# ROS TOOLS

Paloma de la Puente

INDUSTRIALES
ETSII | UPM

# Content

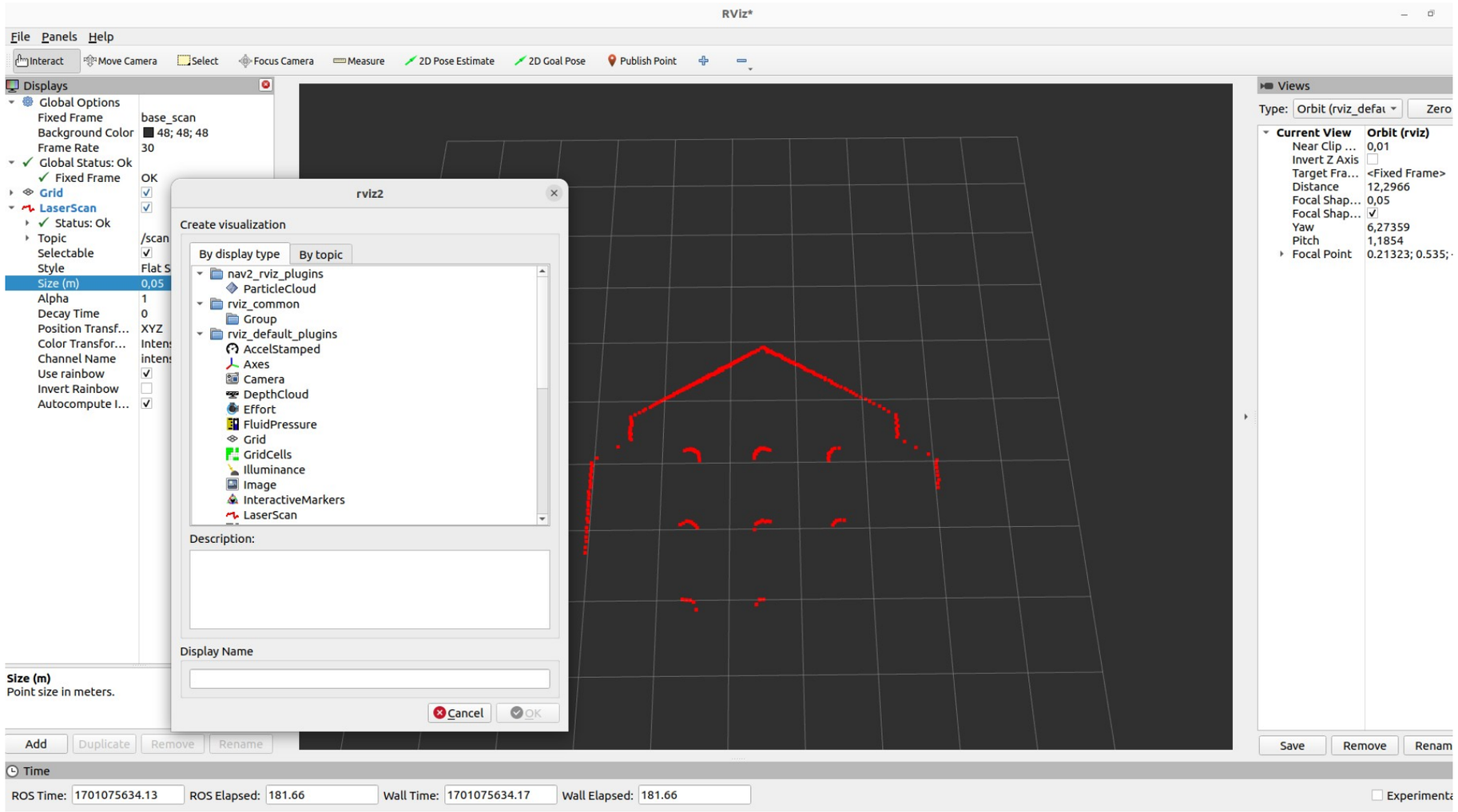
- Rviz visualization
- Tf library
- Rqt
- Rosbag

