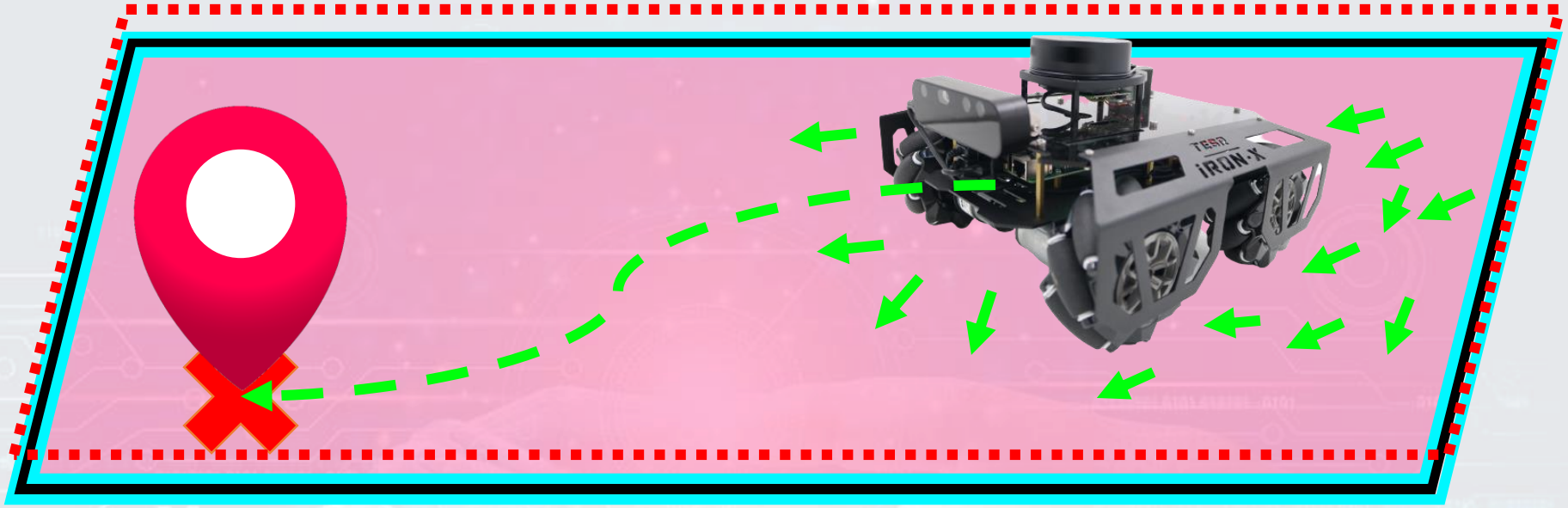


iron-X's SLAM and Navigation

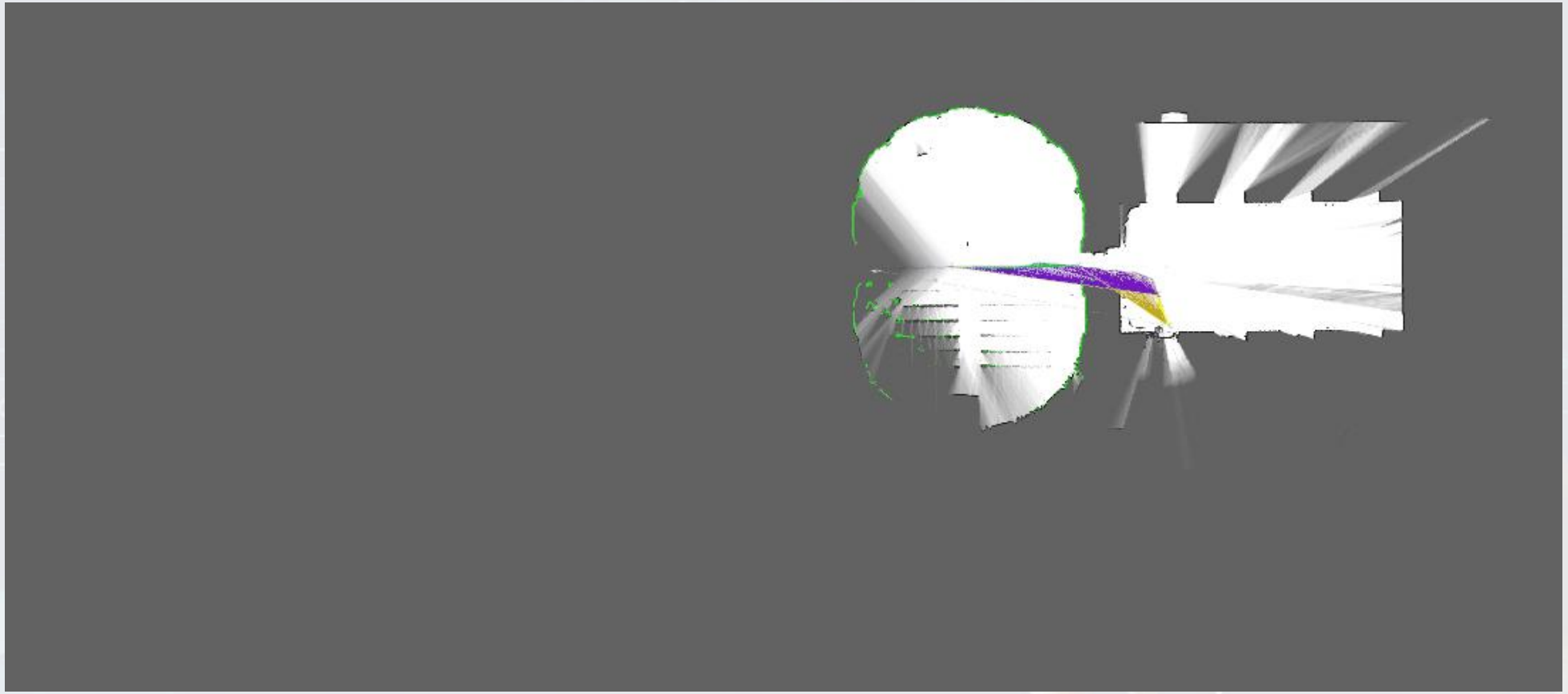
By TESR

ROS2

Thai Embedded

iron-X's SLAM Cartographer

- **Cartographer** is a system that provides real-time simultaneous localization and mapping (SLAM) in 2D and 3D across multiple platforms and sensor configurations.



