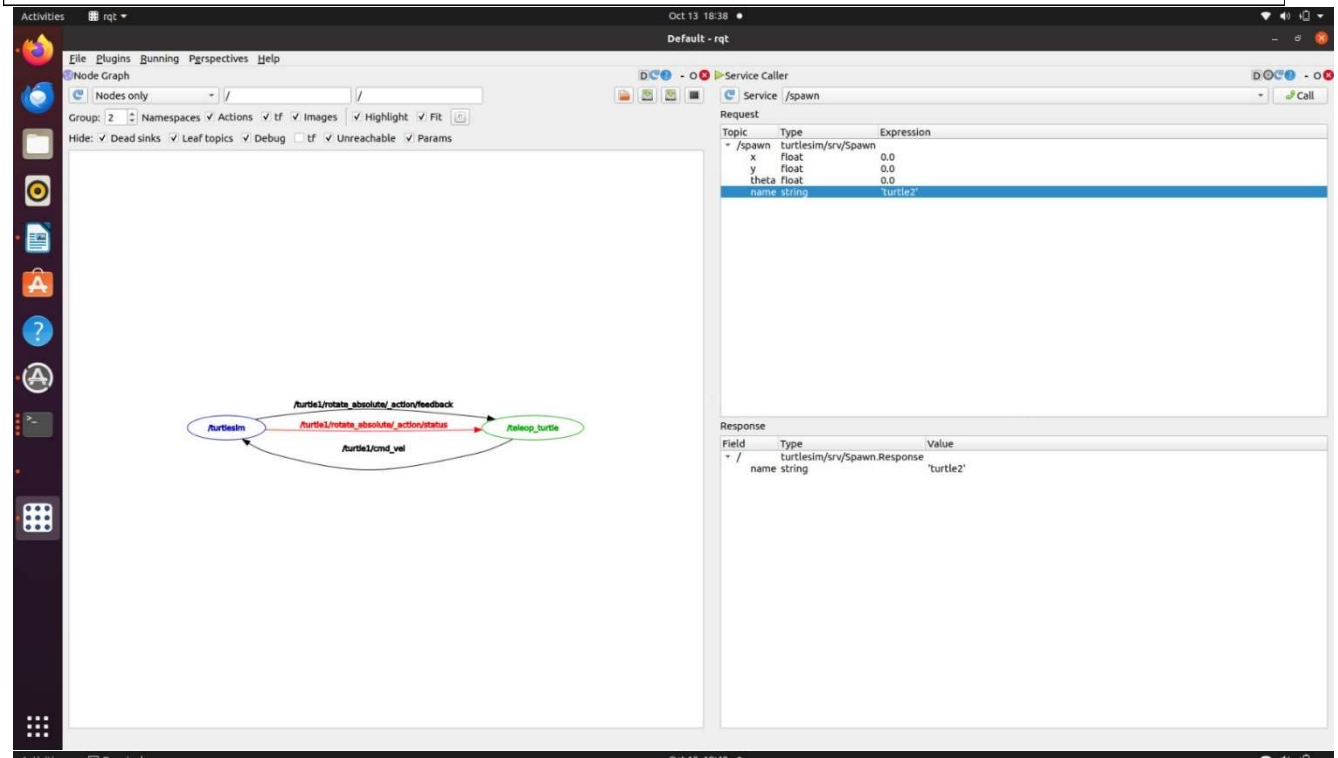
RQT is a graphical tool that provides a userfriendly interface for introspecting and visualizing ROS2 systems. It allows for the inspection of nodes, topics, services, and parameters in a graphical format.