Tools

# RVIZ VISUALIZATION

# RVIZ visualization

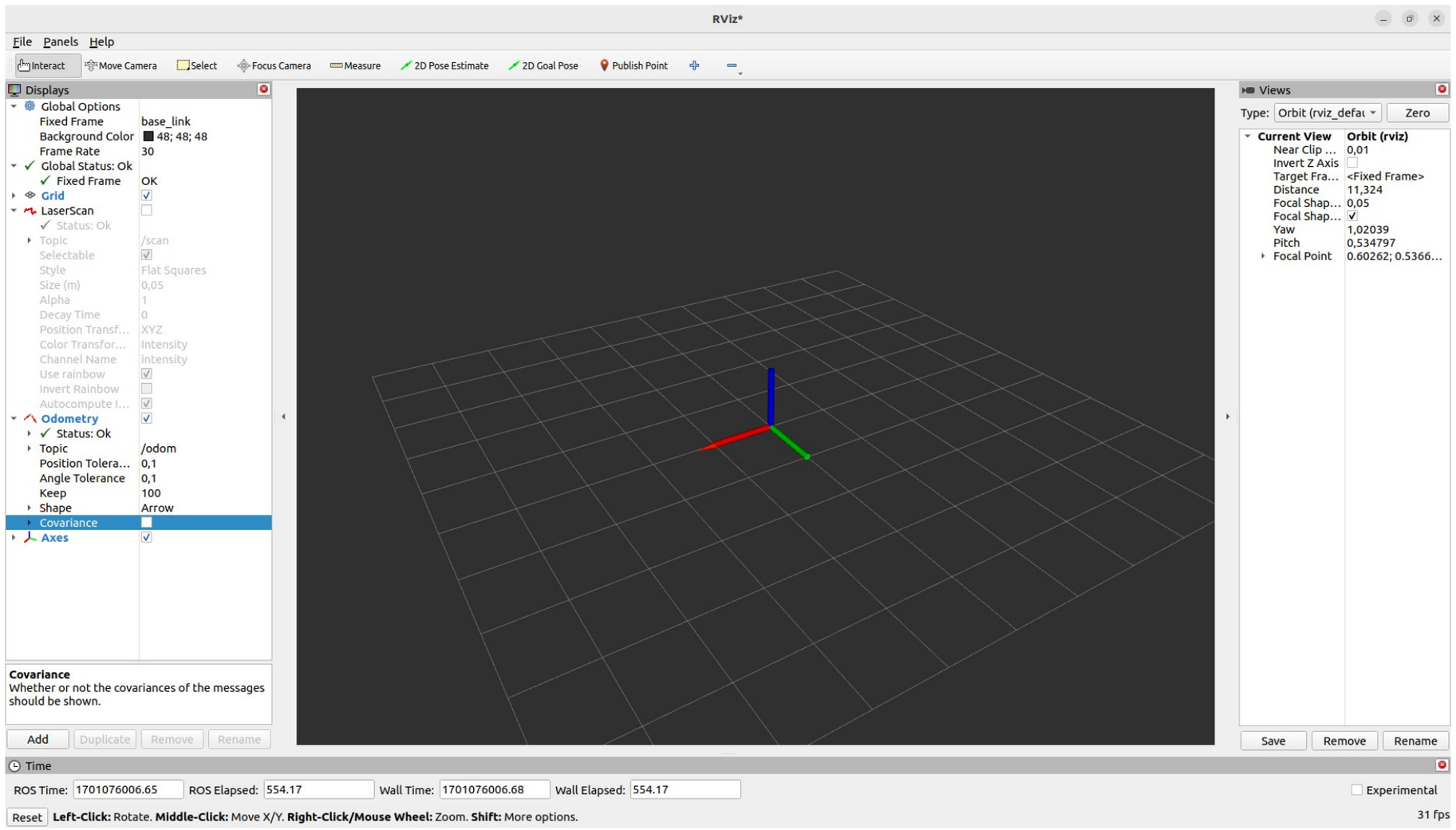
ros2 run rviz2 rviz2




Remember to save your desired configurations!