iron-X's SLAM Cartographer

- To run SLAM you must to active the iron-X's bringup by using:

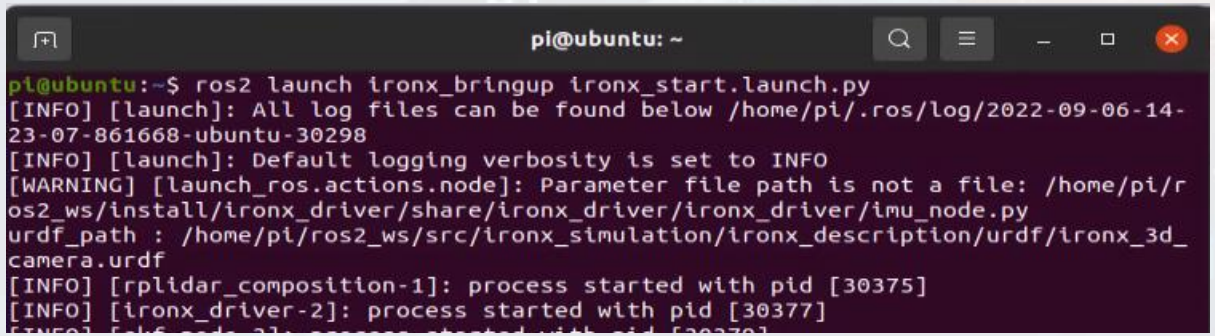
*On the Remote Desktop(NoMachine) or Remote Terminal

```
ros2 launch ironx_bringup ironx_start.launch.py
```

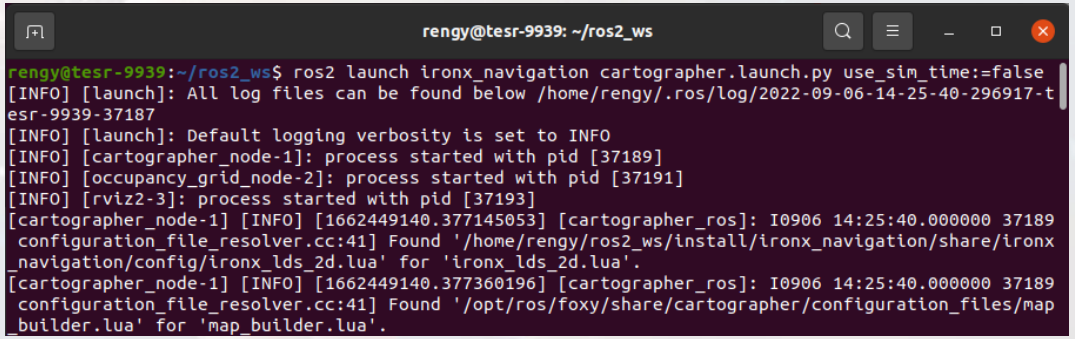
- And then, launch the iron-X's SLAM Cartographer by using:

*On the PC/Laptop Terminal

```
ros2 launch ironx_navigation cartographer.launch.py use_sim_time:=false
```



*On the Remote Desktop(NoMachine) or Remote Terminal



*On the PC/Laptop Terminal

iron-X's SLAM Cartographer

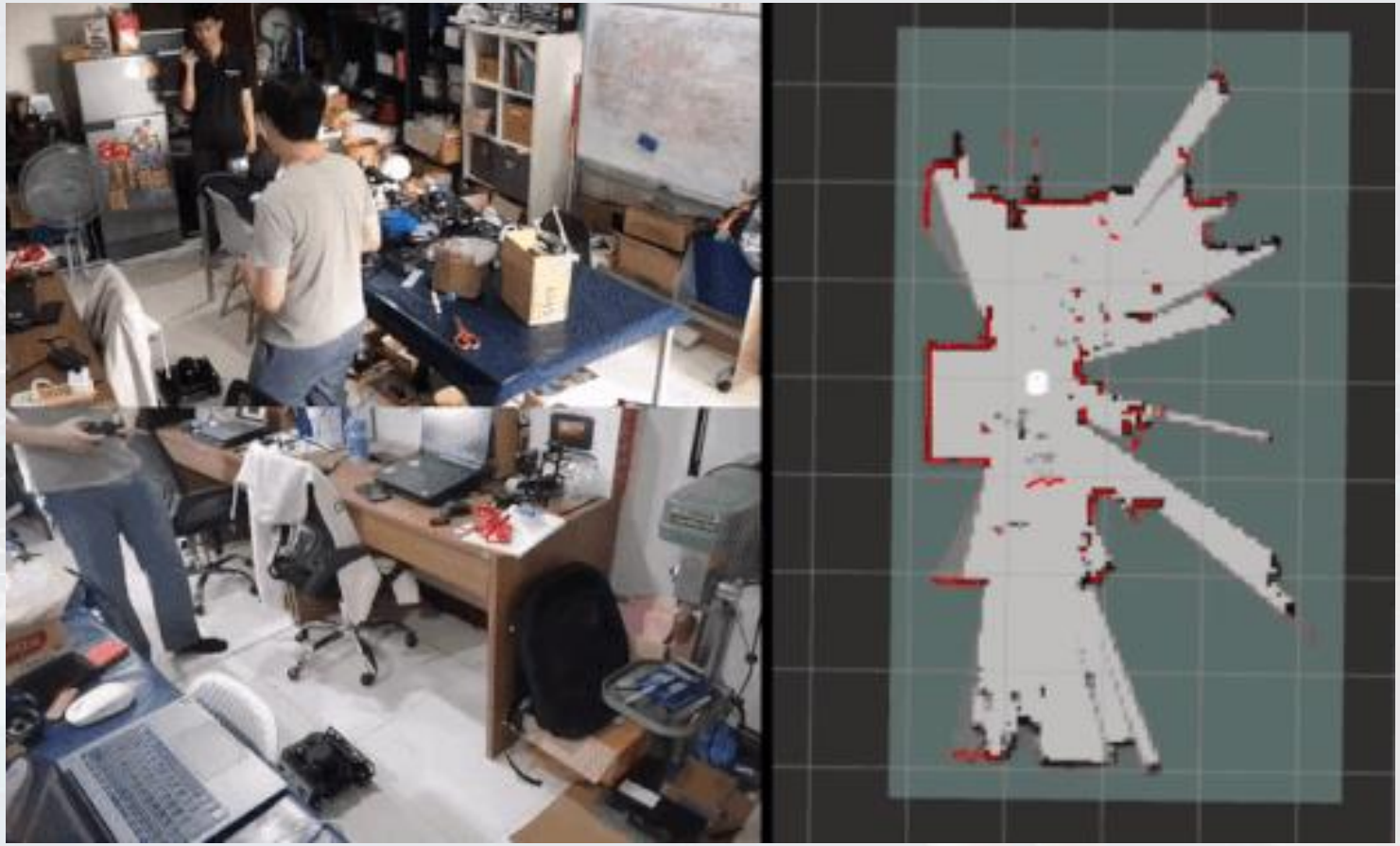
- And then, you may use the keyboard to control iron-X's by type:

```
ros2 run teleop_twist_keyboard teleop_twist_keyboard
```

