**IVDC Autonomy PS-Solutions**

1. Image filtering technique is utilized in image processing to enhance or revise the visual appearance of the image, this uses filter/kernel for every pixel in the image so that new pixels can be obtained based on the existing pixels. Some of the most frequently deployed image filtering techniques include Gaussian filtering, Sobel filtering Laplacian filtering, Median Filtering etc., Gaussian filter helps to blur and eliminate noise and thus act as a smoothening filter by convolving the picture with gaussian filter. The sobel filter computes an image’s gradient and is an edge detecting filter. It is frequently used to draw attention to steep gradients on edges.

Image filtering find many uses such as noise reduction, smoothing and blurring which can be used to picture denoising, spot removal, aesthetics etc., edge detection used in object identification, picture segmentation, to name a few.

Filter techniques like Gaussian technique find their use in pharmaceutical industries as a common method for decreasing noise and smoothing out photographs.

1. The adaptive filter adapts to local variation in image by changing its features based on the local characteristics of the image and its noise(within each image segment). The fixed filter function performs linear filtering of an image source using one of the predefined convolution kernel, whereas adaptive filters can recognize the local resolution and adapt itself to it.
2. Kernal are matrices that are smaller compared to the input images on which they are overlapped on the input image and the corresponding overlapping indices are multiplied and added, a process known as convolution.

Generally, smaller kernel sizes are preferred for detecting fine-grained details while larger kernel sizes are useful for detecting broader patterns. Smaller kernels captures local information while larger ones captures global information. Larger kernel sizes requires more computational resources for processing, both during training and inference.

1. 1. A few popular categories of path planning algorithms are:

• **Grid-based search algorithm:** They find a path that is of minimum cost in a grid map and can be used in applications such as mobile robots in 2D environment. However there’ll be more demand on memory requirements as the dimensions increase. Some of the grid based algorithms are:

1. A\* algorithm: This adds nodes to the environment in a way that prioritizes the nodes that are most likely to find an optimal path and searching their first.
2. D\* algorithm: Optimal and efficient algorithm known for lesser known environment.
3. Focused D\*: it is the focused D\* algorithm for real-time planning.

• **Sampling-based search algorithm:** It creates a searchable tree by randomly sampling new nodes or robot configuration in state space. Unlike Grid-based search these are suitable for higher dimensional search spaces. Generating feasible path for many practical problem makes this algorithm popular, even though it does not provide a complete solution. Some of these are:

1. RRT Algorithm: This algorithm grows a tree rooted at starting configuration by using random samples from the search space. With each sample a connection is attempted to the nearest state to the tree and if it is successful a new state is created. This algorithm both creates a graph of its surrounding and shortest path
2. RRT\*: This is an improvement to the RRT algorithm but with an addition that ensures optimal solution along with the shortest path.
3. FMT: The algorithm is specifically aimed at solved at complex motion planning problems at high-dimensional configuration.

• **Trajectory optimization algorithm:** This formulates the path planning problem that considers the desired vehicle performance, relevant constraints and vehicle dynamics. Along with generating dynamically feasible trajectories, they can also be applied for path planning in uncertain environments.

2. RRT\* algorithm can be used in this, as it can find the shortest distance and optimal path and can be used in static environment.

3. refer code solution.

1. ROS is an open-source framework for building robot application and allows developers to assemble a complex system by connecting existing solutions for a small problem. The ROS provides various functionality which allows developers to build complex software even without having to know certain hardware. They also provide a way to connect network of nodes to the central hub. Due to the way the software runs and the way it communicates ROS is used by developers for building complex systems. Some of the distributions being used for ROS are **Ubuntu**, which is the most popular and widely used, **Arch Linux**, **Debian**, **Fedor**a, **OpenSUSE**.
2. **Service** – they are a method of communication for nodes inside ROS

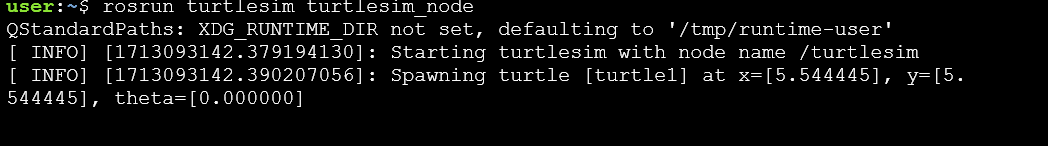
**Messages** – messages are used by nodes to communicate between each other. They are data structure containing typed fields

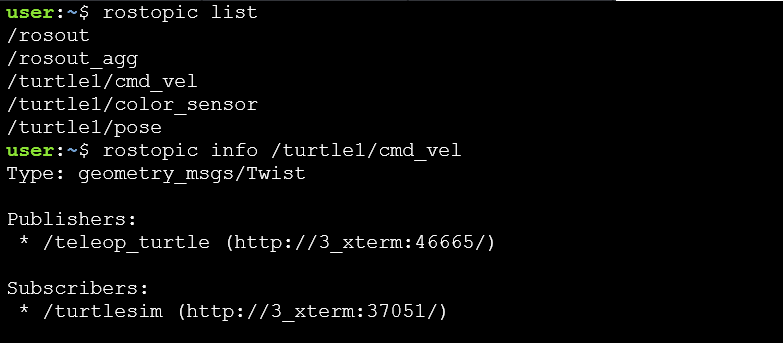
**Action**- one of the communication types which are intended for long running tasks and consist of three parts – a goal, feedback and a result.