# RVIZ visualization

ros2 run rviz2 rviz2

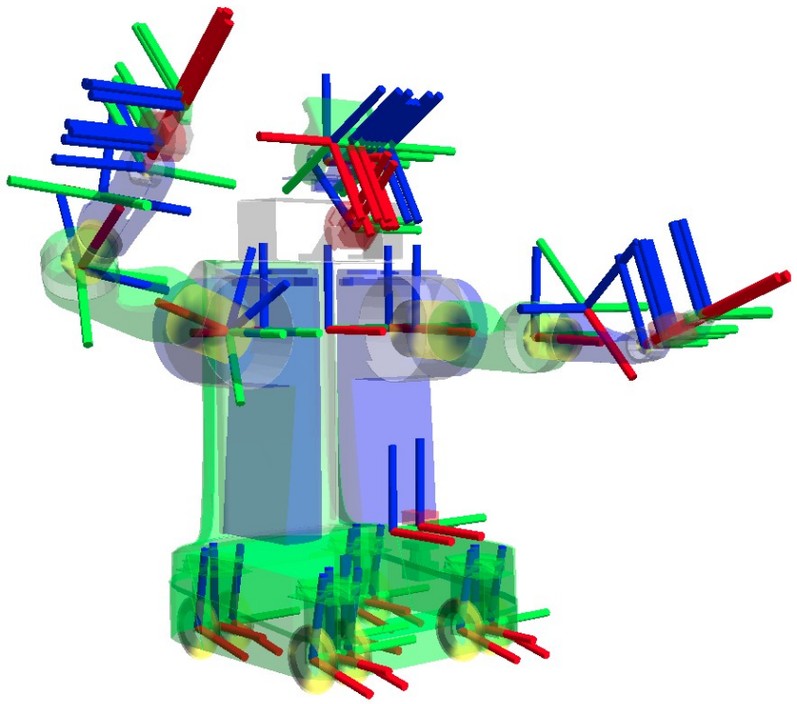


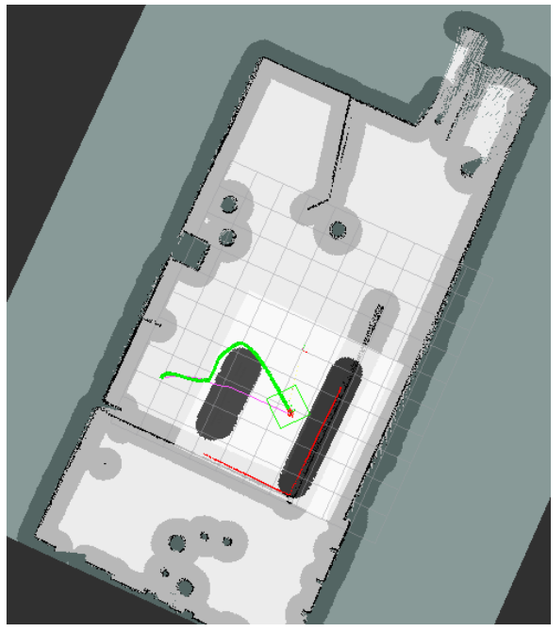
Tools

# TF2

# Tf2

- Different coordinate frames that change over time



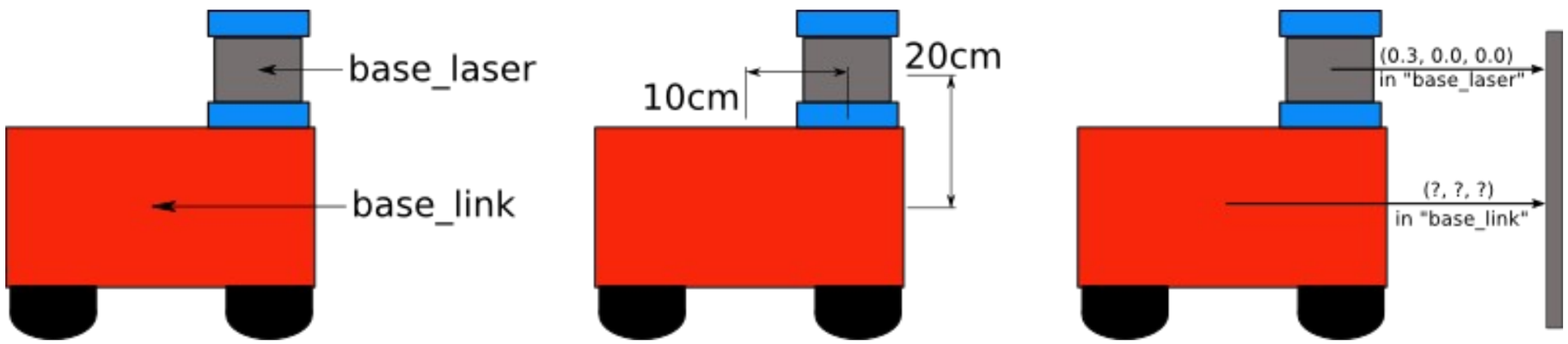