- You can also use the joystick that connected on module on iron-X too.

iron-X's SLAM Cartographer

- Using joystick to move around your place to draw a map.



Speed x6.6

Save a map for Navigation

- And then, you can save a map that you drew using:

```
ros2 run nav2_map_server map_saver_cli -f ~/ros2_ws/src/ironx_navigation/map/map_test
```

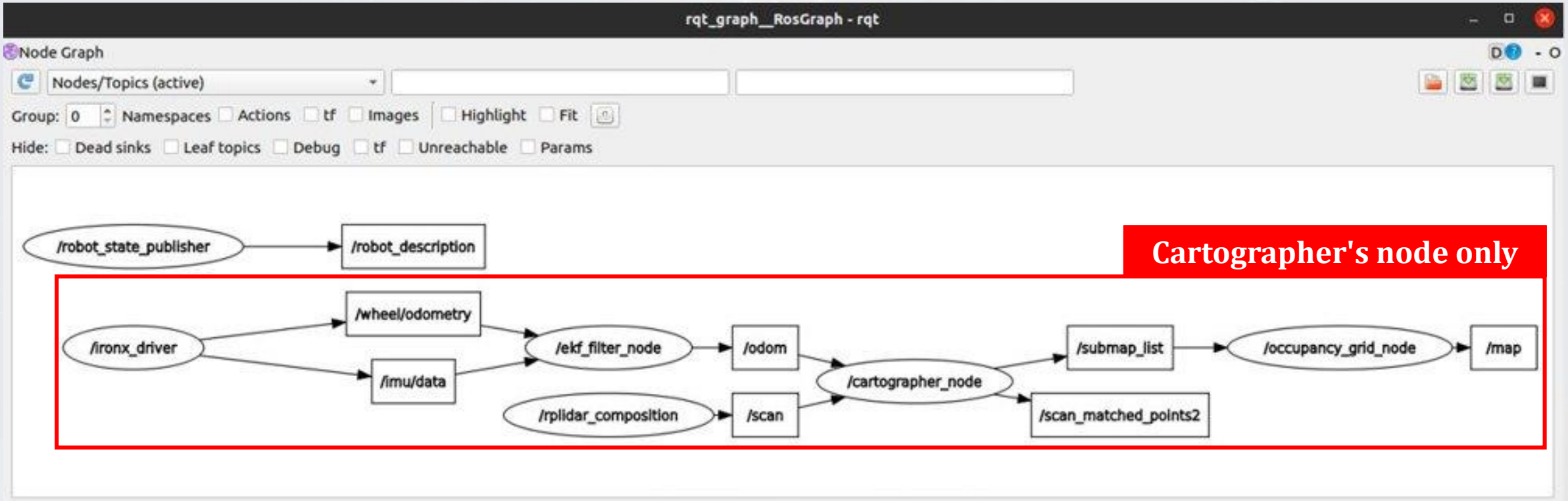
```
rengy@tesr-9939: ~
rengy@tesr-9939:~$ ros2 run nav2_map_server map_saver_cli -f ~/ros2_ws/src/ironx_navigation/map/map_test
[INFO] [1662450140.899949523] [map_saver]:
       map_saver lifecycle node launched.
       Waiting on external lifecycle transitions to activate
       See https://design.ros2.org/articles/node_lifecycle.html for more information.
[INFO] [1662450140.900033273] [map_saver]: Creating
[INFO] [1662450140.900152330] [map_saver]: Saving map from 'map' topic to '/home/rengy/ros2_ws/src/ironx_navigation/map/map_test' file
[WARN] [1662450140.900162460] [map_saver]: Free threshold unspecified. Setting it to default value: 0.250000
[WARN] [1662450140.900171677] [map_saver]: Occupied threshold unspecified. Setting it to default value: 0.650000
[WARN] [map_io]: Image format unspecified. Setting it to: pgm
[INFO] [map_io]: Received a 140 X 85 map @ 0.05 m/pix
[INFO] [map_io]: Writing map occupancy data to /home/rengy/ros2_ws/src/ironx_navigation/map/map_test.pgm
[INFO] [map_io]: Writing map metadata to /home/rengy/ros2_ws/src/ironx_navigation/map/map_test.yaml
[INFO] [map_io]: Map saved
[INFO] [1662450141.425802215] [map_saver]: Map saved successfully
[INFO] [1662450141.425868321] [map_saver]: Destroying
```

After save the map you will get 2 file are map_test.pgm and map_test.yaml at where you are fill to save it. In this case, path of file is ~/ros2_ws/src/ironx_navigation/map