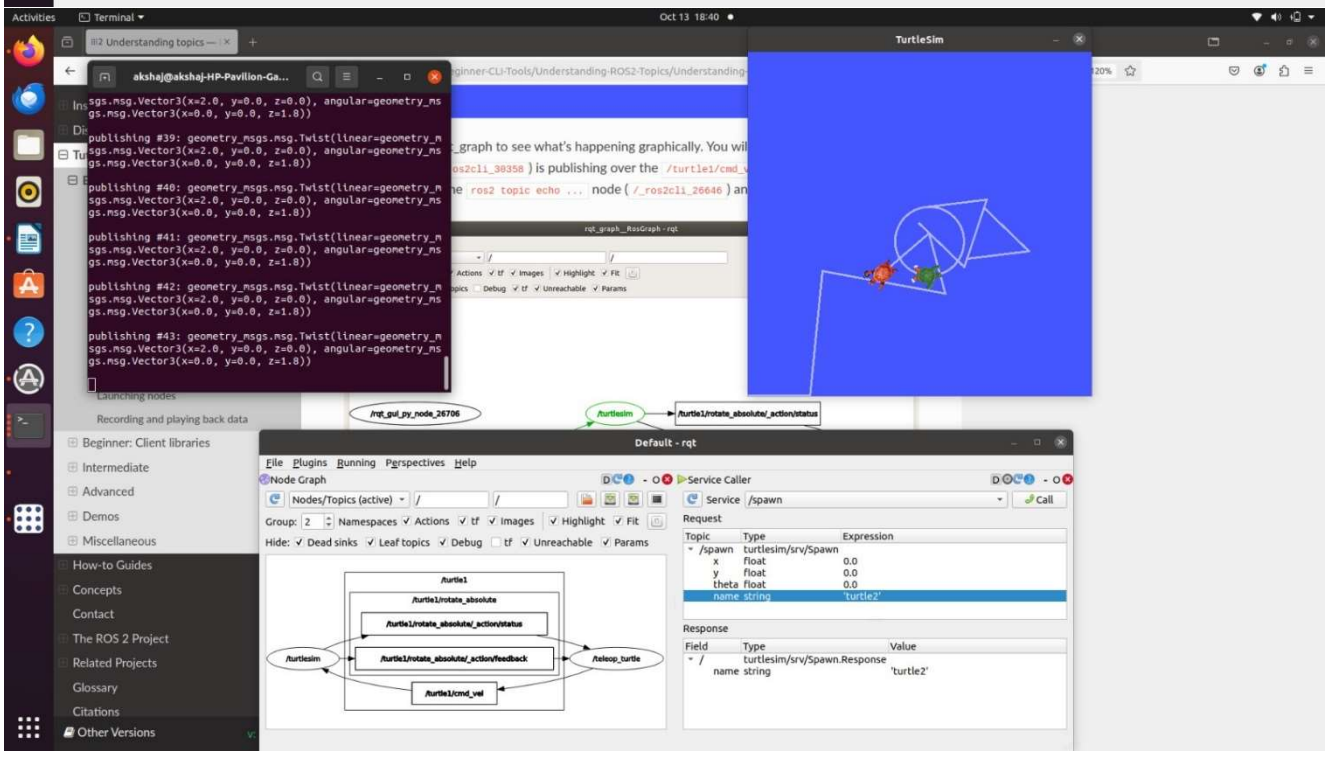
Command to run RQT:

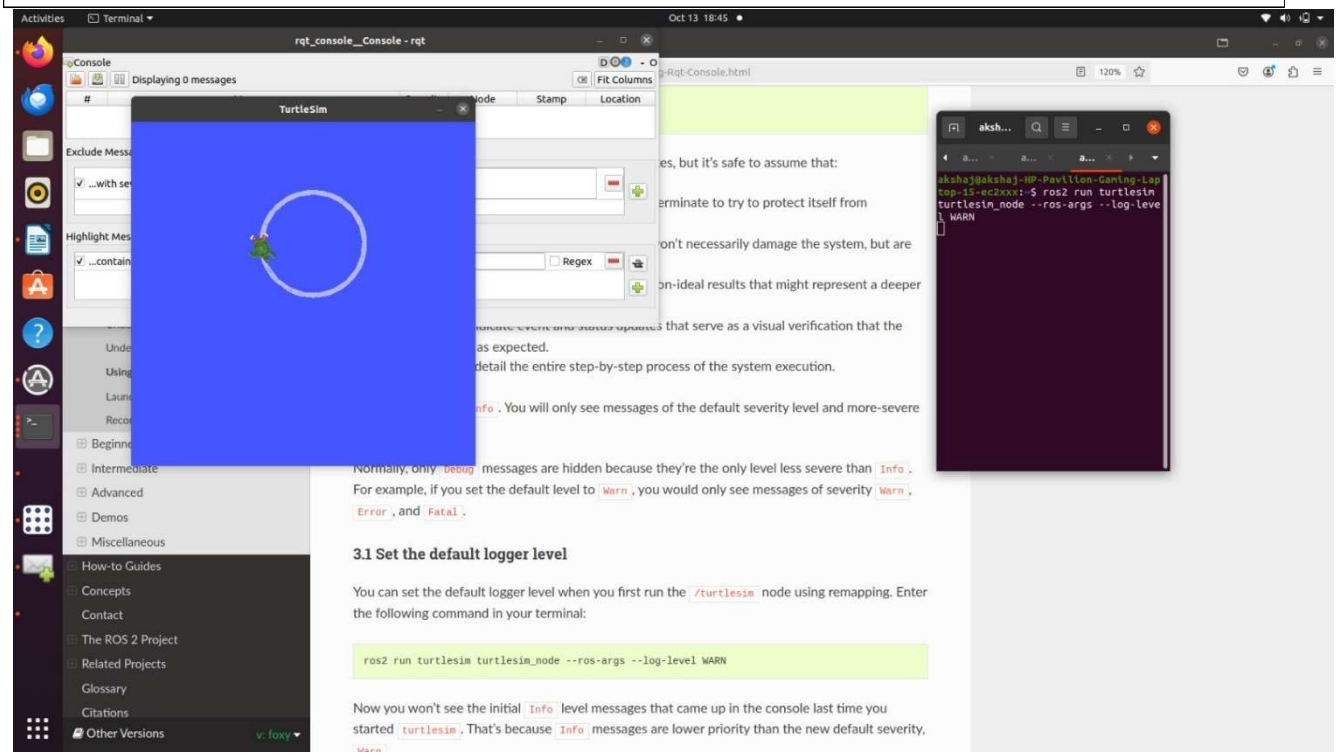
bash

rqt




Activities
rqt
Oct 13 18:38
Default - rqt
File Plugins Running Perspectives Help
Node Graph
Nodes only
Service Caller
Service /spawn
Call
Request
Response
/turtle1/rotate_absolute/_action/feedback
/turtlesim
/turtle1/rotate_absolute/_action/status
/teleop_turtle
/turtle1/cmd_vel


Activities
Terminal
Oct 13 18:40
TurtleSim
akshaj@akshaj-HP-Pavilion-Ga...
Recording and playing back data
Beginner: Client libraries
Intermediate
Advanced
Demos
Miscellaneous
How-to Guides
Concepts
Contact
The ROS 2 Project
Related Projects
Glossary
Citations
Other Versions
Default - rqt
/turtle1
/turtle1/rotate_absolute
/turtle1/rotate_absolute/_action/status
/turtle1/rotate_absolute/_action/feedback
/turtle1/cmd_vel