**tf: The transform library.** By Tully Foote. TePRA 2013: 1-6

https://docs.ros.org/en/humble/Tutorials/Intermediate/Tf2/Tf2-Main.html

# Tf2

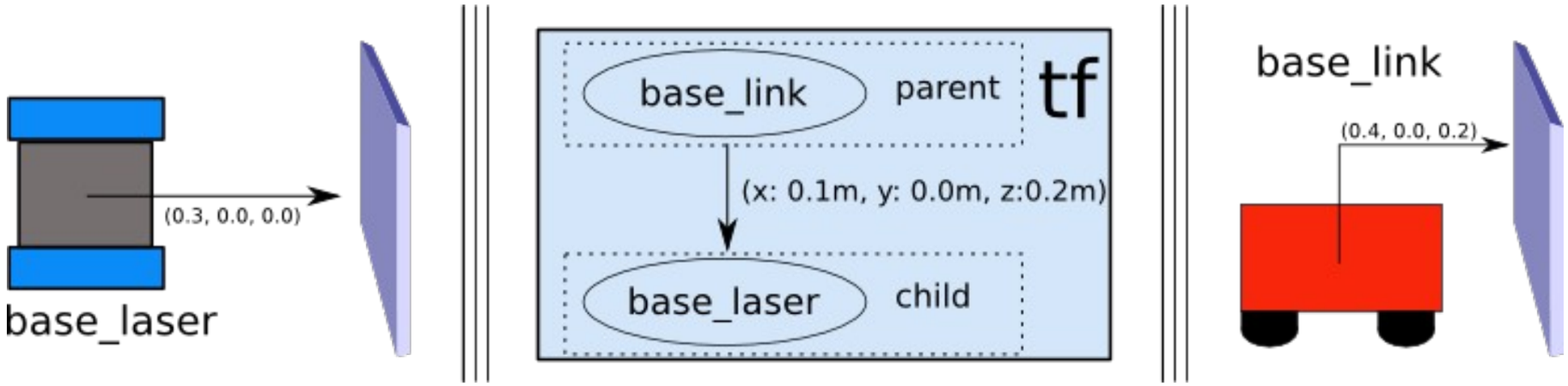
- Different coordinate frames that change over time