RosGraph of iron-X's SLAM Cartographer

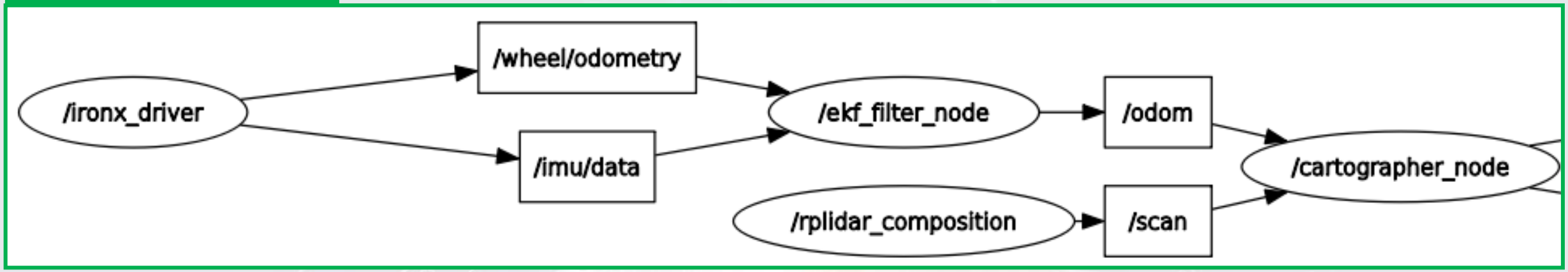
- You can see the RosGraph of iron-X's SLAM Cartographer using:

```
rqt_graph
```

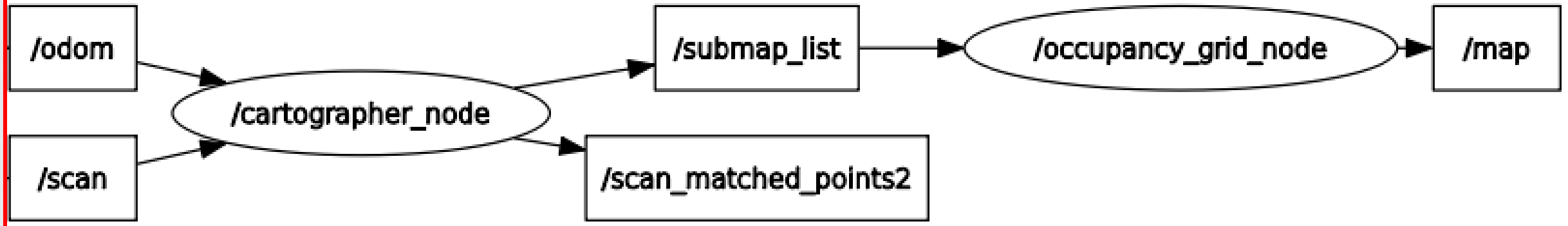


RosGraph of iron-X's SLAM Cartographer

Cartographer's Input



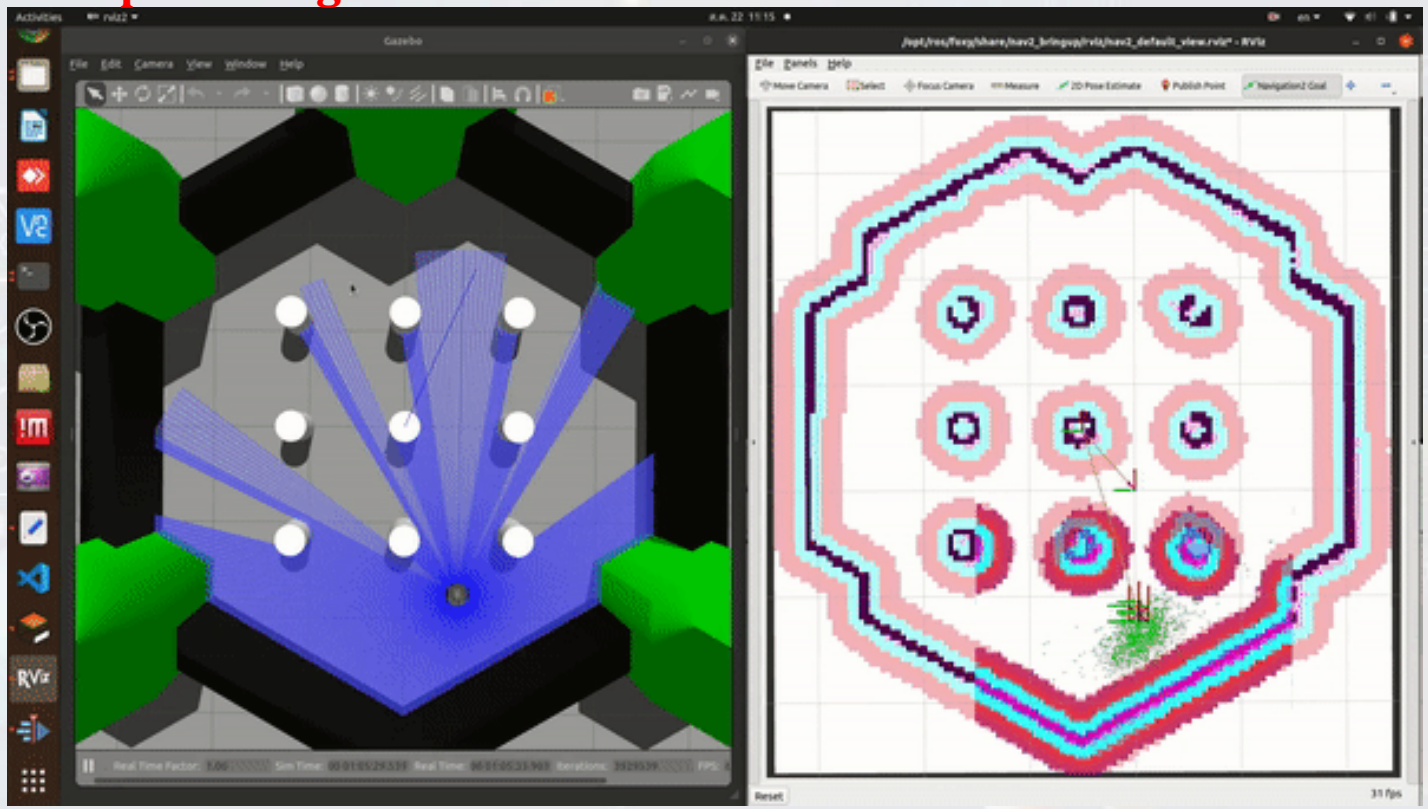
Cartographer's output



iron-X's Navigation

- A 2D navigation stack that takes in information from odometry, sensor streams, and a goal pose and outputs safe velocity commands that are sent to a mobile base

