Abstract: this report explains about the robot programming for agricultural robots and its implementation in detecting the number grape bunches in the vineyard, it also explains about the concepts used in creating the programs, by using the methods such as topological navigation to create wave points to give the robot the co-ordinates to the desired position, using opencv with ROS programming to detect the grape bunches by moving the robot autonomously avoiding obstacles from one wave point to other. The counting of fruits done by detecting individual co -ordinates of every fruit and saving so that it should not repeat counting.

1. Introduction:

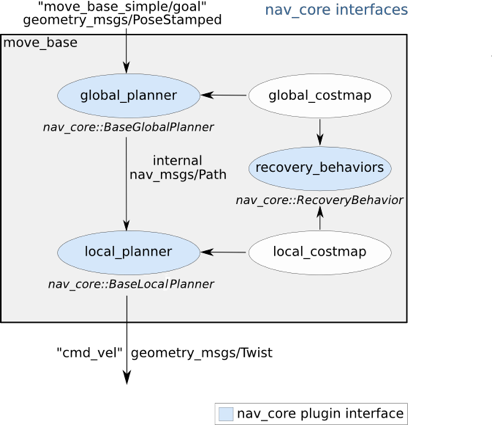
The main characteristics required for the robot is to be adaptable; they can be adapted to different types of tasks without any crucial changes. These adaptability is derived from the generalization of the structure and control, but it can be used if the robot can be programmed easily. In some cases lack of proper programming may lead to perform particular tasks impossible. Due to these reasons the robot programming plays a very important role in robot development [1].

* 1. history:

In the beginning the robots were moved manually to a required position and recording the co-ordinates of the position based on the internal joint co-ordinates, in addition to this moving the robot parts and performing particular tasks at some points like tighten the screws at particular positions, these requires robot programming, these can be done by moving the robot to the specific co-ordinates and giving them the activation signal to perform the task at required position. This method is call teaching by showing [2].The latest advance towards robotic programming provides the ability of robot- level languages without any need programming skills.

2.Navigation in ROS programming:

In navigation the global planner and local planner are required for the path planning. The global planner requires a map of whole environment to calculate the best way possible and the local planner uses the local information obtained from the sensors. The below algorithm shows how global planner and local planner are connected and how it’s used in the maps for navigation by showing how they are connected with one another.



2.1 SLAM: In order to move the robot to a desired position the navigation is very important, one of the most researched areas of robotics is simultaneous localization and mapping (SLAM). Two different approaches exist to the problem of SLAM: topological and metric [3].to use the metric navigation it needs geometric model of the world also it assumes the exact sensor information and used for more precise applications. The topological navigation uses waypoints for. The navigation leads to a quantitative description of navigation goals which is flexible and easy to define a map

2.1.1 Topological navigation: even though the topological map is not precise it is most suitable for agricultural fields where it needs to be moved in certain pattern. In this problem, to count the number grape bunches in the vineyard where the robot needs to be moving in certain pattern in the vineyard the wave points can be created and the directions are given between the nodes to move the robot from one way points to other and load the map to the rviz using a launch file top.nav.launch mentioned in the source code this way the robots can move in the particular path

Waypoints: in Rviz the waypoints can be added using a ROS service call, but first one node should be created in the yaml file by giving the x, y co-ordinates then the map file should be added to a folder. The yaml file added to folder using rosrun topological\_utils load\_yaml\_map.py $(rospack find uol\_cmp9767m\_tutorial)/maps/test. yaml to load the map to Rviz then the additional nodes can be added using below ROS service call

rosservice call /topological\_map\_manager/add\_topological\_node "

name: WayPoint6

pose:

position:

x: -2.0

y: -2.0

z: 0.0

orientation:

x: 0.0

y: 0.0

z: 0.0

w: 1.0

add\_close\_nodes: false"

where x and y are the co-ordinate of the way points the connection between the nodes can also be added using the ROS service call

rosservice call /topological\_map\_manager/add\_edges\_between\_nodes "

origin: WayPoint0

destination: WayPoint6

action: move\_base

edge\_id: WayPoint0\_WayPoint6"

The uni-directional edge between the nodes is added, which is enough for this scenario

Similarly, bi-directional edges can also be added between the nodes

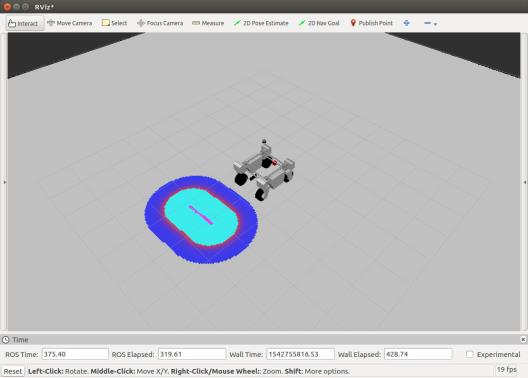
Below picture shows the waypoints added to the map. Where, green arrow mark indicates the

Waypoints and the red arrow mark indicates the edges between them

A screenshot of a computer

Description automatically generated with medium confidence

3.autonomous motion and obstacle avoidance of robot: to move robot from one way points to other python script is used in ROS. Python language is common language used in ROS ,because it is compatible with ROS programming. Where, The MoveBaseGoal class is used to describe the next goal that should be moved where the new X and Y locations will be as well as the final orientation or 'pose' that will be assumed once the endpoint is reached. A SimpleActionClient is formed, and the new goal is sent to the move\_basic server, which is where move\_basic receives new goals. A result is requested from the move\_basic server using our just used client interface. A ROS message will be sent on a ROS topic and the result will be received in another ROS topic. The obstacle will be avoided using the data from the laser scanner or other sensors. it creates the obstacle map from the data collected from the sensors



The blue inflation layer in the above photos shows that it is near to the obstacle.

detection and counting of grape bunches in ROS programming:

1.T. Lozano-Perez, "Robot programming," in Proceedings of the IEEE, vol. 71, no. 7, pp. 821-841, July 1983, doi: 10.1109/PROC.1983.12681.

2.D. D. Grossman, “Programming a computer controlled manip ulator by guiding through the motions,” IBM T. J. Watson Res Cen., Res. Rep. RC6393, 1977 (Declassified 1981).

3.Filliat, D., Meyer, J.A.: Map-based navigation in mobile robots: I. a review of localization strategies. Cog. Sys. Res. 4(4), 243–282 (2003)CrossRefGoogle Scholar