https://navigation.ros.org/setup_guides/transformation/setup_transforms.html

# Tf2

- Different coordinate frames that change over time

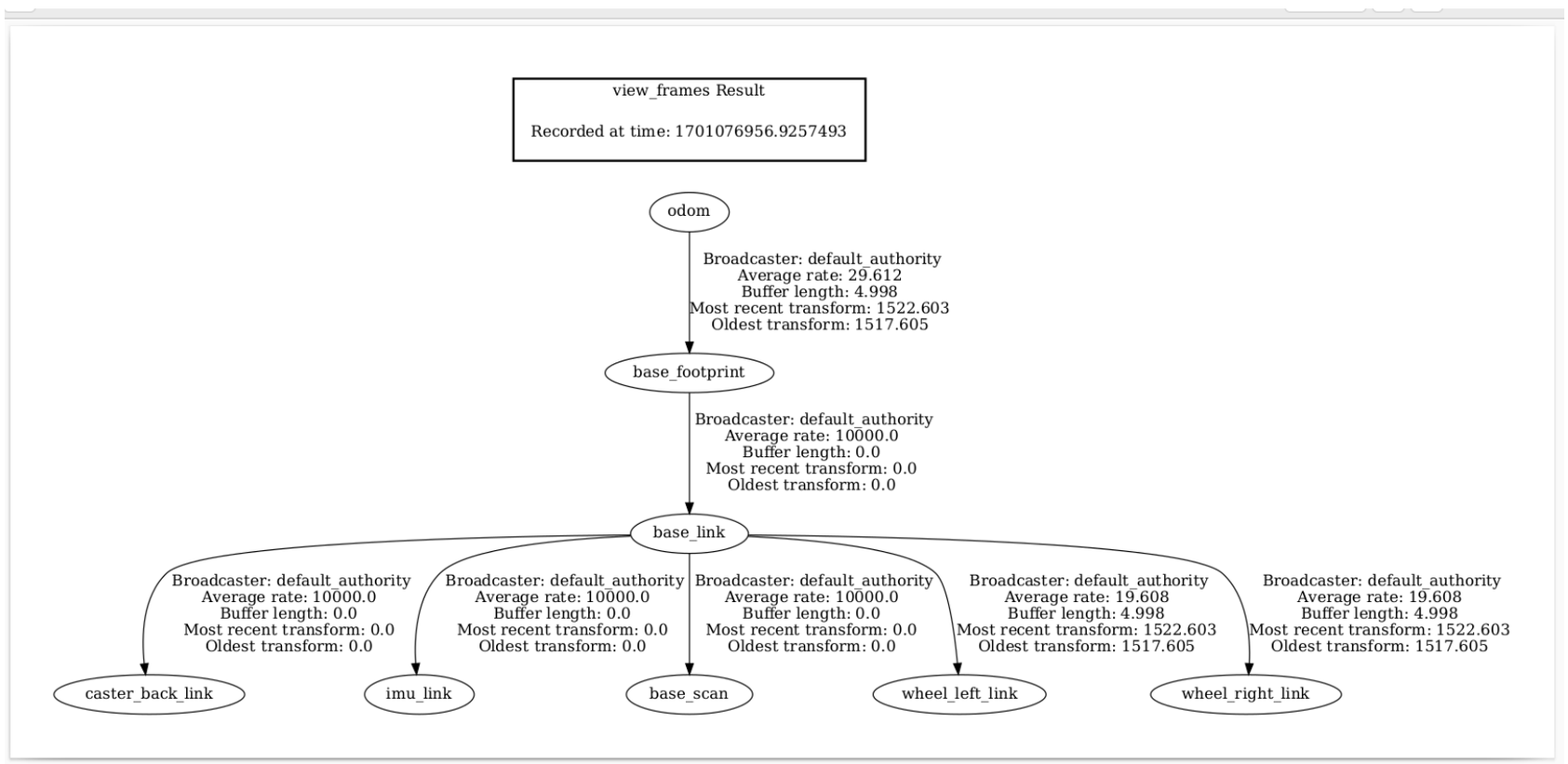



https://navigation.ros.org/setup_guides/transformation/setup_transforms.html

# Tf2

sudo apt-get install ros-humble-tf2-tools

ros2 run tf2_tools view_frames



# Tf2

ros2 run tf2_ros tf2_monitor
Gathering data on all frames for 10 seconds...

RESULTS: for all Frames

Frames:
Frame: base_footprint, published by <no authority available>, Average Delay: 1.70108e+09, Max Delay: 1.70108e+09
Frame: base_link, published by <no authority available>, Average Delay: 1.70108e+09, Max Delay: 1.70108e+09
Frame: base_scan, published by <no authority available>, Average Delay: 1.70108e+09, Max Delay: 1.70108e+09
Frame: caster_back_link, published by <no authority available>, Average Delay: 1.70108e+09, Max Delay: 1.70108e+09
Frame: imu_link, published by <no authority available>, Average Delay: 1.70108e+09, Max Delay: 1.70108e+09
Frame: wheel_left_link, published by <no authority available>, Average Delay: 1.70108e+09, Max Delay: 1.70108e+09
Frame: wheel_right_link, published by <no authority available>, Average Delay: 1.70108e+09, Max Delay: 1.70108e+09

All Broadcasters:
Node: <no authority available> 48.6645 Hz, Average Delay: 1.70108e+09 Max Delay: 1.70108e+09

# Tf2

ros2 run tf2_ros tf2_echo base_link base_scan

[INFO] [1701077588.418222986] [tf2_echo]: Waiting for transform base_link ->  base_scan: Invalid frame ID "base_link" passed to canTransform argument target_frame - frame does not exist
At time 0.0
- Translation: [-0.032, 0.000, 0.172]
- Rotation: in Quaternion [0.000, 0.000, 0.000, 1.000]
- Rotation: in RPY (radian) [0.000, -0.000, 0.000]
- Rotation: in RPY (degree) [0.000, -0.000, 0.000]
- Matrix:
  1.000  0.000  0.000 -0.032
  0.000  1.000  0.000  0.000
  0.000  0.000  1.000  0.172
  0.000  0.000  0.000  1.000

# Tf2

- Static transforms
  - Command line

    ros2 run tf2_ros static_transform_publisher --help

    usage: static_transform_publisher [--x X] [--y Y] [--z Z] [--qx QX] [--qy QY] [--qz QZ] [--qw QW] [--roll ROLL] [--pitch PITCH] [--yaw YAW] --frame-id FRAME_ID --child-frame-id CHILD_FRAME_ID
  - A command line utility for manually sending a static transform.
  - If no translation or orientation is provided, the identity transform will be published. The translation offsets are in meters. The rotation may be provided with roll, pitch, yaw euler angles in radians, or as a quaternion.