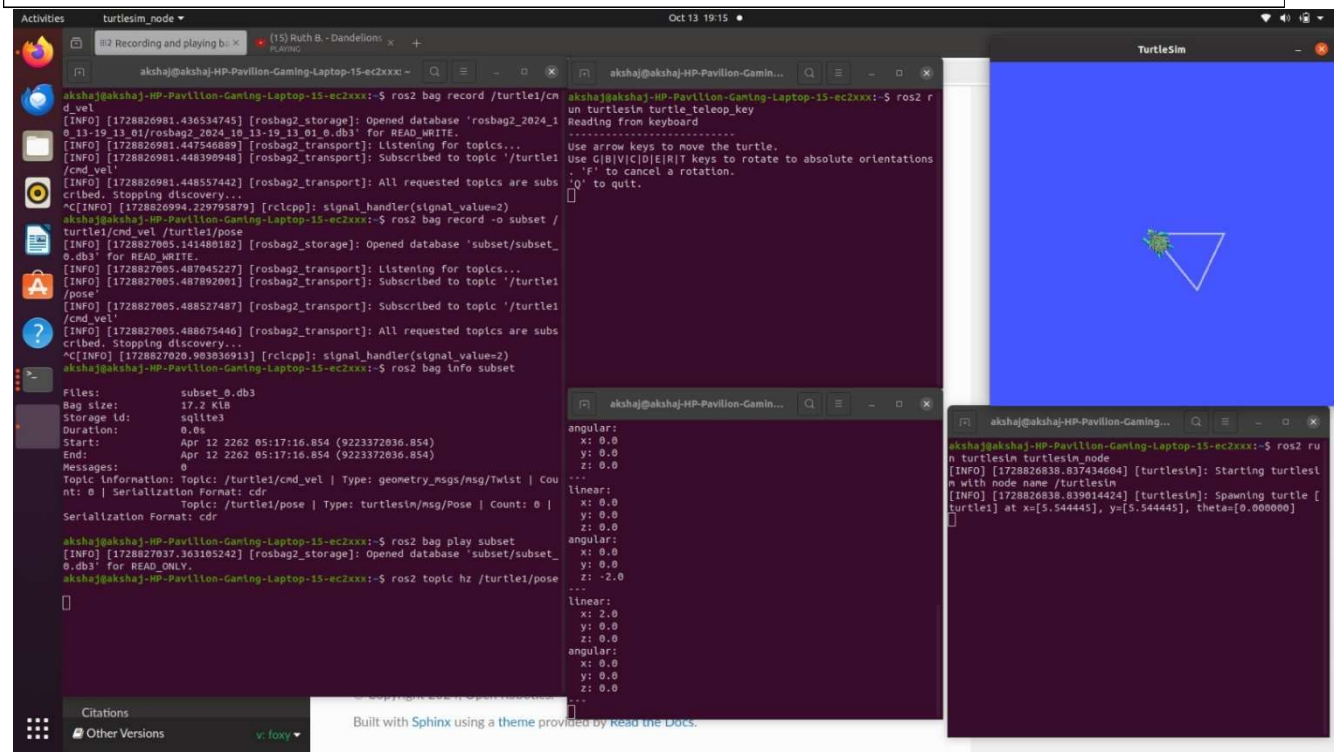


7. Recording and Playing Back Data

ROS2 allows recording data from topics using ros2 bag. This tool is useful for debugging, testing, and sharing experiments. The recorded data can be replayed to reproduce the results.

Command to record data:
bash
ros2 bag record <topic_name>

Command to play back recorded data:
bash
ros2 bag play <bag_file_name>



Basic Commands for Each Topic

Nodes
ros2 node list: Lists all active nodes in the system.
ros2 node info <node_name>: Displays detailed information about a specific node, including its publishers, subscribers, services, and actions.

Topics
ros2 topic list: Lists all active topics.
ros2 topic echo <topic_name>: Displays messages published to a topic in realtime.
ros2 topic pub <topic_name> <msg_type> <args>: Publishes messages to a topic.

Services
ros2 service list: Lists all available services.
ros2 service call <service_name> <service_type>: Calls a service and passes necessary arguments.

Actions
ros2 action list: Lists all actions available in the ROS graph.
ros2 action send_goal <action_name> <action_type> <values>: Sends a goal to an action server.

Parameters
ros2 param list: Lists parameters of a node.
ros2 param set <node_name> <param_name> <value>: Sets a parameter for a node.
ros2 param get <node_name> <param_name>: Retrieves the current value of a parameter.

RQT

rqt: Launches the RQT graphical interface to visualize ROS systems.