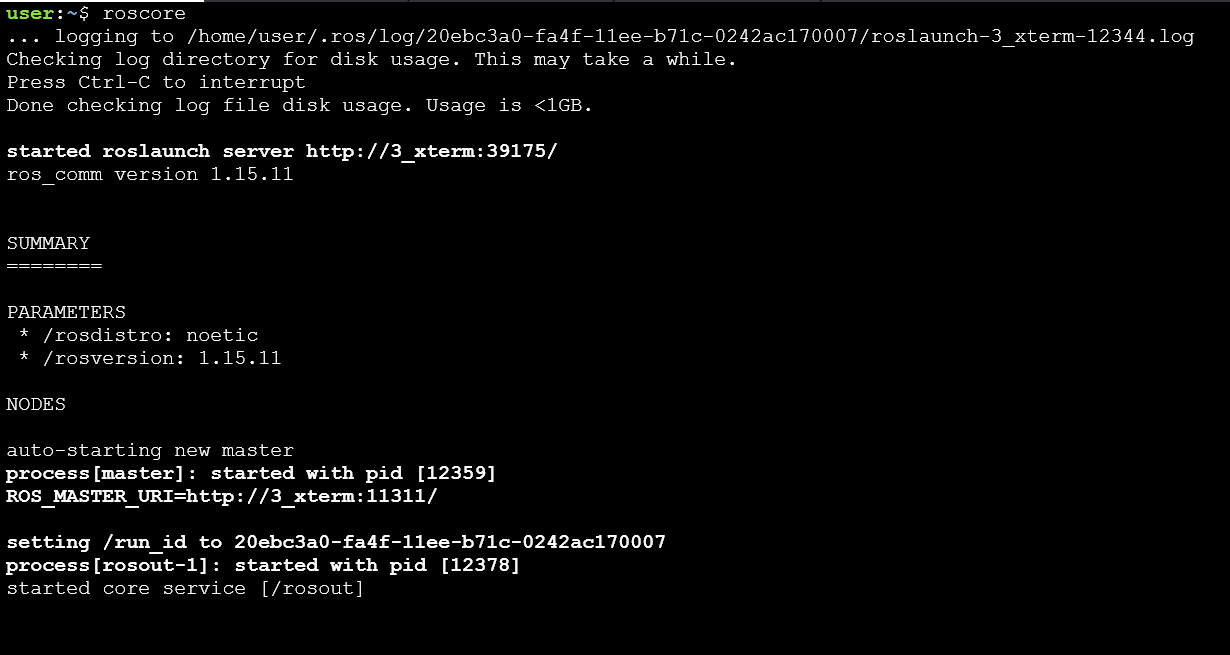
**Nodes** – It is an executable file that represents a subprogram inside the ROS and communicates with the other nodes.

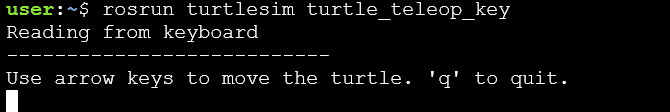
**Topic –** Ros topics transport information between the nodes

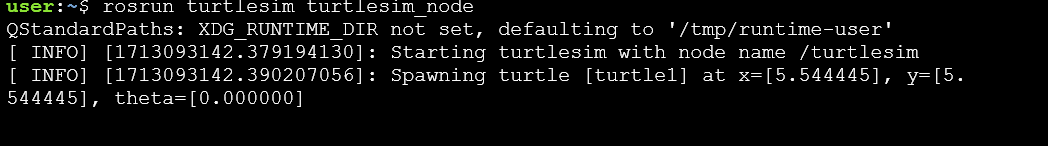
**Rqt\_graph –** rqt\_graph provides a GUI plugin to visualize the ROS computation graph. With this tool one can visualize the ROS graph of one’s application.

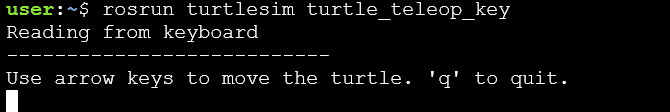
1. 