# Tf2

- Static transforms
  - Launch files

```
from launch import LaunchDescription
from launch_ros.actions import Node

def generate_launch_description():
    return LaunchDescription([
        Node(
            package='tf2_ros',
            executable='static_transform_publisher',
            arguments = ['0', '0', '1', '0', '0', '0', 'world', 'mystaticturtle']
        ),
    ])
```

https://docs.ros.org/en/foxy/Tutorials/Intermediate/Tf2/Writing-A-Tf2-Static-Broadcaster-Cpp.html

# Exercise

1. Run the turtlebot waffle simulation

   $ ros2 launch turtlebot3_gazebo turtlebot3_world.launch.py

2. Run rviz and add the /scan topic. What happens?

   $ ros2 run rviz2 rviz2

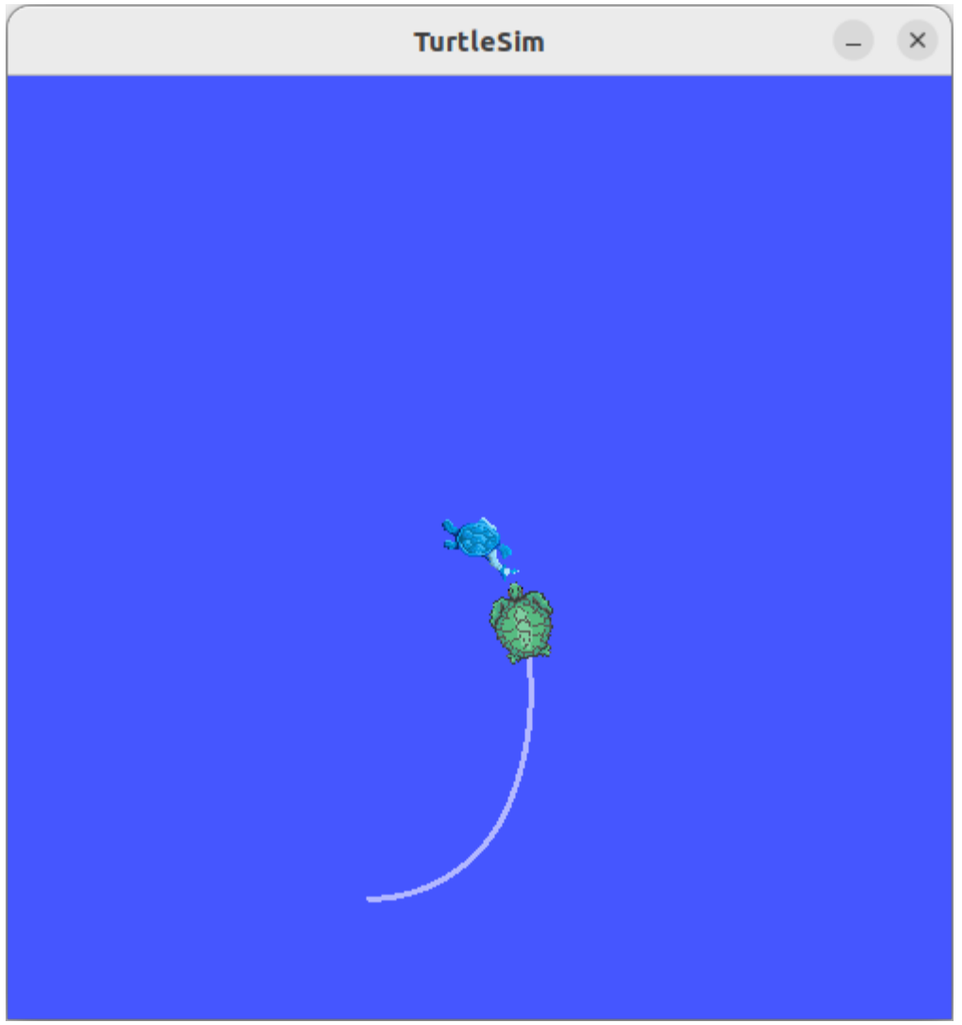
3. Play around: give different values, change the rviz fixed frame, execute view frames...