Example from gazebo



Speed x4

Launch the iron-X's Navigation

- Before run the navigation, you need to active iron-X's bringup using:

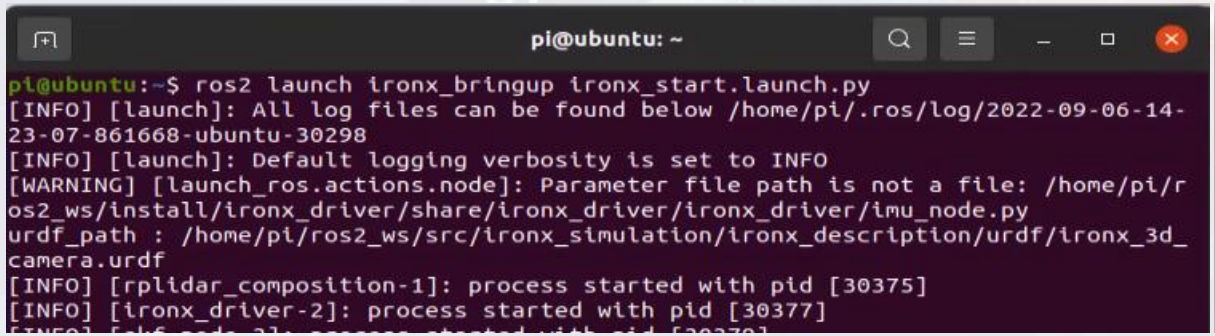
*On the Remote Desktop(NoMachine) or Remote Terminal

```
ros2 launch ironx_bringup ironx_start.launch.py
```

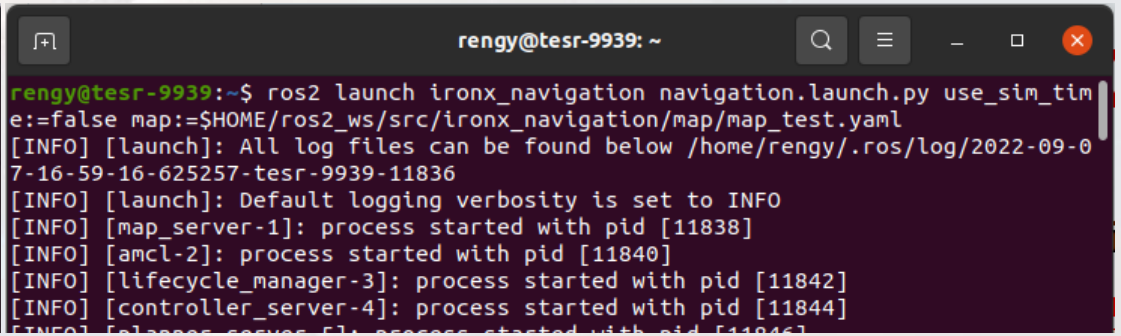
- You can run the iron-X's navigation by using:

*On the PC/Laptop Terminal

```
ros2 launch ironx_navigation navigation.launch.py use_sim_time:=false map:=$HOME/ros2_ws/src/ironx_navigation/map/map_test.yaml
```



*On the Remote Desktop(NoMachine) or Remote Terminal

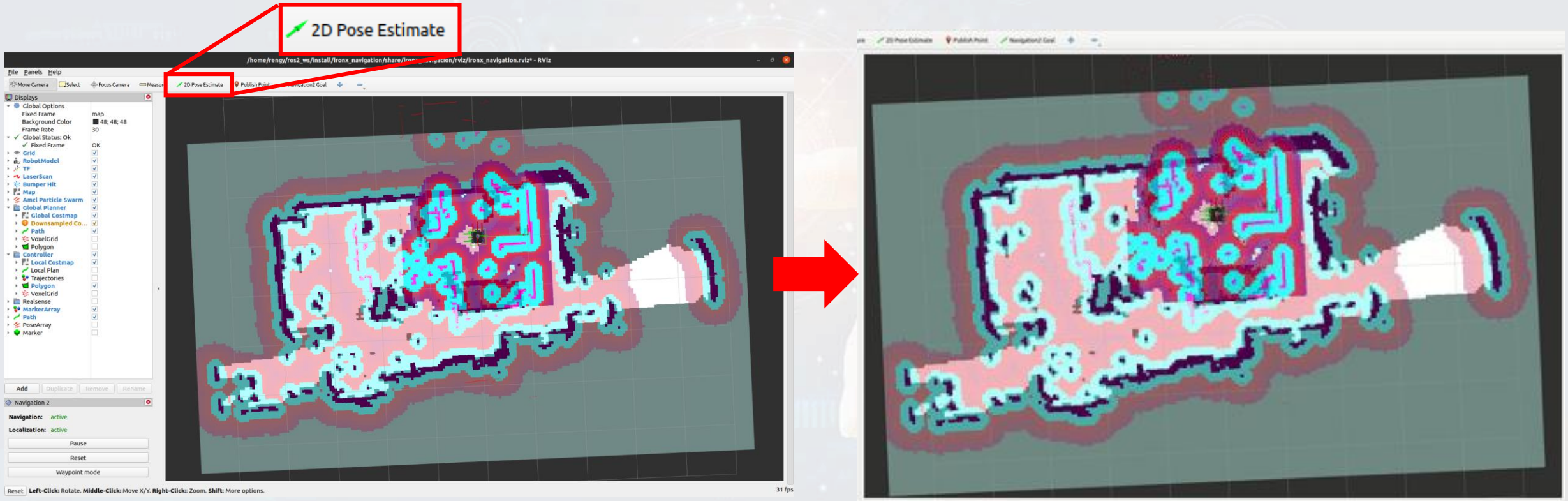


*On the PC/Laptop Terminal

Thai Embedded

Re-position using 2D Pose Estimate

- You can see on Rviz. If the start position of iron-X's model not the same as the real position. So, you can use **"2D Pose Estimate"** to re-position it:



Navigation2 Goal

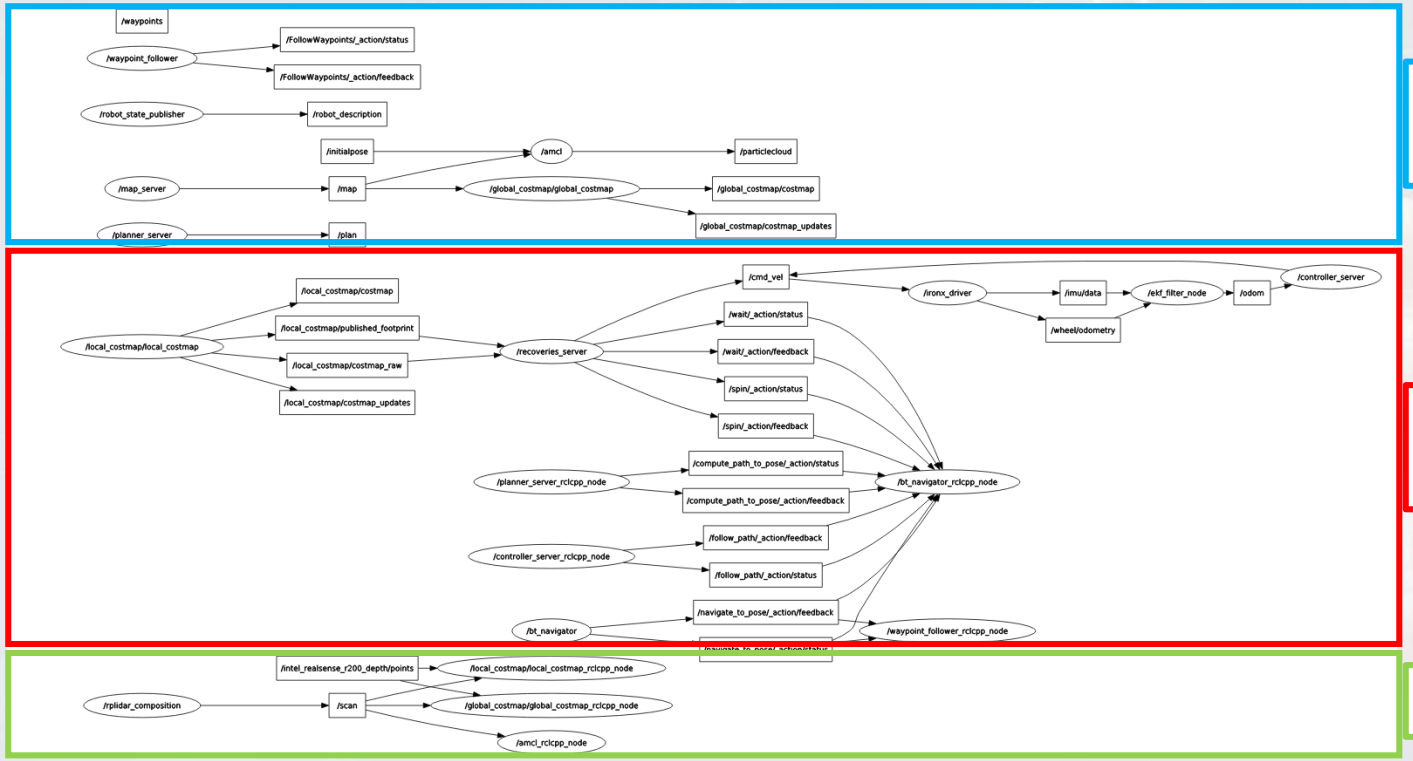
- And then, you can publish the goal for the navigation using **"Navigatoin2 Goal"**:

Speed x2

RosGraph of iron-X's Navigation

- You can see the RosGraph of iron-X's Navigation using:

```
rqt_graph
```



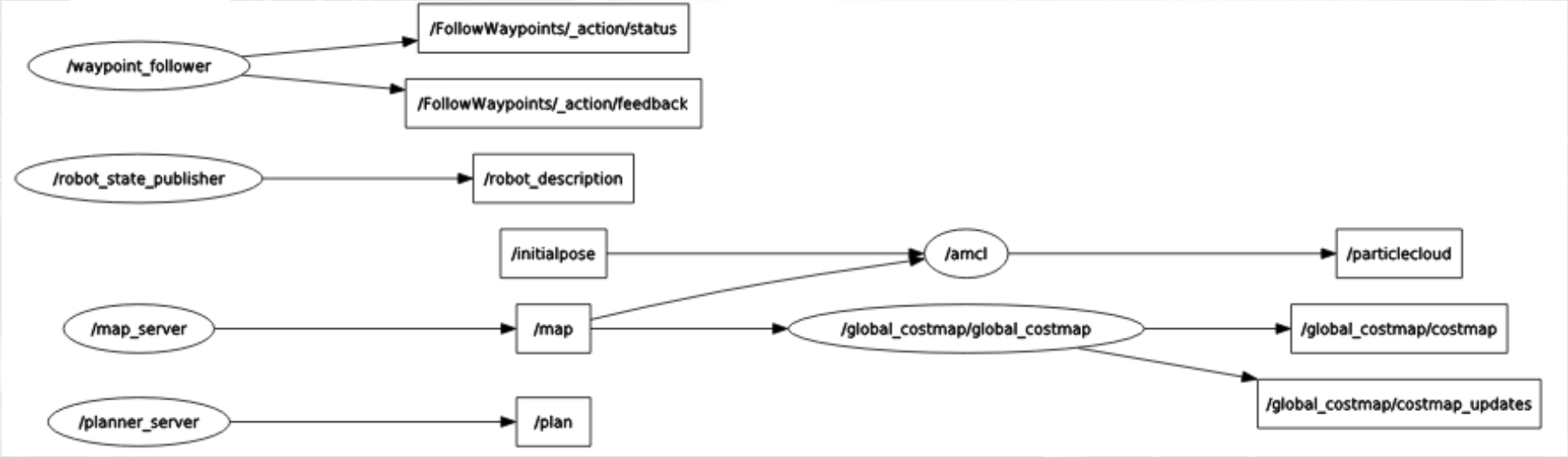
Waypoint_follower, map_server , amcl, global_costmap and planner_server:

Local_costmap, navigation node server and local_plan:

Rplidar_composition node:

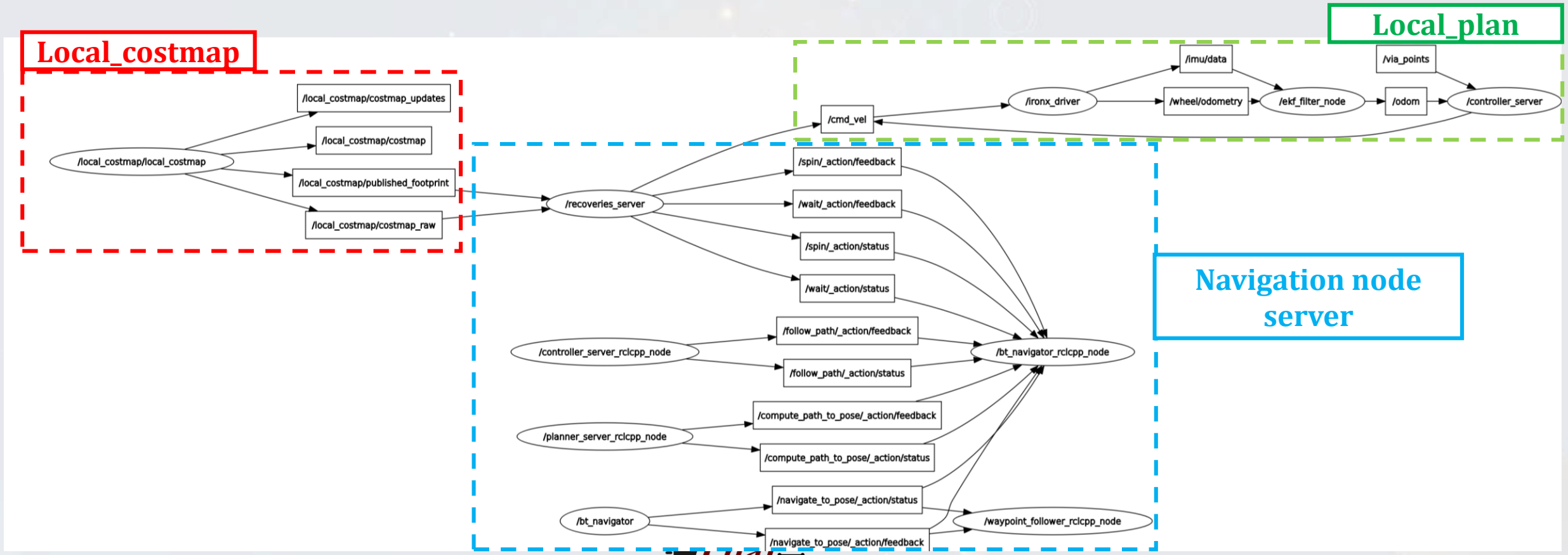
RosGraph of iron-X's Navigation

- **Waypoint_follower, map_server, amcl, global_costmap and planner_server:**
  - **Waypoint_follower** - module implement a way of doing waypoint following.
  - **Map_server** - provide the map to the navigation system used in topic and service of nav.
  - **AMCL** - is a probabilistic localization system for a robot moving in 2D.
  - **Global_costmap** - the environment provided from sensor data and 2d static map data.
  - **Planner_server** - handling the plan that requests for the stack and host a map of plugin implementations.



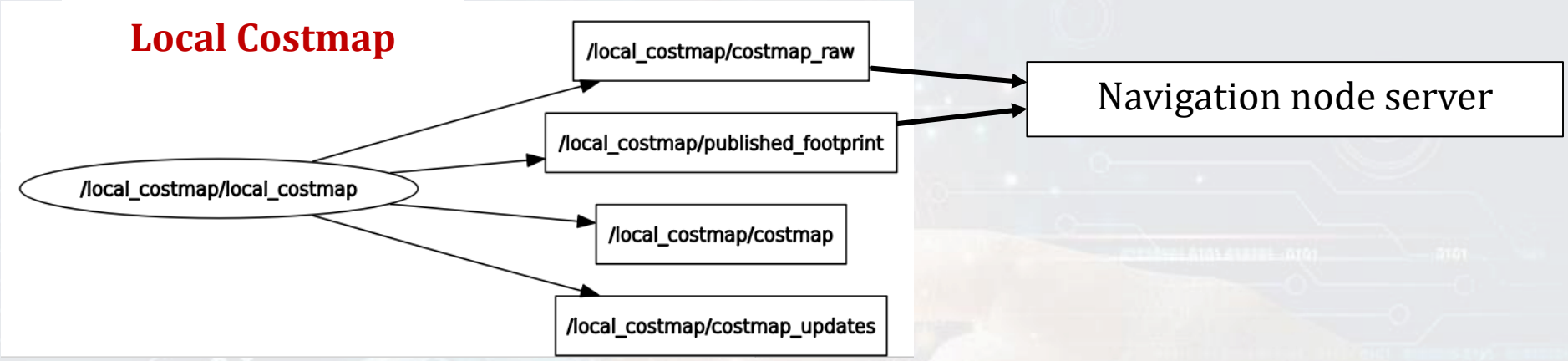
RosGraph of iron-X's Navigation

- Local_costmap, navigation node server and local_plan:

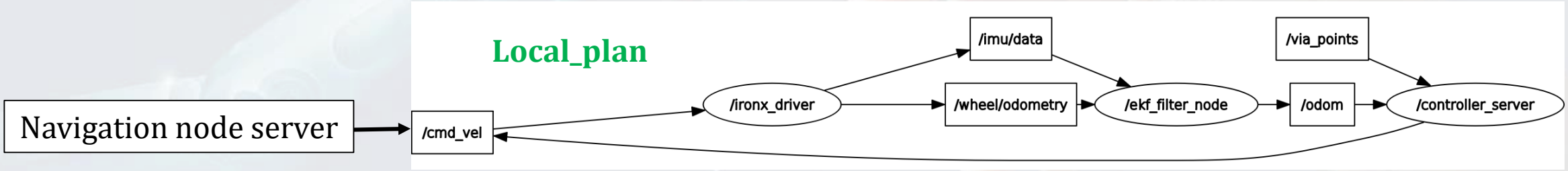


RosGraph of iron-X's Navigation

- **Local_costmap and local_plan:**
  - **Local_costmap –** 2d_costmap layer which is the environment provided from sensor data.