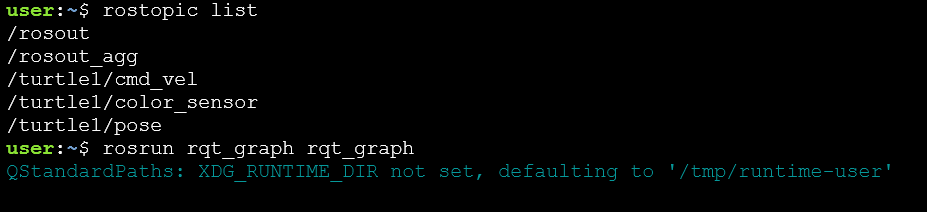


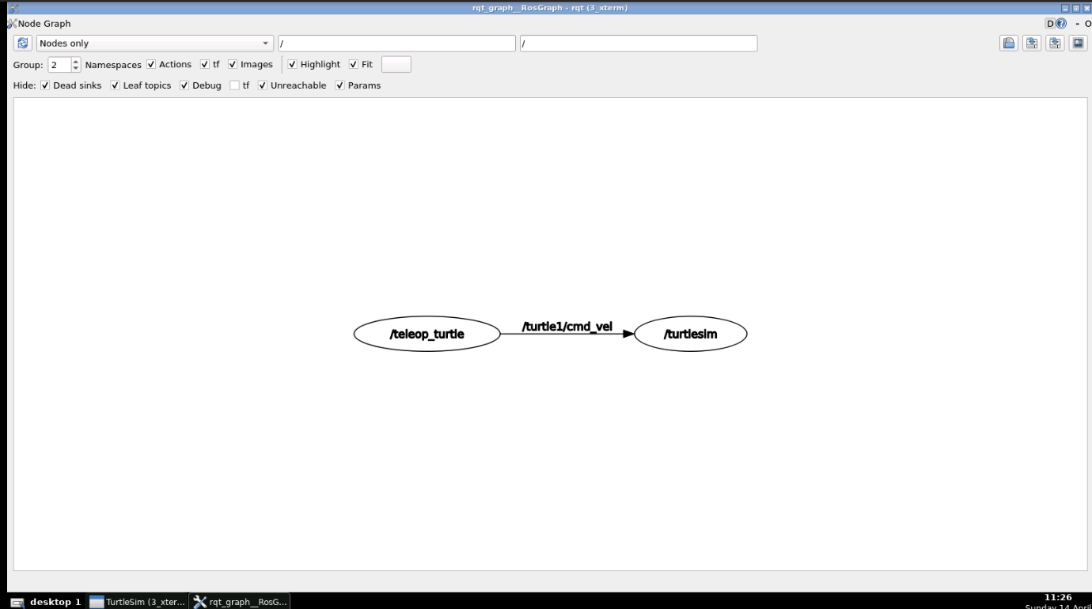




1. 







1. Quaternion is a four element vector composed of two distinct components: a scalar and a 3 element unit vector. The scalar value, w, corresponds to an angle of rotation. The vector term [x y z], corresponds to an axis of rotation, about which the angle or rotation is performed. They are often used in graphics programming, robotics, virtual reality and many things involving 3D orientation as a compact representation of an object in three dimension.

Euler angles are the relevant angles to the rotation axes which are used by programmers to visualize the rotation of object for constructing a rotation matrix for desired orientation. Euler angles is an simpler way to compute 3D orientation compared to Quaternion but has a major drawback of being vulnerable from Gimbal lock.

2. Point cloud data structure is used to represent a collection of multi-dimensional points and is a commonly used way to represent three-dimensional data. These data generally come from 2/3D Laser scans, Laser triangulations etc., Libraries for this in python includes open3d and numpy. For processing and visualizing the data python is used in tasks such as point source visualization, voxel downsampling and voxel normal estimation.
3. In order to ensure that Lidar does not fail safety system following preventative measures can be taken:

Choose the right lidar sensor for the application and environment considering the range, resolution, field of view, scanning frequency, power consumption, size, weight and cost. It should also be ventilated and grounded properly. Redundancy: Even methods like using more than one sensors in the system so that if one fails, others can still provide data.

Detecting lidar failure and acting accordingly is also important apart from taking preventative measures. This includes monitoring lidar sensor status by using indicators like alarms, message to show sensor power, temperature, signal and error code. Additionally, use data fusion technologies to combine and integrate the data from different sources and improve its quality and robustness.

2. The Kalman filter is a powerful tool for estimating and predicting system states in the presence of uncertainty and is widely used as a fundamental concept in application such as target tracking, navigation and control.

*Reference:*

1. <https://www.imageprovision.com/articles/understanding-image-filtering-techniques-in-image-processing>, <https://iq.opengenus.org/convolution-filters/>
2. <https://github.com/zhm-real/PathPlanning?tab=readme-ov-file>, <https://www.awerobotics.com/implementing-path-planning-algorithms-for-robots-using-python/>,
3. <https://www.toptal.com/robotics/introduction-to-robot-operating-system>, <https://roboticsbackend.com/>, <https://wiki.ros.org/>, <https://www.geeksforgeeks.org/ros-subscribers-using-python/>, <https://www.theconstruct.ai/read-laserscan-data/>, <https://www.youtube.com/watch?v=eJ4QPrYqMlw>, https://www.youtube.com/watch?v=zjMuIxRvygQ.
4. <https://www.diag.uniroma1.it/~nardi/Didattica/LabAI/matdid/pcl_intro.pdf>, <https://sigmoidal.ai/en/point-cloud-processing-with-open3d-and-python/>, <https://www.linkedin.com/advice/0/how-can-you-prevent-lidar-from-failing-robot-safety-system-l69xf>, <https://www.kalmanfilter.net/default.aspx>.