### **Project Report: ROS2 Turtlebot Follower and Turtlebot Navigation**

#### **1. Project Title:**

ROS2 Turtlebot Follower and Turtlebot Navigation

#### **2. Objective:**

The primary objective of this project was to develop two distinct functionalities for Turtlebot robots using ROS2:

1. A follower robot that can track and follow a lead robot.
2. Autonomous navigation for a custom-built mobile robot, enabling it to map its environment and navigate autonomously while avoiding obstacles.

#### **3. Introduction:**

The project explores the capabilities of ROS2 (Robot Operating System) in controlling Turtlebot robots, particularly focusing on inter-robot communication and autonomous navigation. The first part of the project involved creating a follower robot that can track and follow a lead robot using TF2 transformations. The second part involved designing a custom mobile robot and enabling it to autonomously navigate an environment, leveraging both local and global mapping techniques.

#### **4. Methodology:**

**First Project: Turtlebot Follower**

* **Robot Setup**: The Turtlebot was set up by downloading and installing the necessary ROS2 packages from the official ROS page.
* **Teleop Control**: The lead robot was manually controlled using the teleop\_twist keyboard operation.
* **Spawning a New Robot**: A second robot was spawned in the simulation environment.
* **TF2 Communication**: Communication between the two robots was established using the TF2 package. A TF Broadcaster (broadcaster.py) was implemented in the lead robot to broadcast its position, while the follower robot used a TF Listener (listener.py) to receive the position data and follow the lead robot.

**Second Project: Turtlebot Navigation**

* **Robot Design**: A basic mobile robot was designed using a .xacro file, integrating components like IMU and GPS.
* **Launch Configuration**: The relevant parameters and sensor configurations were edited in the .launch file, and the robot was built using colcon build.
* **Simulation**: The robot was launched in Gazebo and visualized in RViz.
* **Odometry and Mapping**: Odometry was added to the robot for accurate movement tracking. The robot was controlled using the teleop\_twist keyboard and used Cartographer for creating a map of its environment. The map was saved locally.
* **Autonomous Navigation**: The saved map was used for autonomous navigation, where the robot utilized both local and global maps to navigate the environment and avoid obstacles. The robot's performance was observed in Gazebo and RViz.

#### **5. Results/Findings:**

* **Turtlebot Follower**: The follower robot successfully tracked and followed the lead robot by receiving its position data via TF2. The communication between the two robots was stable, and the follower robot accurately followed the lead robot’s movements.
* **Turtlebot Navigation**: The custom mobile robot was successfully built and simulated. The robot was able to map the environment using Cartographer and navigate autonomously using the local and global maps. The robot effectively avoided obstacles and reached target destinations within the simulated environment.

#### **6. Discussion/Analysis:**

* **Challenges**: While the initial implementations were successful, challenges arose when attempting to replicate the results for further learning. Modifications to certain files during the replication process led to errors, preventing the final results from being re-recorded.
* **Learning Outcomes**: Despite these challenges, the project provided significant learning experiences in ROS2, TF2, robot simulation, and autonomous navigation. The importance of careful file management and version control was highlighted during the troubleshooting process.

#### **7. Conclusion:**

The ROS2 Turtlebot Follower and Turtlebot Navigation project demonstrated the capabilities of ROS2 in facilitating inter-robot communication and autonomous navigation. The project achieved its objectives, though some challenges were encountered during the replication phase. Future work could involve refining the error-prone areas and exploring more advanced navigation techniques.