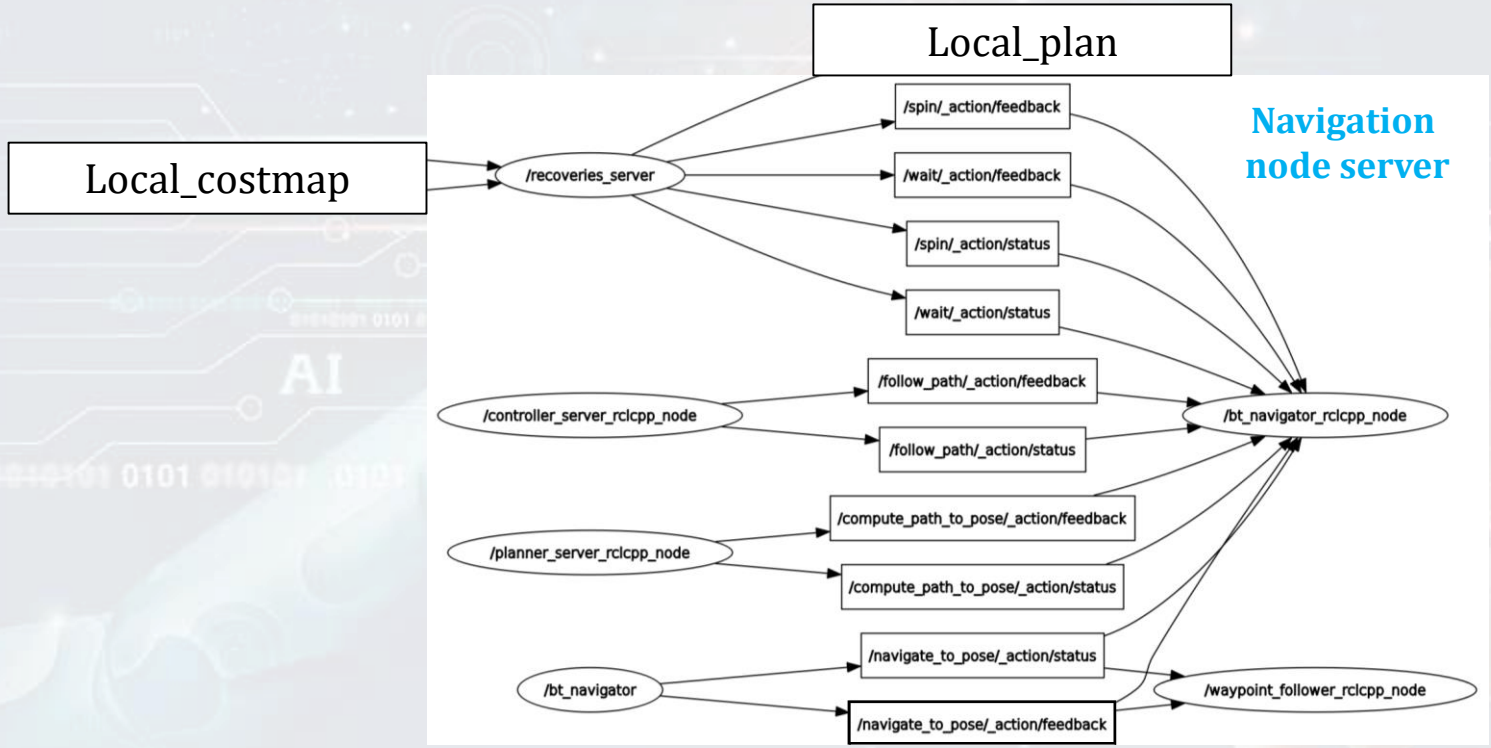


  - **Local_plan -** provides implementations of the Trajectory Rollout and Dynamic Window approaches to local robot navigation on a plane.



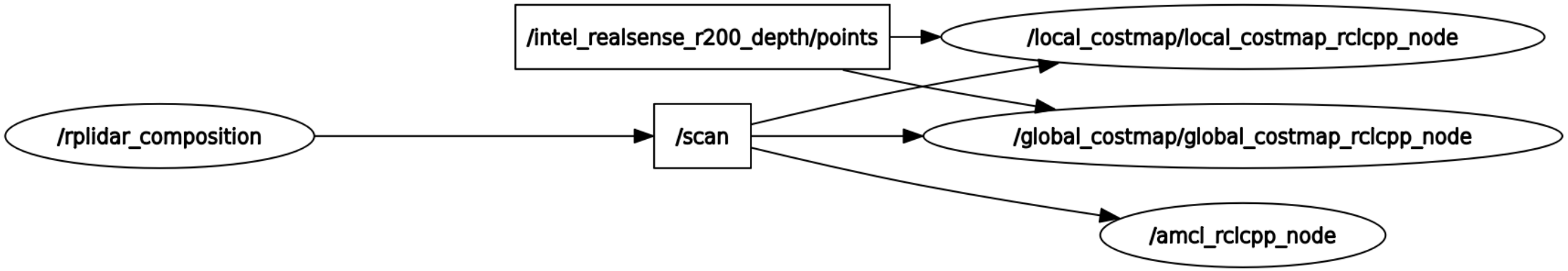
RosGraph of iron-X's Navigation

- **Navigation node server –** the part of the navigation stack system that working as a server to provide the action to the iron-X**:**



RosGraph of iron-X's Navigation

- **Rplidar_composition node –** the rplidarNode which is a actuator driver for RPLIDAR that provide scan data from LIDAR sensor. From this graph, it provide scan data to local_costmap, global_costmap and amcl node**:**



Contact Us

Email: tesrshop@gmail.com
Line official Account: @ion1900z
Facebook fanpage: TESR
Tel. 082-983-7768