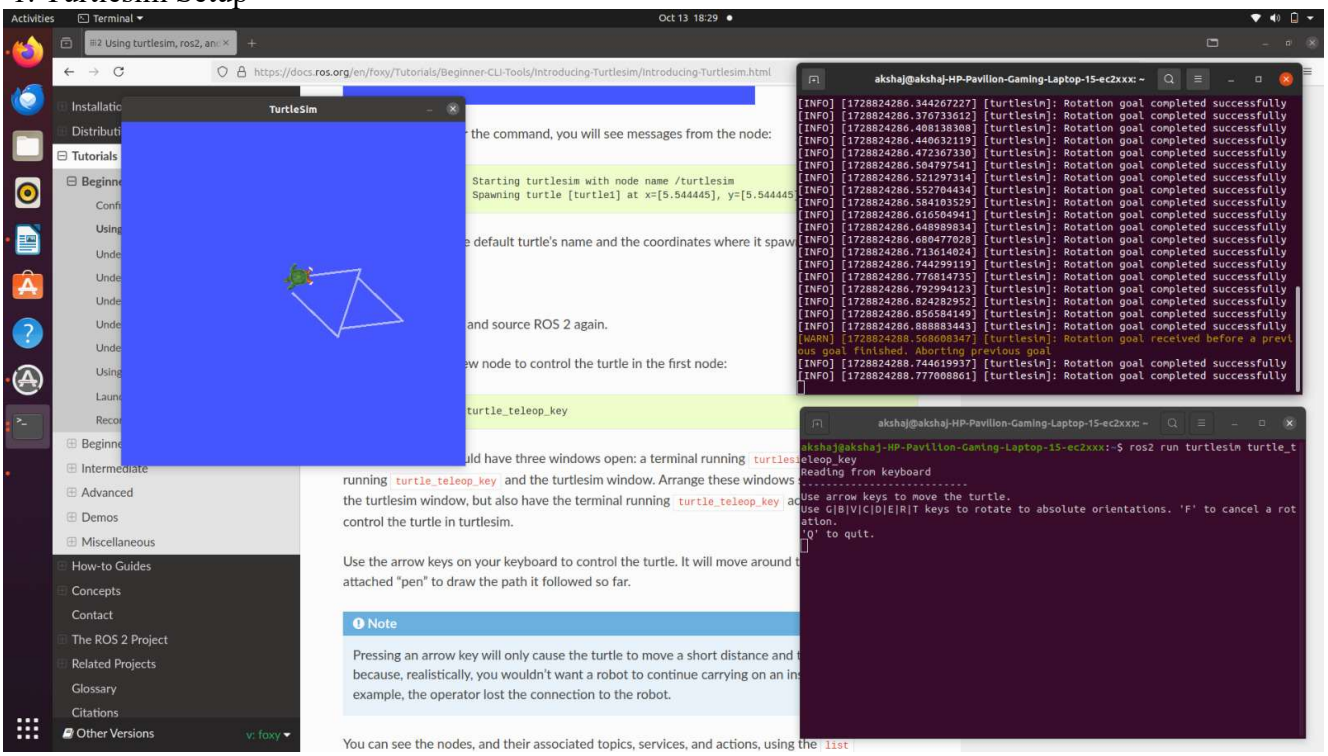
Recording and Playback

ros2 bag record <topic_name>: Records data from a topic.

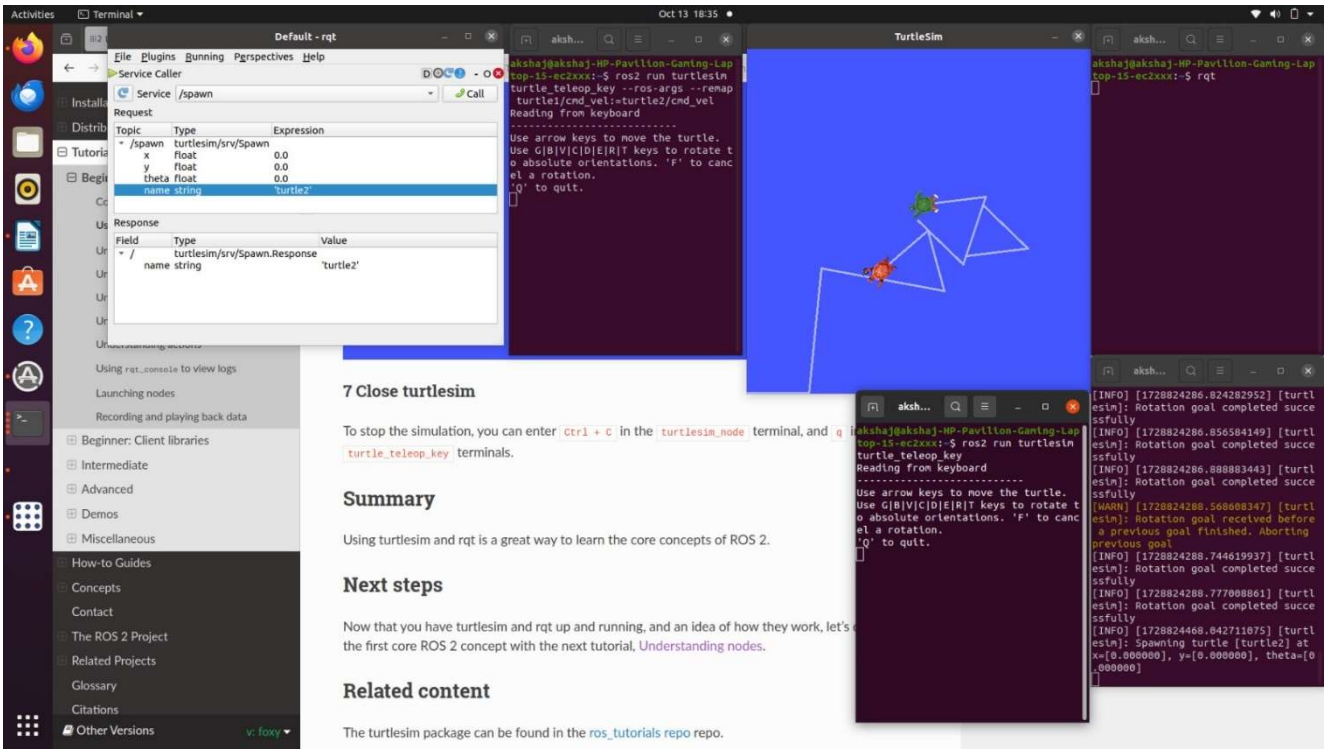
ros2 bag play <bag_file_name>: Plays back recorded data.