# Exercise

sudo apt-get install ros-humble-turtle-tf2-py ros-humble-tf-transformations

ros2 launch turtle_tf2_py turtle_tf2_demo.launch.py

ros2 run turtlesim turtle_teleop_key




https://docs.ros.org/en/humble/Tutorials/Intermediate/Tf2/Introduction-To-Tf2.html

# Tf broadcaster

cd ~/ros2_ws/src
ros2 pkg create --build-type ament_cmake --dependencies geometry_msgs rclcpp tf2 tf2_ros turtlesim -- learning_tf2_cpp
cd learning_tf2_cpp
mkdir src
cd src
gedit turtle_tf2_broadcaster.cpp

https://docs.ros.org/en/foxy/Tutorials/Intermediate/Tf2/Writing-A-Tf2-Broadcaster-Cpp.html

```
add_executable(turtle_tf2_broadcaster src/turtle_tf2_broadcaster.cpp)
ament_target_dependencies(
    turtle_tf2_broadcaster
    geometry_msgs
    rclcpp
    tf2
    tf2_ros
    turtlesim
)
```

```
install(TARGETS
    turtle_tf2_broadcaster
    DESTINATION lib/${PROJECT_NAME})
```

# Tf listener

cd ~/ros2_ws/src/learning_tf2_cpp/src
gedit turtle_tf2_listener.cpp

https://docs.ros.org/en/foxy/Tutorials/Intermediate/Tf2/Writing-A-Tf2-Broadcaster-Cpp.html

```
add_executable(turtle_tf2_listener src/turtle_tf2_listener.cpp)
ament_target_dependencies(
    turtle_tf2_listener
    geometry_msgs
    rclcpp
    tf2
    tf2_ros
    turtlesim
)
```

```
install(TARGETS
    turtle_tf2_listener
    DESTINATION lib/${PROJECT_NAME})
```

Tools

# ROSBAG

# ROS Bag

- Record the messages published on one or more topics to a file
- $ ros2 bag record -o <bag_file_name> <topic1> <topic2>
- Replay the file messages
- $ ros2 bag play <bag_file_name>
- Check the rosbag information

  $ ros2 bag info <bag_file_name>

https://docs.ros.org/en/humble/Tutorials/Beginner-CLI-Tools/Recording-And-Playing-Back-Data/Recording-And-Playing-Back-Data.html

# Exercise

1. Run the turtlebot simulation and record messages on /cmd_vel

   $ mkdir ~/bagfiles

   $ ros2 bag record -o ~/bagfiles/turtle_rosbag cmd_vel

2. Check the file name and the file information

   $  ros2 bag info ~/bagfiles/turtle_rosbag

3. Replay the file messages

   $ ros2 bag play ~/bagfiles/turtle_rosbag

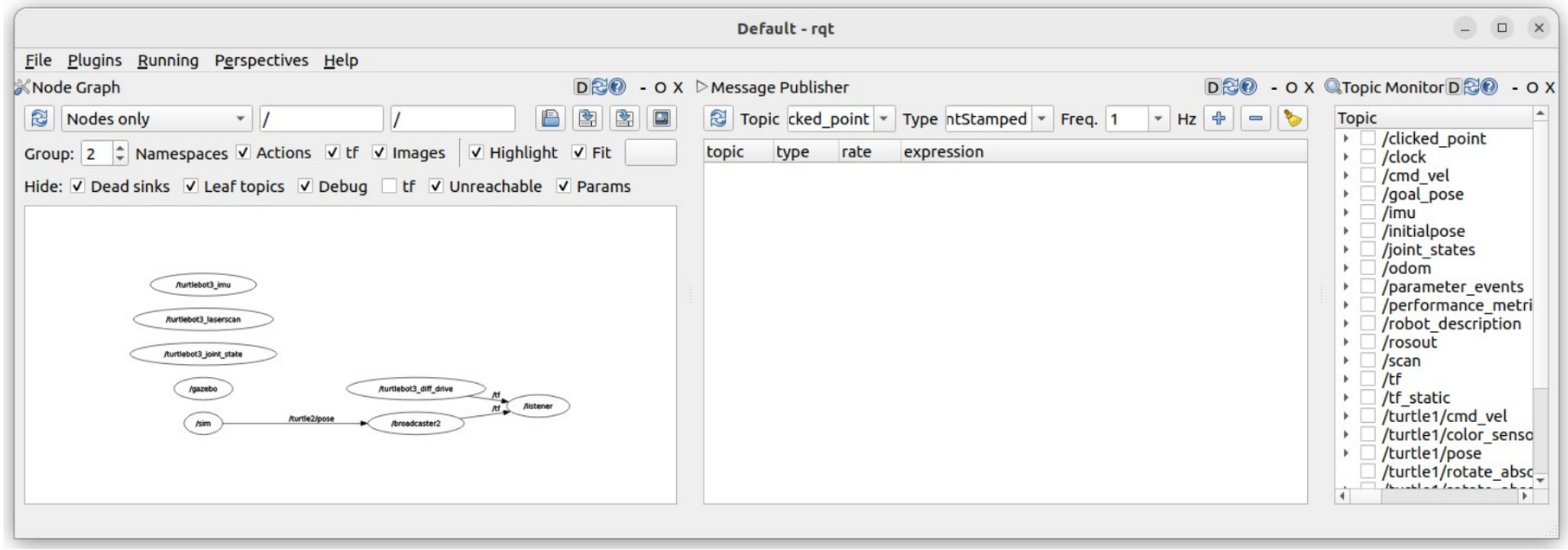
Tools

# RQT

# RQT

- General GUI where different plugins may run.
- Main utilities: introspection, logging, topics, services, actions, visualization...

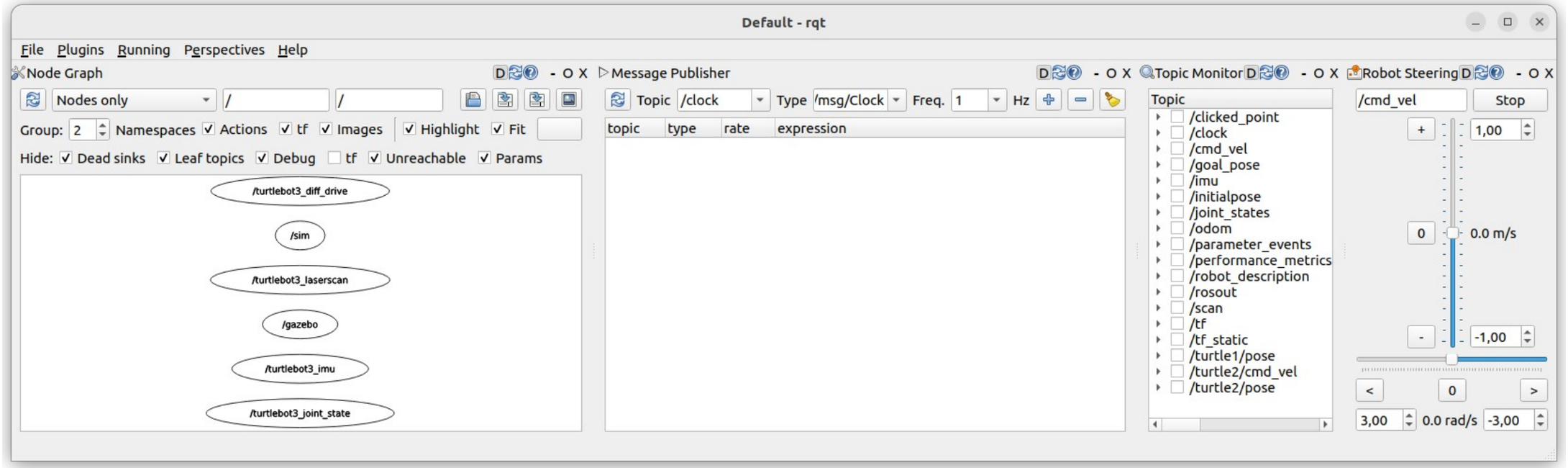
  $rqt



# RQT

$sudo apt-get install ros-humble-rqt-robot-steering

$rqt --force-discover



# ROS TOOLS

End of lesson