Snapshots

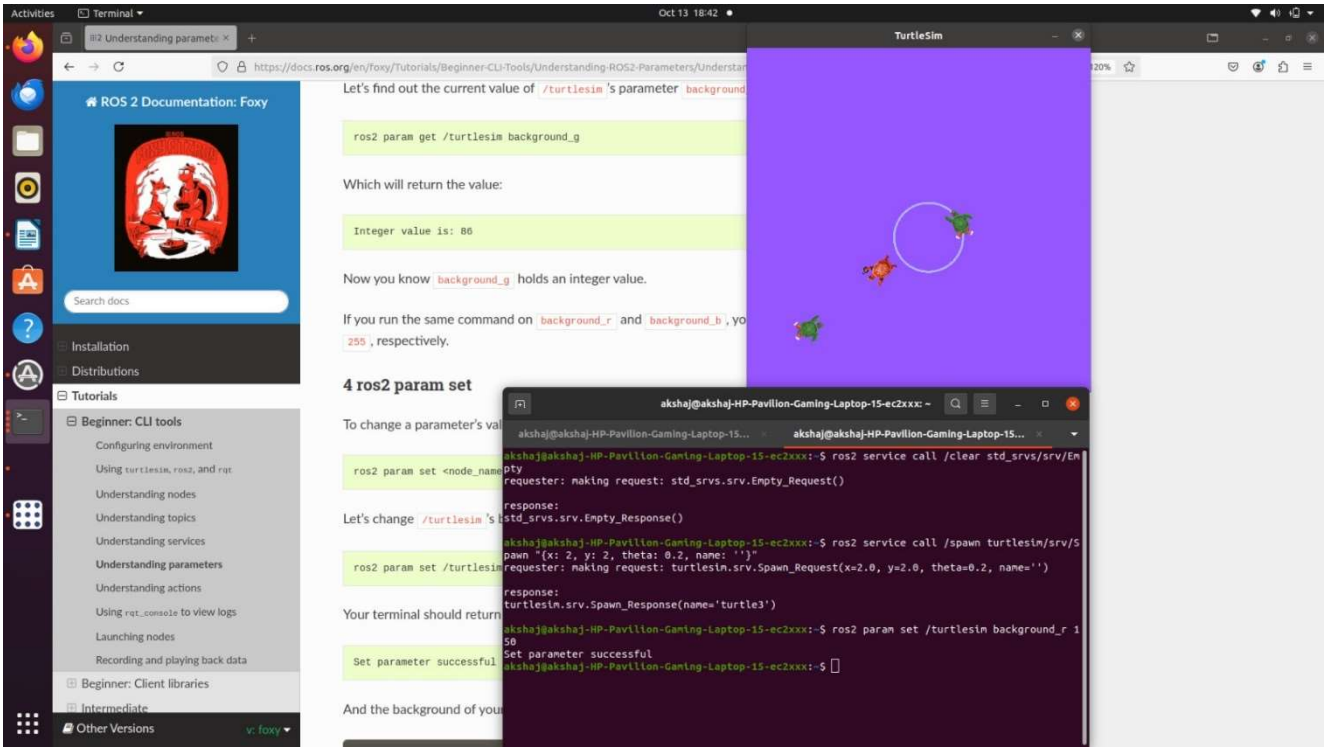
1. Turtlesim Setup

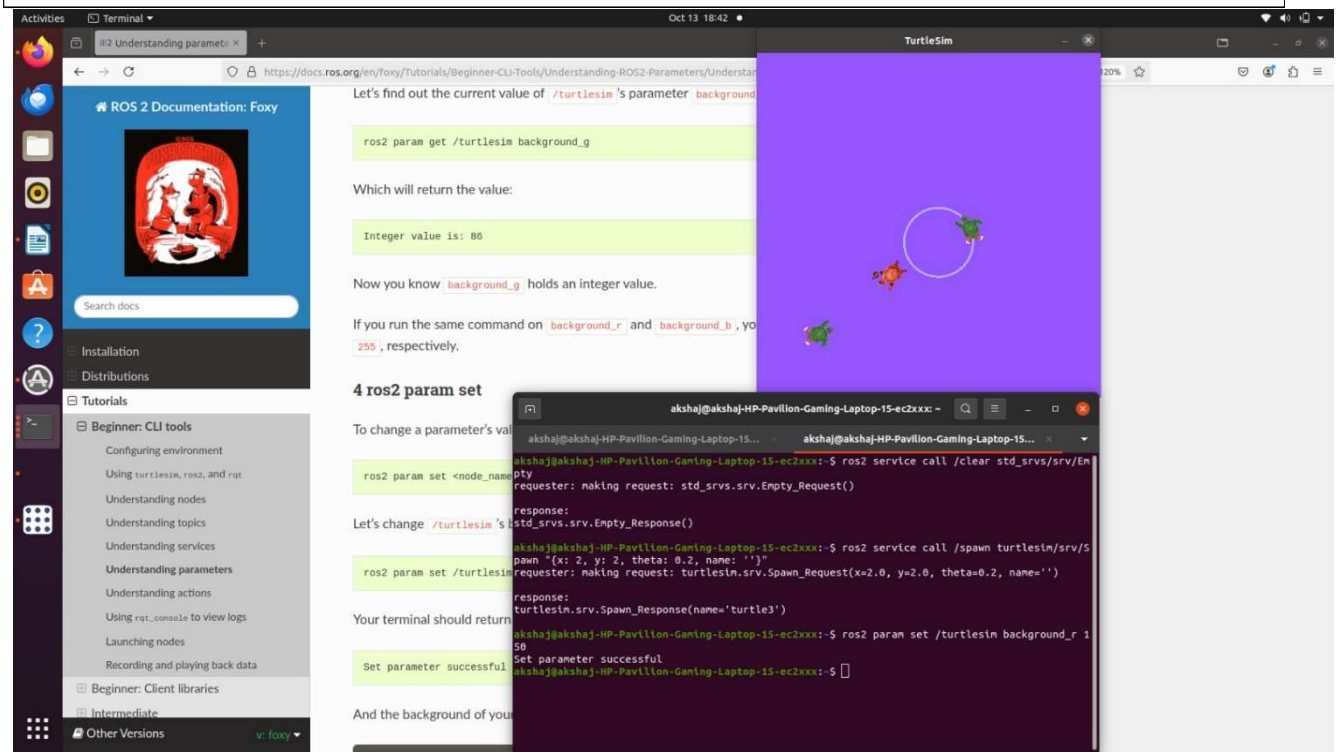


2. RQT Interface

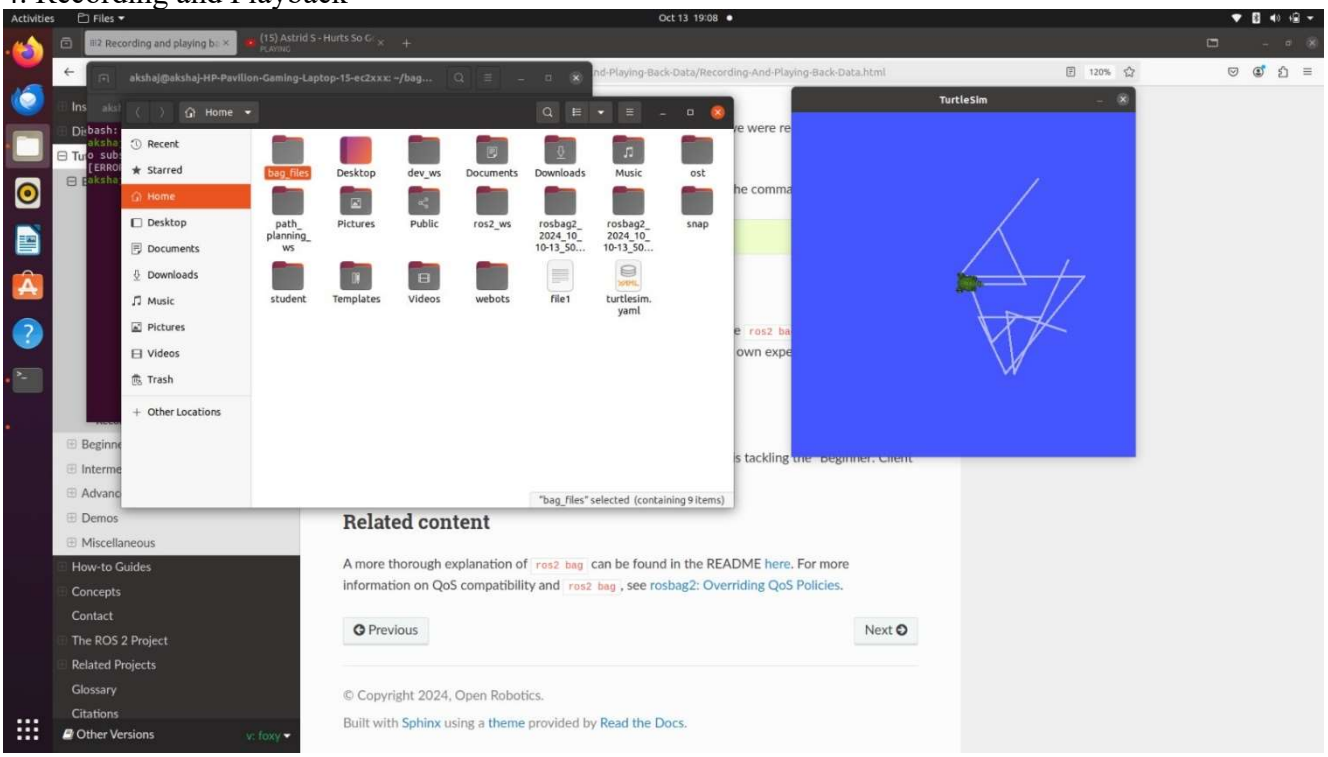


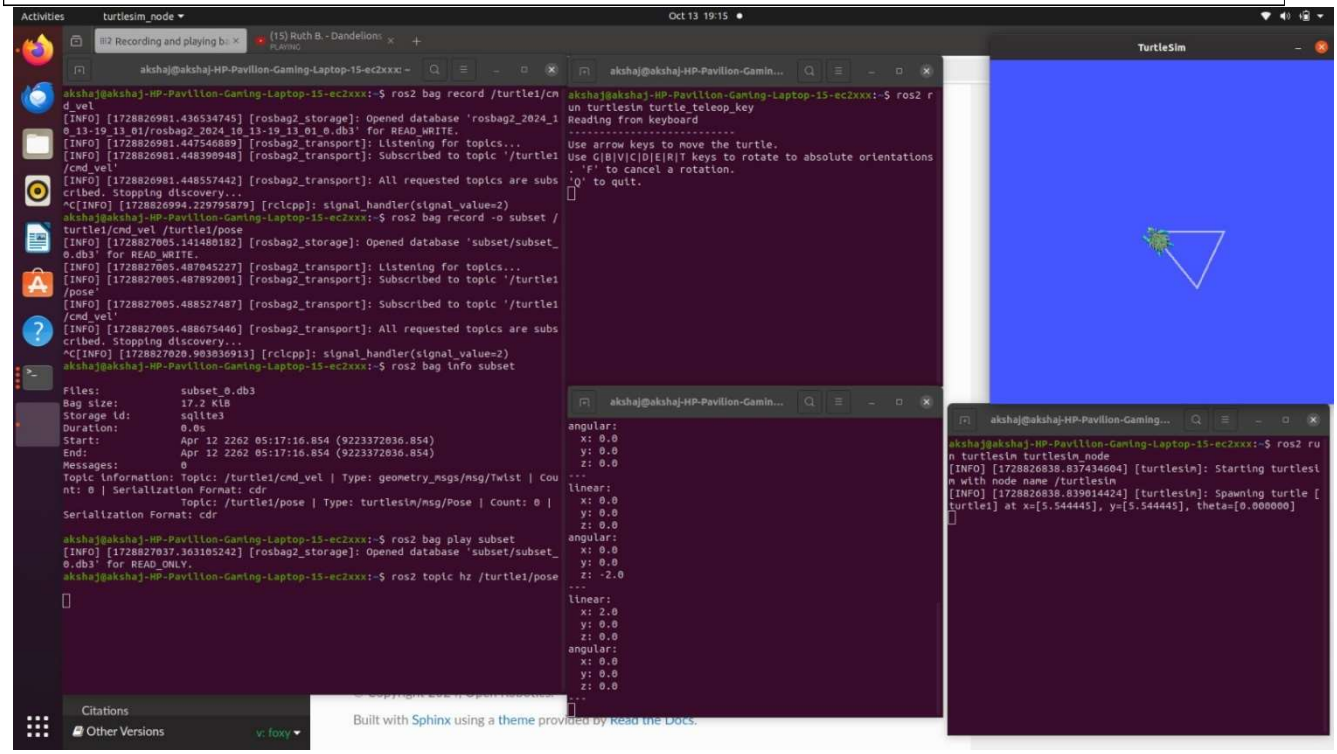
3. Parameter Setting





4. Recording and Playback





Learning Outcomes

Gained a deep understanding of ROS2 nodes, topics, services, actions, and parameters.
Developed proficiency in using RQT to visualize and interact with ROS2 systems.
Learned how to record and replay data using ros2 bag, which is essential for debugging and testing in ROS.
Practiced controlling the Turtlesim simulation using both the terminal and graphical tools.